Matrix and polynomial computation - ctd

16 septembre 2020

Overview

Reminder: multipoint evaluation

Two polynomials of degree at most n in $\mathsf{R}[x]$ are multiplied in at most $\mathsf{M}(n)$ arithmetic operations

Given $P \in \mathsf{K}[x]$, $\deg P < n$,
and $a_0, a_1, \ldots, a_{n-1} \in K$, compute $P(a_0), P(a_1), \ldots P(a_{n-1})$

Theorem: P can be evaluated at n points in $O(\mathsf{M}(n) \log n) = \tilde{O}(n)$ arithmetic operations.

Structured matrices

Multipoint evaluation versus interpolation?

[Bostan & Schost 2004]

With $a_i \neq a_j$ when $i \neq j$,

given $b_0, b_1, \ldots, b_{n-1}$
compute P of degree less than n such that $P(a_0) = b_0, \ldots, P(a_{n-1}) = b_{n-1}$

Theorem: A straight-line program of size L for multipoint evaluation can be transformed into a program for interpolation of size $O(L) + O(\mathsf{M}(n))$

Vandermonde matrix: evaluation

$$V \times P = B$$

$$= \begin{bmatrix} 1 & a_1 & {a_1}^2 & \dots & {a_1}^{n-1} \\ 1 & a_2 & {a_2}^2 & \dots & {a_2}^{n-1} \\ 1 & a_3 & {a_3}^2 & \dots & {a_3}^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & a_{n-1} & {a_{n-1}}^2 & \dots & {a_{n-1}}^{n-1} \end{bmatrix} \times \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ P_{n-1} \end{bmatrix} = \begin{bmatrix} P(a_0) \\ P(a_1) \\ P(a_2) \\ \vdots \\ P(a_{n-1}) \end{bmatrix}$$

Vandermonde matrix: evaluation

$$V \times P = B$$

$$= \begin{bmatrix} 1 & a_1 & {a_1}^2 & \dots & {a_1}^{n-1} \\ 1 & a_2 & {a_2}^2 & \dots & {a_2}^{n-1} \\ 1 & a_3 & {a_3}^2 & \dots & {a_3}^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & a_{n-1} & {a_{n-1}}^2 & \dots & {a_{n-1}}^{n-1} \end{bmatrix} \times \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ P_{n-1} \end{bmatrix} = \begin{bmatrix} P(a_0) \\ P(a_1) \\ P(a_2) \\ \vdots \\ P(a_{n-1}) \end{bmatrix}$$

Vandermonde inverse: interpolation

$$P = V^{-1} \times B$$

Hankel matrix: Newton's sums

$$V = \begin{bmatrix} 1 & a_1 & {a_1}^2 & \dots & {a_1}^{n-1} \\ 1 & a_2 & {a_2}^2 & \dots & {a_2}^{n-1} \\ 1 & a_3 & {a_3}^2 & \dots & {a_3}^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & a_{n-1} & {a_{n-1}}^2 & \dots & {a_{n-1}}^{n-1} \end{bmatrix}$$

$$V^{\mathsf{T}} \times V = H = \begin{bmatrix} 1 & \sum_i a_i & \sum_i {a_i}^2 & \dots & \sum_i {a_i}^{n-1} \\ \sum_i a_i & \sum_i {a_i}^2 & \dots & & \sum_i {a_i}^n \\ \sum_i {a_i}^2 & \dots & & & \\ \vdots & & & & \vdots \\ \sum_i {a_i}^{n-1} & \sum_i a_i^n & \dots & & \sum_i a_i^{2(n-1)} \end{bmatrix}$$

Hankel matrix: Newton's sums

Characteristic greater than n

$$f(x) = (x - a_0)(x - a_1)\dots(x - a_{n-1})$$

$$\frac{\hat{f}'(x)}{\hat{f}(x)} = \sum_{i \geq 0} (\sum_k a_k^i) x^i$$

Exercise: from the partial fraction decomposition of $xf'(x)/f(x)$

Multipoint evaluation $\Longrightarrow$ Interpolation

$V^{\mathsf{T}}V = H$ hence $V^{-1} = H^{-1}V^{\mathsf{T}}$

Hankel matrix inversion: see subsequent courses

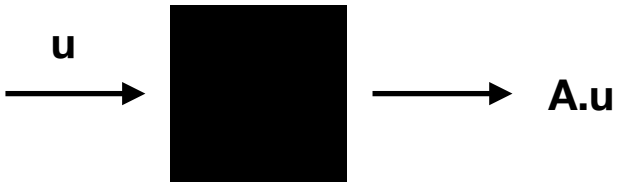
Transposition ?

Black box matrices

[Kaltofen & Trager 1990]

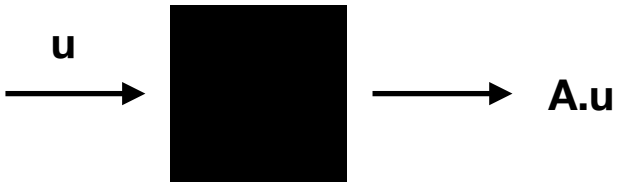
[Kaltofen & Lakshman 1988]

Given

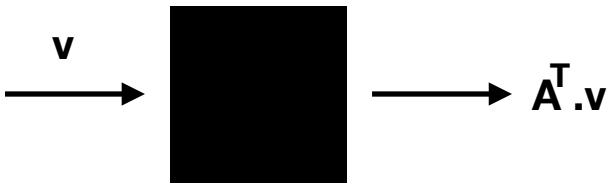


[Kaltofen & Lakshman 1988]

Given



how to generate



Transposition principle

$$A \cdot u \implies v^{\mathsf{T}} \cdot A \cdot u$$

$$f(u_1, \ldots, u_n) = \begin{bmatrix} v_1 & v_2 & \ldots & v_n \end{bmatrix} \cdot A \cdot \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$$

$$f(u_1, \ldots, u_n) = \sum_j \sum_i v_i \, a_{ij} \, u_j$$

$$\partial_{u_j} = \sum_i v_i a_{ij} = (v^{\mathsf{T}} \cdot A)_j$$

Transposition principle

$$A \cdot u \implies v^{\mathsf{T}} \cdot A \cdot u$$

$$f(u_1, \ldots, u_n) = \begin{bmatrix} v_1 & v_2 & \ldots & v_n \end{bmatrix} \cdot A \cdot \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$$

$$f(u_1, \ldots, u_n) = \sum_j \sum_i v_i \; a_{ij} \; u_j$$

$$\partial_{u_j} = \sum_i v_i a_{ij} = (v^{\mathsf{T}} \cdot A)_j$$

Theorem: If a linear map (invertible) can be computed by a program of length L, then the transpose map can be computed by a program of length L

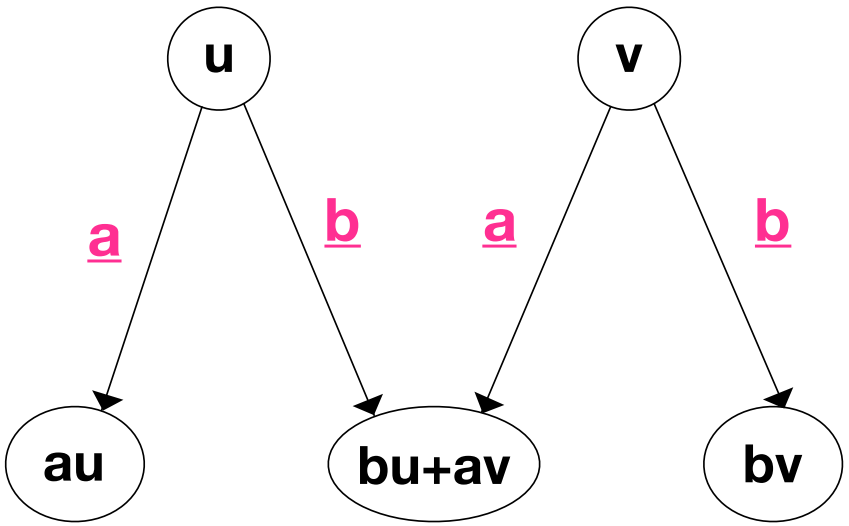
(NB: linear program or transformation required)

$$(a+bx)*(u+vx) = (au)+(bu+av)x+(bv)x^2$$

$$\begin{bmatrix} a & 0 \\ b & a \\ 0 & b \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} au \\ bu+av \\ bv \end{bmatrix}$$

$$(a + bx) * (u + vx) = (au) + (bu + av)x + (bv)x^2$$

$$\begin{bmatrix} a & 0 \\ b & a \\ 0 & b \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} au \\ bu + av \\ bv \end{bmatrix}$$



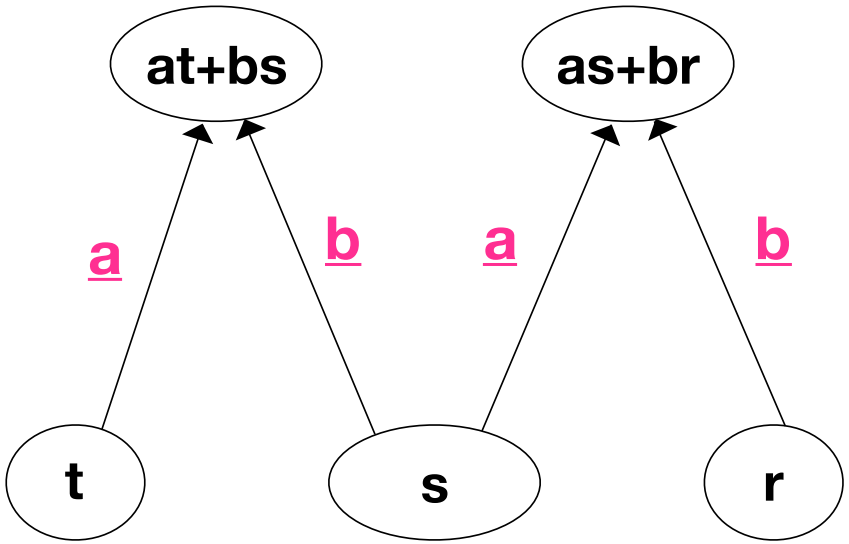
$$\begin{bmatrix} t & s & r \end{bmatrix} \begin{bmatrix} a & 0 \\ b & a \\ 0 & b \end{bmatrix} = \begin{bmatrix} at + bs \\ as + br \end{bmatrix}$$

$$\begin{bmatrix} t & s & r \end{bmatrix} \begin{bmatrix} a & 0 \\ b & a \\ 0 & b \end{bmatrix} = \begin{bmatrix} at+bs \\ as+br \end{bmatrix}$$

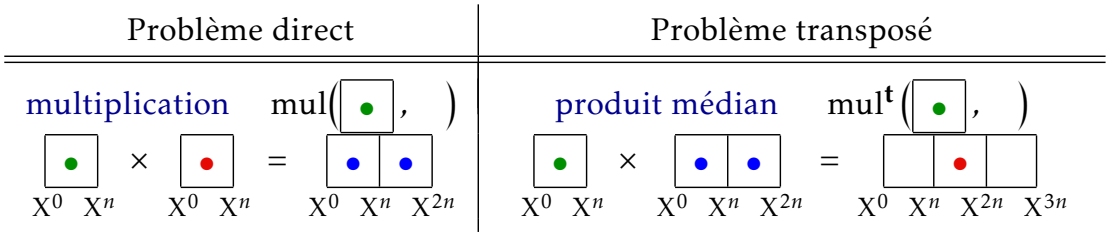
$$(a+bx) * (r+sx+tx^2) = (ar) + (as+br)x + (at+bs)x^2 + (bt)x^3$$

$$\begin{bmatrix} t & s & r \end{bmatrix} \begin{bmatrix} a & 0 \\ b & a \\ 0 & b \end{bmatrix} = \begin{bmatrix} at+bs \\ as+br \end{bmatrix}$$

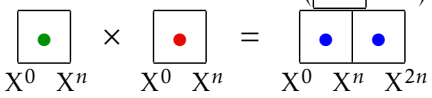
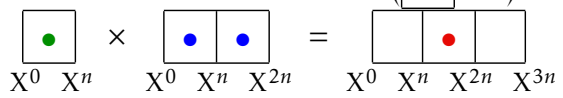
$$(a+bx) * (r+sx+tx^2) = (ar) + (as+br)x + (at+bs)x^2 + (bt)x^3$$



Transposition principle dictionary for univariate polynomials



Transposition principle dictionary for univariate polynomials

Problème direct	Problème transposé
multiplication $\mathrm{mul}(\square\bullet, \square\bullet)$ 	produit médian $\mathrm{mul}^{\mathbf{t}}(\square\bullet, \square\bullet)$ 
division euclidienne $A \mapsto A \bmod P$	extension de récurrences $(a_0, \ldots, a_{n-1}) \mapsto (a_0, \ldots, a_{2n-1})$
évaluation multipoint $P \mapsto (P(a_0), \ldots, P(a_{n-1}))$	sommes de Newton pondérées $(p_0, \ldots, p_{n-1}) \mapsto (\sum p_i, \ldots, \sum p_i a_i^{n-1})$
interpolation (systèmes de Vandermonde)	décomposition en éléments simples (systèmes de Vandermonde transposés)
décalage de polynômes $P(X) \mapsto P(X+1)$	évaluation de factorielles descendantes $P = \sum a_i X^{\underline{i}} \mapsto (P(0), \ldots, P(n-1))$
extrapolation sur $0, 1, 2, \ldots$	division modulo $(X-1)^n$
q-décalage	évaluation de q-factorielles descendantes
composition modulaire	projection des puissances

Randomization

Monte Carlo algorithm (see also BPP class, bounded-error)

Incorrect output with a certain (small) probability

[Freivalds 1979] [Kaltofen, Nehring & Saunders 2011]

Given A, B, P in $\mathsf{K}^{n\times n}$ decide whether $A \times B = P$?

$s \in \mathsf{K}$, $u = [1\ s\ s^2\ \ldots\ s^{n-1}]^{\mathsf{T}} \in \mathsf{K}^n$

check whether $A \times (B \times u) = P \times u$ **?**

Random bits $+ O(n^2)$ operations

Application: Given an algorithm that reduces to matrix multiplication, run the algorithm but check the matrix products instead of computing them

Las Vegas algorithm (see also ZPP class, zero error)

Returns the correct result or "fail"

Exercise - Continued

$A \in \mathsf{K}^{n \times n}$, splitting with $(n/2) \times (n/2)$ invertible blocks

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

$$\begin{bmatrix} \mathrm{I} & 0 \\ -CA^{-1} & \mathrm{I} \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \mathrm{I} & -A^{-1}B \\ & \mathrm{I} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & D - CA^{-1}B \end{bmatrix}$$

$$M^{-1} = R \cdot \begin{bmatrix} A^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \cdot L$$

Given $A \in \mathsf{K}^{n \times n}$ invertible the elimination may fail if some block is singular

Ensure non singularity of the blocks involved ?

Algebraic preconditioning

- Choose a random $U \in \mathsf{K}^{n \times n}$, such that computing $A \times U$ is cheap
- Solve the problem with input $A \times U$
- Deduce the answer for A
  eg by inverting U for computing A^{-1}

Given $A \in \mathsf{K}^{n \times n}$ invertible the elimination may fail if some block is singular

Ensure non singularity of the blocks involved ?

Algebraic preconditioning

- Choose a random $U \in \mathsf{K}^{n \times n}$, such that computing $A \times U$ is cheap
- Solve the problem with input $A \times U$
- Deduce the answer for A
  eg by inverting U for computing A^{-1}

Schwartz–Zippel lemma

The determinant of an $r \times r$ submatrix of $A \times U$ is a polynomial in the entries of U, **of degree** r:

$$P(u_{11}, \ldots, u_{ij}, \ldots, u_{nn}) \neq 0 \in \mathsf{K}[u_{11}, \ldots, u_{ij}, \ldots, u_{nn}]$$

Theorem: Given $P(x_1, \ldots, x_l) \neq 0$ polynomial of degree d, $\alpha_1, \ldots, \alpha_l$ chosen uniformly and independently from a finite subset S of K, one has

$$\text{Prob}\{\ P(\alpha_1, \ldots, \alpha_l) \neq 0 \in \mathsf{K}\} \geq 1 - \frac{d}{\text{card}(S)}$$

NB: does not depend on the number of variables

$P(x_1, \dots, x_\ell)$ $\qquad \sigma = \mathrm{card}(S)$

By induction : $\leq \sigma^{\ell-1} d$ zeros in S^ℓ

- one variable $\ell = 0$, d zeros
- for ℓ

$$P(x_1, \dots, x_\ell) = q_0 + q_1(x_1, \dots x_{\ell-1}) + \dots + \underbrace{q_k(x_1, \dots x_{\ell-1})}_{d^\circ d-k} x_\ell^k$$

$P(x_1, \dots, x_\ell)$ $\quad \sigma = \text{card}(S)$

By induction: $\leq \sigma^{\ell-1} d$ zeros in S^ℓ

- one variable $\ell = 0$, d zeros
- for ℓ

$$P(x_1, \dots, x_\ell) = q_0 + q_1(x_1 \cdots x_{\ell-1}) + \cdots + \underbrace{q_k(x_1 \cdots x_{\ell-1})}_{d^\circ d-k} x_\ell^k$$

Two types of zeros:

a) $q_k(x_1 \cdots x_{\ell-1}) = 0$ at most $\sigma^{\ell-2}(d-k)$

zero for P possibly for every zero of q_k and all values for x_ℓ

$$\longrightarrow \sigma^{\ell-2}(d-k) \times \sigma = (d-k)\sigma^{\ell-1}$$

b) $q_k(x_1 \cdots x_{\ell-1}) \neq 0$ then in one variable of deg k

for at most $\sigma^{\ell-1}$ choices

$$\Rightarrow (d-k)\sigma^{\ell-1} + k\sigma^{\ell-1} = d s^{\ell-1}$$