

TD : ALGORITHMES POUR L'INTELLIGENCE ARTIFICIELLE
== ALGORITHME DES K MOYENNES ==

I) ALGORITHME DES K-MOYENNES

1. Recopier et remplir le tableau suivant en appliquant l'algorithme des k-moyennes en prenant $k = 2$ et d_2 et d_5 comme centres de classe initiaux. On utilisera la distance $\delta(d_i, d_j)$:

On notera c_i les centres de classe et $\mathcal{A}_i$ les classes correspondantes.

Itération	c_1	c_2	$\mathcal{A}_1$	$\mathcal{A}_2$
1	d_2	d_5	$\{d_2, d_3\}$	$\{d_1, d_4, d_5\}$
2	$(1, 1, 0)$	$(1, 2, 5/3)$	$\{d_2, d_3, d_5\}$	$\{d_1, d_4\}$
3	$(1, 2/3, 1/3)$	$(1, 3, 2)$	$\{d_2, d_3, d_5\}$	$\{d_1, d_4\}$

Itération n°0 – Clusters initiaux :

- $\mathcal{A}_1 = \{d_2\}$ de centre $c_1 = d_2$;
- $\mathcal{A}_2 = \{d_5\}$ de centre $c_2 = d_5$

Itération n°1 :

- **Étape 1 : Affectation des vecteurs aux clusters $\mathcal{A}_1$ et $\mathcal{A}_2$:**
 - **Choix pour l'affectation de d_1 :**
 - $\delta(d_1, c_1) = |1 - 2| + |3 - 1| + |3 - 0| = 1 + 2 + 3 = 6$
 - $\delta(d_1, c_2) = |1 - 1| + |3 - 0| + |3 - 1| = 0 + 3 + 2 = 5$
 - $\mathcal{A}_1 = \{d_2\}$; $\mathcal{A}_2 = \{d_1, d_5\}$
 - **Choix pour l'affectation de d_3 :**
 - $\delta(d_3, c_1) = |0 - 2| + |1 - 1| + |0 - 0| = 2 + 0 + 0 = 2$
 - $\delta(d_3, c_2) = |0 - 1| + |1 - 0| + |0 - 1| = 1 + 1 + 1 = 3$
 - $\mathcal{A}_1 = \{d_2, d_3\}$; $\mathcal{A}_2 = \{d_1, d_5\}$
 - **Choix pour l'affectation de d_4 :**
 - $\delta(d_4, c_1) = |1 - 2| + |3 - 1| + |1 - 0| = 1 + 2 + 1 = 4$
 - $\delta(d_4, c_2) = |1 - 1| + |3 - 0| + |1 - 1| = 0 + 3 + 0 = 3$
 - $\mathcal{A}_1 = \{d_2, d_3\}$; $\mathcal{A}_2 = \{d_1, d_4, d_5\}$
- **Étape 2 : Calcul des centres :**
 - $c_1 = \frac{1}{|\mathcal{A}_1|} \sum_{x_i \in \mathcal{A}_1} x_i = (1, 1, 0)$
 - $c_2 = \frac{1}{|\mathcal{A}_2|} \sum_{x_i \in \mathcal{A}_2} x_i = (1, 2, 5/3)$

Itération n°2 :

- **Étape 1 : Affection des vecteurs aux clusters $\mathcal{A}_1$ et $\mathcal{A}_2$:**
 - **Choix pour l'affectation de d_1 :**
 - $\delta(d_1, c_1) = |1 - 1| + |3 - 1| + |3 - 0| = 0 + 2 + 3 = 5$
 - $\delta(d_1, c_2) = |1 - 1| + |3 - 2| + |3 - 5/3| = 0 + 1 + 4/3 = 7/3 = 2,33$
$$\rightarrow \mathcal{A}_1 = \{ \quad \}; \mathcal{A}_2 = \{d_1\}$$
 - **Choix pour l'affectation de d_2 :**
 - $\delta(d_2, c_1) = |2 - 1| + |1 - 1| + |0 - 0| = 1 + 0 + 0 = 1$
 - $\delta(d_2, c_2) = |2 - 1| + |1 - 2| + |0 - 5/3| = 1 + 1 + 5/3 = 11/3 = 3,67$
$$\rightarrow \mathcal{A}_1 = \{d_2\}; \mathcal{A}_2 = \{d_1\}$$
 - **Choix pour l'affectation de d_3 :**
 - $\delta(d_3, c_1) = |0 - 1| + |1 - 1| + |0 - 0| = 1 + 0 + 0 = 1$
 - $\delta(d_3, c_2) = |0 - 1| + |1 - 2| + |0 - 5/3| = 1 + 1 + 5/3 = 11/3 = 3,67$
$$\rightarrow \mathcal{A}_1 = \{d_2, d_3\}; \mathcal{A}_2 = \{d_1\}$$
 - **Choix pour l'affectation de d_4 :**
 - $\delta(d_4, c_1) = |1 - 1| + |3 - 1| + |1 - 0| = 0 + 2 + 1 = 3$
 - $\delta(d_4, c_2) = |1 - 1| + |3 - 2| + |1 - 5/3| = 0 + 1 + 2/3 = 5/3 = 1,67$
$$\rightarrow \mathcal{A}_1 = \{d_2, d_3\}; \mathcal{A}_2 = \{d_1, d_4\}$$
 - **Choix pour l'affectation de d_5 :**
 - $\delta(d_5, c_1) = |1 - 1| + |0 - 1| + |1 - 0| = 0 + 1 + 1 = 2$
 - $\delta(d_5, c_2) = |1 - 1| + |0 - 2| + |1 - 5/3| = 0 + 2 + 2/3 = 8/3 = 2,67$
$$\rightarrow \mathcal{A}_1 = \{d_2, d_3, d_5\}; \mathcal{A}_2 = \{d_1, d_4\}$$
- **Étape 2 : Calcul des centres :**
 - $c_1 = \frac{1}{|\mathcal{A}_1|} \sum_{x_i \in \mathcal{A}_1} x_i = (1, 2/3, 1/3)$
 - $c_2 = \frac{1}{|\mathcal{A}_2|} \sum_{x_i \in \mathcal{A}_2} x_i = (1, 3, 2)$

Itération n°3 :

- **Étape 1 : Affection des vecteurs aux clusters $\mathcal{A}_1$ et $\mathcal{A}_2$:**
 - **Choix pour l'affectation de d_1 :**
 - $\delta(d_1, c_1) = |1 - 1| + |3 - 2/3| + |3 - 1/3| = 0 + 7/3 + 8/3 = 15/3 = 5$
 - $\delta(d_1, c_2) = |1 - 1| + |3 - 3| + |3 - 2| = 0 + 0 + 1 = 1$
$$\rightarrow \mathcal{A}_1 = \{ \quad \}; \mathcal{A}_2 = \{d_1\}$$
 - **Choix pour l'affectation de d_2 :**
 - $\delta(d_2, c_1) = |2 - 1| + |1 - 2/3| + |0 - 1/3| = 1 + 1/3 + 1/3 = 2/3 = 0,67$
 - $\delta(d_2, c_2) = |2 - 1| + |1 - 3| + |0 - 2| = 1 + 2 + 2 = 5$
$$\rightarrow \mathcal{A}_1 = \{d_2\}; \mathcal{A}_2 = \{d_1\}$$

- **Choix pour l'affectation de d_3 :**
 - $\delta(d_3, c_1) = |0 - 1| + |1 - 2/3| + |0 - 1/3| = 1 + 1/3 + 1/3 = 2/3 = 0,67$
 - $\delta(d_3, c_2) = |0 - 1| + |1 - 3| + |0 - 2| = 1 + 2 + 2 = 5$
 $\rightarrow \mathcal{A}_1 = \{d_2, d_3\}; \mathcal{A}_2 = \{d_1\}$
- **Choix pour l'affectation de d_4 :**
 - $\delta(d_4, c_1) = |1 - 1| + |3 - 2/3| + |1 - 1/3| = 0 + 7/3 + 2/3 = 9/3 = 3$
 - $\delta(d_4, c_2) = |1 - 1| + |3 - 3| + |1 - 2| = 0 + 0 + 1 = 1$
 $\rightarrow \mathcal{A}_1 = \{d_2, d_3\}; \mathcal{A}_2 = \{d_1, d_4\}$
- **Choix pour l'affectation de d_5 :**
 - $\delta(d_5, c_1) = |1 - 1| + |0 - 2/3| + |1 - 1/3| = 0 + 2/3 + 2/3 = 4/3 = 1,33$
 - $\delta(d_5, c_2) = |1 - 1| + |0 - 3| + |1 - 2| = 0 + 3 + 1 = 4$
 $\rightarrow \mathcal{A}_1 = \{d_2, d_3, d_5\}; \mathcal{A}_2 = \{d_1, d_4\}$
- **mêmes résultats qu'en itération 2 $\rightarrow$ CONVERGENCE.**

2. Donner la complexité de l'algorithme en fonction du nombre de documents n , de la dimension caractéristique des documents p et du nombre de classes (clusters) k et du nombre d'itérations I .

L'algorithme des k-moyennes comporte deux phases :

- **Affectation :** pour chaque vecteur d_i , on calcul sa distance à chacun des k centres.
 - Une distance L1 coûte $O(p)$ (somme des p valeurs absolues)
 - Le coût total est donc de $O(n \cdot k \cdot p)$
- **Recalcul des centres :** on moyenne, pour chaque cluster, les vecteurs qui lui sont affectés :
 - Pour un cluster i , le coût vaut $O(|\mathcal{A}_i| \cdot p)$;
 - Pour l'ensemble des clusters : $\sum_{i=1}^k O(|\mathcal{A}_i| \cdot p) = O\left(p \underbrace{\sum_{i=1}^k |\mathcal{A}_i|}_{=n}\right) = O(np)$

Le coût total d'une itération est donc dominé par $O(n \cdot k \cdot p) = O(n \cdot p)$ si k est constant.

Avec I itérations, le coût total est de $O(I \cdot n \cdot k \cdot p) = O(n \cdot p)$ si k et I sont constants.

3. Implémenter la fonction distance $\delta(d_i, d_j)$ calculant la distance de Manhattan entre deux vecteurs NumPy. Tester avec $\delta(d_1, d_2) = 6.0$ et $\delta(d_2, d_3) = 2.0$.

Voir fichier « 6. TD4 - K-moyennes - Correction.py »

4. Écrire une fonction `affectation(documents, centres)` qui retourne, pour chaque document, l'indice du centre le plus proche au sens de δ . La fonction retourne une liste d'entiers de même longueur que documents.

Voir fichier « 6. TD4 - K-moyennes - Correction.py »

5. Écrire une fonction `recalcul_centres(documents, indices_centres, k)` qui calcule les nouveaux centres de classe comme la moyenne des documents affectés à chaque classe.

Voir fichier « 6. TD4 - K-moyennes - Correction.py »

6. À l'aide des fonctions précédentes, implémenter l'algorithme complet `k_moyennes(documents, k, centres_init)`. L'algorithme s'arrête quand les affectations des documents dans les classes ne changent plus entre deux itérations. Retourner les affectations finales et les centres.

Voir fichier « 6. TD4 - K-moyennes - Correction.py »

7. Appliquer `k_moyennes` sur `documents_grands`. Pour `k_moyennes`, utiliser `k=3` et trois documents du début de chaque groupe comme centres initiaux.

Voir fichier « 6. TD4 - K-moyennes - Correction.py »

Vérifier : Labels finaux `doc_grands`: [0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 0, 0, 0]

⇒ On retrouve bien les bons labels

II) ALGORITHME « SINGLE-PASS »

8. Appliquer l'algorithme 1 avec $\theta = 5.0$. Détailler les résultats des étapes de l'algorithme.

- **Initialisation : Un seul centre $c_1 = d_1$; Un seul cluster $\mathcal{A}_1 = \{d_1\}$**
- **Calcul de $\delta(d_2, c_1) = |2 - 1| + |1 - 3| + |0 - 3| = 1 + 2 + 3 = 6 > \theta$**
 - ⇒ **Création d'une nouvelle classe $\mathcal{A}_2 = \{d_2\}$, de centre $c_2 = d_2$**
 - **Centres : $c_1 = d_1 = (1, 3, 3)$; $c_2 = d_2 = (2, 1, 0)$**
 - **Clusters : $\mathcal{A}_1 = \{d_1\}$; $\mathcal{A}_2 = \{d_2\}$**
- **Calcul de $\delta(d_3, c_1) = |0 - 1| + |1 - 3| + |0 - 3| = 1 + 2 + 3 = 6 > \theta$**
- **Calcul de $\delta(d_3, c_2) = |0 - 2| + |1 - 1| + |0 - 0| = 2 + 0 + 0 = 2 \leq \theta$**
 - ⇒ **Affecte d_3 à $\mathcal{A}_2$: $\mathcal{A}_2 = \{d_2 ; d_3\}$**
 - ⇒ **Calcul $c_2 = \frac{1}{|\mathcal{A}_2|} \sum_{x_i \in \mathcal{A}_2} x_i = (1, 1, 0)$**
 - **Centres $c_1 = (1, 3, 3)$; $c_2 = (1, 1, 0)$**
 - **Clusters : $\mathcal{A}_1 = \{d_1\}$; $\mathcal{A}_2 = \{d_2 ; d_3\}$**
- **Calcul de $\delta(d_4, c_1) = |1 - 1| + |3 - 3| + |1 - 3| = 0 + 0 + 2 = 2 \leq \theta$**
- **Calcul de $\delta(d_4, c_2) = |1 - 1| + |3 - 1| + |1 - 0| = 0 + 2 + 1 = 3 \leq \theta$**
 - ⇒ **Affecte d_4 à $\mathcal{A}_1$: $\mathcal{A}_1 = \{d_1 ; d_4\}$**
 - ⇒ **Calcul $c_1 = \frac{1}{|\mathcal{A}_1|} \sum_{x_i \in \mathcal{A}_1} x_i = (1, 3, 2)$**
 - **Centres $c_1 = (1, 3, 2)$; $c_2 = (1, 1, 0)$**
 - **Clusters : $\mathcal{A}_1 = \{d_1 ; d_4\}$; $\mathcal{A}_2 = \{d_2 ; d_3\}$**
- **Calcul de $\delta(d_5, c_1) = |1 - 1| + |0 - 3| + |1 - 2| = 0 + 3 + 1 = 4 \leq \theta$**
- **Calcul de $\delta(d_5, c_2) = |1 - 1| + |0 - 1| + |1 - 0| = 0 + 1 + 1 = 2 \leq \theta$**
 - ⇒ **Affecte d_5 à $\mathcal{A}_2$: $\mathcal{A}_2 = \{d_2 ; d_3 ; d_5\}$**
 - ⇒ **Calcul $c_2 = \frac{1}{|\mathcal{A}_2|} \sum_{x_i \in \mathcal{A}_2} x_i = (1, 2/3, 1/3)$**
 - **Centres $c_1 = (1, 3, 2)$; $c_2 = (1, 0.67, 0.33)$**
 - **Clusters : $\mathcal{A}_1 = \{d_1 ; d_4\}$; $\mathcal{A}_2 = \{d_2 ; d_3 ; d_5\}$**

Les deux algorithmes produisent la même partition $\{d_1 ; d_4\} / \{d_2 ; d_3 ; d_5\}$ même si les centres ne sont pas exactement identiques car le « Single Pass » recalcule les centres à chaque pas (pas en fin d'itération).

9. Analyser la complexité temporelle de l'algorithme « Single Pass » en fonction de n (nombre de documents), p (dimension des vecteurs) et j (nombre de classes créées). Comparer avec la complexité des k -moyennes.

Pour chaque document d_i (n documents en tout), on calcule sa distance à tous les j centres existants, chaque distance coûtant $O(p)$. La boucle principale est donc $O(n \cdot j \cdot p)$.

Dans le pire cas $j = n$ (tous les documents créent une classe), ce qui donne $O(n^2 \cdot p)$.

Dans le meilleur cas (k classes finales fixées), c'est $O(n \cdot k \cdot p)$, soit $O(n \cdot p)$ si k est constant.

Comparaison entre les deux algorithmes :

- Les deux algorithmes sont linéaires en n si k et l sont constants. La différence clé est que « Single Pass » ne nécessite qu'une seule passe sur les données, ce qui le rend plus adapté aux grands corpus : on gagne un facteur l (nombre d'itérations dans l'algorithme des k moyennes).
 - Le « single pass » est également plus adapté aux données en flux (streaming) car il traite les vecteurs dans l'ordre d'arrivée, une seule fois. La mémoire nécessaire est donc en $O(j \cdot p)$: j centres de dimension p en $O(j \cdot p)$, tailles des j clusters en $O(j)$ et sommes en $O(j \cdot p)$.
 - L'algorithme « Single pass » démarre naturellement et crée des classes au besoin via θ . Il n'y a pas besoin de choisir le nombre de cluster k . Mais il est cependant sensible au choix de θ .
 - La qualité du « Single pass » est souvent inférieure à k moyennes (qui optimise une fonction objectif là où « Single Pass » est plus sur une heuristique).
10. Écrire une fonction `dist_max(d_i, centres, theta)` qui vérifie si le document d_i est plus éloigné que θ de tous les centres existants. La fonction retourne `True` si $\delta(d_i, c_k) > \theta$ pour tout c_k , et `False` sinon.
Voir fichier « 6. TD4 - K-moyennes - Correction.py »
11. Écrire une fonction `dist_min(d_i, centres)` qui retourne l'indice ℓ du centre le plus proche de d_i au sens de δ (étape (iii) de l'algorithme « Single Pass »).
Voir fichier « 6. TD4 - K-moyennes - Correction.py »
12. Écrire une fonction `recalcule_centre_sp(centres, l, classe_l)` qui met à jour le centre c_l après l'ajout du document d_i à la classe $\mathcal{A}_l$. Le nouveau centre est la moyenne de tous les documents de $\mathcal{A}_l$.
Voir fichier « 6. TD4 - K-moyennes - Correction.py »
13. Implémenter l'algorithme « Single Pass » complet. La fonction `single_pass(documents, theta)` prend en entrée la liste des documents et le seuil θ , et retourne les centres finaux et les classes.
Voir fichier « 6. TD4 - K-moyennes - Correction.py »