Big Data
TD 4 : Complexité algorithmique
BUT 3

Guillaume Metzler et Antoine Rolland
Institut de Communication (ICOM)
Université de Lyon, Université Lumière Lyon 2
Laboratoire ERIC UR 3083, Lyon, France
guillaume.metzler@univ-lyon2.fr; antoine.rolland@univ-lyon2.fr

Les exercices de cette fiche permette de mettre en avant l'importance de la façon dont les traitements statistiques doivent être effectués. On aborde la notion de complexité des algorithmes développés. Cette notion est primordiale en milieu industrielle, notamment quand il s'agit de développer des solutions qui *passent à l'échelle*.

Exercice 1 : Un peu de complexite

1. Calculez le nombre minimal d'opérations élémentaires pour calculer la moyenne d'un n échantillon de variables quantitatives continues.
2. Calculez le nombre minimal d'opérations élémentaires pour calculer la moyenne d'un n échantillon de variables quantitatives discrètes avec K modalités effectives si nous connaissons le tri à plat.
3. Comment appelle-t-on la complexité du calcul d'une moyenne ?
4. Donnez la complexité de chacun des algorithmes suivants codés en R.

```
## Algorithme 1 ##

x <- rnorm(n)
xbar<-mean(n)
s<-0
for (i in 1:n){
    s<-s+(x[i]-xbar)^2
}
s<-sqrt(s/n)

## Algorithme 2 ##
x <- rnorm(n)
sd(x)*(n-1)/n

## Algorithme 3 ##
x <- rnorm(n)
```

```
s<-0
for (i in 1:n){
    xbar<-mean(n)
    s<-s+(x[i]-xbar)^2
}
s<-sqrt(s/n)
```

5. Étant donné un n-échantillon, quelle est la complexité pour extraire toutes les parties de cette échantillon (c'est-à-dire, l'ensemble vide, tous les individus, tous les couples, tous les triplets et ainsi de suite jusqu'à l'échantillon complet) ?
6. On revient maintenant sur l'algorithme de l'Analyse en Composantes Principales, dont les différentes étapes sont données ci-dessous :
7. On revient maintenant sur l'algorithme de l'Analyse en Composantes Principales appliquée à une matrice de données $mX \in \mathscr{M}n, p(\mathbb{R})$, dont les différentes étapes sont données ci-dessous :

   (a) Centrage et réduction des variables de la matrice $\mathbf{X}$.
   (b) Calcul du produit $\mathbf{X}^\top\mathbf{X}$.
   (c) Recherche des valeurs propres de la matrice $\mathbf{X}^\top\mathbf{X}$ ainsi que les vecteurs propres associés.
   (d) Projection sur les sous-espaces propres des variables en calculant les $\mathbf{X}u_k,\ k = 1, \ldots s$.
   (e) Projection des individus sur les sous-espaces en calculant les $\frac{1}{\lambda_s}\mathbf{X}^\top\mathbf{X}u_k,\ k = 1, \ldots, s$.

   Déterminer la complexité de chaque étape ainsi que la complexité globale de la procédure. Que se passe-t-il si p est plus petit que n ?

Exercice 2 : Suite de Fibonacci

L'histoire des Mathématiques est parfois surprenante, et décidément toujours inattendue. Le vieux nombre d'or (qui apparaît dans la suite), à l'origine géométrique, s'apparenta des siècles plus tard avec des fractions issues d'une suite purement arithmétique. L'artisan de cette union fut le plus remarquable mathématicien du Moyen Âge, Leonardo Pisano, plus connu sur le nom de Fibonacci [1]. Le plus célèbre de tout les problèmes qui fait apparaître ce nombre d'or se trouve certainement dans le **Livre de l'abaque**. Il s'agit du fameux problème des lapins, dont la solution est la suite aujourd'hui connue sous le nom de Fibonacci.

Le problème est posé de la façon suivante : combien de couples de lapins aurons-nous à la fin de l'année si nous commençons avec un couple qui engendre chaque mois un autre couple qui procrée à son tour au bout de deux mois de vie ?

L'objectif de cet exercice est alors d'étudier les solutions de ce problème et plus largement la suite de Fibonacci $(F_n)_{\in\mathbb{N}}$ définie par $F_0 = 1$ et $F_1 = 1$ et pour tout $n \geq 1$.

$$F_{n+2} = F_{n+1} + F_n.$$

1. **Leonardo Pisano, Fibonacci (1170 - 1250)**. Il naquit à Pise en 1170. Son surnom renseigne sur son origine familiale : Fibonacci signifie tout simplement «fils de Bonacci» (figlio di Bonacci). Cependant, ce nom est d'origine moderne ; on ne dispose d'aucune preuve permettant d'affirmer qu'il était connu sous le patronyme de Fibonacci. Il s'initia aux mathématiques à partir de la comptabilité, car son père était un marchand italien qui avait des activités commerciales internationales. Rapidement, Leonardo montra un vif intérêt pour les mathématiques qui allait bien au-delà de leurs applications mercantiles. Ses voyages marchands en Afrique du Nord lui offrirent l'opportunité de s'initier aux mathématiques arabes aux côtés de maîtres musulmans. Il connut ainsi le système de numérotation arabo-hindou et en comprit immédiatement les énormes avantages. En Europe, il en devint le défenseur le plus zélé et tenta de le vulgariser. C'est à lui que nous devons apparition dans notre culture.

Plus précisément, on s'intéressera à des algorithmes permettant de déterminer les différents termes de cette suite, avec des complexités différentes.

1. Une version récursive de l'algorithme est donnée par la procédure suivante

---

Algorithm 1: Une approche récursive

Paramètre : Un entier $n \geq 0$
Sortie : Valeur de F_n
Fibo(n)
if $n \leq 1$ **then**
    retourner n
else
    retourner Fibo($n-1$) + Fibo($n-2$)
end if

Expliquer le fonctionnement de l'algorithme et évaluer sa complexité.

2. Une autre version, dite par recensement, peut également être employée pour évaluer les termes de cette suite. La procédure est décrite ci-après :

Algorithm 2: Une approche par recensement

Paramètre: Un entier $n \geq 0$
Sortie : Valeur de F_n
Table de Stockage : T de taille n Fibo(n)
if $n \leq 1$ **then**
    retourner n
else
    retourner Fibo($n-1$) + Fibo($n-2$)
    T[$n-1$] = Fibo($n-1$)
end if

Expliquer le fonctionnement de l'algorithme et évaluer sa complexité.

3. Proposer une version du calcul des termes de la suite $(F_n)_{n \in N}$ par une approche itérative, de complexité linéaire en temps et constante en mémoire.

Remarque Il est possible de calculer les différents terme de cette suite à l'aide d'une procédure de complexité logarithmique ($\log_2(n)$) en considérant la matrice

$$\mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

On pourrait même trouver une procédure de complexité constante ! Mais la solution nécessite de résoudre une petite équation.