Réduire les couleurs d'une photo

l'algorithme des k-moyennes

Mise en situation. Une photo numérique est formée de millions de **pixels**. Chaque pixel est codé par trois nombres entre 0 et 255 : la quantité de **rouge**, de **vert** et de **bleu** (codage RVB). Cela permet plus de 16 millions de couleurs différentes !
Pour alléger un fichier, obtenir un effet « affiche » ou imprimer avec peu d'encres, on peut décider de ne garder que k couleurs (par exemple 2, 8 ou 16). Mais **lesquelles choisir** pour que la photo reste ressemblante ? Une intelligence artificielle peut les trouver toute seule, sans même savoir ce que représente la photo, grâce à l'algorithme des **k-moyennes** (*k-means* en anglais).

1. Faire des groupes... sans étiquettes

Définition

Avec la méthode des k plus proches voisins, la machine apprenait à partir d'exemples **déjà classés** (des chansons dont on connaissait le genre) : c'est de l'apprentissage **supervisé**.
Avec les k-moyennes, on ne donne à la machine **aucune étiquette** : elle doit former elle-même des **groupes** de données qui se ressemblent. On parle d'apprentissage **non supervisé**. Chaque groupe est représenté par son **centre**, qui est la **moyenne** des données du groupe.

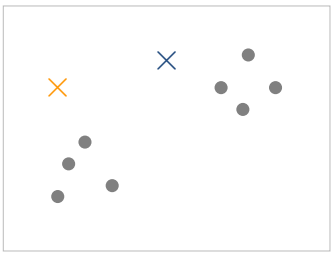
L'algorithme des k-moyennes

On choisit le nombre k de groupes, puis k centres de départ (souvent au hasard).

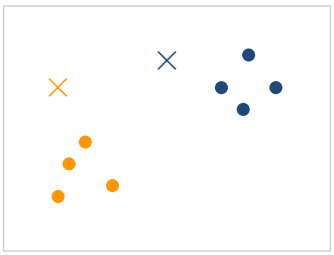
Étape A. Affecter : chaque donnée rejoint le groupe du centre **le plus proche**.

Étape B. Déplacer : chaque centre est remplacé par la **moyenne** des données de son groupe.

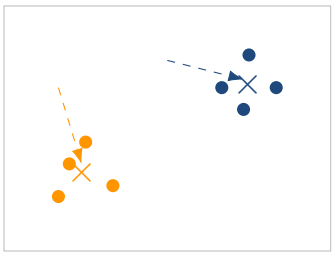
On répète les étapes A et B jusqu'à ce que les groupes **ne changent plus**.



Départ : 2 centres au hasard



A : on affecte



B : on déplace les centres

2. Une petite image en niveaux de gris

Remarque

Dans une image en **niveaux de gris**, chaque pixel est codé par un seul nombre entre 0 (noir) et 255 (blanc). L'**écart** entre deux niveaux de gris est la distance entre ces deux nombres : on soustrait le plus petit au plus grand.

Une image de $4 \times 3 = 12$ pixels

190	210	230	240
200	10	20	220
30	40	50	60

On veut réduire cette image à $k = 2$ niveaux de gris. On choisit comme centres de départ $c_1 = 20$ et $c_2 = 70$. *Pour l'exercice, les centres de départ sont imposés. En pratique, l'ordinateur les choisit au hasard, par exemple en tirant k pixels de l'image.*

▶ À vous de jouer !

Premier tour.

1. **Étape A.** Compléter le tableau : pour chaque pixel, calculer son écart à $c_1 = 20$ et son écart à $c_2 = 70$, puis indiquer son groupe (1 ou 2) : c'est celui du centre le plus proche.

Valeur	10	20	30	40	50	60	190	200	210	220	230	240
Écart à 20												
Écart à 70												
Groupe												

2. **Étape B.** Calculer le nouveau centre de chaque groupe, c'est-à-dire la moyenne des valeurs du groupe.

Coup de pouce

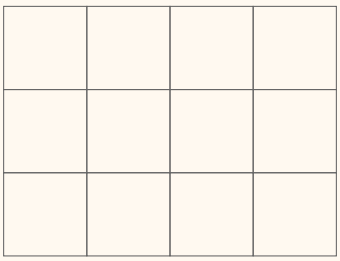
Pour le pixel de valeur 50 : l'écart à 20 vaut $50 - 20 = 30$ et l'écart à 70 vaut $70 - 50 = 20$. Comme $20 < 30$, le pixel est plus proche de c_2 : il va dans le groupe 2.
Pour calculer une moyenne : on additionne toutes les valeurs du groupe, puis on divise par le nombre de valeurs.

▶ À vous de jouer !

Deuxième tour. On reprend avec les nouveaux centres $c_1 = 25$ et $c_2 = 175$.

1. **Étape A.** Quels pixels vont dans le groupe 1 ? Lesquels vont dans le groupe 2 ?
2. **Étape B.** Calculer les nouveaux centres.
3. Faire un troisième tour avec ces nouveaux centres. Les groupes changent-ils ? Que doit faire l'algorithme ?
4. **L'image réduite.** On remplace chaque pixel par le centre de son groupe. Compléter l'image ci-dessous en écrivant la nouvelle valeur de chaque pixel.

Image réduite à 2 niveaux de gris



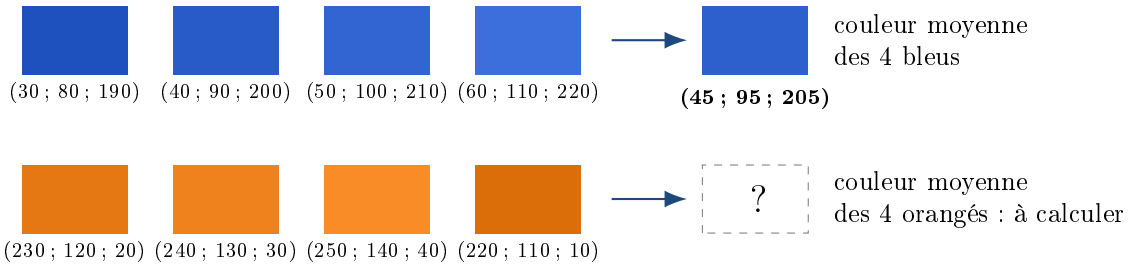
Coup de pouce

Astuce : un pixel est plus proche de c_1 que de c_2 lorsqu'il est situé avant le **milieu** des deux centres. Au deuxième tour, ce milieu vaut $\dfrac{25+175}{2} = \dfrac{200}{2} = 100$.

3. Et en couleurs ?

Définition

En couleurs, chaque pixel est décrit par **trois nombres** $(R\,;V\,;B)$. L'algorithme est exactement le même : on calcule des écarts entre couleurs, puis des moyennes. La moyenne de plusieurs couleurs se calcule **composante par composante** : moyenne des rouges, moyenne des verts, moyenne des bleus.



Exemple

Moyenne des quatre bleus, composante par composante :

$$\text{rouge} : \frac{30+40+50+60}{4} = \frac{180}{4} = 45$$
$$\text{vert} : \frac{80+90+100+110}{4} = \frac{380}{4} = 95$$
$$\text{bleu} : \frac{190+200+210+220}{4} = \frac{820}{4} = 205$$

La couleur moyenne des quatre bleus est donc $(45\,;95\,;205)$.

▶ À vous de jouer !

1. Calculer la couleur moyenne des quatre pixels orangés : $(230\,;120\,;20)$, $(240\,;130\,;30)$, $(250\,;140\,;40)$ et $(220\,;110\,;10)$.
2. Une photo mesure $1\,000 \times 1\,000$ pixels. Chaque pixel occupe 3 octets (un octet par couleur). Combien la photo compte-t-elle de pixels ? Quelle est sa taille en octets ?
3. On réduit la photo à 16 couleurs. Chaque pixel n'a alors besoin que d'un demi-octet (4 bits) pour indiquer laquelle des 16 couleurs il utilise. Quelle est la nouvelle taille de la photo (on néglige la liste des 16 couleurs) ? Combien de fois est-elle plus petite ?

Coup de pouce

Pourquoi 4 bits suffisent-ils ? Avec 4 bits, on peut écrire $2 \times 2 \times 2 \times 2 = 16$ numéros différents (de `0000` à `1111`) : un numéro par couleur.

4. Un peu de Python

Exemple

En Python, les fonctions `sum` et `len` donnent la somme et le nombre d'éléments d'une liste. On peut ainsi calculer un centre :

```
groupe1 = [10, 20, 30, 40]
centre1 = sum(groupe1) / len(groupe1)
print(centre1)
```

Et pour savoir dans quel groupe va un pixel (la fonction `abs` donne l'écart entre deux nombres) :

```
c1 = 25
c2 = 175
v = 60
if abs(v - c1) < abs(v - c2):
    print("groupe 1")
else:
    print("groupe 2")
```

▶ À vous de jouer !

1. Qu'affiche le premier programme ?
2. Compléter les lignes suivantes pour calculer le centre du groupe 2 au premier tour (partie 2) :

```
groupe2 = [...]
centre2 = ...
```

3. Qu'affiche le second programme ? Et si on remplace la ligne `v = 60` par `v = 120` ?

Coup de pouce

Pour `v = 120` : l'écart à `c1` vaut $120 - 25 = 95$ et l'écart à `c2` vaut $175 - 120 = 55$. Lequel est le plus petit ?

5. Pour aller plus loin

Remarque

L'algorithme des k-moyennes est très utilisé : regrouper des clients qui ont les mêmes habitudes d'achat, des articles qui parlent du même sujet, des zones d'une image médicale, etc. Il a deux limites : il faut **choisir** k à l'avance, et le résultat peut dépendre des **centres de départ** choisis au hasard.

★ Question étoilée

On reprend l'image de 12 pixels de la partie 2.

1. Que devient l'image si l'on choisit $k = 1$? Calculer la valeur de l'unique centre.
2. Que devient l'image si l'on choisit $k = 12$? A-t-on gagné de la place ?
3. Expliquer pourquoi le choix de k est un **compromis**.