Transformée de Fourier rapide

Maximilien Dreveton

July 8, 2016

Références Demazure, Peyré

0.1 Recasages

Passe à l'aise 102 Groupe des nombres complexes de module 1. Sous-groupes des racines de l'unité. Applications.

110 Caractère d'un groupe abélien fini et transformée de Fourier discrète. Applications.

155 Endomorphismes diagonalisables en dimension finie. (Dans rapport du jury. Pourquoi ?)

Arnaques

0.2 Le développement

Remarque : La preuve dans Demazure est probablement trop compliquée (TF sur fonctions à valeurs dans un anneau quelconque...) pour finalement pas grand chose. Préférer celle du Peyré, mais en changeant un peu la méthode de présentation.

Définition 1. *Soit N un entier naturel, et f un élément de $\mathbb{C}^N$. On définit la transformée de Fourier de f par*

$$\tilde{f}[j] := \sum_{n=0}^{N-1} f[n] w_n^{jn} \, for \, j = 0, \dots N-1$$

où l'on a posé $w_N = exp(2i\pi/N)$ racine (primitive) N-ième de l'unité.

Proposition 0.1. *A priori, un algorithme naïf de calcul de la TF est en $O(N^2)$. Dans le cas $N = 2^k$, on propose un algo dit FFT en Nlog(N)*

Proof. Idée de la preuve : Supposons dans un premier temps N pair, et décomposons la somme en termes pairs / termes impairs. Il vient :

$$\tilde{f}[j] = \sum_{n=0}^{N/2-1} f[2n] w_N^{2nj} + \sum_{n=0}^{N/2-1} f[2n+1] w_N^{(2n+1)j}$$

Posons $f^0 := \{f[0], f[2], \dots f[N-2]\}$ et $f^1 := \{f[1], f[3], \dots f[N-1]\}$. Alors, en remarquant que $w_n^{2jn} = w_{N/2}^{jn}$ et $w_N^{(2n+1)j} =$, il vient :

$$\tilde{f}[j] = \tilde{f^0}[j] + w_N^j \tilde{f^1}[j]$$
$$\tilde{f}[j + N/2] = \tilde{f^0}[j] + w_N^j \tilde{f^1}[j]$$

pour j variant de 0 à N/2-1.

Ainsi, le calcul de la TF de f (vecteur de N composantes) se calcule comme deux TF de N/2 composantes + un nombre linéaire en N de multiplication par racines n-ième et additions.

Bien sûr, si N/2 est pair, belotte et rebelotte. Le cas le plus sympathique est lorsque N est une puissance de deux, disons $N = 2^k$: on itère k-1 fois et on tombe sur le calcul de 2^{k-1} TF de taille 2. Montrons maintenant que cette manière de faire est efficace. □

Dans tout ce qui suit, on suppose $N = 2^k$

Proposition 0.2. *Le temps de calcul de l'algorithme précédement introduit est en $Nlog_2(N)$.*

Proof. Notons □

0.3 Applications

- Calcul TF d'une série de donnée exp
- Calcul TF d'un produit de convolution
- Calcul TF d'un produit de deux polynômes

On développe cette dernière application. La donnée d'un polynôme de degré N-1 revient en la donnée de ces coefficients dans la base canonique. Pour $a = (a_0, \dots, a_{N-1}) \in \mathbb{K}^N$, on pose $P_a = \sum_{n=0}^{N-1} a_i X^i$. De même, introduisons $b = (b_0, \dots, b_{N-1})$ et P_b.

Le produit $P_a P_b = \sum_{i=0}^{2N-2} (\sum_{r=0}^{i} a_{i-r} b_i) X^i = P_(a * b)$.

La dernière égalité est vicieuse. En effet, le produit de convolution de deux séquences de taille N est une séquence de taille N, or $P_a P_b$ est un polynôme de degré 2N-2 : il y a donc quelque chose qui cloche. En fait, on a sans le dire rajouter des zéros sur les termes n'existant pas dans la deuxième expressions (ie la_{i-r} ou b_r si r trop grand / trop petit n'est pas défini). Regardons de plus près ce qui se passe.