



Polynomprodukt und Fast Fourier Transformation



Polynome

Reelles Polynom p in einer Variablen x

$$p(x) = a_n x^n + \dots + a_1 x^1 + a_0$$

$a_0, \dots, a_n \in R, a_n \neq 0$: **Koeffizienten** von p

Grad von p : höchste Potenz in p ($= n$)

Beispiel:

$$p(x) = 3x^3 - 15x^2 + 18x$$

Menge aller reellen Polynome: $R[x]$

$$p, q \in R[x]$$

$$\begin{aligned} p(x) &= a_n x^n + \dots + a_1 x^1 + a_0 \\ q(x) &= b_n x^n + \dots + b_1 x^1 + b_0 \end{aligned}$$

1. Addition

$$\begin{aligned} p(x) + q(x) &= (a_n x^n + \dots + a_0) + (b_n x^n + \dots + b_0) \\ &= (a_n + b_n) x^n + \dots + (a_1 + b_1) x^1 + (a_0 + b_0) \end{aligned}$$

2. Produkt

$$\begin{aligned} p(x)q(x) &= (a_n x^n + \dots + a_0)(b_n x^n + \dots + b_0) \\ &= c_{2n} x^{2n} + \dots + c_1 x^1 + c_0 \end{aligned}$$

c_i : Welche Monomprodukte haben Grad i ?

$$\Rightarrow c_i = \sum_{j=0}^i a_j b_{i-j} \quad i = 0, \dots, 2n.$$

$$a_{n+1} = \dots = a_{2n} = 0, b_{n+1} = \dots = b_{2n} = 0$$

Polynomring $R[x]$

3. Auswerten an der Stelle x_0 : **Horner-Schema**

$$p(x_0) = (\dots(a_n x_0 + a_{n-1})x_0 + \dots + a_1)x_0 + a_0$$

Zeit: $O(n)$

Repräsentation eines Polynoms

3. Punkt/Wertdarstellung

Interpolationslemma

Jedes Polynom $p(x)$ aus $R[x]$ vom Grad n ist eindeutig bestimmt durch $n+1$ Paare $(x_i, p(x_i))$, mit $i = 0, \dots, n$ und $x_i \neq x_j$ für $i \neq j$

Beispiel:

Das Polynom

$$p(x) = 3x(x-2)(x-3)$$

wird durch die Paare $(0,0)$, $(1,6)$, $(2,0)$, $(3,0)$ eindeutig festgelegt.

Operationen auf Polynomen

Addition:

$$p + q = (x_0, y_0 + z_0), (x_1, y_1 + z_1), \dots, (x_n, y_n + z_n)$$

Zeit: $O(n)$

Produkt:

$$p \cdot q = (x_0, y_0 \cdot z_0), (x_1, y_1 \cdot z_1), \dots, (x_n, y_n \cdot z_n)$$

(Voraussetzung: $n \geq \text{Grad}(pq)$)

Zeit: $O(n)$

Auswerten an der Stelle x' : ??

Umwandeln in Koeffizientendarstellung

(Interpolation)

Polynomprodukt



Berechnung des Produkts zweier Polynome p, q vom Grade $< n$

p, q Grad $n-1$, n Koeffizienten



Auswertung: $x_0, x_1, \dots, x_{2n-1}$

$2n$ Punkt/Wertpaare $(x_i, p(x_i))$ und $(x_i, q(x_i))$



Punktweise Multiplikation

$2n$ Punkt/Wertpaare $(x_i, pq(x_i))$



Interpolation

pq Grad $2n-2$, $2n-1$ Koeffizienten

Ansatz für Divide and Conquer

Idee: (n sei gerade)

$$\begin{aligned} p(x) &= a_0 + a_1x + \dots + a_{n-1}x^{n-1} \\ &= a_0 + a_2x^2 + \dots + a_{n-2}x^{n-2} + \\ &\quad a_1x + a_3x^3 + \dots + a_{n-1}x^{n-1} \\ &= a_0 + a_2x^2 + \dots + a_{n-2}(x^2)^{(n-2)/2} + \\ &\quad x(a_1 + a_3x^2 + \dots + a_{n-1}(x^2)^{(n-2)/2}) \\ &= p_0(x^2) + xp_1(x^2) \end{aligned}$$

$$p_0(x) = a_0 + a_2x + \dots + a_{n-2}x^{(n-2)/2}$$

$$p_1(x) = a_1 + a_3x + \dots + a_{n-1}x^{(n-2)/2}$$

Wähle x_0, \dots, x_{2n-1} , so dass Berechnung von $p(x_k)$ und $p(x_{k+n})$ fast identisch.

Repräsentation von $p(x)$

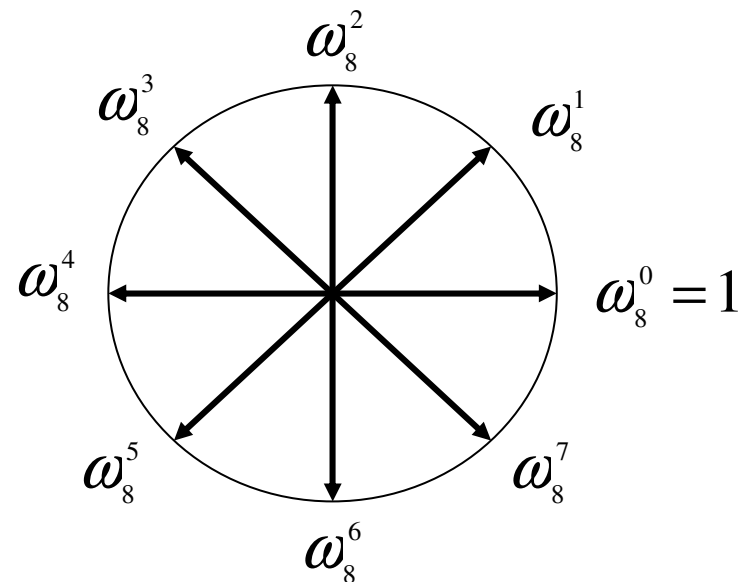
Annahme: $\text{Grad}(p) < n$

**3a. Werte an den n Potenzen der n -ten komplexen
Haupteinheitswurzel** $\omega_n = e^{2\pi i/n}$

$$i = \sqrt{-1} \quad e^{2\pi i} = 1$$

Potenz von ω_n (Einheitswurzeln):

$$1 = \omega_n^0, \omega_n^1, \dots, \omega_n^{n-1}$$



Diskrete Fourier Transformation

Werte von p für die n Potenzen von ω_n legen p eindeutig fest, falls $\text{Grad}(p) < n$.

Diskrete Fourier Transformation (DFT)

$$DFT_n(p) = (p(\omega_n^0), p(\omega_n^1), \dots, p(\omega_n^{n-1}))$$

Beispiel: $n=4$

$$e^{ix} = \cos x + i \sin x$$

$$\omega_4^0 = e^{0i} = \cos(0) + i \sin(0) = 1$$

$$\omega_4^1 = e^{2\pi i/4} = \cos(\pi/2) + i \sin(\pi/2) = i$$

$$\omega_4^2 = (e^{2\pi i/4})^2 = \cos \pi + i \sin \pi = -1$$

$$\omega_4^3 = (e^{2\pi i/4})^3 = \cos(3\pi/2) + i \sin(3\pi/2) = -i$$

Polynomprodukt

Berechnung des Produkts zweier Polynome p, q vom Grade $< n$

p, q Grad $n-1$, n Koeffizienten



Auswertung: $\omega_{2n}^0, \omega_{2n}^1, \dots, \omega_{2n}^{2n-1}$

$2n$ Punkt/Wertpaare $(\omega_{2n}^i, p(\omega_{2n}^i))$ und $(\omega_{2n}^i, q(\omega_{2n}^i))$



Punktweise Multiplikation

$2n$ Punkt/Wertpaare $(\omega_{2n}^i, pq(\omega_{2n}^i))$



Interpolation

pq Grad $2n-2$, $2n-1$ Koeffizienten

Eigenschaften der Einheitswurzel

$\omega_{2n}^0, \omega_{2n}^1, \dots, \omega_{2n}^{2n-1}$ bilden eine **multiplikative Gruppe**

Kürzungslemma

Für alle $n > 0$, $0 \leq k \leq n$, und $d > 0$ gilt:

$$\omega_{dn}^{dk} = \omega_n^k$$

Beweis:

$$\omega_{dn}^{dk} = e^{2\pi i dk / (dn)} = e^{2\pi i k / n} = \omega_n^k$$

Folge:

$$\omega_{2n}^n = \omega_2^1 = -1$$

Diskrete Fourier Transformation

$$DFT_n(p) = (p(\omega_n^0), p(\omega_n^1), \dots, p(\omega_n^{n-1}))$$

Fast Fourier Transformation:

Berechnung von $DFT_n(p)$ mittels
eines Divide-and-Conquer Ansatzes

Idee: (n sei gerade)

$$\begin{aligned} p(x) &= a_0 + a_1x + \dots + a_{n-1}x^{n-1} \\ &= a_0 + a_2x^2 + \dots + a_{n-2}x^{n-2} + \\ &\quad a_1x + a_3x^3 + \dots + a_{n-1}x^{n-1} \\ &= a_0 + a_2x^2 + \dots + a_{n-2}(x^2)^{(n-2)/2} + \\ &\quad x(a_1 + a_3x^2 + \dots + a_{n-1}(x^2)^{(n-2)/2}) \\ &= p_0(x^2) + xp_1(x^2) \end{aligned}$$

$$p_0(x) = a_0 + a_2x + \dots + a_{n-2}x^{(n-2)/2}$$

$$p_1(x) = a_1 + a_3x + \dots + a_{n-1}x^{(n-2)/2}$$

Diskrete Fourier Transformation



Auswertung für $k = 0, \dots, n-1$

$$p(\omega_n^k) = p_0((\omega_n^k)^2) + \omega_n^k p_1((\omega_n^k)^2) = \begin{cases} p_0(\omega_{n/2}^k) + \omega_n^k p_1(\omega_{n/2}^k), \\ \text{falls } k < n/2 \\ p_0(\omega_{n/2}^{k-n/2}) + \omega_n^k p_1(\omega_{n/2}^{k-n/2}) \\ \text{falls } k \geq n/2 \end{cases}$$

$$\begin{aligned} DFT_n(p) &= (p_0(\omega_{n/2}^0), \dots, p_0(\omega_{n/2}^{n/2-1}), p_0(\omega_{n/2}^0), \dots, p_0(\omega_{n/2}^{n/2-1})) \\ &+ (\omega_n^0 p_1(\omega_{n/2}^0), \dots, \omega_n^{n/2-1} p_1(\omega_{n/2}^{n/2-1}), \omega_n^{n/2} p_1(\omega_{n/2}^0), \dots, \omega_n^{n-1} p_1(\omega_{n/2}^{n/2-1})) \end{aligned}$$

Beispiel:

$$p(\omega_4^0) = p_0(\omega_2^0) + \omega_4^0 p_1(\omega_2^0)$$

$$p(\omega_4^1) = p_0(\omega_2^1) + \omega_4^1 p_1(\omega_2^1)$$

$$p(\omega_4^2) = p_0(\omega_2^0) + \omega_4^2 p_1(\omega_2^0)$$

$$p(\omega_4^3) = p_0(\omega_2^1) + \omega_4^3 p_1(\omega_2^1)$$

Berechnung von DFT_n

$$DFT_n(p) = (p(\omega_n^0), p(\omega_n^1), \dots, p(\omega_n^{n-1}))$$

Einfachster Fall: $n = 1$ ($\text{Grad}(p) = n - 1 = 0$)

$$DFT_1(p) = a_0$$

Sonst :

Divide:

Teile p in p_0 und p_1 auf

Conquer:

Berechne $DFT_{n/2}(p_0)$ und $DFT_{n/2}(p_1)$ rekursiv

Merge:

Berechne für $k = 0, \dots, n-1$:

$$DFT_n(p)_k = (DFT_{n/2}(p_0), DFT_{n/2}(p_0))_k + \omega_n^k \cdot (DFT_{n/2}(p_1), DFT_{n/2}(p_1))_k$$

Eine kleine Verbesserung



$$p(\omega_n^k) = \begin{cases} p_0(\omega_{n/2}^k) + \omega_n^k p_1(\omega_{n/2}^k), \\ \text{falls } k < n/2 \\ p_0(\omega_{n/2}^{k-n/2}) + \omega_n^k p_1(\omega_{n/2}^{k-n/2}), \\ \text{falls } k \geq n/2 \end{cases}$$
$$= \begin{cases} p_0(\omega_{n/2}^k) + \omega_n^k p_1(\omega_{n/2}^k), \\ \text{falls } k < n/2 \\ p_0(\omega_{n/2}^{k-n/2}) - \omega_n^{k-n/2} p_1(\omega_{n/2}^{k-n/2}), \\ \text{falls } k \geq n/2 \end{cases}$$

Also falls $k < n/2$:

$$p_0(\omega_{n/2}^k) + \omega_n^k p_1(\omega_{n/2}^k) = p(\omega_n^k)$$
$$p_0(\omega_{n/2}^k) - \omega_n^k p_1(\omega_{n/2}^k) = p(\omega_n^{k+n/2})$$

Eine kleine Verbesserung



Beispiel:

$$p(\omega_4^0) = p_0(\omega_2^0) + \omega_4^0 p_1(\omega_2^0)$$

$$p(\omega_4^1) = p_0(\omega_2^1) + \omega_4^1 p_1(\omega_2^1)$$

$$p(\omega_4^2) = p_0(\omega_2^0) - \omega_4^0 p_1(\omega_2^0)$$

$$p(\omega_4^3) = p_0(\omega_2^1) - \omega_4^1 p_1(\omega_2^1)$$

Fast Fourier Transformation

Algorithmus *FFT*

Input: Ein Array a mit den n Koeffizienten eines Polynoms p
und $n = 2^k$

Output: $DFT_n(p)$

1. **if** $n = 1$ **then** /* p ist konstant */
2. **return** a
3. $d^{[0]} = FFT([a_0, a_2, \dots, a_{n-2}], n/2)$
4. $d^{[1]} = FFT([a_1, a_3, \dots, a_{n-1}], n/2)$
5. $\omega_n = e^{2\pi i/n}$
6. $\omega = 1$
7. **for** $k = 0$ **to** $n/2 - 1$ **do** /* $\omega = \omega_n^k$ */
8. $d_k = d_k^{[0]} + \omega \cdot d_k^{[1]}$
9. $d_{k+n/2} = d_k^{[0]} - \omega \cdot d_k^{[1]}$
10. $\omega = \omega_n \cdot \omega$
11. **return** d

FFT : Beispiel

$$p(x) = 3x^3 - 15x^2 + 18x + 0$$

$$a = [0, 18, -15, 3]$$

$$a^{[0]} = [0, -15] \quad a^{[1]} = [18, 3]$$

$$\begin{aligned} FFT([0, -15], 2) &= (FFT([0],1) + FFT([-15],1), \quad FFT([0],1) - FFT([-15],1)) \\ &= (-15, 15) \end{aligned}$$

$$\begin{aligned} FFT([18, 3], 2) &= (FFT([18],1) + FFT([3],1), \quad FFT([18],1) - FFT([3],1)) \\ &= (21, 15) \end{aligned}$$

$$k = 0 ; \omega = 1$$

$$d_0 = -15 + 1 * 21 = 6$$

$$d_2 = -15 - 1 * 21 = -36$$

$$k = 1 ; \omega = i$$

$$d_1 = 15 + i * 15$$

$$d_3 = 15 - i * 15$$

$$FFT(a, 4) = (6, 15+15i, -36, 15-15i)$$

$T(n)$ = Zeit um ein Polynom vom Grad $< n$ an den Stellen
 $\omega_{2n}^0, \omega_{2n}^1, \dots, \omega_{2n}^{2n-1}$ auszuwerten.

$$T(1) = O(1)$$

$$\begin{aligned} T(n) &= 2 T(n/2) + O(n) \\ &= O(n \log n) \end{aligned}$$

Polynomprodukt



Berechnung des Produkts zweier Polynome p, q vom Grade $< n$

p, q Grad $n-1$, n Koeffizienten



Auswertung durch FFT: $\omega_{2n}^0, \omega_{2n}^1, \dots, \omega_{2n}^{2n-1}$

$2n$ Punkt/Wertpaare $(\omega_{2n}^i, p(\omega_{2n}^i))$ und $(\omega_{2n}^i, q(\omega_{2n}^i))$



Punktweise Multiplikation

$2n$ Punkt/Wertpaare $(\omega_{2n}^i, pq(\omega_{2n}^i))$



Interpolation

pq Grad $2n-2$, $2n-1$ Koeffizienten

Interpolation

Umrechnung der Punkt/Wert-Darstellung in die Koeffizientendarstellung

Gegeben: $(x_0, y_0), \dots, (x_{n-1}, y_{n-1})$ mit $x_i \neq x_j$, für alle $i \neq j$

Gesucht: Polynom p mit Koeffizienten a_0, \dots, a_{n-1} ,
so dass

$$\begin{aligned} p(x_0) &= a_0 + a_1 x_0 + \dots + a_{n-1} x_0^{n-1} = y_0 \\ p(x_1) &= a_0 + a_1 x_1 + \dots + a_{n-1} x_1^{n-1} = y_1 \\ p(x_2) &= a_0 + a_1 x_2 + \dots + a_{n-1} x_2^{n-1} = y_2 \\ &\vdots \\ p(x_{n-1}) &= a_0 + a_1 x_{n-1} + \dots + a_{n-1} x_{n-1}^{n-1} = y_{n-1} \end{aligned}$$

Matrixschreibweise

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^{n-1} \\ 1 & x_1 & \cdots & x_1^{n-1} \\ & & \vdots & \\ 1 & x_{n-1} & \cdots & x_{n-1}^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

Gleichungssystem

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^{n-1} \\ 1 & x_1 & \cdots & x_1^{n-1} \\ & & \ddots & \\ 1 & x_{n-1} & \cdots & x_{n-1}^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

ist lösbar für $x_i \neq x_j$, für alle $i \neq j$

Spezialfall (hier) : $x_i = \omega_n^i$

Definition:

$$V_n = (\omega_n^{ij})_{i,j}, \quad a = (a_i), \quad y = (y_i)$$

$$V_n a = y \quad \Rightarrow \quad a = V_n^{-1} y$$

Interpolation

Satz

Für alle $0 \leq i, j \leq n - 1$ gilt:

$$(V_n^{-1})_{ij} = \frac{\omega_n^{-ij}}{n}$$

Beweis

$$V_n^{-1} = \left(\frac{\omega_n^{-ij}}{n} \right)_{i,j}$$

zu zeigen:

$$V_n^{-1} V_n = I_n$$

Betrachte Eintrag von $V_n^{-1}V_n$ in Zeile i , Spalte j :

$$(V_n^{-1}V_n)_{ij} =$$

$$\begin{pmatrix} \dots & \dots & \dots \\ \frac{1}{n} & \frac{\omega_n^{-i}}{n} & \dots & \frac{\omega_n^{-i(n-1)}}{n} \\ \dots & \dots & \vdots & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix} \begin{pmatrix} \dots & 1 & \dots \\ \dots & \omega_n^j & \dots \\ \dots & \omega_n^{2j} & \dots \\ \dots & \vdots & \dots \\ \dots & \omega_n^{(n-1)j} & \dots \end{pmatrix}_{ij}$$

$$(V_n^{-1}V_n)_{ij} = \sum_{k=0}^{n-1} \frac{\omega_n^{-ik}}{n} \omega_n^{jk} = \frac{1}{n} \sum_{k=0}^{n-1} \omega_n^{(-i+j)k}$$

Fall 1: $i = j$

$$\frac{1}{n} \sum_{k=0}^{n-1} \omega_n^{(-i+j)k} = \frac{1}{n} \sum_{k=0}^{n-1} \omega_n^{0 \cdot k} = 1$$

Fall 2: $i \neq j$, d.h. $-(n-1) \leq -i + j \leq n-1$ und damit $n \nmid -i + j$:

$$\frac{1}{n} \sum_{k=0}^{n-1} \omega_n^{(-i+j)k} = 0$$

Summationslemma

Für alle $n > 0$, $l \geq 0$ mit $n \nmid l$

$$\sum_{k=0}^{n-1} \omega_n^{lk} = 0$$

Beweis:

$$\sum_{k=0}^{n-1} (\omega_n^l)^k = \frac{(\omega_n^l)^n - 1}{\omega_n^l - 1} = \frac{(\omega_n^n)^l - 1}{\omega_n^l - 1} = 0$$

Interpolation



$$\begin{aligned} a_i &= (V_n^{-1} \mathbf{y})_i \\ &= \left(\frac{1}{n}, \frac{\omega_n^{-i}}{n}, \dots, \frac{\omega_n^{-i(n-1)}}{n} \right) \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} \\ &= \sum_{k=0}^{n-1} y_k \frac{\omega_n^{-ik}}{n} \\ &= \frac{1}{n} \sum_{k=0}^{n-1} y_k (\omega_n^{-i})^k \end{aligned}$$

Interpolation



$$a = \frac{1}{n} \left(\sum_{k=0}^{n-1} y_k (\omega_n^{-0})^k, \sum_{k=0}^{n-1} y_k (\omega_n^{-1})^k, \dots, \sum_{k=0}^{n-1} y_k (\omega_n^{-(n-1)})^k \right)$$

$$r(x) = y_0 + y_1 x + y_2 x^2 + \dots + y_{n-1} x^{n-1}$$

$$a = \frac{1}{n} (r(\omega_n^{-0}), r(\omega_n^{-1}), \dots, r(\omega_n^{-(n-1)}))$$

$$a = \frac{1}{n} (r(\omega_n^{-0}), r(\omega_n^{-1}), \dots, r(\omega_n^{-(n-1)}))$$

$$a = \frac{1}{n} (r(\omega_n^n), r(\omega_n^{n-1}), \dots, r(\omega_n^1)) \quad \text{denn } \omega_n^n = 1$$

$$a_i = \frac{1}{n} (DFT_n(r))_{n-i} \quad (i \neq 0)$$

$$a_0 = \frac{1}{n} (DFT_n(r))_0$$

Polynomprodukt durch FFT

Berechnung des Produkts zweier Polynome p, q vom Grade $< n$

p, q Grad $n-1$, n Koeffizienten

Auswertung durch FFT: $\omega_{2n}^0, \omega_{2n}^1, \dots, \omega_{2n}^{2n-1}$

$2n$ Punkt/Wertpaare $(\omega_{2n}^i, p(\omega_{2n}^i))$ und $(\omega_{2n}^i, q(\omega_{2n}^i))$

Punktweise Multiplikation

$2n$ Punkt/Wertpaare $(\omega_{2n}^i, pq(\omega_{2n}^i))$

Interpolation durch FFT

pq Grad $2n-2$, $2n-1$ Koeffizienten