Ayudantía
SCT - Semana - 10

Mínimos Cuadrados - Ecuaciones Normales y Factorización QR

1. Use mínimos cuadrados, las ecuaciones normales, el proceso de Gram-Schmidt y la factorización QR para encontrar la ecuación lineal $y = ax + b$ que mejor ajuste los tres puntos $(1, 3)$, $(2, 4)$ y $(3, 4)$.

2. Considere la siguiente secuencia de matrices tridiagonales $T_n \in \mathbb{R}^{(n+1)\times n}$,

$$
T_n = \begin{bmatrix}
a_1 & c_1 & 0 & \dots & \dots & \dots & 0 \\
b_1 & a_2 & c_2 & 0 & \dots & \dots & \vdots \\
0 & b_2 & a_3 & c_3 & 0 & \dots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\
\vdots & \ddots & \ddots & 0 & b_{n-2} & a_{n-1} & c_{n-1} \\
\vdots & \ddots & \ddots & \ddots & 0 & b_{n-1} & a_n \\
0 & \ddots & \ddots & \ddots & \ddots & 0 & b_n
\end{bmatrix},
$$

donde los coeficientes a_i, b_i y c_i son conocidos. Note que T_n no es una matriz cuadrada, dado que tiene una fila más que la cantidad de columnas. Considere que usted tiene acceso a la factorización QR reducida de la matriz T_n, es decir, tiene acceso a,

$$
T_n = \widehat{Q}_n\,\widehat{R}_n = \begin{bmatrix} \mathbf{q}_1^{[n]} & \mathbf{q}_2^{[n]} & \dots & \mathbf{q}_n^{[n]} \end{bmatrix}
\begin{bmatrix}
r_{1,1} & r_{1,2} & r_{1,3} & \dots & r_{1,n} \\
0 & r_{2,2} & r_{2,3} & \dots & r_{2,n} \\
0 & 0 & r_{3,3} & \dots & r_{3,n} \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
0 & \dots & \dots & 0 & r_{n,n}
\end{bmatrix},
$$

donde $\widehat{Q}_n \in \mathbb{R}^{(n+1)\times n}$ y $\widehat{R}_n \in \mathbb{R}^{n\times n}$. En este caso se usa el súper-índice "$[n]$" para denotar que el vector $\mathbf{q}_i^{[n]} \in \mathbb{R}^{n+1}$, y está asociado a la matriz T_n.

Como se indicó al principio, T_n denota una secuencia de matrices tridiagonales, esto significa que la siguiente matriz de esta secuencia es,

$$
T_{n+1} = \begin{bmatrix}
a_1 & c_1 & 0 & \dots & \dots & \dots & \dots & 0 \\
b_1 & a_2 & c_2 & 0 & \dots & \dots & \dots & \vdots \\
0 & b_2 & a_3 & c_3 & 0 & \dots & \dots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots\ddots & 0 & \\
\vdots & \ddots & \ddots & 0 & b_{n-2} & a_{n-1} & c_{n-1} & 0 \\
\vdots & \ddots & \ddots & \ddots & 0 & b_{n-1} & a_n & c_n \\
0 & \ddots & \ddots & \ddots & \ddots & 0 & b_n & a_{n+1} \\
0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & b_{n+1}
\end{bmatrix},
$$

la cual se diferencia de T_n solamente porque se agregó una columna y una fila adicional. Este patrón puede representarse de la siguiente forma,

$$
T_{n+1} = \left(\begin{array}{ccc|c}
 & & & 0 \\
 & & & \vdots \\
 & T_n & & 0 \\
 & & & c_n \\
 & & & a_{n+1} \\
\hline
0 & \dots & 0 & b_{n+1}
\end{array}\right),
$$

es decir, la mayor parte de la matriz T_{n+1} corresponde a la matriz T_n. La tarea a resolver es obtener la factorización QR reducida de la matriz T_{n+1} dado que conocemos la factorización QR reducida de la matriz T_n. Para su desarrollo considere la siguiente identidad,

$$
\begin{aligned}
T_{n+1} &= \left(\begin{array}{ccc|c} & & & 0 \\ & & & \vdots \\ & T_n & & 0 \\ & & & c_n \\ & & & a_{n+1} \\ \hline 0 & \dots & 0 & b_{n+1} \end{array}\right) \\
&= \left(\begin{array}{ccc|c} & & & 0 \\ & & & \vdots \\ & \widehat{Q}_n \, \widehat{R}_n & & 0 \\ & & & c_n \\ & & & a_{n+1} \\ \hline 0 & \dots & 0 & b_{n+1} \end{array}\right) \\
&= \widehat{Q}_{n+1} \, \widehat{R}_{n+1} \\
&= \begin{bmatrix} \mathbf{q}_1^{[n+1]} & \mathbf{q}_2^{[n+1]} & \dots & \mathbf{q}_n^{[n+1]} & \mathbf{q}_{n+1}^{[n+1]} \end{bmatrix}
\begin{bmatrix}
r_{1,1} & r_{1,2} & r_{1,3} & \dots & r_{1,n} & r_{1,n+1} \\
0 & r_{2,2} & r_{2,3} & \dots & r_{2,n} & r_{2,n+1} \\
0 & 0 & r_{3,3} & \dots & r_{3,n} & r_{2,n+1} \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
0 & \dots & \dots & 0 & r_{n,n} & r_{n,n+1} \\
0 & \dots & \dots & 0 & 0 & r_{n+1,n+1}
\end{bmatrix}.
\end{aligned}
$$

A partir de las identidades anteriores, usted debe determinar los vectores $\mathbf{q}_i^{[n+1]}$ para $i \in \{1, 2, 3, \dots, n+1\}$ y la última columna de R_{n+1}.

Recuerde que $\mathbf{q}_i^{[n]} \in \mathbb{R}^{n+1}$, pero $\mathbf{q}_i^{[n+1]} \in \mathbb{R}^{n+2}$. Es decir, tiene una diferencia de una unidad en la dimensión del espacio vectorial asociado. Lo importante es que esta diferencia puede ser resuelta convenientemente para obtener las primeras n columnas de la matriz $\widehat{Q}_{n+1}$.

- *Hint 1: You may consider useful this relationship:* $\mathbf{q}_j^{[n+1]} = [(\mathbf{q}_j^{[n]})^T, 0]^T$, *for* $j \in \{1, 2, 3, \dots, n\}$ *and* ^{T} *is the transpose operator.*
- *Hint 2: Notice that the coefficients* $r_{i,j}$ *do not have a super index, this means that the only missing part of* R_{n+1} *is actually its last column!*

Preguntas

(a) Explique teóricamente como obtendrá la factorización QR reducida de T_{n+1} a partir de la factorización QR reducida de T_n.

(b) Construya la función "‘incrementalQR"‘ que obtiene la factorización QR reducida de la matriz T_{n+1} a partir de la factorización reducida de T_n. El input de esta función corresponde a Q_n, R_n, a_{n+1}, b_{n+1}, y c_n. IMPORTANTE: Esta función se considerará correcta si obtiene adecuadamente la factorización QR reducida de T_{n+1} a partir de la factorización QR reducida de T_n. Si se obtiene la factorización QR reducida directamente de la matriz T_{n+1}, no se considerará correcta. La importancia de esto es que obtener la factorización QR reducida de esta forma reduce significativamente la cantidad de operaciones elementales requerida, acelerando la computación significativamente.

```
'''
INPUT:
Qn    : (ndarray) Matriz unitaria Q_n de la factorización QR reducida de la matriz
         T_n = Q_n * R_n.
Rn    : (ndarray) Matriz triangular superior R_n de la factorización QR reducida de
         la matriz T_n = Q_n * R_n.
a     : (float)   Coeficiente a_{n+1} de la matriz T_{n+1}.
b     : (float)   Coeficiente b_{n+1} de la matriz T_{n+1}.
c     : (float)   Coeficiente c_n de la matriz T_{n+1}.

OUTPUT:
Q_next: (ndarray) Matriz unitaria Q_{n+1} de la factorización QR reducida de la
          matriz T_{n+1} = Q_{n+1} * R_{n+1}.
```

```
R_next: (ndarray) Matriz triangular superior R_{n+1} de la factorización QR
        reducida de la matriz T_{n+1} = Q_{n+1} * R_{n+1}.
'''
def incrementalQR(Qn, Rn, a, b, c):
    # Your own code.
    Q_next = np.zeros(2, 1) # Remover esta línea si usted implementa esta función.
    R_next = np.zeros(1, 1) # Remover esta línea si usted implementa esta función.
    return Q_next, R_next
```