

VI – Algorithme des K plus proches voisins

M. Solnon

PT, Lycée Gustave Eiffel, Dijon

6 octobre 2026

Cadre : problème de classification

Données / échantillon

- Objets $X_0, \dots, X_{n-1}$
- Étiquettes $e_0, \dots, e_{n-1} \in \mathcal{E}$
- Échantillon : $\mathcal{S} = \{(X_0, e_0), \dots, (X_{n-1}, e_{n-1})\}$
- $\mathcal{E}$: ensemble fini ($\text{Card}(E) \ll n$)

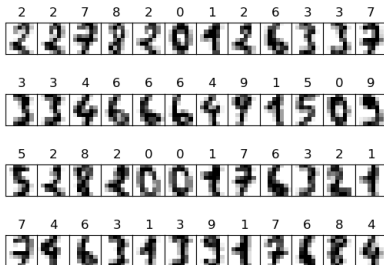
Hypothèse

Si X_i et X_j sont « proches », e_i et e_j ont tendance à être égales.

Objectif

Étant donné un nouvel objet Y , prédire son étiquette « la plus probable », au vu de l'échantillon.

Exemple : détection de chiffres



Problème

- $d = 64$, $n = 1797$, $\text{Card}(E) = 10$
- Classe digits de sklearn.datasets

Exemple : détection de visages



Problème

- $d = 5828$, $n = 13233$, $\text{Card}(E) = 5749$
- Classe lfw de sklearn.datasets
- Extrêmement difficile

Autres exemples de sklearn.datasets

Détection de cancers du sein

- Décrit géométriquement les cellules d'une biopsie
- $d = 30$, $n = 569$, $\text{Card}(E) = 2$

Articles de journaux

- $d = ??$, $n = 18846$, $\text{Card}(E) = 20$
- Possibilité de transformer les articles en vecteurs ($d > 30000$)

Détection d'attaques informatiques (KDD Cup 1999)

- $d = 41$ (dont quelques variables discrètes), $n = 4\,898\,431$, $\text{Card}(E) = 2$

Quelques hypothèses sous-jacentes

Données

- Peuvent appartenir à un espace $\mathcal{D}$ « trop gros »
- Nos méthodes ne vont fonctionner que pour des données vectorielles ($\mathbb{R}^d$ avec d « raisonnable »)

Prétraitement

- Fonction (de description) $f : \mathcal{D} \rightarrow \mathbb{R}^d$
- Cachée ou inexistante dans le cadre de ce cours
- Étape importante et potentiellement difficile

Qualité des données

On n'apprend que ce qui est contenu dans les données

Choix de la géométrie

Distance euclidienne usuelle

$$X = \begin{pmatrix} x_0 \\ \vdots \\ x_{d-1} \end{pmatrix}, Y = \begin{pmatrix} y_0 \\ \vdots \\ y_{d-1} \end{pmatrix} \in \mathbb{R}^d$$

$$d(X, Y) = \sqrt{\sum_{i=0}^{d-1} (x_i - y_i)^2} = \sqrt{(X - Y)^\top (X - Y)}.$$

Complexité

Calcul de $d(X, Y)$ en temps $\mathcal{O}(d)$.

Algorithme des K plus proches voisins

Principe

Affecter à un nouvel objet Y l'étiquette majoritaire parmi celles de K objets de l'échantillon les plus proches de Y .

Classificateur

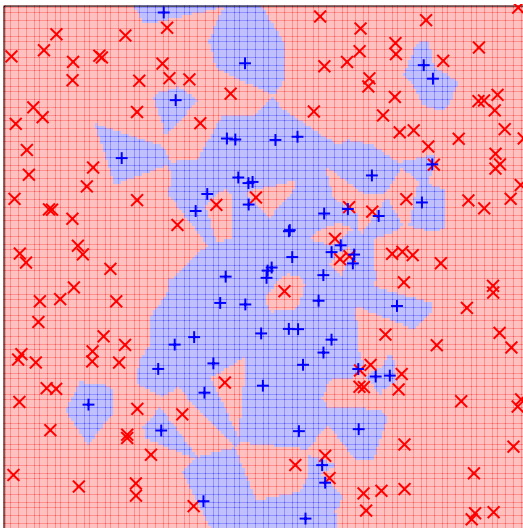
On construit ainsi une fonction $\hat{f} : \mathbb{R}^d \rightarrow \mathcal{E}$.

Demande de savoir :

- récupérer les K plus proches voisins (donc de trier l'échantillon par distance croissante avec Y) ;
- réaliser un vote parmi les étiquettes des K plus proches voisins.

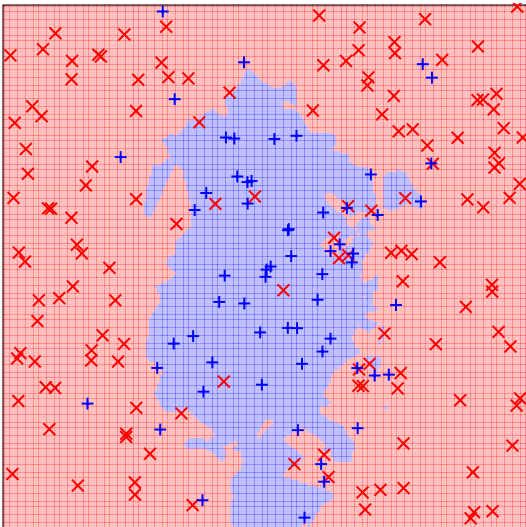
Résultats sur notre exemple

$K = 1$



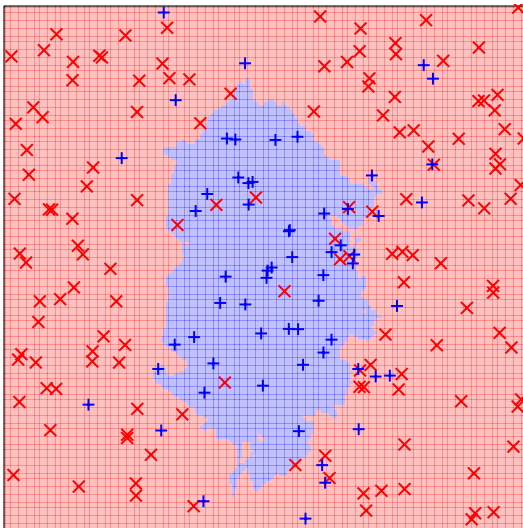
Résultats sur notre exemple

$K = 5$



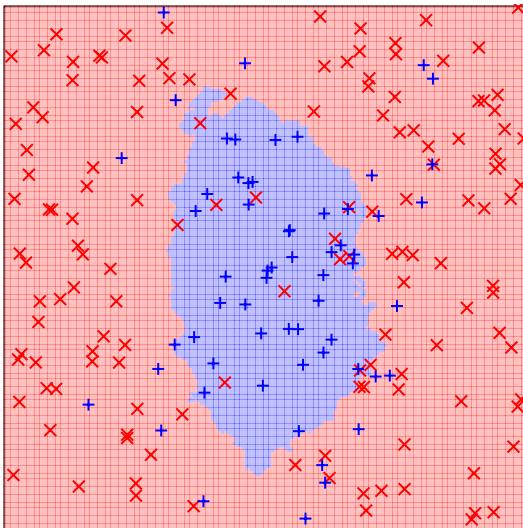
Résultats sur notre exemple

$K = 10$



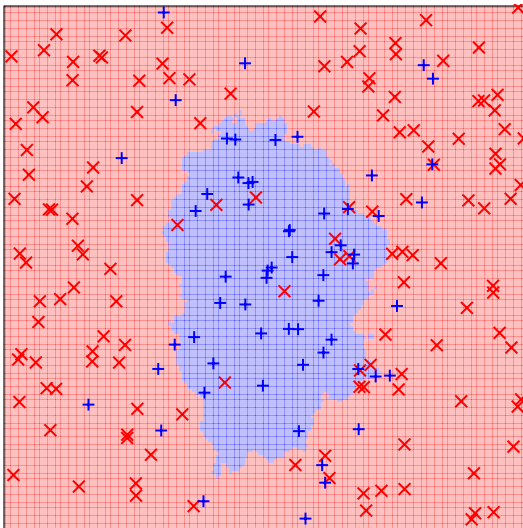
Résultats sur notre exemple

K = 20



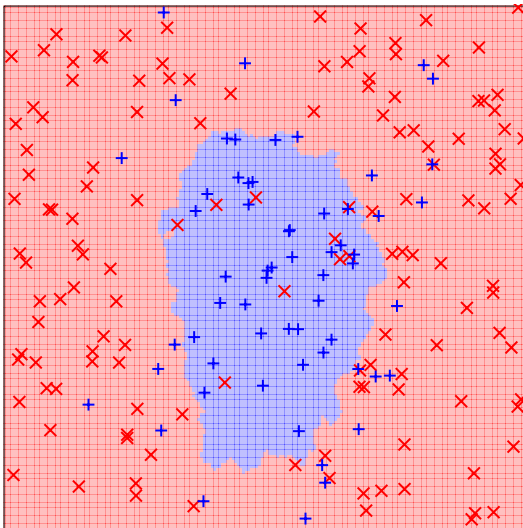
Résultats sur notre exemple

$K = 30$



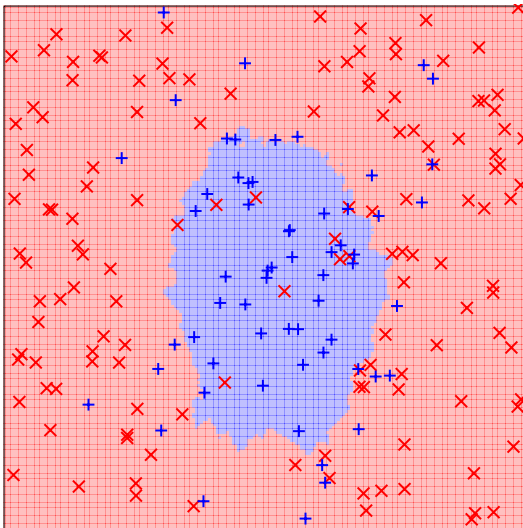
Résultats sur notre exemple

$K = 40$



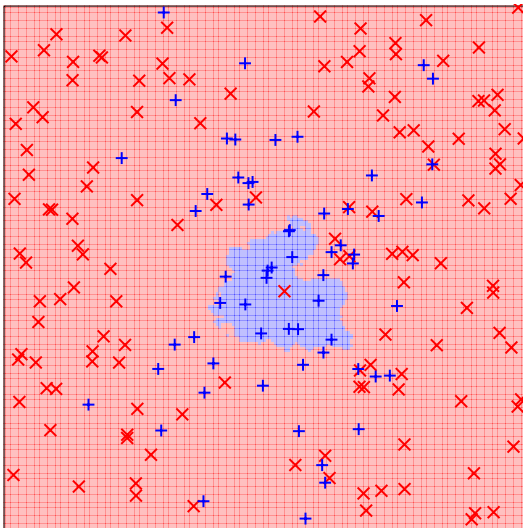
Résultats sur notre exemple

$K = 50$



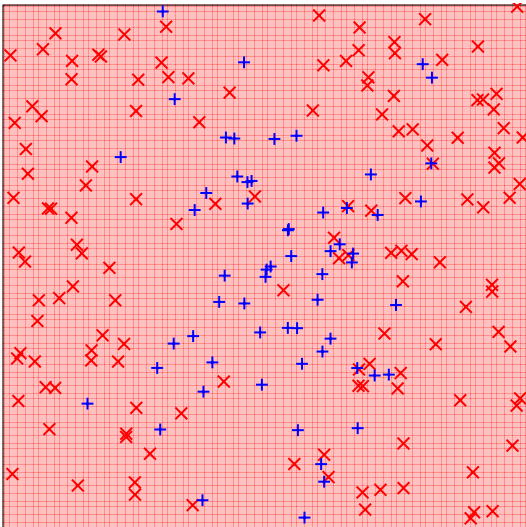
Résultats sur notre exemple

K = 100



Résultats sur notre exemple

K = 200



Calcul de la distance

Fonction $d(p1, p2)$:

$d \leftarrow$ dimension de $p1$

$s \leftarrow 0$

Pour i variant de 0 à $d-1$, **Faire**

$s \leftarrow s + (p1[i] - p2[i])^2$

Renvoyer $\sqrt{s}$

Fin Fonction

Remarque

Peut se faire en une expression si les données sont des vecteurs
numpy

Complexité

$\mathcal{O}(d)$

Format des données

Notre convention

- `listeX` : la liste des vecteurs $[X_0, \dots, X_{n-1}]$;
- `listeEtiq` : la liste des étiquettes $[e_0, \dots, e_{n-1}]$, où e_i est l'étiquette la donnée X_i .

Même convention qu'en TP de SII

Tri rapide de l'échantillon (début)

Fonction TRI_RAPIDE(listeX, listeEtiq, Y):

Si listeX a au plus 1 élément **Alors**

Renvoyer listeX, listeEtiq

Sinon

 piv, n $\leftarrow$ listeX[0], taille de l'échantillon

 lX1, lX2, lE1, lE2 $\leftarrow$ listes vides

Pour i variant de 1 à n-1, **Faire**

Si $d(\text{listeX}[i], Y) \leq d(\text{piv}, Y)$ **Alors**

 ajouter listeX[i] à lX1

 ajouter listeEtiq[i] à lE1

Sinon

 ajouter listeX[i] à lX2

 ajouter listeEtiq[i] à lE2

Tri rapide de l'échantillon (fin)

```
lX1, lE1 ← TRI_RAPIDE(lX1, lE1, Y)
lX2, lE2 ← TRI_RAPIDE(lX2, lE2, Y)
listeXtriee ← Concat. de lX1, [piv], LX2
listeEtriee ← Concat. de lE1, [listeE[0]], LE2
Renvoyer listeXtriee, listeEtriee
```

Fin Fonction

Exercice

Consigne

Écrire ce tri en utilisant l'algorithme de tri fusion.

Clef de valeur maximale dans un dictionnaire

Fonction clef_maxi(d) :

*## Précondition : les valeurs du dictionnaire d sont
des entiers positifs*

 cmax $\leftarrow \emptyset$

 vmax $\leftarrow -1$

Pour chaque clef **De** d, **Faire**

Si d[clef] > vmax **Alors**

 cmax $\leftarrow$ clef

 vmax $\leftarrow$ d[clef]

Renvoyer cmax

Fin Fonction

Vote parmi les K plus proches voisins

Fonction vote(listeEtiquettesTriees, K):

*## Précondition : echantillon trié par distance
croissante par rapport au point*

$d \leftarrow$ dictionnaire vide
 Pour chaque etiquette **De** K premiers éléments
de listeEtiquettesTriees, **Faire**

Si etiquette est déjà une clef de d **Alors**
 $d[\text{etiquette}]$ incrémenté de 1
 Sinon
 $d[\text{etiquette}] \leftarrow 1$
 Renvoyer clef_maxi(d)

Fin Fonction

Fonction finale

```
Fonction knn(listeX, listeEtiquettes, Y, K):  
|  _, listeEtiquettesTriees ← trie(listeX,  
listeEtiquettes, Y)  
|  Renvoyer vote(listeEtiquettesTriees, K)  
Fin Fonction
```

Exercice

Données

- `fleches_donnees.py`
- $X_i \in \mathbb{R}^{256}$ (images 16×16 « empilées » en niveaux de gris)
- étiquettes : haut, bas, gauche, droite
- fonction `affiche` : affiche une image
- deux échantillons (entraînement et test)

Consignes

- Implémenter en Python les algorithmes vus ci dessus.
- Pour $K = 1$: étiquette prédite pour `Xtest[0]` ? Et pour `Xtest[1]` ? Sont-elles correctes ?
- Obtenir la liste des erreurs.

Et maintenant ?

Question restantes

- Évaluation de la qualité du classificateur f ainsi construit.
- Choix du paramètre K en fonction des données.

Critère d'évaluation : erreur de classification

- On suppose que $(X_0, e_0), \dots, (X_{n-1}, e_{n-1})$ sont indépendants et suivent une même loi P .
- Si $(X, e) \sim P$ est indépendant de l'échantillon, minimiser

$$\text{Err}(f) = P(f(X) \neq e).$$

- Fonction f^* minimisant ceci : *classificateur de Bayes*.

Erreur empirique

Erreur empirique

Si f est un classificateur, sur un échantillon

$\mathcal{S} = \{(X_0, e_0), \dots, (X_{n-1}, e_{n-1})\}$:

$$\widehat{\text{Err}}(f, \mathcal{S}) = \frac{1}{n} \sum_{i=0}^{n-1} \mathbf{1}_{f(X_i) \neq e_i} = \frac{\text{Card} \{ (X, e) \in \mathcal{S} \mid f(X) \neq e \}}{\text{Card}(\mathcal{S})}$$

Conséquence de la loi des grands nombres

Si f et $\mathcal{S}$ sont indépendants

$$\widehat{\text{Err}}(f, \mathcal{S}) \xrightarrow[n \rightarrow +\infty]{\mathbb{P}} \text{Err}(f).$$

Mais... $\hat{f}$ n'est pas indépendant de $\mathcal{S}$

Entraînement et test

Règle d'or

Ne jamais évaluer la qualité d'un processus sur les données qui ont servi à le construire.

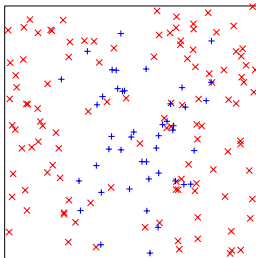
Solution : séparation de l'échantillon

- Séparer $\mathcal{S}$ en deux échantillons $\mathcal{S}_E$ (classiquement $\approx 80\%$ des données) et $\mathcal{S}_T$ (classiquement $\approx 20\%$ des données)
- Construire le classificateur $\hat{f}$ sur l'échantillon $\mathcal{S}_E$ (échantillon d'entraînement)
- Calculer $\widehat{\text{Err}}(\hat{f}, \mathcal{S}_T)$ (échantillon de test)

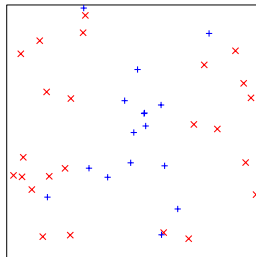
Étape de calibration

Choisir K qui minimise $\widehat{\text{Err}}(\hat{f}, \mathcal{S}_T)$.

Exemple : séparation entraînement / test



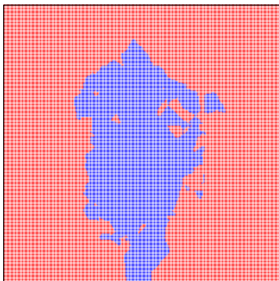
(a) Échantillon d'entraînement



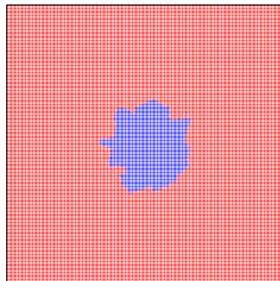
(b) Échantillon de test

Exemple : entraînement

K = 5

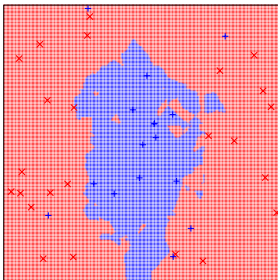


K = 60

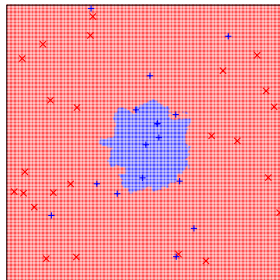


Exemple : évaluation

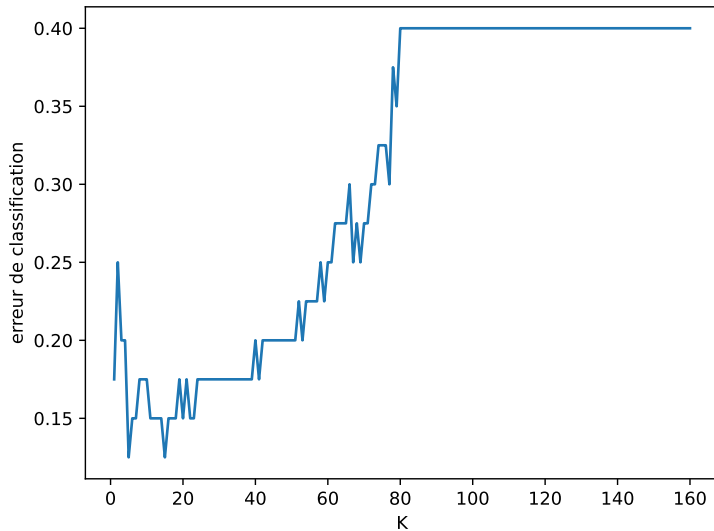
K = 5, erreurs : 12.5 %



K = 60, erreurs : 25.0 %



Exemple : calibration



Matrice de confusion

Définition

Si $\mathcal{E} = \{E_1, \dots, E_p\}$

$$M_{i,j} = \text{Card} \{ k \in \llbracket 0, n \rrbracket \mid e_k = E_i \text{ et } f(X_k) = E_j \}.$$

Interprétation

$M_{i,j}$: compte le nombre d'éléments de l'échantillon d'étiquette i classifiés en l'étiquette j .

Exemple

$$K = 5 : \begin{pmatrix} 24 & 0 \\ 5 & 11 \end{pmatrix}$$

$$K = 60 : \begin{pmatrix} 24 & 0 \\ 10 & 6 \end{pmatrix}.$$

Utilisation

- Nombre d'erreurs : $\sum_{0 \leq i \neq j < p} M_{i,j}$
- Nombre de prédictions correctes : $\sum_{i=0}^{p-1} M_{i,i}$
- Nombre de données d'étiquette i : $\sum_{j=0}^{p-1} M_{i,j}$
- Nombre de données classifiées en j : $\sum_{i=0}^{p-1} M_{i,j}$

Exercice

Pour chacune des matrices de confusion suivantes, déterminer

- le nombre d'étiquettes ;
- la taille de l'échantillon de test ;
- le nombre d'observations de chaque étiquette dans l'échantillon de test (on pourra les numéroter à partir de 0) ;
- l'erreur de classification.

1. $A = \begin{pmatrix} 10 & 2 & 1 \\ 3 & 8 & 4 \\ 2 & 7 & 1 \end{pmatrix}$

2. $B = \begin{pmatrix} 1 & 4 & 0 & 3 \\ 5 & 10 & 1 & 0 \\ 1 & 5 & 7 & 2 \\ 1 & 6 & 3 & 7 \end{pmatrix}$

Exercice

On considère une tâche de classification à trois étiquettes, et l'on construit deux classificateurs f_1, f_2 dessus, dont on donne les matrices de confusion, calculées sur un même échantillon de test.

$$M_1 = \begin{pmatrix} 10 & 2 & 3 \\ 0 & 7 & 1 \\ 2 & 1 & 5 \end{pmatrix}, \quad M_2 = \begin{pmatrix} 11 & 1 & 3 \\ 2 & 2 & 4 \\ 1 & 0 & 7 \end{pmatrix}.$$

Quel classificateur choisiriez-vous ?