Matriz de la Transformada Discreta de Seno Tipo-I

Los autores de estos apuntes son: Issis Fragoso Martínez, Oscar García Hernández, Egor Maximenko, Juan José Sánchez Meneses.

Herramienta auxiliar: fórmulas para sumas de cosenos

1 Proposición (fórmula para el núcleo de Dirichlet)**.** *Sea* $\theta \in \mathbb{R} \setminus (2\pi\mathbb{Z})$*. Entonces*

$$1 + 2\sum_{k=1}^{n} \cos(k\theta) = \frac{\sin\frac{(2n+1)\theta}{2}}{\sin\frac{\theta}{2}}. \tag{1}$$

Demostración. Primero escribimos la suma original en la forma compleja:

$$1 + 2\sum_{k=1}^{n} \cos(k\theta) = \sum_{k=-n}^{n} e^{ik\theta},$$

luego factorizamos $e^{-in\theta}$ y aplicamos la fórmula para la suma parcial de la serie geométrica:

$$\sum_{k=-n}^{n} e^{ik\theta} = e^{-in\theta}\sum_{k=0}^{2n} e^{ik\theta} = e^{-in\theta}\frac{(e^{i\theta})^{2n+1}-1}{e^{i\theta}-1}$$

y trabajando con exponenciales regresamos a funciones trigonométricas:

$$\begin{aligned} &= \frac{e^{-i(n+1)\theta/2}}{e^{-i\theta/2}}\,\frac{e^{(2n+1)i\theta}-1}{e^{i\theta}-1} = \frac{e^{i(2n+1)\theta/2}-e^{-i(2n+1)\theta/2}}{e^{i\theta/2}-e^{-i\theta/2}} \\ &= \frac{2i\,\sin\frac{(2n+1)\theta}{2}}{2i\,\sin\frac{\theta}{2}} = \frac{\sin\frac{(2n+1)\theta}{2}}{\sin\frac{\theta}{2}}. \end{aligned}$$

□

2 Corolario. *Sea* $m \in \mathbb{Z}$ *tal que* $0 < |m| \le 2n$*. Entonces*

$$1 + 2\sum_{k=1}^{n} \cos\frac{km\pi}{n+1} = (-1)^{m+1}. \tag{2}$$

Demostración. Primero aplicamos (1) con $\alpha = \frac{m\pi}{n+1}$:

$$1 + 2\sum_{k=1}^{n} \cos\frac{km\pi}{n+1} = \frac{\sin\frac{(2n+1)m\pi}{2(n+1)}}{\sin\frac{m\pi}{2(n+1)}}.$$

Luego notamos que

$$\frac{(2n+1)m\pi}{2(n+1)} = \frac{(2n+2-1)m\pi}{2(n+1)} = m\pi - \frac{m\pi}{2(n+1)},$$

usamos las fórmulas $\sin(m\pi+\beta) = (-1)^m\sin(\beta)$ y $\sin(-\beta) = -\sin(\beta)$:

$$\frac{\sin\frac{(2n+1)m\pi}{2(n+1)}}{\sin\frac{m\pi}{2(n+1)}} = \frac{\sin\left(m\pi - \frac{m\pi}{2(n+1)}\right)}{\sin\frac{m\pi}{2(n+1)}} = \frac{(-1)^{m+1}\sin\frac{m\pi}{2(n+1)}}{\sin\frac{m\pi}{2(n+1)}} = (-1)^{m+1}.$$

□

Matriz de la Transformada Discreta de Seno Tipo I

3 Definición. Denotemos por S_n a la siguiente matriz:

$$S_n = \sqrt{\frac{2}{n+1}}\left[\sin\frac{jk\pi}{n+1}\right]_{j,k=1}^{n}.$$

La transformación lineal $\mathbb{R}^n \to \mathbb{R}^n$ definida mediante la regla $x \mapsto S_n x$ se llama la *transformada discreta de seno* (de tipo I).

4 Ejemplo. Para $n = 2$ y $n = 3$,

$$S_2 = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}, \qquad S_3 = \begin{bmatrix} \frac{1}{2} & \frac{\sqrt{2}}{2} & \frac{1}{2} \\ \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\ \frac{1}{2} & -\frac{\sqrt{2}}{2} & \frac{1}{2} \end{bmatrix}.$$

5 Algoritmo (algoritmo para crear la matriz S_n)**.** Escribimos una función en el lenguaje MATLAB para crear la matriz S_n.

```
function S = dstmatrix(n)
   S = sqrt(2.0/(n+1)) * sin(pi * (1:n)' * (1:n) / (n+1));
end
```

La misma función en el lenguaje Python+Numpy:

```
def dstmatrix(n):
   a = linspace(1, n, n)
   return sqrt(2.0 / (n + 1)) * sin(outer(a, a) * pi / (n + 1))
```

Una versión más lenta, sin usar el producto externo:

```
def dstmatrix(n):
   return [[ sqrt(2.0/(n+1))*sin((j*k*pi)/(n+1))\
      for j in range(1,n+1)] for k in range(1,n+1)]
```

Propiedades principales de la matriz S_n

6 Teorema. *La matriz S_n es involutiva, simétrica y ortogonal:*

$$S_n^2 = I_n, \qquad S_n^\top = S_n, \qquad S_n^\top S_n = I_n.$$

Demostración. Sean $p, q \in \{1, \ldots, n\}$. Calculemos la entrada (p, q) de la matriz S_n^2:

$$\begin{aligned}
(S_n^2)_{p,q} &= \frac{2}{n+1} \sum_{k=1}^{n} \sin \frac{pk\pi}{n+1} \sin \frac{kq\pi}{n+1} \\
&= \frac{1}{n+1} \sum_{k=1}^{n} \cos \frac{(p-q)k\pi}{n+1} - \frac{1}{n+1} \sum_{k=1}^{n} \cos \frac{(p+q)k\pi}{n+1} \\
&= \frac{1}{2(n+1)} \left(1 + 2 \sum_{k=1}^{n} \cos \frac{(p-q)k\pi}{n+1} \right) - \frac{1}{2(n+1)} \left(1 + 2 \sum_{k=1}^{n} \cos \frac{(p+q)k\pi}{n+1} \right).
\end{aligned}$$

Si $p = q$, entonces los sumandos en la primera suma son $\cos \frac{(p-q)k\pi}{n+1} = 1$, y para la segunda suma aplicamos (2):

$$(S_n^2)_{p,p} = \frac{1}{2(n+1)} (1 + 2n) - \frac{1}{2(n+1)} (-1)^{2p+1} = 1.$$

Si $p \neq q$, entonces utilizamos (2) para ambos sumandos:

$$(S_n^2)_{p,p} = \frac{1}{2(n+1)} ((-1)^{p-q+1} - (-1)^{p+q+1}) = \frac{1}{2(n+1)} (-1)^{p-q+1} (1 - (-1)^{2q}) = 0.$$

Acabamos de demostrar que $(S_n^2)_{p,q} = \delta_{p,q}$ para cualesquiera $p, q \in \{1, \ldots, n\}$, por lo cual significa que $S_n^2 = I_n$. La propiedad simétrica se cumple por definición, y la propiedad ortogonal se sigue de las dos propiedades anteriores. □

Notamos que hay algoritmos muy rápidos, de orden $O(n \log n)$, para calcular la transformada discreta de seno. En este texto no estudiamos estos algoritmos. La siguiente página muestra el algoritmo trivial que utiliza $O(n^2)$ operaciones.

El siguiente programa calcula la transformada discreta de seno de un vector aleatorio $x \in \mathbb{R}^n$. También se verifica que la norma del vector no cambia al aplicar la transformada.

Programa 1: Calcula el producto de $S_n x$, $x \in \mathbb{R}^n$

```
#include <iostream>
#include <cstdlib>
#include <cmath>
using namespace std;

void printVector(const double *vector, size_t n) {
   cout << "{";
   for (size_t i = 0; i < n; ++i)
      cout << vector[i] << ((i + 1 == n) ? endl : ",\t");
   cout << "}" << endl;
}

double norm(const double *a, size_t n) {
   double N = 0.0;
   for (size_t k = 0; k < n; ++k) N += a[k] * a[k];
   return sqrt(N);
}

void DST(double *b, const double *a, size_t n) {
   double coef = sqrt(2.0 / (n + 1)), t;
   for (size_t i = 0; i < n; ++i) {
      t = (i + 1) * M_PI / (n + 1);
      for (size_t j = 0; j < n; j++)
         b[i] += coef * sin(temp * (j + 1)) * a[j];
   }
}

int main() {
   size_t n = 5;
   double *a = new double [n](), *b = new double [n]();
   srand((unsigned)time(NULL));
   for (size_t i = 0; i < n; ++i)
      a[i] = rand() % n + 1;
   DST(b, a, n);
   cout << "vector a = "; printVector(a, n);
   cout << endl << "vector b = DST(a) = "; printVector(b, n);
   cout << "||a|| = " << norm(a, n) << ",\t";
   cout << "||b|| = " << norm(b, n) << endl;
   delete[] a; delete[] b;
   return EXIT_SUCCESS;
}
```