Stage LIESSE
Introduction à l'IA
A. Apprentissage Supervisé

Thomas Bonald

Avril 2021

Objectif

Prédire la **classe** (classification) ou la **valeur** (régression) d'un échantillon à partir d'exemples connus.

En pratique, il s'agit d'apprendre une fonction $f : x \mapsto y$ minimisant:

$$\frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) + \Omega(f)$$

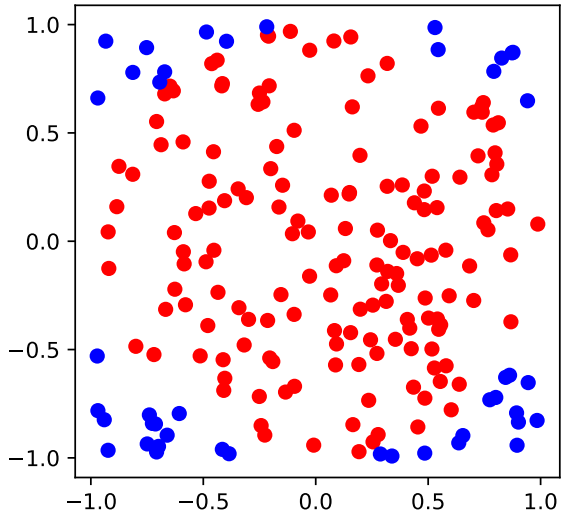
où

- $x \in \mathbb{R}^d$
- $y \in \{0, 1\}, \{1, \ldots, K\}$ ou $\mathbb{R}$
- $(x_1, y_1), \ldots, (x_n, y_n)$ sont les données d'apprentissage
- ℓ est la fonction de perte
- Ω est une fonction de régularisation (optionnelle)

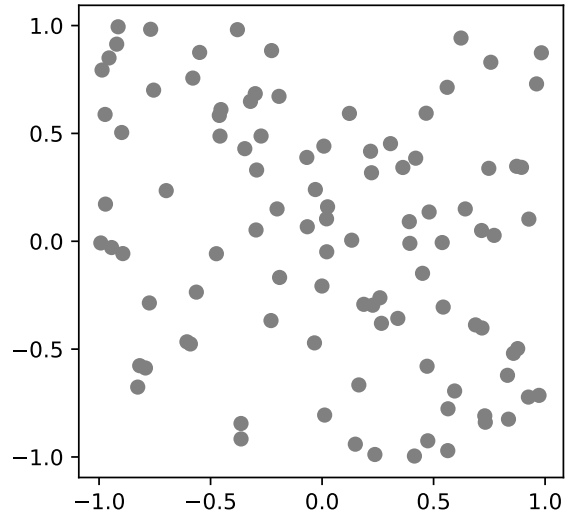
Exemple (classification)

$$x \in \mathbb{R}^2,\ y \in \{0, 1\}$$

Données d'entrainement
($n = 200$)



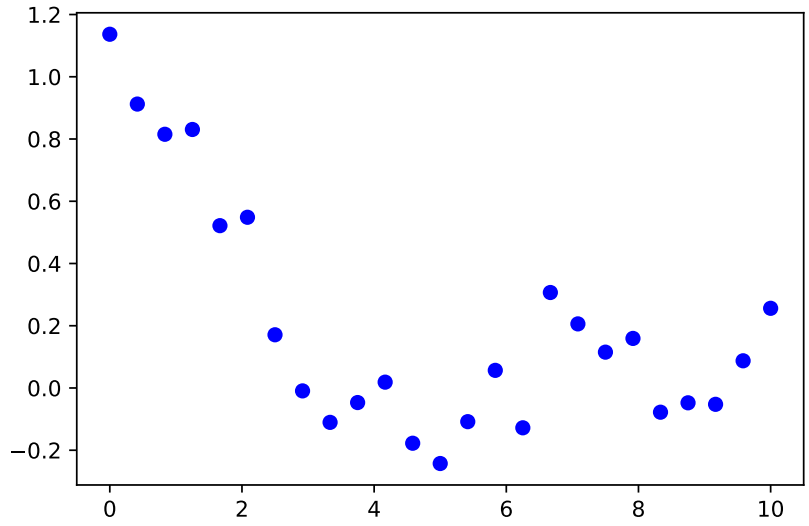
Données de test
($n' = 100$)



Exemple (régression)

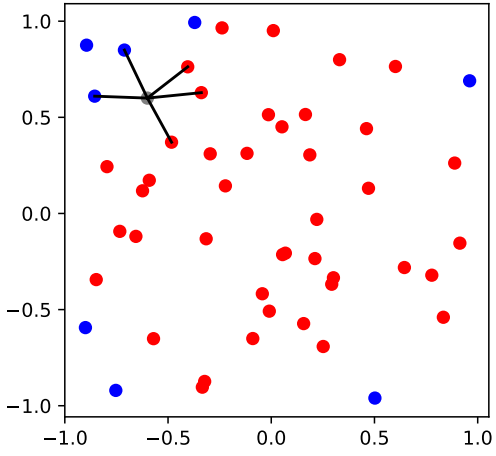
$$x \in \mathbb{R},\ y \in \mathbb{R}$$

Données d'entrainement
$(n = 25)$

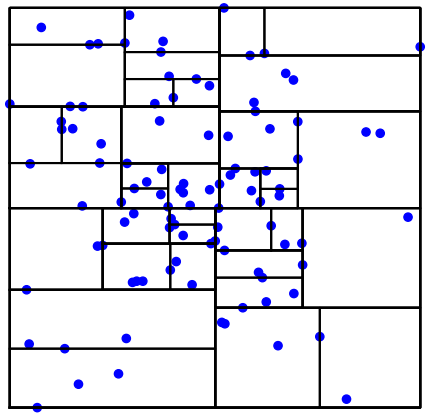


Plan

1. Plus proches voisins



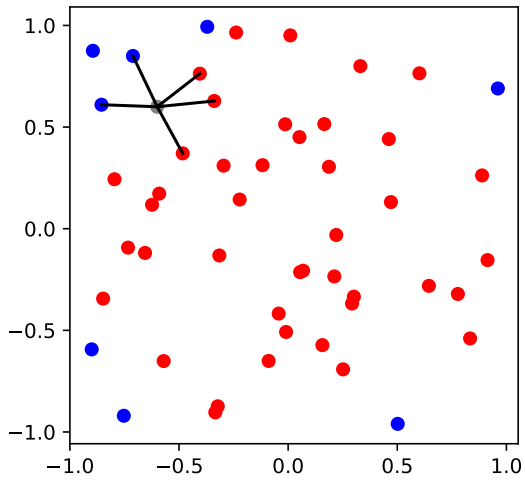
2. Recherche arborescente



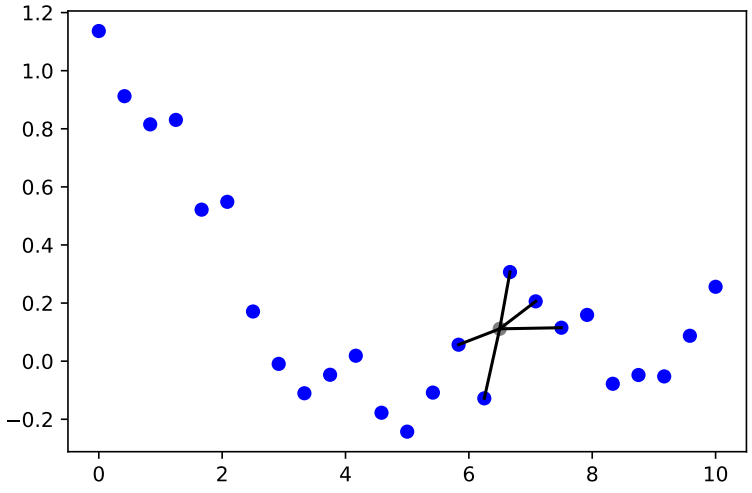
Algorithme des k plus proches voisins

Une méthode **heuristique**, simple et intuitive.

Classification
Vote majoritaire

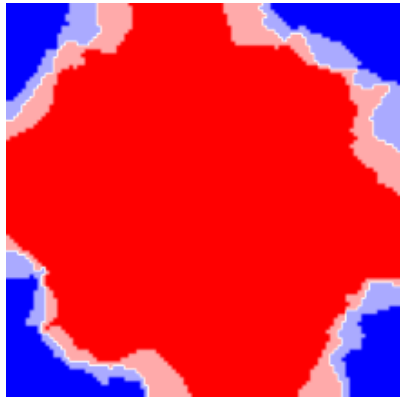


Régression
Moyenne

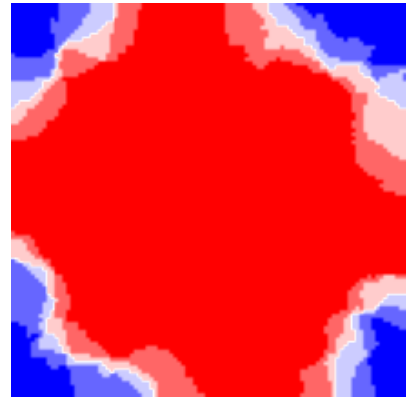


Exemple en classification

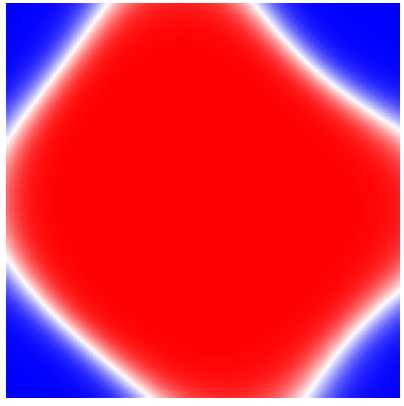
$k = 3$



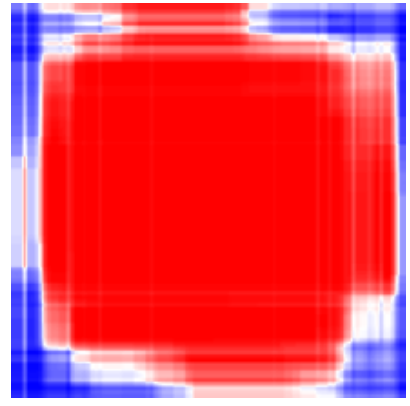
$k = 5$



SVM

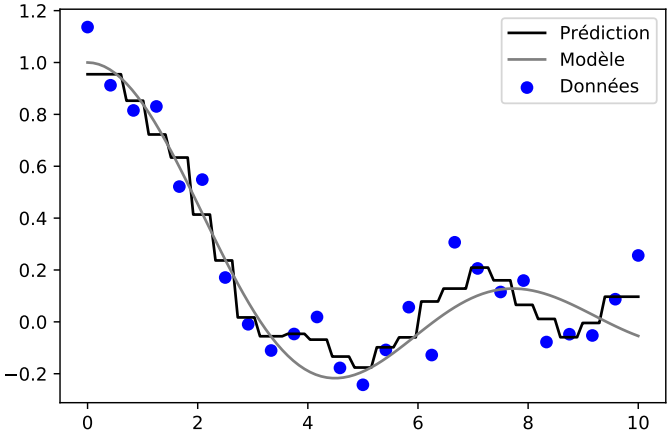


Forêt aléatoire

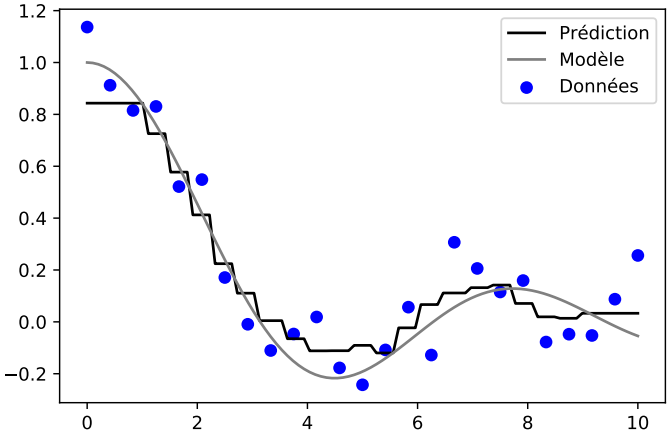


Exemple en régression

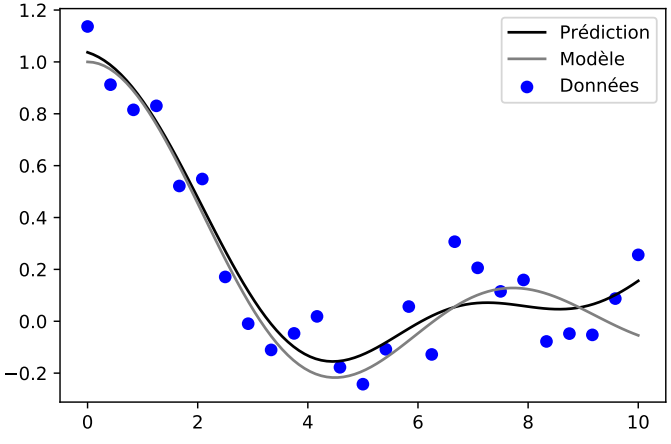
$k = 3$



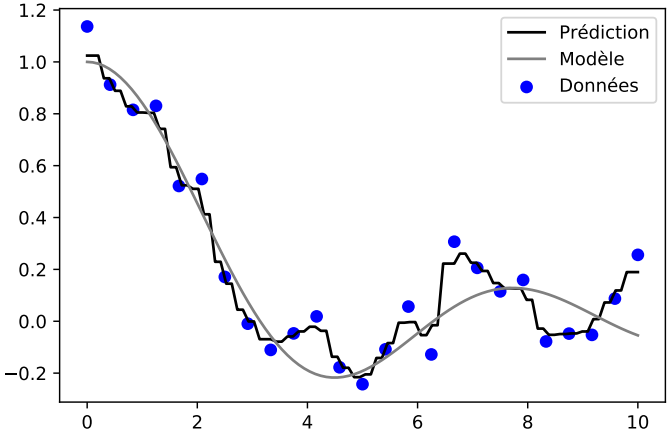
$k = 5$



SVM

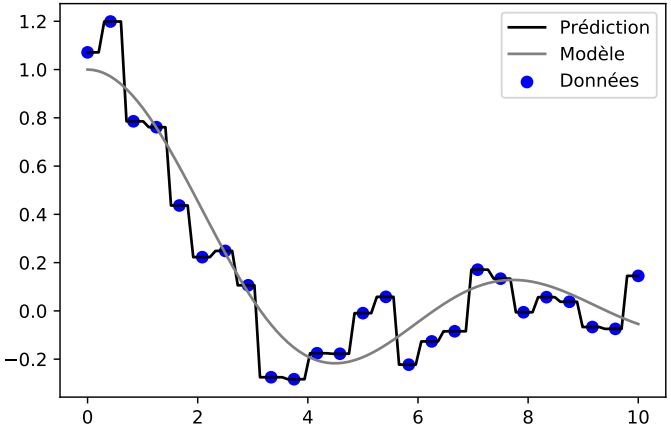


Forêt aléatoire

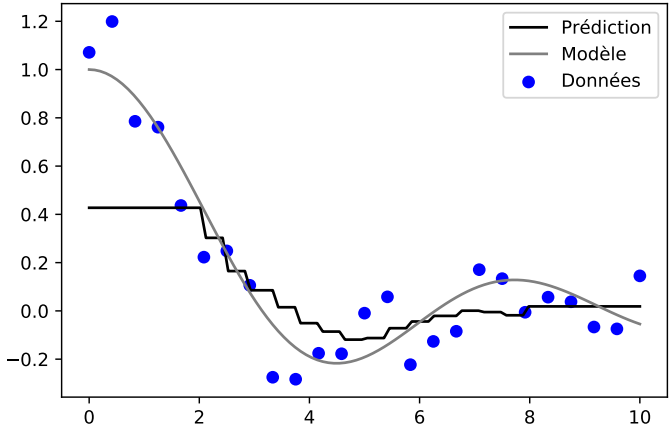


Dilemne biais-variance

$k = 1$

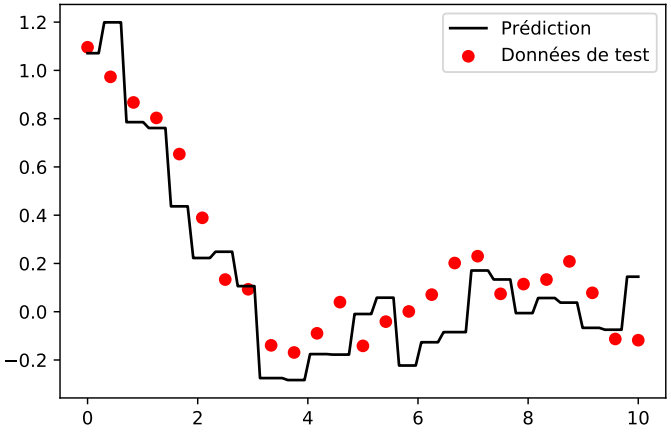


$k = 10$



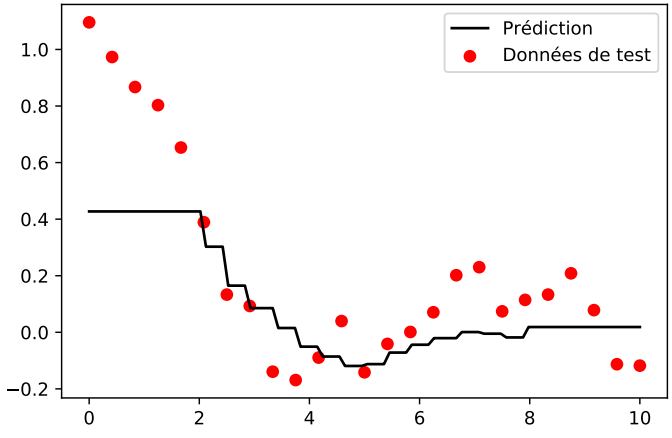
Variance élevée

Sur-apprentissage



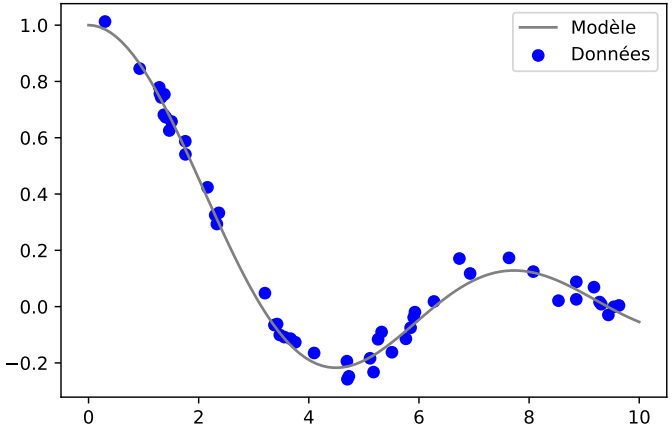
Biais élevé

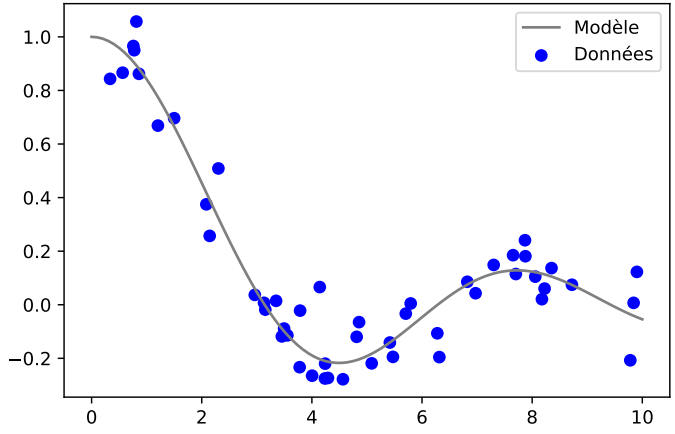
Sous-apprentissage



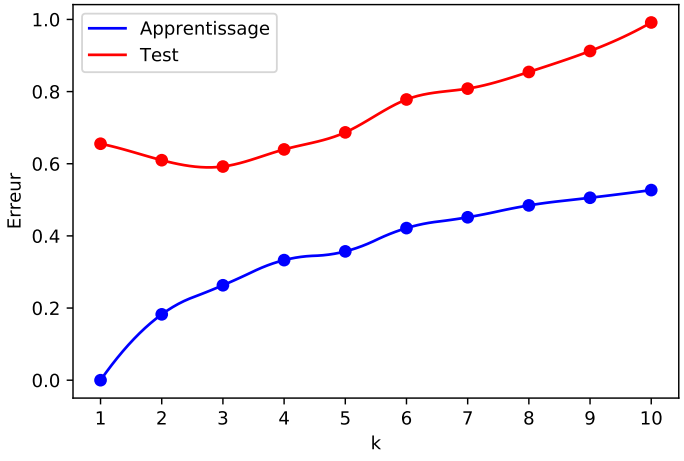
Quelle valeur de k ?

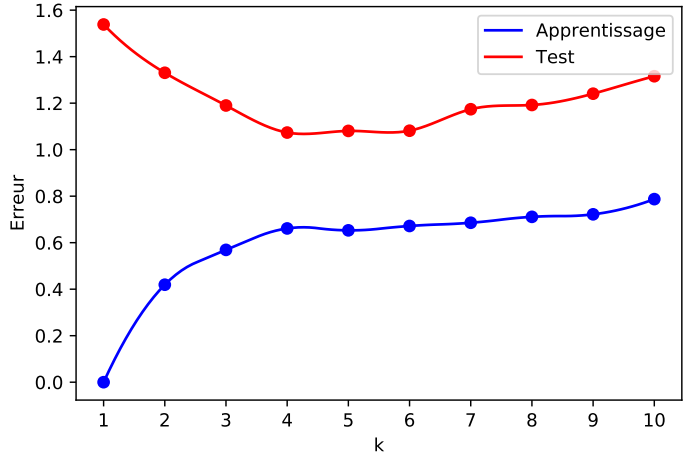
Données





Prédiction





Application: Digits

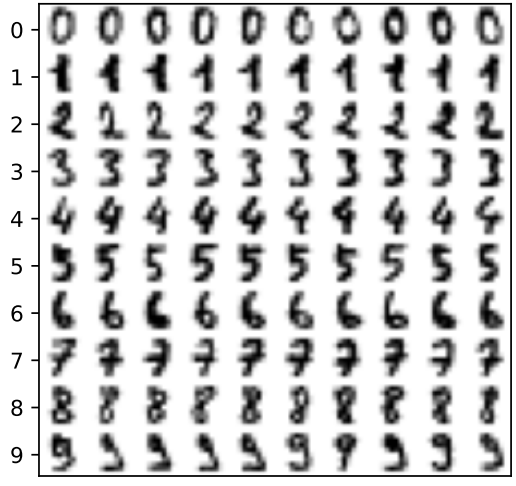
1797 images de chiffres

8×8 pixels

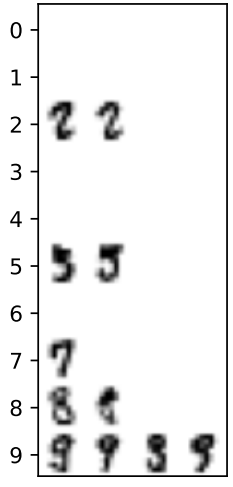
Apprentissage sur 50% des exemples

Précision = 98% (pour $k = 1$)

Apprentissage

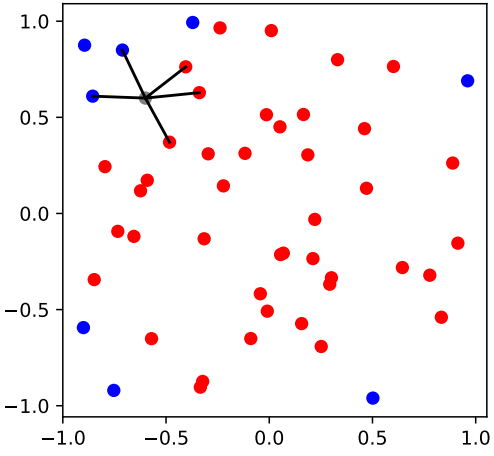


Erreurs sur le test

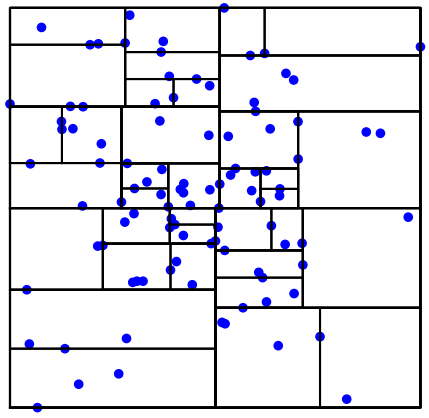


Plan

1. Plus proches voisins



2. Recherche arborescente



Recherche dans un tableau (1D)

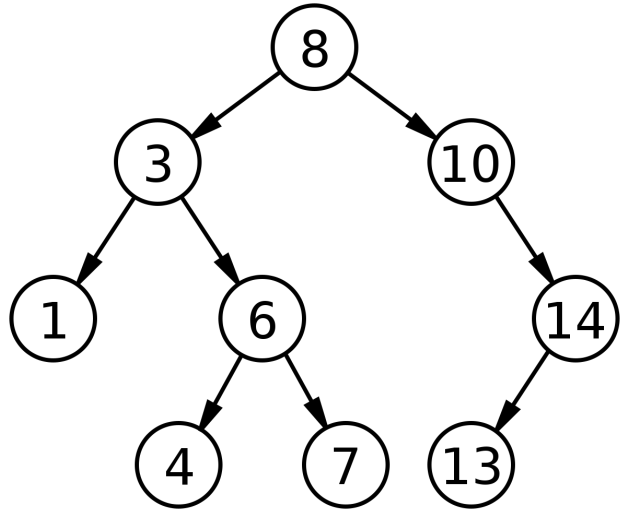
Données

8	3	1	6	10	14	4	13	7

Requête

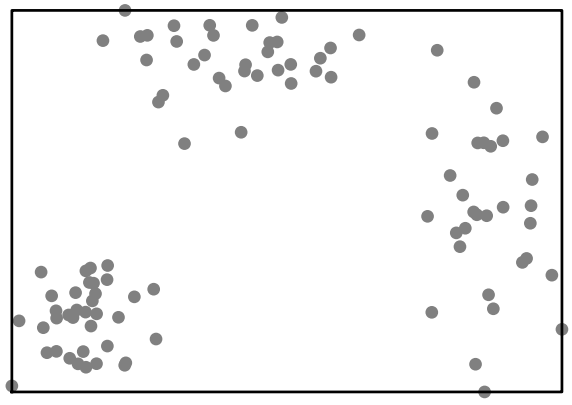
5

Arbre binaire de recherche

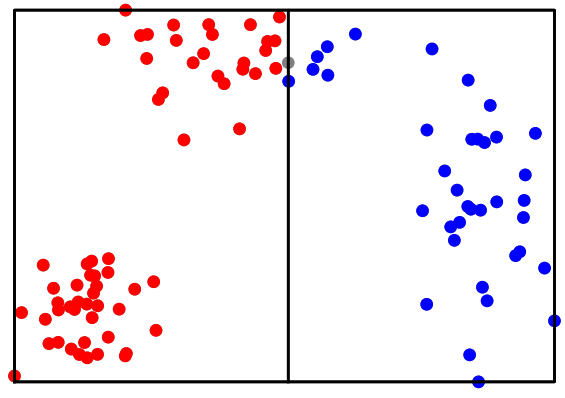


Arbre k-dimensionnel

Données

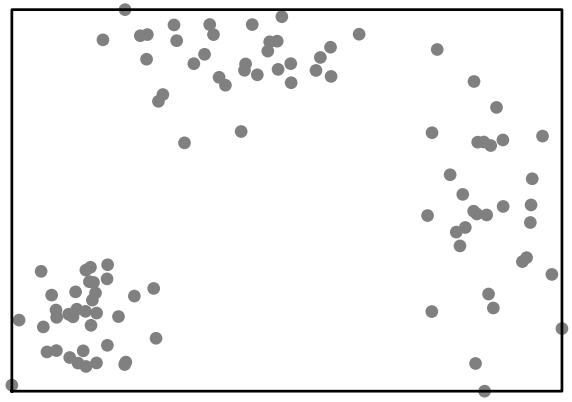


Première coupe

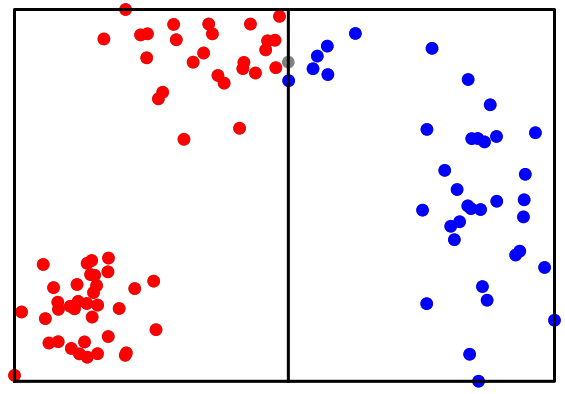


Arbre k-dimensionnel

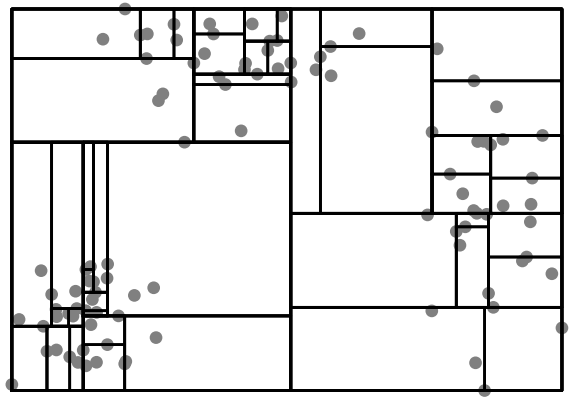
Données



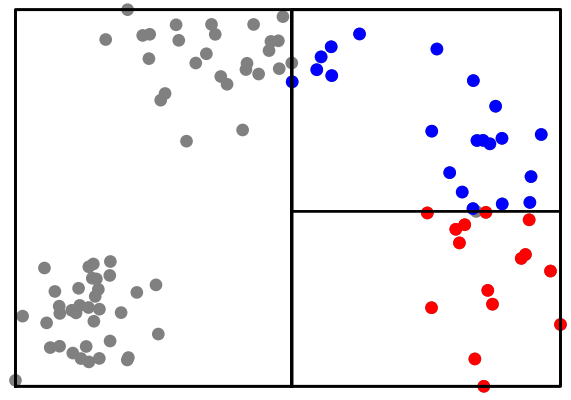
Première coupe



Partition finale

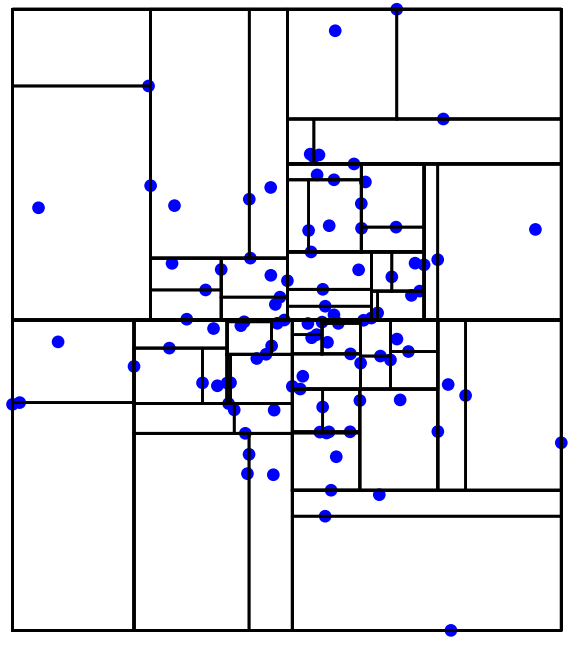


Seconde coupe

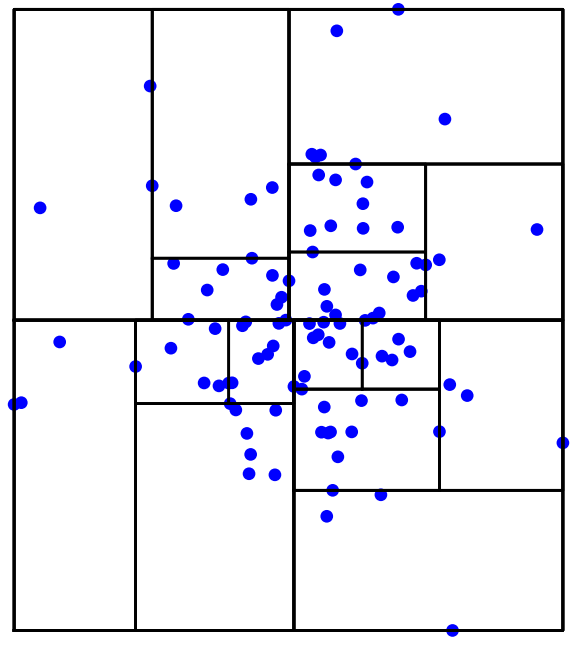


Élagage

Arbre complet
Taille des feuilles $= 1$

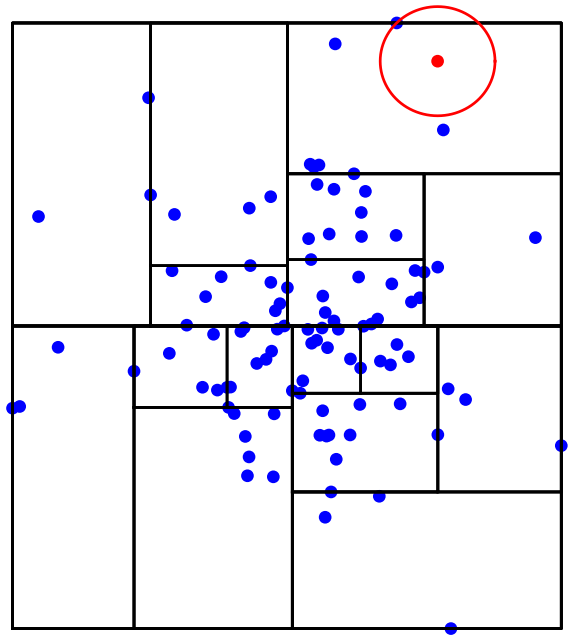


Arbre élagué
Taille des feuilles ≤ 10

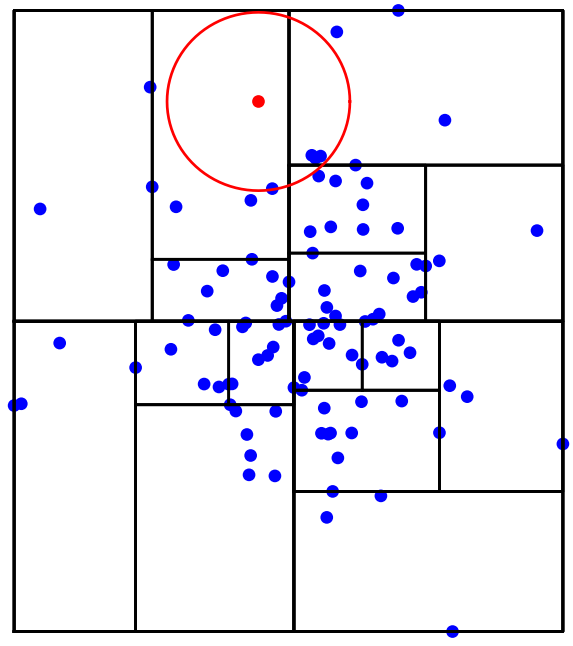


Recherche dans un arbre k-dimensionnel

Cas favorable
Recherche locale

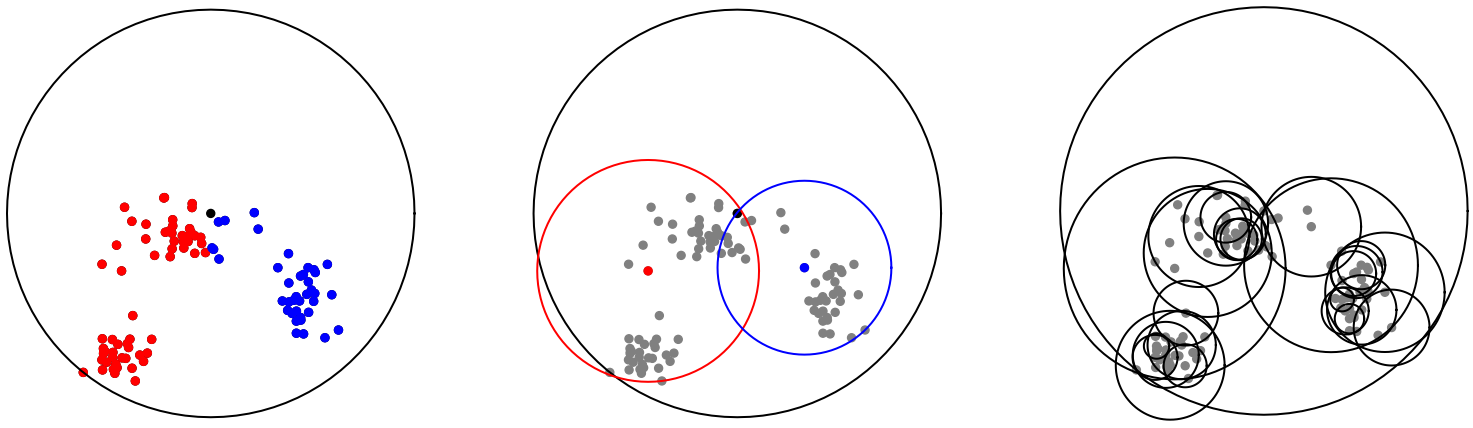


Cas défavorable
Intersection



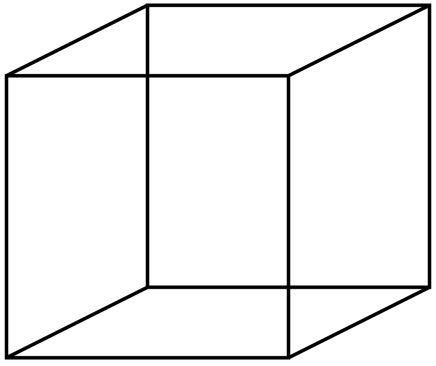
Arbre boule

Même principe, mais on garde en mémoire la **boule** dans laquelle se trouve chaque partie des données (centre et rayon).



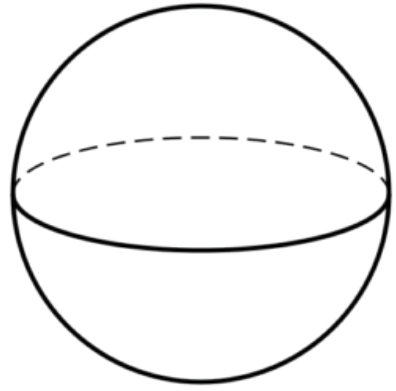
Compacité

Cube unitaire



volume $= 1$

Boule unitaire



$$\text{volume} = \begin{cases} \frac{\pi^p}{p!} & d = 2p \\ \frac{2^{p+1}\pi^p}{(2p+1)!!} & d = 2p+1 \end{cases}$$

Complexité

- Recherche en $O(\ln n)$ dans les cas favorables
- Sensible à la stratégie de coupe (choix de l'axe et du pivot)
- En pratique, peu efficace en grande dimension ($d > 20$)
  $\rightarrow$ **fléau de la dimension**