Clustering

Épreuve pratique d'algorithmique et de programmation
Concours commun des Écoles normales supérieures

Durée de l'épreuve: 3 heures 30 minutes

Juin/Juillet 2013

ATTENTION !

N'oubliez en aucun cas de recopier votre u_0
à l'emplacement prévu sur votre fiche réponse

Important.

Sur votre table est indiqué un numéro u_0 qui servira d'entrée à vos programmes. Les réponses attendues sont généralement courtes et doivent être données sur la fiche réponse fournie à la fin du sujet. À la fin du sujet, vous trouverez en fait deux fiches réponses. La première est un exemple des réponses attendues pour un $\widetilde{u_0}$ particulier (précisé sur cette même fiche et que nous notons avec un tilde pour éviter toute confusion!). Cette fiche est destinée à vous aider à vérifier le résultat de vos programmes en les testant avec $\widetilde{u_0}$ au lieu de u_0. Vous indiquerez vos réponses (correspondant à votre u_0) sur la seconde et vous la remettrez à l'examinateur à la fin de l'épreuve.

En ce qui concerne la partie orale de l'examen, lorsque la description d'un algorithme est demandée, vous devez présenter son fonctionnement de façon schématique, courte et précise. Vous ne devez en aucun cas recopier le code de vos procédures!
Quand on demande la complexité en temps ou en mémoire d'un algorithme en fonction d'un paramètre n, on demande l'ordre de grandeur en fonction du paramètre, par exemple: $O(n^2)$, $O(n \log n)$,...
Il est recommandé de commencer par lancer vos programmes sur de petites valeurs des paramètres et de ***tester vos programmes sur des petits exemples que vous aurez résolus préalablement à la main ou bien à l'aide de la fiche réponse type fournie en annexe***. Enfin, il est recommandé de lire l'intégralité du sujet avant de commencer afin d'effectuer les bons choix de structures de données dès le début.

1 Introduction

En analyse de données statistiques, le clustering décrit des méthodes de classification de données : méthode de regroupement hiérarchique ou méthode de partitionnement de données. Sa tâche consiste à grouper un ensemble d'objets de telle sorte que les objets d'une même classe (cluster) sont similaires entre eux de ceux des autres classes. Son but principal est l'exploration de données en vue d'extraire de la connaissance et est très utile en analyse statistique de données, en analyse d'image, reconnaissance de motifs, apprentissage automatique et bioinformatique.

Ici, on s'intéresse au problème de partitionner un ensemble P de N points en un certain nombre de sous-ensembles. Dans tous les cas (sauf Section 3.1), les sous-ensembles seront des *segments* représentant un ensemble de points d'indices consécutifs de la forme $\{p_i, p_{i+1}, \ldots, p_{j-1}, p_j\}$ pour $i \leq j$.

Étant donné un ensemble de N points $P = \{p_1, \ldots, p_N\}$ et un entier $K \leq N$, on s'intéresse au problème de minimisation de la valeur :

$$\sum_{i=1}^{K} \sum_{p_j \in C_i} dist(p_j, C_i),$$

où C_i est une classe qui sera représentée soit par un point soit par une fonction et *dist* sera une fonction. La fonction *dist* sera le carré de la norme euclidienne sauf à la Setion 3.2.

1.1 Préliminaires

Considérons $(u_k)_{k\geq 0}$ et $(v_k)_{k\geq 0}$ les suites d'entiers définies par :

$$u_k = \begin{cases} \text{votre } u_0 \quad (\textit{À reporter sur votre fiche réponse}) & \text{si } k = 0 \\ 6679 \times u_{k-1} \mod 10007 & k > 0. \end{cases}$$

et $v_k = u_k \bmod 100$ pour $k \geq 0$.

Question 1 *Que valent :*
a) v_3 ***b)*** v_{50} ***c)*** v_{100}.

2 En dimension 1

Soit un ensemble $P = \{x_1, x_2, \ldots, x_N\}$ de N entiers et un entier $K \leq N$. On cherche à classifier les N entiers dans K segments $\mathbf{S} = \{S_1, \ldots, S_K\}$ représentés par les nombres $\{\mu_1, \ldots, \mu_K\}$ où μ_i est la moyenne des valeurs dans S_i de sorte à minimiser la valeur

$$\sum_{i=1}^{K} \sum_{x_j \in S_i} (x_j - \mu_i)^2,$$

où $\mu_i = (\sum_{x_j \in S_i} x_j)/\#S_i$ avec $\#S_i$ le cardinal de S_i.
Pour générer les valeurs de P, on trie les valeurs $v_0, \ldots, v_N$ dans l'ordre croissant et on définit les valeurs de $x_1, \ldots, x_N$ comme les N plus grandes valeurs. On n'éliminera pas les doublons.

2.1 Approche gloutonne

Question à développer pendant l'oral : L'intervalle fermé entre les valeurs x_i sera représenté par sa taille, son point de départ et son point d'arrivée. La taille du segment $[x_i, x_j]$ est $|x_j - x_i|$. Décrire un algorithme qui retourne les intervalles entre les x_i classés par taille croissante. En cas d'égalité sur la taille, on considérera également le point de départ pour les départager. Quelle est sa complexité en temps ?

Question 2 *Pour $N = 10$, on donnera les intervalles de la forme $[x_i, x_{i+1}]$ dans l'ordre croissant des tailles :*
a) *les 3 premiers* ***b)*** *les 3 du milieu* ***c)*** *les 3 derniers.*

Question à développer pendant l'oral : Si $K = N$, alors le minimum est obtenu en créant N classes contenant chacune un élément. En remarquant que les points (classes) les plus proches doivent être dans la même classe, proposer une méthode gloutonne qui fusionne en priorité les classes les plus proches. Une classe sera représentée par la moyenne des éléments contenus. En cas d'égalité, on considérera le début du segment dans l'ordre croissant pour les départager. Analyser la complexité en temps et en mémoire de votre algorithme.

Question 3 *Que vaut la valeur de la somme $\sum_{i=1}^{K} \sum_{x_j \in S_i} (x_j - \mu_i)^2$ dans le cas :*
a) $N = 8, K = 3$ ***b)*** $N = 10, K = 3$ ***c)*** $N = 20, K = 6$.

2.2 Solution optimale

Pour rechercher la solution optimale, on va considérer les sous-problèmes suivants : classifier $\{x_1, \ldots, x_i\}$ en m classes et on stockera le minimum dans l'entrée $D[m, i]$ d'une matrice $(K + 1) \times (N + 1)$.
Question à développer pendant l'oral : Donner une relation de récurrence entre $D[m, i]$ et des valeurs de $D[m - 1, j - 1]$ et $d(x_j, \ldots, x_i)$ où cette valeur représente la somme des distances au carré des $x_j, \ldots, x_i$ à leur moyenne. Quelle est la complexité en temps et en mémoire de votre algorithme ?

Question 4 *Donner la valeur de $D[K, N]$ pour :*
a) $N = 8, K = 3$ ***b)*** $N = 10, K = 3$ ***c)*** $N = 20, K = 6$.

Question à développer pendant l'oral : Comment modifier votre algorithme pour qu'il retourne également les classes satisfaisant la valeur optimale ?

Question 5 *Pour $N = 50$ et $K = 3$, donner*
a) *le premier segment* ***b)*** *le second segment* ***c)*** *le troisième segment.*

Question à développer pendant l'oral : Trouver une relation entre $m_i = \sum_{j=1}^{i} x_j / i$ et m_{i-1} et x_i et montrer la relation

$$d(x_1, \ldots, x_i) = d(x_1, \ldots, x_{i-1}) + \frac{i-1}{i}(x_i - m_{i-1})^2.$$

En déduire un algorithme plus rapide et donner sa complexité en temps.

Question 6 *On considère les points $x_1, \ldots, x_N$ correspondant aux N plus grandes valeurs de $u_0, u_1, \ldots, u_N$.*
Attention : *Certaines questions peuvent nécessiter un temps de calcul assez long.*
Que vaut $D[K, N]$ pour :
a) $N = 1000, K = 100$ ***b)*** $N = 2000, K = 100$ ***c)*** $N = 4000, K = 100$.

3 En dimension 2

On s'intéresse maintenant au même problème dans un plan. On considère un ensemble P de N points dans un plan représentés par $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$. On notera $p_i = (x_i, y_i)$.
Les valeurs μ_i représentant chaque classe seront maintenant des points $\mu_i = (x^{\mu_i}, y^{\mu_i})$ du plan définis par

$$x^{\mu_i} = \frac{\sum_{p_j \in C_i} x_j}{\#C_i} \text{ et } y^{\mu_i} = \frac{\sum_{p_j \in C_i} y_j}{\#C_i},$$

où $\#C_i$ dénote le cardinal de l'ensemble C_i.

3.1 Recherche exhaustive

Question à développer pendant l'oral : Proposer une méthode naïve pour résoudre le problème quand on cherche à classifier l'ensemble en deux classes $K = 2$. Quelle est la complexité en temps et mémoire de votre algorithme.

Question 7 *Que vaut la valeur minimale de $\sum_{x_j \in S_1} \|p_j - \mu_1\|^2 + \sum_{x_j \in S_2} \|p_j - \mu_2\|^2$ où $\|\cdot\|^2$ est le carré de la distance euclidienne pour les points $\{p_1, \ldots, p_N\}$ définis par $x_i = v_{2i}$ et $y_i = v_{2i+1}$ pour $i = 1$ à N :*
a) $N = 10$ ***b)*** $N = 16$ ***c)*** $N = 20$.

3.2 Approximation de points par des segments

Soit une droite L définie par l'équation $y = ax + b$. On dit que l'erreur de L par rapport à P est la somme des carrés des distances aux points de P :

$$Erreur(L, P) = \sum_{i=1}^{N} (y_i - ax_i - b)^2.$$

Question à développer pendant l'oral : Montrer que la solution

$$a = \frac{N \sum_i x_i y_i - (\sum_i x_i)(\sum_i y_i)}{N \sum_i x_i^2 - (\sum_i x_i)^2} \text{ et } b = \frac{\sum_i y_i - a \sum_i x_i}{N}$$

est une solution optimale au problème quand on recherche une droite qui approxime l'ensemble de points par la méthode des moindres carrés.

On s'intéresse ici à une généralisation de ce problème par plusieurs droites. Pour chaque segment S de la partition de P, on calcule la droite minimisant l'erreur par rapport à S selon la formule suivante. La *pénalité* d'une partition est définie comme la somme des termes suivants :

i) le nombre de segments de la partition de P multiplié par une constante $PEN > 0$ (on pénalise l'ajout de segments en donnant une grande valeur à PEN),

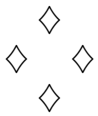
ii) pour chaque segment, la valeur de l'erreur de la droite optimale correspondant à ce segment.

Notre but est de trouver une partition dont la pénalité est minimale.
Question à développer pendant l'oral : Donner un algorithme pour résoudre ce problème. Quel est sa complexité en temps et en mémoire. (On pourra comme à la section 2.2 chercher une relation de récurrence pour la valeur OPT(j) qui représente la solution optimale pour les points $p_1, \ldots, p_j$ et $e_{i,j}$ l'erreur minimale de toute droite par rapport à $p_i, p_{i+1}, \ldots, p_j$.)

On considère les points générés par l'équation $f_{i,M}(x) = \prod_{j=0}^{i-1}(x - v_j)/M$ et $p_k = (k, f_i(k))$ pour $k = 1, \ldots, N$. On prendra $PEN = 1000$ dans les questions suivantes.

Question 8 *Pour la suite de points générés par la fonction $f_{5,100000}$, que vaut la valeur optimale quand :*
a) $N = 60$ ***b)*** $N = 80$ ***c)*** $N = 100$.

Question 9 *Donner le nombre de segments pour la valeur optimale quand $N = 100$ et pour les fonctions :*
a) $f_{3,100}$ ***b)*** $f_{4,100000}$ ***c)*** $f_{5,100000}$.



Fiche réponse type: Clustering

$\widetilde{u_0}$: 26

Question 1

a) 80

b) 24

c) 52

Question 2

a) (80,80), (88,90), (52,57)

b) (45,52), (80,88), (26,35)

c) (35,45), (7,26), (57,80)

Question 3

a) 463.83

b) 717.00

c) 222.05

Question 4

a) 463.83

b) 536.25

c) 201.80

Question 5

a) (1, 30)

b) (35, 66)

c) (75, 99)

Question 6

a) 591849.54

b) 1378186.32

c) 3042049.00

Question 7

a) 9235.83

b) 13386.76

c) 21579.45

Question 8

a) 5649.55

b) 8377.28

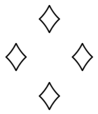
c) 15326.38

Question 9

a) 10

b) 3

c) 13



Fiche réponse: Clustering

Nom, prénom, u_0: .....

Question 1

a)

b)

c)

Question 2

a)

b)

c)

Question 3

a)

b)

c)

Question 4

a)

b)

c)

Question 5

a)

b)

c)

Question 6

a)

b)

c)

Question 7

a)

b)

c)

Question 8

a)

b)

c)

Question 9

a)

b)

c)