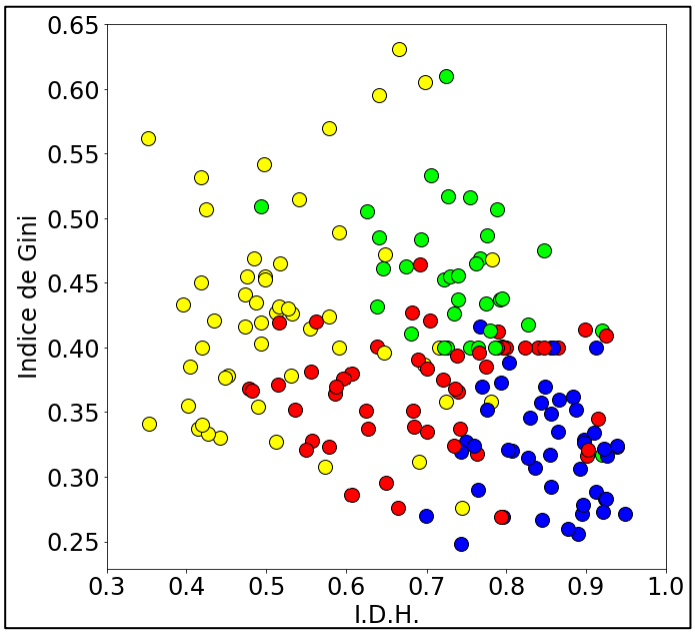
Algorithme des k plus proches voisins

Comprendre le problème posé



Voici un graphique représentant 186 pays avec :

- en abscisse l'IDH (indice de développement humain),
- en ordonnée l'indice de Gini qui mesure les inégalités économiques au sein d'un pays (indice de Gini élevé : inégalités importantes),
- en couleur le continent du pays (vert : Amériques, jaune : Afrique, bleu : Europe, rouge : Asie et Océanie).

Les données peuvent être représentées sous forme d'une table contenant 186 enregistrements ayant cette forme là : `{'idh' : 0.897, 'gini' : 0.326, 'continent' : 'europe'}`
Chaque enregistrement contient ce qu'on appellera pour toute la suite :

- N = 2 descripteurs `'idh'` et `'gini'` qui donnent la position du point,
- une unique *étiquette* `'continent'` qui donne la couleur du point.

Le problème de *classification* auquel l'algorithme des k plus proches voisins tente d'apporter une réponse est le suivant (les trois formulations sont équivalentes) :

Un nouveau pays a pour IDH 0,55 et pour indice de Gini 0,40. Quel est son continent ?
Un nouveau point a pour position (0,55 ; 0,40). Quelle est sa couleur ?"
Un nouvel enregistrement a pour descripteurs `'idh':0.55` et `'gini':0.40`. Quelle est son étiquette ?

Un humain peut rapidement répondre : "Sans doute rouge (Asie-Océanie) ou jaune (Afrique)". On peut dès lors se demander quel est l'intérêt d'avoir un algorithme pour faire cela. L'intérêt est triple :

- Automatiser la réponse,
- Répondre rapidement à des milliers ou millions de questions du même type,
- Généraliser à des cas avec plus de deux descripteurs

Rappel mathématique

La distance entre deux points dont les positions sont $(x_A; y_A)$ et $(x_B; y_B)$ est donnée par la formule :

$$dist(A,B) = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

Si on a trois descripteurs numériques pour chaque point : $(x_A; y_A; z_A)$ et $(x_B; y_B; z_B)$ on obtient :

$$dist(A,B) = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2}$$

Cela se généralise à quatre descripteurs, cinq descripteurs ... et N descripteurs.

Présentation de l'algorithme

Commencer par regarder ce gif animé : algo_1_sur_2.gif .

Appelons `pays_A` le dictionnaire du pays dont on cherche l'étiquette et `table_pays` la table des dictionnaires des pays dont on connait les étiquettes. Voici une proposition d'algorithme :

```
(1) créer une copie table_pays_distances  de la table table_pays (*)
(2) pour chaque dictionnaire pays_B dans table_pays_distances :
(3)     calculer la distance d_AB entre le pays_B et le pays_A
(4)     ajouter la paire 'distance':d_AB au dictionnaire pays_B
(5)
(6) trier la table_pays_distances par distance croissante
(7) créer k_plus_proches_continents = liste des k premiers continents de cette table
(8) attribuer à pays_A le continent majoritaire de la liste k_plus_proches_continents
```

Avec `k = 5` (5 plus proches voisins) et `pays_A = {'idh':0.55, 'gini':0.40}` on obtient :

```
table_pays_distances au début de l'algorithme (1)
[{'continent': 'asia', 'idh': 0.824, 'gini': 0.4},
 {'continent': 'europe', 'idh': 0.848, 'gini': 0.37},
 {'continent': 'asia', 'idh': 0.701, 'gini': 0.384},
 {'continent': 'asia', 'idh': 0.736, 'gini': 0.368},
 {'continent': 'asia', 'idh': 0.774, 'gini': 0.385},
 {'continent': 'americas', 'idh': 0.722, 'gini': 0.4},
 {'continent': 'europe', 'idh': 0.89, 'gini': 0.256}, etc.
```

```
table_pays_distances après l'ajout du champ 'distance' (5)
[{'continent': 'asia', 'idh': 0.824, 'gini': 0.4, 'distance': 0.274},
 {'continent': 'europe', 'idh': 0.848, 'gini': 0.37, 'distance': 0.3},
 {'continent': 'asia', 'idh': 0.701, 'gini': 0.384, 'distance': 0.152},
 {'continent': 'asia', 'idh': 0.736, 'gini': 0.368, 'distance': 0.189},
 {'continent': 'asia', 'idh': 0.774, 'gini': 0.385, 'distance': 0.225},
 {'continent': 'americas', 'idh': 0.722, 'gini': 0.4, 'distance': 0.172},
 {'continent': 'europe', 'idh': 0.89, 'gini': 0.256, 'distance': 0.369}, etc.
```

```
table_pays_distances une fois triée selon la distance croissante (6)
[{'continent': 'africa', 'idh': 0.555, 'gini': 0.415, 'distance': 0.016},
 {'continent': 'asia', 'idh': 0.556, 'gini': 0.381, 'distance': 0.02},
 {'continent': 'asia', 'idh': 0.563, 'gini': 0.42, 'distance': 0.024},
 {'continent': 'africa', 'idh': 0.531, 'gini': 0.378, 'distance': 0.029},
 {'continent': 'africa', 'idh': 0.533, 'gini': 0.426, 'distance': 0.031},
 {'continent': 'africa', 'idh': 0.527, 'gini': 0.43, 'distance': 0.038},
 {'continent': 'africa', 'idh': 0.579, 'gini': 0.424, 'distance': 0.038}, etc.
```

```
liste k_plus_proches_continents (7)
['africa', 'asia', 'asia', 'africa', 'africa']
```

```
On peut prédire que pays_A a pour étiquette 'continent' : 'africa' (8)
```

(*) : Si la table de données est lourde il faut éviter la recopie intégrale et dans ce cas procéder un peu différemment.

Nous verrons en TP des exemples plus utiles : en botanique et en médecine. Plus généralement, les problèmes de *classification* sont très nombreux et font l'objet de recherches intenses à l'ère du big data, du développement des IA et des techniques d'apprentissage. La question plus générale que l'on cherche à résoudre est celle-ci :

Comment à partir de données, en déduire une étiquette qui m'intéresse ?

- Quelle est la classe de cet animal sur cette photo ? chat ? chien ? perroquet ?
- Quel est, au vu de ses données médicales, le diagnostic que l'on peut faire sur ce patient ? malade ou pas ?
- Quel est, au vu de sa navigation web, le profil de ce consommateur ? intéressé ou pas par mes produits ?
- Quel est, au vu de son dossier administratif, le profil de ce contribuable ? fraudeur ou pas ?

Rappel : "Coller une étiquette à quelqu'un" : catégoriser, attribuer grossièrement une appartenance, classer socialement, politiquement .

Pour finir, regarder la seconde image animée : algo_2_sur_2.gif