Techniques algébriques

Objectif : Utiliser des résultats (simples) d'algèbre et théorie des nombres pour concevoir et analyser des algorithmes probabilistes.

Structure commune : Calcul d'empreintes (fingerprinting) pour passer d'objets de grande taille à des objets "petits".

1. Algorithme de Freivalds (1977)

Entrée Trois matrices A, B, C de taille $n \times n$ à coefficients dans un corps ($\mathbb{Q}$, $\mathbb{F}_p$, ...)

Sortie Vrai/Faux : est-ce que $A \times B = C$

1. Choisir aléatoirement un vecteur r de bits, de taille n
2. Calculer $A \times (B \times r)$ et $C \times r$
3. Répondre "Vrai" si les résultats sont égaux, "Faux" sinon

Exemple $A = \begin{bmatrix} 2 & 3 \\ 3 & 4 \end{bmatrix}$ $B = \begin{bmatrix} 1 & 0 \\ 1 & 2 \end{bmatrix}$ $C = \begin{bmatrix} 6 & 5 \\ 9 & 7 \end{bmatrix}$

- $r = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $A \times (B \times r) = \begin{bmatrix} 2 & 3 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 11 \\ 15 \end{bmatrix} = C \times r$
- $r = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $A \times (B \times r) = \begin{bmatrix} 2 & 3 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \end{bmatrix} \neq \begin{bmatrix} 6 \\ 9 \end{bmatrix} = C \times r$

Complexité 3 produits matrice x vecteur : $\Theta(n^2)$.

↳ le produit de matrices se calcule en $\Theta(n^3)$ par l'algo naïf, $\Theta(n^{2,8\ldots})$ par l'algo de Strassen et $\Theta(n^{2,3728639\ldots})$ pour le meilleur algo connu (Le Gall 2014).

⇒ Freivalds permet de vérifier rapidement et simplement.

Correction Si $A \times B = C$, alors $A \times (B \times r) = (A \times B) \times r = C \times r$.
Donc l'algo est correct si $AB = C$.

Lemme Si $A \times B \neq C$, l'algo répond "Vrai" avec proba $\leq \frac{1}{2}$.

Preuve Soit $D = AB - C \neq 0$. On écrit $D = (D_{ij})_{i,j}$ et on fixe un vecteur $r = \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}$.

Alors $D \cdot r = \begin{pmatrix} d_1 \\ \vdots \\ d_n \end{pmatrix}$ où $d_i = \sum_{k=1}^{n} D_{ik} r_k$.

Soit j tq $D_{ij} \neq 0$. Alors $d_i = D_{ij} r_j + y$.

$$\mathbb{P}[d_i = 0] = \underbrace{\mathbb{P}[d_i = 0 \mid y = 0]}_{\mathbb{P}[r_j = 0] = 1/2} \mathbb{P}[y = 0] + \underbrace{\mathbb{P}[d_i = 0 \mid y \neq 0]}_{\mathbb{P}[r_j = 1] = 1/2} \mathbb{P}[y \neq 0]$$

$$= \tfrac{1}{2}\mathbb{P}[y = 0] + \tfrac{1}{2}\mathbb{P}[y \neq 0] = \tfrac{1}{2}.$$

$$\Rightarrow \mathbb{P}[d = 0] = \mathbb{P}\left[\bigwedge_{i=1}^{n} d_i = 0\right] \leq \mathbb{P}[d_i = 0] \leq \tfrac{1}{2}. \quad \square$$

Conclusion L'algo de Freivalds permet de tester une égalité entre 2 matrices en $\Theta(n^2)$ opérations sans calculer explicitement les matrices.

Il ne peut se tromper que si l'égalité n'a pas lieu, avec proba $\leq 1/2$. En répétant k fois, proba d'erreur $\leq 1/2^k$.

Remarque $A\times(B\times r)$ et $C\times r$ sont les empreintes de $A\times B$ et C.

Fausse bonne idée Tirer aléatoirement r dans un ensemble plus grand: par exemple, $r \in S^n$ où S est un ensemble quelconque. Alors $\mathbb{P}[\text{erreur}] \leq 1/|S|$.

↳ Avec $n\cdot\log|S|$ bits, on atteint $1/|S|$. Si on prend $n\cdot\log|S|$ bits dans l'algo. de Freivalds, on obtient une borne $1/2^{\log|S|} = 1/|S|$. → Pareil!

Oui mais: multiplications plus coûteuses!

2. Egalité d'entiers — Application à la recherche de motifs

Exemple Vous avez (légalement !) téléchargé votre livre préféré d'algos probabilistes. Vous voulez vérifier son intégrité. Comment faire ?

Modélisation $A = a_0 \dots a_N$ et $B = b_0 \dots b_N$ sont 2 chaînes de caractères sur un alphabet Σ de taille β. On veut tester l'égalité $A \stackrel{?}{=} B$ avec le moins d'échanges possibles.

Idée 1. Voir A et B comme des entiers en base β : on numérote les éléments de Σ de 0 à $\beta-1$, et on définit $n_A = \sum_{i=0}^{N} a_i \beta^i$ et $n_B = \sum_{i=0}^{N} b_i \beta^i$.

2. On choisit un nombre premier p aléatoirement et on calcule $n_A \bmod p$ et $n_B \bmod p$.

3. On prétend que $A = B$ si $n_A = n_B \bmod p$.

Questions Combien de bits échangés ? Quelle garantie sur le résultat ?

Théorème . Si $A = B$, l'algo fonctionne à tous les coups.
. Si $A \neq B$ et que p est choisi $\leq tN \log(tN)$,
la proba. de mauvaise réponse est $\leq 1/t$
. le nombre de bits échangés est $\Theta(\log N + \log t)$.

Preuve . $n_A \bmod p = n_B \bmod p$ ssi p divise $n_A - n_B$
. le nb de diviseurs premiers de $n_A - n_B$ est $\leq N$
↳ $n_A, n_B \leq 2^{N+1}$, donc $n_A - n_B < 2^{N+1}$. Or chaque diviseur premier est ≥ 2. Donc $2^{\#\text{diviseurs}} \leq n_A - n_B < 2^{N+1}$.

. Thm des nombres premiers : $\pi(n) \sim n/\log n$

↳ $\mathbb{P}[p \text{ divise } n_A - n_B] \leq \dfrac{\#\text{ diviseurs de } n_A - n_B}{\#\text{ premiers} \leq tN \log(tN)}$

$$\leq \frac{N}{\pi(tN \log tN)} \leq \Theta(1/t).$$

□

⇒ Avec $t = N$: on peut tester l'égalité avec $\Theta(\log N)$ bits échangés, avec proba d'erreur $\leq \Theta(1/N)$.

(+ répétitions ⇝ $\Theta(1/N^k)$)

Application Recherche de motif dans un texte.

Entrée Texte $X = x_1 \cdots x_n$, motif $Y = y_1 \cdots y_m$

Sortie(s) Y apparaît-il dans X ? Combien de fois ? Où ?...

Formalisation On note, pour $0 \leq j \leq n-m+1$,

$$X(j) = x_j \cdots x_{j+m-1}.$$

↳ Trouver (compter) j tq $X(j) = Y$.

Algos déterministes Naïf en $\Theta(m \times n)$: on teste tous les j
Optimal en $\Theta(n+m)$ [KMP]

Idée $Y \rightsquigarrow n_y$, $X(j) \rightsquigarrow n_j$, + test $n_y \stackrel{?}{=} n_j [p]$

Algo

1. Tirer p aléatoirement $\leq \tau$ (à préciser)
2. $n_y \leftarrow \sum_{i=1}^{m} y_i \beta^{i-1} \mod p$
3. $n_x \leftarrow \sum_{i=1}^{m} x_i \beta^{i-1} \mod p$
4. Pour $j = 1$ à $n-m+1$:
5. | Si $n_y = n_x$: Renvoyer j
6. | $n_x \leftarrow \beta \times (n_x - \beta^{m-1} x_j) + x_{j+m} \mod p$
7. Renvoyer "Pas trouvé"

. Complexité 2,3. $\Theta(m)$ si opérations mod p coûtent 1. (OK pour p petit)

6. Temps constant : si $\beta = 2$ par exemple, ce ne sont que des décalages et une addition mod p.

↳ $\Theta(n+m)$ si arithmétique en $\Theta(1)$.

. Correction Probabilité de faux positif :

$$\mathbb{P}[n_x = n_y \mid X(j) \neq Y] \leq \frac{m}{\pi(\tau)} = \Theta\left(\frac{m \log \tau}{\tau}\right)$$

↳ Pour tester l'existence d'un motif, faux positif avec proba $\leq \Theta\left(N \frac{m \log \tau}{\tau}\right)$.

$\Rightarrow$ Avec $\tau = N^2 m \log(N^2 m)$ $\rightarrow$ $\leq \Theta(1/N)$.

Variante On a un algo Monte Carlo : très rapide, souvent juste.

Si à chaque égalité $n_x = n_y [p]$, on teste (bêtement) l'égalité des chaînes : l'espérance du nombre de tests est petit si $X(j) \neq Y$.

Solutions : 1. Au premier test, si c'est un faux positif, on branche l'algo naïf. Espérance $\Theta((n+m)(1-1/n) + 1/n \, nm) = \Theta(n+m)$.

2. On change p : après t chgts $\Theta(1/n^t)$ risques.

3. Egalités polynomiales - Complexes produits

- **Polynôme** $12x^4 + 5x^3 + 7x^2 - 3x - 1$

↳ $p(X) = \sum_{i=0}^{d} a_i X^i$: polynôme univarié de degré d.

- **Vérification du produit de polynômes:**

× $p(X) = \sum_{i=0}^{d} a_i X^i \quad q(X) = \sum_{i=0}^{d} b_i X^i$

↳ $p(X) \times q(X) = r(X) = \sum_{i=0}^{2d} c_i X^i$

où $c_i = \sum_{j+k=i} a_j b_k$. → Complexité (op. arithmétiques) → $\Theta(d^2)$ ou $\Theta(d \log d)$

× Algo de vérification:

1. Choisir a aléatoirement dans un ensemble S
2. Evaluer $p(a)$, $q(a)$ et $r(a)$.
3. Accepter si $p(a)\,q(a) = r(a)$.

→ Complexité : d opérations arithmétiques.

→ $\mathbb{P}[\text{échec}] = \mathbb{P}[(p \times q - r)(a) = 0 \mid pq \neq r]$

$\overset{\text{def}}{=} \mathbb{P}[a \text{ racine de } pq - r \mid pq \neq r] = \frac{\#\text{racines}}{|S|}$.

↳ Si $|S| > 2d+1$, $\mathbb{P}[\text{échec}] < 1/2$.

- Problème : les évalués de P, Q, r peuvent être grands !

→ Solution : Faire les calculs modulo p_1 où p_1 est un premier choisi aléatoirement :

Comme $(a \times b) \bmod p_1 = [(a \bmod p_1) \times (b \bmod p_1)] \bmod p_1$
et $(a+b) \bmod p = [(a \bmod p_1) + (b \bmod p_1)] \bmod p_1$

$P(a) \bmod p_1$ peut se calculer en faisant des calculs modulo p_1.

↳ $\mathbb{P}[P(a) \bmod p_1 = 0 \mid P(a) \neq 0] \leq \frac{\#\text{diviseurs premiers de } P}{\text{taille de l'ensemble où on tire } p_1}$.

- Généralisation Considérons $\Pi = \begin{pmatrix} 1 & x_1 & \dots & x_1^{n-1} \\ 1 & x_2 & \dots & x_2^{n-1} \\ & & \vdots & \\ 1 & x_n & \dots & x_n^{n-1} \end{pmatrix}$

la matrice de Vandermonde. Je prétends que $\det(\Pi) = \prod_{i<j} (x_i - x_j)$. Comment vérifier ?

⇒ On veut un analogue du test de polynômes mais pour les polynômes à plusieurs variables.

. **Déf** . $P(X_1,\dots,X_n) = \sum_{j=0}^{k} \beta_j X_1^{\alpha_{1j}} \dots X_n^{\alpha_{nj}}$

Ex exemple : $p(x)=3x^2yz+2xy^2-z^3+2y+1$

. **Addition** $\sum_{j=0}^{k} (\beta_j+\gamma_j) X^{\alpha_j}$

. **Mult** $\sum_{j=0}^{k}\sum_{\ell=0}^{k} \beta_j \gamma_\ell X^{\alpha_{1j}+\alpha_{1\ell}} \dots X_n^{\alpha_{nj}+\alpha_{n\ell}}$.

Ex exemple : $p(x)\times(3xy+z) = 9x^3y^2z + 6x^2y^3 + \dots + 2yz + z$

Thm Soit $P \in \mathbb{K}[X_1,\dots,X_n]$ de degré d et $S \subseteq \mathbb{K}$.

Alors $\mathbb{P}_{r_1,\dots,r_n \in S}[P(r_1,\dots,r_n)=0 \mid P\neq 0] \leq \frac{d}{|S|}$

Preuve par récurrence sur n

$n=1$: $P(X)$ de degré $\leq d$ a au plus d racines.

$$\mathbb{P}[P(r)=0 \mid P\neq 0] = \mathbb{P}[r \in Z(P)] = \frac{|Z(P)|}{|S|} \leq \frac{d}{|S|}$$

$n>1$: $P = \sum_{i=0}^{k} X_1^i P_i(X_2,\dots,X_n)$ avec $P_k \neq 0$.

$$\mathbb{P}[P_k(r_2,\dots,r_n)=0 \mid P_k\neq 0] \leq \frac{d-k}{|S|} \quad [\text{par récurrence}].$$

On définit $q(X_1) = P(X_1, r_2, \dots, r_n)$. $\mathbb{P}[q(r_1)=0] \leq k/|S|$

$$\Rightarrow \mathbb{P}[P(r_1,\dots,r_n)=0 \mid P\neq 0] \leq \mathbb{P}[P_k(r_2,\dots,r_n)=0 \mid P_k\neq 0] + \mathbb{P}[P(X_1,r_2,\dots,r_n)\neq 0 \mid P_k(r_2,\dots,r_n)\neq 0]$$
$$\leq \frac{d-k}{|S|} + k/|S| = d/|S| . \qquad \square$$

- Application : test d'égalité entre polys $PQ \stackrel{?}{=} R$

1. Choisir $r_1, \dots, r_n \in S$.
2. Évaluer $P(r), Q(r)$ et $R(r)$.
3. Accepter si $P(r)Q(r) = R(r)$.

↳ Correction $\mathbb{P}[\text{échec}] \leq \frac{\deg(R)}{|S|}$.

↳ Complexité × Coût de l'évaluation → ça dépend
× Si besoin, calcul modulo un premier.

- Vraie application : Couplage parfait dans un graphe biparti

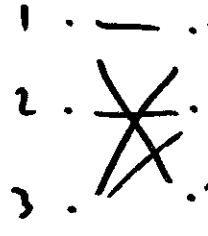
Entrée $G = (U, V, E)$ avec $E \subseteq U \times V$ et $|U| = |V|$.

Sortie $C \subseteq E$ tq $|C| = |U| = |V|$ et

$$\begin{cases} \forall u \in U \ \exists! c \in C \ "u \in c" \\ \forall v \in V \ \exists! c \in C \ "v \in c" \end{cases}.$$

Déf Matrice d'adjacence symbolique de G :

$A = (A_{ij})_{ij}$ avec $A_{ij} = \begin{cases} x_{ij} \text{ si } (u_i, v_j) \in E \\ 0 \text{ sinon.} \end{cases}$



$\to \begin{pmatrix} x_{11} & 0 & x_{13} \\ 0 & x_{22} & 0 \\ x_{31} & x_{32} & 0 \end{pmatrix}$

Thm G a un couplage parfait

$\iff \det(A)$ est un polynôme non nul.

Rappel $\det\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = -2$. $\det A = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} A_{i,\sigma(i)}$

Preuve Si $\sigma_1 \neq \sigma_2$, $\prod_{i=1}^{n} A_{i,\sigma_1(i)} \neq \prod_{i=1}^{n} A_{i,\sigma_2(i)}$

(sauf si nuls)

Donc : $\det(A) \neq 0 \iff \exists \sigma \prod_{i=1}^{n} A_{i,\sigma(i)} \neq 0$

$\iff \exists \sigma \; \forall i : A_{i,\sigma(i)} \neq 0$

$\iff \exists \sigma \; \forall i : A_{i,\sigma(i)} = x_{i,\sigma(i)}$

$\iff \exists \sigma \; \forall i : (u_i, v_{\sigma(i)}) \in E$

$\iff G$ admet un couplage parfait. □

Algo 1. Choisir des poids aléatoires $r_{ij} \in S$ sur chaque arête

2. Evaluer $\det\left(A(r_{11}, \dots, r_{nn})\right) =: \Delta$

3. Accepter si $\Delta \neq 0$.

Complexité Coût d'un déterminant $\sim \Theta(n^3)$ ou $\Theta(n^\omega)$.

Ça pas top, mais parallélisable.

Correction $\mathbb{P}[\text{échec}] \leq \dfrac{\deg(\det(A))}{|S|} = \dfrac{n}{|S|}$.

4. Conclusion : comparaisons

- Freivalds = cas particulier de Schwartz-Zippel :

Soit $z_1, \dots, z_n$ des variables, et $P_{AB} = AB\begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix}$, et $P_C = C \cdot \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix}$. Alors $A \cdot B = C \iff P_{AB} = P_C$.

- Vision commune de 2. et 3.

Etant donné $A = a_0 \dots a_n$, construisons $P_A = \sum_{i=0}^{n} a_i X^i$.

Alors 2. $\iff$ $P_A(\beta) \bmod q = 0$?
(β : base ; q : aléatoire)

3. $\iff$ $P_A(\alpha) \bmod p = 0$?
(α : aléatoire ; p : base)

- Troisième possibilité :

Prendre P aléatoirement et évaluer $P(A)$ et $P(B)$ (modulo un nb premier) $\rightarrow$ avec grde proba, $P(A) \neq P(B)$ si $A \neq B$.

$\Rightarrow$ Idée des fonctions de hachage (spoiler).