Transformée de Fourier rapide

Léo Gayral

2017-2018

ref : Cormen – Introduction to algorithms, third edition – p.901

Remarque 1. Dans l'espace de Schwarz $\mathcal{S}(\mathbb{R})$, on peut facilement montrer $\widehat{f * g} = \hat{f} \times \hat{g}$. Ceci permet de simplifier un certain nombre de calculs, que ce soit en théorie ou en pratique.

En outre, si on considère deux polynômes $P, Q \in \mathbb{C}[X]$, on peut leur associer des séries $(p_i), (q_j) \in \mathbb{C}^{(\mathbb{Z})}$. Les coefficients de PQ correspondent alors à la série convolée $(p_i) * (q_j)$.

Théorème 1. Soient $P, Q \in \mathbb{C}_{n-1}[X]$. On peut calculer les coefficients du polynôme PQ en $O(n \ln(n))$ opérations.

Démonstration.
Par interpolation de Lagrange on sait que, pour $x_0, \dots, x_{n-1} \in \mathbb{C}$ deux à deux distincts, l'application $\Phi : \begin{array}{ccc} \mathbb{C}_{n-1}[X] & \to & \mathbb{C}^n \\ P = (p_0, \dots, p_{n-1}) & \mapsto & (P(x_0), \dots, P(x_{n-1})) \end{array}$ est un isomorphisme de $\mathbb{C}$-espaces vectoriels.

A isomorphisme près, déterminer les coefficients de PQ équivaut à calculer le produit terme à terme de $\Phi(P) \cdot \Phi(Q)$, en temps linéaire. A priori, calculer $\Phi(P)$ (explicitement) ou $\Phi^{-1}(X)$ (avec les polynômes interpolateurs) se fait en temps $O(n^2)$, ce qui n'apporte pas de gain conséquent par rapport au calcul direct des coefficients, également en temps quadratique.

Cela revient à considérer le produit par une matrice de Vandermonde :

$$\Phi(P) = \begin{pmatrix} 1 & x_1 & \cdots & x_1^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \cdots & x_n^{n-1} \end{pmatrix} \begin{pmatrix} p_0 \\ \vdots \\ p_{n-1} \end{pmatrix}$$

On va maintenant chercher à ajouter de la structure sur le $(x_i) \in \mathbb{C}^n$ afin de réduire le nombre d'opérations nécessaire. Soit w une racine n-ième primitive de l'unité – typiquement $e^{\pm\frac{2i\pi}{n}}$ – et $x_i = w^i$, de sorte que $\{x_i\} = \mathbb{U}_n \cong \mathbb{Z}/n\mathbb{Z}$. On nomme $V(w) = (w^{ij})_{0\leq i,j\leq n-1}$ la matrice de Vandermonde associée.

Dans ce cas, w^{-1} est aussi une racine primitive et $V(w)^{-1} = \frac{1}{n}V(w^{-1})$:

$$\begin{aligned}
\left[V(w)V\left(w^{-1}\right)\right]_{i,j} &= \sum_{k=0}^{n-1} V(w)_{i,k}V\left(w^{-1}\right)_{k,j} \\
&= \sum_{k=0}^{n-1} w^{ik-kj} \\
&= \sum_{k=0}^{n-1} \left(w^{i-j}\right)^k \\
&= n \times \delta_{i,j}
\end{aligned}$$

donc le calcul de $\Phi(P) = V(w)P$ et de $\Phi^{-1}(X) = \frac{1}{n}V\left(w^{-1}\right)X$ devraient idéalement se faire selon la même méthode.

Lorsque n est pair, on remarque que $w^{i+\frac{n}{2}} = -w^i$, ce qui ajoute des symétries dans V : $V(w)_{\frac{n}{2}+i,j} = (-1)^j V(w)_{i,j}$. Pour $i \in [\![0, \frac{n}{2} - 1]\!]$ on définit les vecteurs :

$$\begin{aligned}
u_i &= \sum_{j\equiv 0[2]} w^{ij}X_j = \sum_{k=0}^{\frac{n}{2}-1} w^{i\times 2k}X_{2k} = \sum_{k=0}^{\frac{n}{2}-1} \left(w^2\right)^{ik} X_{2k} \\
v_i &= \sum_{j\equiv 1[2]} w^{ij}X_j = \sum_{k=0}^{\frac{n}{2}-1} w^{i\times(2k+1)}X_{2k+1} = w^i \sum_{k=0}^{\frac{n}{2}-1} \left(w^2\right)^{ik} X_{2k+1}
\end{aligned}$$

de sorte que $(VX)_i = u_i + v_i$ et $(VX)_{\frac{n}{2}+i} = u_i - v_i$.

On définit les vecteurs $X_p = (X_{2k})_{0\leq k\leq \frac{n}{2}-1}$ et $X_i = (X_{2k+1})_{0\leq k\leq \frac{n}{2}-1} \in \mathbb{C}^{\frac{n}{2}}$. Comme w est une racine primitive de $\mathbb{U}_n$, w^2 est une racine primitive de $\mathbb{U}_{\frac{n}{2}}$ et $V\left(w^2\right)$ est bien définie comme précédemment. On pose $Y_p = V\left(w^2\right)X_p$, $Y_i = V\left(w^2\right)X_i$, de sorte que :

$$V(w)X = \begin{pmatrix} Y_p \\ Y_p \end{pmatrix} + \operatorname{diag}\left(w^i, 0 \leq n\right)\begin{pmatrix} Y_i \\ Y_i \end{pmatrix}.$$

Ces calculs se font en temps linéaire, donc lorsqu'on part de $n = 2^m$ on a un algorithme avec une complexité $T(n) = 2 \times T\left(\frac{n}{2}\right) + O(n)$, donc $T(n) = O\left(n\ln(n)\right)$.

Remarquons que dans cet algorithme particulier, il est nécessaire de considérer $n = 2^m$ pour obtenir le bon résultat à la fin, et pas simplement

pour simplifier le calcul de la complexité. Dans un cas général, si on pose $d = \max(\deg(P), \deg(Q))$, alors quitte à se placer sur $\mathbb{C}_{2^{\lceil \log_2(d+1) \rceil}-1}[X]$ on peut bien effectuer les calculs avec une complexité en $O(d \ln(d))$. $\square$