MPI: Apprentissage

Cours

- Différece entre apprentissage supervisé et non supervisé.
- Algorithme des k-plus proches voisins.
- Arbre de décision, Matrice de confusion.
- Entropie de Shannon, gain, algorithme ID3.
- Algorithme CHA (Classification hiérachique ascendente)
- L'Algorithme des k-moyennes.

Petites questions

1. Donner un algorithme qui ajoute un sommet $\vec{x} \in \mathbb{R}^k$ à un arbre k-dimensionnel. Quel est sa complexité ?
2. Écrire une fonction `int* voisins1D(float x, float* X, int k, int n)` permettant de trouver efficacement les k plus proches voisins de x dans l'ensemble de n données X trié par ordre croissant, où chaque donnée est un réel (en dimension 1). La fonction renvoie un tableau d'entiers contenant les indices des k plus proches voisins de x dans X. Quelle est sa complexité ?
3. Rappeler l'intêret et le fonctionnement de l'algorithme de classification hiérarchique ascendante.

Single-pass (CCINP 2024)[1]

On considère l'algorithme suivant pour catégoriser des données $X = \{\vec{x_1}, ..., \vec{x_n}\} \subseteq \mathbb{R}^d$. On note $\delta(\vec{x}, \vec{y}) = \sum_{i=1}^{d} |\vec{x}_i - \vec{x}_i|$ la distance entre $\vec{x}$ et $\vec{y}$. La distance de $\vec{x}$ à une classe C est la distance de $\vec{x}$ au centre de C. On note $\mathrm{Cl}(\vec{x}, C)$ la classe dont le centre est le plus proche de $\vec{x}$ pour δ.

Soit $\theta \in \mathbb{R}_+$, on considère:

$C \leftarrow \emptyset$
Pour i de 1 à n :
    Si $\forall \vec{c} \in C, \delta(\vec{x}_i, \vec{c}) > \theta$:
        $C \leftarrow C \cup \{\{\vec{x}_i\}\}$ // On crée une nouvelle classe
    Sinon:
        $\overline{C} \leftarrow \mathrm{Cl}(\vec{x}_i, C)$
        $\overline{C}' \leftarrow \overline{C} \cup \{\vec{x}_i\}$
        $C \leftarrow C \setminus \left\{\overline{C}\right\} \cup \left\{\overline{C}'\right\}$
Renvoyer C

Question 1 Proposer une implémentation en C de la fonction δ. On représentera un $\vec{x} \in \mathbb{R}^d$ par un `float* x`, et d sera traité comme une constante.

Question 2 Montrer que l'ordonnancement des $\vec{x}_1, ..., \vec{x}_n$ importe.

Question 3 Soit $\vec{x} \in X$, on note C_{ini} la classe de $\vec{x}$ au moment ou il a été ajouté, et C^* sa classe à la fin. Montrer que

$$\delta(\vec{x}, C^*) \leq \theta \ln(|C^*| - |C_{\text{ini}}|)$$

On admettera que $\frac{1}{2} + \frac{1}{3} + ... + \frac{1}{k} \leq \ln(k)$

Question 4 Quel sont les avantages de cet algorithme par rapport aux k-moyennes ? Proposer des améliorations à l'algorithme.

[1] https://prepas-mp2i.fr/documents/sujets/2024/CCINP-INFO.pdf

Clustering en dimension 1[2]

Définition Soit $K \in \mathbb{N}^*$ et $E = \{x_0, ..., x_{N-1}\}$ un ensemble de réels avec $x_0 \leq ... \leq x_{N-1}$. Pour $X \subseteq E$, on note μ_X son centre et $S(X)$ son score, formellement définit par

$$\mu_X := \frac{1}{|X|} \sum_{x \in X} x \qquad\qquad S(X) := \sum_{x \in X} (x - \mu_X)^2$$

On pose $S(i,j) = S(\{x_i, ..., x_j\})$. On cherche à résoudre l'algorithme des k-moyennes sur E de manière exacte. C'est à dire que l'on cherche une partition $\mathcal{P} = \{X_1, ..., X_K\}$ de $[\![0; N-1]\!]$ qui minimise la somme des scores de chaque X_i

Question 2 Montrer que l'algorithme CHA ne permet pas de résoudre de manière exacte le problème. On pourra prendre $N = 4$ et $K = 2$.

Question 3 On note $I(n,k)$ le score minimale d'une partition de $\{x_0, ..., x_n\}$ en k classes. Que vaux $I(n,1)$?

Question 4 Montrer que

$$\forall n > 0, \forall k > 1, I(n,k) = \min_{k-1 \leq m \leq n-1} (I(m, k-1) + S(m,n))$$

Question 5 En déduire une fonction `double inertie(double* E, int N, int K)` qui calcule le score minimale possible d'une parition de E en K classes non vides.

Question 6 Montrer que l'on peut calculer l'ensemble des $S(i,j)$ en $O(N^2)$

[2] tiré de https://mpi-lamartin.github.io/mpi-info/docs/ia/non_supervise/td_apprentissage

k-centres

Définition On fixe $V \subseteq \mathbb{R}^d$ un jeu de donnée. Pour $\vec{y} \in \mathbb{R}^d$ on définit $d(\vec{y}, V) = \max_{\vec{x} \in V} \|\vec{y} - \vec{x}\|$. On cherche à choisir k points de V tel qu'ils minimise $\max_{v \in V} d(v, X)$. On regarde donc le problème k-CENTRE suivant:

Entrée: $v_1, ..., v_n \subseteq \mathbb{R}^d$ une liste de points finis **Sortie:** $X := \{x_1, ..., x_k\} \subseteq V$ qui minimise $\max_{v \in V} d(v, X)$

Question 1 Rapeller le fonctionnement de l'algorithme des k-moyennes. Quel sont les différences avec le problème des k-centres ?

Question 2 On cherche à montrer que l'algorithme du cours de la question ne donne pas de bonne approximation du problème des k-centres. Pour cela on pose $k = 2$, $d = 1$ et on pose N points en 0, N points en 1 et un point en D. Montrer que pour des bonnes valeurs de N et D on peut avoir le ratio $\frac{\text{opt-k-moyennes}}{\text{opt-k-centre}}$ arbitrairement large.

Question 3 Proposer un algorithme glouton pour le problème des k-centres.

Question 4 On note $p_1, ..., p_k$ les k points retrourné par notre algorithme et $p_1^*, ..., p_k^*$ la solution optimale. On partitionne $V = \bigsqcup_{i \in \mathbb{N}} V_i$ avec V_i l'ensemble des points les plus proches de p_i^*. Montrer que

- Si $\forall i, V_i \cap \{p_1, ..., p_n\} \neq \emptyset$ alors on a une 2-approximation
- Sinon, on a aussi une 2-approximation

Algorithme ID3 avec intervalles

On cherche ici à effectuer l'algorithme ID3 pour des attributs non discret (un intervalle des réels par exemple). Pour cela, on considère des attributs $(A_i)_i$ tel que A_i est soit fini, soit $A_i = [\alpha_i, \beta_i]$, et un ensemble de classes C fini. Pour $S \subseteq A_1 \times .... \times A_k \times C$ un ensemble fini d'apprentisage, et pour A_i de la forme $A_i = [\alpha, \beta]$, pour $k \in]\alpha, \beta[$, on définit $S_{<k}, S_{\geq k}$ la coupure à k telle que $S_{<k} = \{\vec{s} \in S \mid \vec{s}_i < k\}$ et $S_{\geq k} = \{\vec{s} \in S \mid \vec{s}_i \geq k\}$

Question 1 On suppose que on a un attribut "taille (en cm)" qui est un naturel. Quel est le problème avec ID3 ?

Question 2 On cherche à trouver une "bonne coupure" : une qui maximise le gain d'information sur la coupure. Proposer une formule pour le gain d'information dans ce contexte similaire.

Question 3 Proposer un algorithme en $O(|S| \log |S| + |C| \times |S|)$ qui trouve le meilleur $k \in [\alpha; \beta]$ pour effectuer une coupure.

Question 4 Quelle est la complexité de l'algorithme ID3 avec notre généralisation en fonction de $|S|$, de $|C|$ et des $(A_i)_i$?