Ayudantía
SCT - Semana - 07

Transformada discreta de coseno. El Teorema de interpolación de la transformada discreta de coseno indica lo siguiente:

Thm 1. *Sea* $\mathbf{x} = [x_0, x_1, \ldots, x_{n-1}]^T$ *un vector con* n *coeficientes reales. Se define* $\mathbf{y} = [y_0, y_1, \ldots, y_{n-1}]^T = C\,\mathbf{x}$*, donde* C *corresponde a la matriz asociada a la Transformada Discreta de Coseno (DCT del inglés Discrete Cosine Transform) de orden* n*. Entonces la función:*

$$P_n(t) = \frac{1}{\sqrt{n}}\,y_0 + \sqrt{\frac{2}{n}}\sum_{k=1}^{n-1} y_k\,\cos\left(\frac{k\,(2\,t+1)\,\pi}{2\,n}\right),$$

satisface que $P_n(j) = x_j$ *para todo* $j \in \{0, 1, \ldots\, n-1\}$ *y los coeficientes de la matriz* $C \in \mathbb{R}^{n\times n}$ *se define de la siguiente forma:*

$$C_{i,j} = \sqrt{\frac{2}{n}}\,a_i\,\cos\left(\frac{i\,(2\,j+1)\,\pi}{2\,n}\right),$$

para $i \in \{0, 1, \ldots\, n-1\}$ *y* $j \in \{0, 1, \ldots\, n-1\}$*, donde*

$$a_i = \begin{cases} \dfrac{1}{\sqrt{2}}, & si\ i = 0,\\ 1, & si\ i \in \{1, 2, \ldots, n-1\}.\end{cases}$$

Por simplicidad se incluye de forma explícita la matriz C,

$$C = \sqrt{\frac{2}{n}}\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\ \cos\left(\frac{\pi}{2\,n}\right) & \cos\left(\frac{3\,\pi}{2\,n}\right) & \cdots & \cos\left(\frac{(2\,n-1)\,\pi}{2\,n}\right) \\ \cos\left(\frac{2\,\pi}{2\,n}\right) & \cos\left(\frac{6\,\pi}{2\,n}\right) & \cdots & \cos\left(\frac{2\,(2\,n-1)\,\pi}{2\,n}\right) \\ \vdots & \vdots & \vdots & \vdots \\ \cos\left(\frac{(n-1)\,\pi}{2\,n}\right) & \cos\left(\frac{(n-1)\,3\,\pi}{2\,n}\right) & \cdots & \cos\left(\frac{(n-1)\,(2\,n-1)\,\pi}{2\,n}\right) \end{bmatrix}$$

En resumen, podemos concluir que necesitamos evaluar recurrentemente la función $f(x) = \cos(\pi\,x)$ al utilizar la DCT. Más aún, es posible notar que para construir la transformada discreta de coseno para $n = 100$ necesitamos evaluar el coeficiente $C_{n-1,n-1} = \cos\left(\frac{(n-1)\,(2\,n-1)\,\pi}{2\,n}\right)$, lo que es $\cos\left(\frac{19701}{200}\,\pi\right) \approx \cos(309.4625843418625829 54) \approx -0.015707317311820$. Entonces, si consideramos que trabajaremos con vectores de dimensión menor o igual a 100, significa que necesitamos evaluar la función $\cos(\cdot)$ para valores en el intervalo $[0, 100\,\pi]$. Adicionalmente sabemos que,

$$\left|\frac{\mathrm{d}^m f(x)}{\mathrm{d}x^m}\right| = \begin{cases} \pi^m\,|\sin(\pi\,x)|, & \text{si } m \text{ es impar},\\ \pi^m\,|\cos(\pi\,x)|, & \text{si } m \text{ es par}.\end{cases}$$

Preguntas:

(a) Proponga un algoritmo basado en interpolación polinomial que permita evaluar la función $f(x)$ para $x \in [0, 100]$ con la ayuda de $GEN(n)$: un costoso generador de n puntos equispaciados $(\widehat{x}, \widehat{y})$ para la función $cos(x)$ en el intervalo $[0{,}2\pi]$. Debe garantizar un error de interpolación menor o igual a $\varepsilon > 0$ minimizando el costo computaciónal que supone utilizar $GEN(n)$. Usted debe explicar qué se debe hacer para asegurar esto y cómo planea determinar lo que necesite para cumplir con lo solicitado.

Puede considerar útil saber que para una interpolación con puntos equispaciados se cumple que:

$$|\prod_{i=0}^{n}(x - x_i)| = \prod_{i=0}^{n}|x - x_i| \le \frac{n!}{4}h^{n+1}$$

Su respuesta debe:

- Explicar claramente cómo **construirá** su interpolador polinomial.
- Explicar claramente cómo se debe **utilizar** su interpolador polinomial.

(b) Explique cómo puede utilizar la interpolación polinomial anterior para evaluar $\sin(\pi\,x)$ con los recursos que tiene disponibles.

- Hint 1: It is important to recall that the cosine function is periodic, so, what can you do if you want to evaluate it for values greater that $2\,\pi$? The same applies for negative values, but they are not needed for this problem.
- Hint 2: You may find useful to consider the modulus operator, which returns the reminder of the operation. For instance in NumPy we can performe the following computation `np.mod(6.2,2.5)` which returns `1.2`.
- Hint 3: The lowest upper bound should contain the coefficient π^n but not $\pi^{2\,n}$ nor other additional coefficients that makes it larger. In this case n corresponds to the number of points used in the interpolation.