#1 14.4

Prework: So, $\min \|X^TB - Y\| \Leftrightarrow \min \|Y - X^TB\|$, which is the minimization of error E

$$Y = X^TB + E \;\&\; E \leq Y - X^TB$$

Therefore, $E(x) = (Y - X^TB)^T(Y - X^TB)$
$= (Y^T - (X^TB)^T)(Y - X^TB)$
$= (Y^T - (B^TX))(Y - X^TB)$
$= Y^TY - Y^TX^TB - YB^TX + B^TXX^TB$

Note: $Y^TXB$ is a scalar
So, $Y^TXB = C = C^T = (Y^TXB)^T = B^TX^TY$
Thus, $Y^TXB = B^TX^TY$

We want to minimize $E(x)$ so

$$\frac{\partial E(x)}{\partial B} = -X^TY + XX^TB = 0$$

Then, $-X^TY + XX^TB = 0$
$\Rightarrow XX^TB = X^TY$
$\Rightarrow B = (XX^T)^{-1}X^TY$

#1 14.4a $Y$ is the truth value for each of the variables in $X$ for all samples $N$. The $i^{th}$ Row of $Y$ is the truth for each of the variables in the $i^{th}$ row of $X$. i.e. the actual response to $i^{th}$ question for each of the $N$ responses.

#1 14.4b Since, the rows of $X$ are lin. indep. we can use the pseudo inverse
$* \; X^+ = X(XX^T)^{-1}$ where $(X^+)^T = (X(XX^T)^{-1})^T = (XX^T)^{-T}X^T = (X^TX)^{-1}X^T = (X^T)^+$

From my prework I found that:

$$B = (XX^T)^{-1}X^TY$$

Thus, by $*$ we see that

$$\hat{B} = (X^T)^+Y$$

#2 15.4 Here we want to minimize $\|A^{(1)}x - b\|^2 + \cdots + \|A^{(k)}x - b\|^2$

Let $F(x) = \sum_{i=0}^{k} \|A^ix - b\|^2$

we want to minimize $F(x)$ with respect to $x$

So, $\frac{\partial F(x)}{\partial x} = \sum_{i=1}^{k} 2\langle A_j^{(k)}, A^{(k)}x - b\rangle = 0$, where $A_j^{(k)}$ is the $j^{th}$ column of $A^{(k)}$

Then, $\sum_{i=1}^{k} \langle A_j^{(k)}, A^{(k)}x - b\rangle = 0$
$\Rightarrow (A^{(1)}x - b) + \cdots + (A^{(k)}x - b) = 0$
$\Rightarrow (A^{(1)}x + \cdots + A^{(k)}x) - kb = 0$
$\Rightarrow \sum_{i=1}^{k}(A^{(i)})x = kb$

Therefore, $x_k^{rob} = \left(\sum_{i=1}^{k}(A^{(i)})\right)^{-1}kb$

To verify that $x^{rob} = (A^{(1)})^{-1}b$ for $k=1$,
we set $k=1$ in the above equation and get:

$$x_1^{rob} = \left(\sum_{i=1}^{1} A^{(i)}\right)^{-1}(1)b = (A^{(1)})^{-1}b$$

Thus, for $k=1$ $\quad x^{rob} = (A^{(1)})^{-1}b$

# 15.11 a Let $A = 0_{m\times n}$ WLOG $m > n$. other case $m=n$ & $n>m$

$$0^+ = \lim_{\lambda\to 0} 0^T_{m\times n}(0_{m\times n}0^T_{m\times n} + \lambda I)^{-1} = \lim_{\lambda\to 0} 0_{m\times n} = 0_{m\times n}$$

#15.11 b Consider the first formula

Since, the columns of $A$ are lin. indep $(A^TA)^{-1}$ exists (*1)

Thus, $A^+ = \lim_{\lambda \to 0} (A^TA + \lambda I)^{-1} A^T = (A^TA)^{-1}A^T$ which holds by (*1)

So, the limit reduces to $A^+ = (A^TA)^{-1}A^T$

#15.11c Consider the second formula

Since, the rows of $A$ are lin. indep $(AA^T)^{-1}$ exists (*2)

Thus, $A^+ = \lim_{\lambda \to 0} A^T(AA^T + \lambda I)^{-1} = A^T(AA^T)^{-1}$ which holds by (*2)

So, the limit reduces to $A^+ = A^T(AA^T)^{-1}$

#4 16.11 Assume $Cx = d$ & let $y = x - a$

Then, $C(y+a) = d$

$\Rightarrow Cy + Ca = d$

$\Rightarrow -Cy = Ca - d$

$\Rightarrow y = C^+(Ca - d)$

$\Rightarrow x - a = C^+(Ca - d)$

$\Rightarrow x = a + C^+(Ca - d)$

Note: $C^+$ exists since the rows of $C$ are lin. indep. and $C^+$ is the left pseudo inverse of $C$.

#5 (1) Given that the columns of $A$ are lin. indep.

First, we can compute $A = QR$ using the Gram-Schmidt algorithm

Then with $Q$ & $R$, we can compute $A^+ = R^{-1}Q^T$

So, we solve $x = A^+b = R^{-1}Q^Tb$

#5 (2) First, compute $A = LL^T$ using cholesky factorization where $L$ is a lower triangular matrix.

Then, use forward substitution to solve $Lv = b$

Finally, use backwards substitution to solve $L^Tx = v$

#6

Let $E = \lambda I_n : n \overset{s}{\times} n$, $F = A^T : n \times m$, $G = I_m : m \overset{s}{\times} m$, & $H = A : n \times m$

Consider $(A^TA + \lambda I_n)^{-1} = (\lambda I_n + A^TA)^{-1} = (\lambda I_n + A^T I_m A)^{-1} = (E + FGH)^{-1}$

By matrix inversion lemma then

$$(E + FGH)^{-1} = E^{-1} - E^{-1}F(G^{-1} + HE^{-1}F)^{-1}HE^{-1}$$

$$= (\lambda I_n)^{-1} - (\lambda I_n)^{-1}A^T(I_m^{-1} + A(\lambda I_n)^{-1}A^T)^{-1}A(\lambda I)^{-1}$$

$$= \lambda^{-1}I - \lambda^{-1}I\, A^T(I + \lambda^{-1}AA^T)^{-1}\lambda^{-1}A$$

$$= (\lambda^{-1})^2 I(I - A^T(I + \lambda^{-1}AA^T)^{-1}A)$$

- $J = (I + \lambda^{-1}AA^T)^{-1}$ takes $m^2n + m^2 + m + m^3$ flops to compute

$$= (\lambda^{-1})^2 I(I - A^TJA)$$

- $K = (I - A^TJA)$ takes $m^2n + m^2n + n$ flops to compute

$L = (\lambda^{-1})^2 IK$ takes $m^3 + m^2$ flops to compute

Then, $\hat{x} = Lb$ takes $m^2 - m$ flops to compute

Therefore, the total flops required is: $3m^2 + 3m^2n + 2m^3 + n$

which is $O(m^2n + m^3)$

**#7a** Using $P_A(\lambda) = \det(\lambda I_n - A)$ (i.e. formal definition of characteristic polynomial)

$$P_C(\lambda) = \det(\lambda I_n - C) = \det\begin{bmatrix} \lambda & 0 & \cdots & 0 & c_0 \\ -1 & \lambda & \cdots & 0 & c_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda & c_{n-2} \\ 0 & 0 & \cdots & -1 & \lambda + c_{n-1} \end{bmatrix}$$

$$= \lambda \cdot \det\begin{bmatrix} \lambda & 0 & \cdots & 0 & c_1 \\ -1 & \lambda & \cdots & 0 & c_0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda & c_{n-2} \\ 0 & 0 & \cdots & -1 & \lambda + c_{n-1} \end{bmatrix} + (-1)^{1+n} \cdot c_0 \cdot \det\begin{bmatrix} -1 & \lambda & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda \\ 0 & 0 & \cdots & -1 \end{bmatrix}$$

$$= \lambda \cdot \det\begin{bmatrix} \lambda & 0 & \cdots & 0 & c_1 \\ -1 & \lambda & \cdots & 0 & c_0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda & c_{n-2} \\ 0 & 0 & \cdots & -1 & \lambda + c_{n-1} \end{bmatrix} + (-1)^{1+n} \cdot c_0 (-1)^{n-1}$$

$$= \lambda \cdot \det\begin{bmatrix} \lambda & 0 & \cdots & 0 & c_1 \\ -1 & \lambda & \cdots & 0 & c_0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda & c_{n-2} \\ 0 & 0 & \cdots & -1 & \lambda + c_{n-1} \end{bmatrix} + (-1)^{2n} c_0$$

$$= \lambda \cdot \det\begin{bmatrix} \lambda & 0 & \cdots & 0 & c_1 \\ -1 & \lambda & \cdots & 0 & c_0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda & c_{n-2} \\ 0 & 0 & \cdots & -1 & \lambda + c_{n-1} \end{bmatrix} + c_0$$

$$\vdots$$

$$= \lambda^n + \lambda^{n-1} c_{n-1} + \cdots + \lambda c_1 + c_0$$

**#7b** Assume $C$ has $n$ distinct eigenvalues $(\lambda_1, \lambda_2, \ldots, \lambda_n)$

Then,
$$VC = \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{n-1} \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \lambda_n & \lambda_n^2 & \cdots & \lambda_n^{n-1} \end{bmatrix} \begin{bmatrix} 0 & 0 & \cdots & 0 & -c_0 \\ 1 & 0 & \cdots & 0 & -c_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & -c_{n-2} \\ 0 & 0 & \cdots & 1 & -c_{n-1} \end{bmatrix}$$

$$= \begin{bmatrix} \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{n-1} & -p(\lambda_1) + \lambda_1^n \\ \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{n-1} & -p(\lambda_2) + \lambda_2^n \\ \vdots & \vdots & & \vdots & \\ \lambda_n & \lambda_n^2 & \cdots & \lambda_n^{n-1} & -p(\lambda_n) + \lambda_n^n \end{bmatrix}$$

since: $-c_0 - c_1\lambda_i - \cdots - c_{n-1}\lambda_i^{n-1} - \lambda_i^n = 0$

$\Rightarrow -c_0 - c_1\lambda_i - \cdots - c_{n-1}\lambda_i^{n-1} - \lambda_i^n + \lambda_i^n = \lambda_i^n$

$\Rightarrow -p(\lambda_i) + \lambda_i^n = 0 + \lambda_i^n = \lambda_i^n$

$$= \begin{bmatrix} \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{n-1} & \lambda_1^n \\ \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{n-1} & \lambda_2^n \\ \vdots & \vdots & & \vdots & \vdots \\ \lambda_n & \lambda_n^2 & \cdots & \lambda_n^{n-1} & \lambda_n^n \end{bmatrix}$$

$$= \begin{bmatrix} \lambda_1 & & & & \\ & \lambda_2 & & & \\ & & \ddots & & \\ & & & \lambda_{n-1} & \\ & & & & \lambda_n \end{bmatrix} \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{n-1} \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \lambda_n & \lambda_n^2 & \cdots & \lambda_n^{n-1} \end{bmatrix}$$

$$= \mathrm{Diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) \cdot V$$

$$= \Lambda V$$

#8a Consider, $w = e^{2\pi i/n}$

Note: $w^n = (e^{2\pi i/n})^n = e^{2\pi i} = 1.$ So, $w^{j+n} = w^j w^n = w^j$

Using the primitive roots of unity of $w$

The $i^{th}$ normalized eigenvector of $C$ is

$$x^{(i)} = \begin{bmatrix} 1/\sqrt{n}\, w^{0(i-1)} \\ 1/\sqrt{n}\, w^{1(i-1)} \\ \vdots \\ 1/\sqrt{n}\, w^{(n-1)(i-1)} \end{bmatrix},$$ and its corresponding eigenvalue is $\lambda_i$

Therefore, the eigenmatrix of $C$ is $V = [x^{(1)}\ x^{(2)} \cdots x^{(n)}] = F^* = F^{-1}$

and $\Lambda = \mathrm{Diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) = \mathrm{Diag}\left(\begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \\ & & & \lambda_n \end{bmatrix}\right) = \mathrm{Diag}(x^{(1)}\ x^{(2)} \cdots x^{(n)}) = \mathrm{Diag}(Fc)$

Thus,

$$C = V\Lambda V^{-1} = F^{-1}\Lambda(F^{-1})^{-1} = F^*\Lambda F$$

#8b Suppose $C = F^*\Lambda F$

Then, I would solve $x = (F^*\Lambda F)^{-1} b = (F\Lambda^{-1}F^{-1})b = (F\Lambda F^*)b$

Step 1: $x = F^* b$ takes $n\log n$ flops

Step 2: $y = \Lambda x$ takes $n\log n$ flops for $\Lambda$ & $n$ flops for $\Lambda x$

Step 3: $z = Fy$ takes $n\log n$ flops

Total: takes $3n\log n + n$ flops

which is $O(n\log n)$

#9a Suppose $\nabla f(x) = 0$. Then, $A^TA\hat{x} = A^Tb$

$A^TA\hat{x} = A^TA(V\Sigma^{-1}U^Tb) = A^T(U\Sigma V^T)(V\Sigma^{-1}U^T)b$

$= A^T(U\Sigma(V^TV)\Sigma^{-1}U^T)b = A^T(U\Sigma(I)\Sigma^{-1}U^T)b$

$= A^T(U(\Sigma\Sigma^{-1})U^T)b = A^T(U(I)U^T)b$

$= A^T(UU^T)b = A^T(I)b$

$= A^Tb$

#9b Assume $x \neq \hat{x}$

Then, $\|Ax-b\|^2 = \|(Ax-A\hat{x}) + (A\hat{x}-b)\|^2$

$= \|(Ax-A\hat{x})\|^2 + \|(A\hat{x}-b)\|^2 + 2(Ax-A\hat{x})^T(A\hat{x}-b)$

Looking at the third term on the RHS

Then, $(Ax-A\hat{x})^T(A\hat{x}-b) = (A(x-\hat{x}))^T(A\hat{x}-b)$

$= (x-\hat{x})^TA^T(A\hat{x}-b)$

$= (x-\hat{x})^T(A^TA\hat{x}-A^Tb)$

$= (x-\hat{x})^T(0)$ (by #9a)

$= 0$

So, $\|Ax-b\|^2 = \|(Ax-A\hat{x})\|^2 + \|(A\hat{x}-b)\|^2$

Additionally, the first term on the RHS is non-negative.

Therefore, we have

$\|Ax-b\|^2 \geq \|(A\hat{x}-b)\|^2$

#10a

## Part A

In [210]:
```
a = np.random.rand(6, 3)
u, s, vh = np.linalg.svd(a, full_matrices=False)
q, r = np.linalg.qr(a)
list_r = [r]
list_q = [q]

for i in range(100000):
    q, r = np.linalg.qr(np.transpose(r))
    list_r.append(r)
    list_q.append(q)

print(a)
```

Starting matrix

```
[[0.68607795 0.801398   0.48055485]
 [0.11252787 0.75508562 0.45226404]
 [0.75385057 0.53701423 0.02117887]
 [0.0853504  0.02602419 0.12403826]
 [0.62733978 0.19914588 0.47546382]
 [0.6633118  0.07862518 0.18284602]]
```

In [213]:
```
print(V)
print('\n')
print(np.transpose(vh))
```

V algorithm vs V SVD

```
[[-0.68474985 -0.72202029 -0.09901686]
 [-0.61224018  0.64361845 -0.45925728]
 [-0.39532216  0.25385425  0.88276747]]


[[-0.68474985 -0.72202029 -0.09901686]
 [-0.61224018  0.64361845 -0.45925728]
 [-0.39532216  0.25385425  0.88276747]]
```

In [214]:
```
print(U)
print('\n')
print(u)
```

U algorithm vs U SVD

```
[[ 0.61850168 -0.19841316 -0.02806078]
 [ 0.38609503 -0.72379565  0.09857747]
 [ 0.45879178  0.26927238 -0.72180028]
 [ 0.06635039  0.01865035  0.21253559]
 [ 0.39755746  0.28430806  0.63490049]
 [ 0.30893704  0.53203906  0.14223016]]


[[-0.61850168  0.19841316 -0.02806078]
 [-0.38609503  0.72379565  0.09857747]
 [-0.45879178 -0.26927238 -0.72180028]
 [-0.06635039 -0.01865035  0.21253559]
 [-0.39755746 -0.28430806  0.63490049]
 [-0.30893704 -0.53203906  0.14223016]]
```

In [215]:
```
print(list_r[99999])
print('\n')
print(s)
```

Σ algorithm vs Σ SVD

```
[[ 1.86000111e+000 -9.88131292e-324  0.00000000e+000]
 [ 0.00000000e+000 -7.17811429e-001  0.00000000e+000]
 [ 0.00000000e+000  0.00000000e+000  4.19195739e-001]]


[1.86000111 0.71781143 0.41919574]
```

---

Based on the results of the test I can conclude the hypothesis that

$Q_1 Q_3 \cdots Q_{2k+1} \cdots$ converges to $V$

$Q_0 Q_2 \cdots Q_{2k} \cdots$ converges to $U$

&

$R_{2k}$ converges to $\Sigma$

Note that if $\lambda$ is the eigenvalue for the eigenvector $v$ then $(-\lambda)$ is the eigenvalue for the eigenvector $(-v)$.

#10b Prove $Q_1(R_1R_0)$ is the QR factorization of $R_0^TR_0$

$Q_1(R_1R_0) = (Q_1R_1)R_0 = R_0^TR_0$

#10c ① Prove $(R_1R_0)Q_1 = R_1R_1^T$

$$\begin{aligned}(R_1R_0)Q_1 &= R_1(Q_1R_1)^TQ_1\\ &= R_1R_1^TQ_1^TQ_1\\ &= R_1R_1^T(I)\\ &= R_1R_1^T\end{aligned}$$

② Prove $R_1R_1^T = R_2^TR_2$

$$\begin{aligned}R_1R_1^T &= (Q_2R_2)^T(Q_2R_2)\\ &= R_2^TQ_2^TQ_2R_2\\ &= R_2^T(I)R_2\\ &= R_2^TR_2\end{aligned}$$

③ Prove $B_0 = R_0^TR_0$ & $B_1 = R_2^TR_2$ have the same eigenvalues

In ② we showed that:

- $R_iR_i^T = R_{i+1}^TR_{i+1}$

Applying ② to $R_{i+1}^TR_{i+1}$ we see that:

- $R_iR_i^T = R_{i+2}R_{i+2}^T$

So, $B_0 = R_0R_0^T = R_2R_2^T = B_1$

Therefore, $B_0$ and $B_1$ are similar and that:

- $B_1 = S^{-1}B_0S$, where $S$ is the change of basis matrix

Then, the characteristic polynomial of $B_1$ is:

$$\begin{aligned}- P_{B_1}(\lambda) = \det(B_1 - \lambda I) &= \det(S^{-1}B_0S - \lambda I)\\ &= \det(S^{-1}(B_0 - \lambda I)S)\\ &= \det(S^{-1})\det(B_0 - \lambda I)\det(S)\\ &= \det(B_0 - \lambda I) = P_{B_0}(\lambda)\end{aligned}$$

Thus, since $B_0$ & $B_1$ have the same characteristic then they have the same eigenvalues.

#10d Prove $B_K = R_{2K}R_{2K}^T$ are the ...

In #10c I showed applying the QR factorization to $B_K$ an odd number of times (i.e. once) results in $B_K = R_{K+1}R_{K+1}^T = (B_{K+1})^T$.

Applying it an even number of times (i.e. twice) resuts in $B_K = (B_{K+1})^T = B_{K+2}$

So, $B_K = B_t = R_tR_t^T$, where $t = 2n$ & $n \in \mathbb{Z}^+$ (i.e. $t$ is even)

Therefore, if we choose $t = 2K$, where $K \in \mathbb{Z}^+$

Thus, $B_K = R_{2K}R_{2K}^T$