Synthèse du cours 10 : Introduction aux algorithmes probabilistes

26 novembre 2024

François Laroussinie

NB : Ces synthèses ont pour but de compléter les notes prises en cours. Elles ne les remplacent pas ! En particulier, la plupart des preuves n'y figurent pas. Rappel : il faut programmer les algorithmes vus en cours.

1 Algorithme de Freivalds

Pour ce problème, on dispose de 3 matrices $n \times n$ A, B et C et on veut vérifier si $C = A \cdot B$. C'est donc un problème de décision.

L'algorithme naïf consiste à vérifier coefficient par coefficient et cela se fait en $O(n^3)$ (ou avec Strassen en $O(n^{\log_2 7}) \approx O(n^{2.8})$ ou avec un autre algorithme de multiplication de matrices...). La procédure de base de l'algorithme de Freivalds est la suivante :

Procédure `Base-Freivalds` (A, B, C)
begin
  Générer aléatoirement un vecteur $\bar{x} = (x_i)_{1 \leq i \leq n}$ avec $x_i \in \{0, 1\}$ pour tout i
  Calculer $P = A \cdot (B \cdot \bar{x}) - C \cdot \bar{x}$
  retourner Oui ssi $P = \bar{0}$

La complexité de la procédure est en $O(n^2)$ (grâce à l'ordre indiqué par les parenthèses... et on ne veut surtout pas faire $(A \cdot B) \cdot \bar{x}$ qui serait en $O(n^3)$!).

Notons que si $C = A \cdot B$, alors l'algorithme retourne toujours le bon résultat (car $P = (A \cdot B - C) \cdot \bar{x}$). Et sinon, on va voir que l'algorithme retourne Oui (une erreur) avec une probabilité inférieure ou égale à $\frac{1}{2}$.
Si on itère l'algorithme k fois, on obtient alors un algorithme en $O(k \cdot n^2)$ avec une probabilité d'erreur inférieure ou égale à $\frac{1}{2^k}$, ce qui est très bien !

L'algorithme complet est donc :

Procédure `Algo-Freivalds` (A, B, C)
begin
  pour $i = 1 \ldots k$ **faire**
    Générer aléatoirement un vecteur $\bar{x} = (x_i)_{1 \leq i \leq n}$ avec $x_i \in \{0, 1\}$ pour tout i
    Calculer $P = A \cdot (B \cdot \bar{x}) - C \cdot \bar{x}$
    si $P \neq \bar{0}$ **alors**
      retourner $\bot$
  retourner $\top$

1.1 Analyse de l'algorithme de Freivalds

Supposons que $A \cdot B \neq C$. Soit $\bar{x}$ un vecteur de 0/1 choisi aléatoirement. Soit $P = A \cdot (B \cdot \bar{x}) - C \cdot \bar{x}$, on a :

$$P = A \cdot (B \cdot \bar{x}) - C \cdot \bar{x} \;=\; (A \cdot B) \cdot \bar{x} - C \cdot \bar{x} \;=\; (A \cdot B - C) \cdot \bar{x}$$

Soit $D = A \cdot B - C$.

Notons que l'on peut avoir $D \neq \bar{0}$ et $D \cdot \bar{x} = 0$, par exemple :

$$D = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix} \quad \bar{x} = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} \quad \text{ou} \quad D = \begin{pmatrix} 0 & 1 & -1 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix} \quad \bar{x} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

On va montrer que $\mathsf{Pr}(D \cdot \bar{x} \neq 0) \geq \frac{1}{2}$ (c'est-à-dire que la probabilité que $D \cdot \bar{x}$ soit différent du vecteur nul est supérieure ou égale à $\frac{1}{2}$).
Comme D est non nulle, il existe au moins une colonne j non nulle. Soit $\bar{x}'$le vecteur obtenu à partir de $\bar{x}$ en changeant seulement sa j-ème composante ($0 \to 1$ et $1 \to 0$), c'est à dire : $\bar{x}' = \bar{x} + \bar{u_j}$ ou $\bar{x}' = \bar{x} - \bar{u_j}$ avec u_j le vecteur composé de 0 partout sauf en j-ème position où il y a un 1. Dans la suite on note $\bar{x}' = \bar{x} \pm \bar{u_j}$. On a aussi $\bar{x} = \bar{x}' \pm \bar{u_j}$.

On peut observer que $D \cdot \bar{x}$ et $D \cdot \bar{x}'$ ne peuvent pas être nuls tous les deux. En effet, supposons $D \cdot \bar{x} = \bar{0}$, alors $D \cdot \bar{x}' = D \cdot (\bar{x} \pm \bar{u_j}) = D \cdot \bar{x} \pm D \cdot \bar{u_j} = \bar{0} \pm D \cdot \bar{u_j} \neq \bar{0}$ car la colonne j est non nulle (et elle correspond précisément à $D \cdot \bar{u_j}$). Idem si $D \cdot \bar{x}' = \bar{0}$.

Donc lorsqu'on choisit aléatoirement un $\bar{x}$, on a toujours au moins autant de vecteurs qui rendent $D \cdot \bar{x} \neq \bar{0}$ (car à chaque vecteur qui rend $D \cdot \bar{x}$ nul, il y a un $\bar{x}'$ pour lequel ce n'est pas le cas). Donc $\mathsf{Pr}(D \cdot \bar{x} \neq 0) \geq \frac{1}{2}$.

1.2 Complexité

Avec l'algorithme de Freivalds et une marge d'erreur inférieure à $0,01$, il faut faire 7 tests :

$$(\frac{1}{2})^k < \frac{1}{100} \Leftrightarrow \frac{1}{2^k} < \frac{1}{100} \Leftrightarrow 2^k > 100 \Leftrightarrow k > \log_2 100 = 6.6$$

Pour une précision de $0,0001$, il faut $k > \log_2(10000)$, c'est à dire $k \geq 14$.

La complexité de l'algorithme est en $O(k \cdot n^2)$. Pour $n = 100$, l'algorithme naïf (en n^3) donnerait de l'ordre de 1 million d'opérations. Avec l'algorithme de Freivalds et une précision de $0,0001$, on aurait besoin de de l'ordre de 140.000 opérations.