



# Chapitre 1: k plus proches voisins (KNN)

**Ibrahima Sy**

**Université Cheikh Anta Diop de Dakar(UCAD)  
Master Modélisation Statistique et Informatique(MSI)  
Faculté des Sciences et Techniques (FST)**

April 14, 2021

# Plan

Rappels Mathématiques

Fonctionnement de L'algorithme

Comment choisir la valeur K ?

Avantages & Limitations de K-NN

References

# Distance

## Distance

On appelle distance sur un ensemble  $E$  toute application  $d$  définie sur le produit  $E^2 = E \times E$  et à valeurs dans l'ensemble  $\mathbb{R}^+$  des réels positifs ou nuls, vérifiant les propriétés suivantes :

- ▶  $(P_1) : \forall (a, b) \in E^2, d(a, b) = d(b, a)$  (symétrie )
- ▶  $(P_2) : \forall (a, b) \in E^2, d(a, b) = 0 \Leftrightarrow a = b$  (séparation )
- ▶  $(P_3) : \forall (a, b, c) \in E^3, d(a, c) \geq d(a, b) + d(b, c)$  (inégalité triangulaire )

Un ensemble muni d'une distance s'appelle un **espace métrique**.

# Norme

## Notion de Norme

Une application  $N : E \mapsto \mathbb{R}$  est appelé norme, si les trois propriétés suivantes sont vérifiées

- ▶  $\|x\| = 0 \Leftrightarrow x = 0$ , pour tout  $x \in E$
- ▶ Soit  $\lambda \in \mathbb{R}$ ,  $\|\lambda x\| = |\lambda| \|x\|$
- ▶  $\forall (x, y) \in \mathbb{E}^2$ ,  $\|x + y\| \leq \|x\| + \|y\|$  (Inégalité triangulaire)

## Distance sur des espaces vectoriels

Sur un espace vectoriel normé  $(\mathbb{E}, ||.||)$ , la distance  $d$  « induite par » la norme  $||.||$  est définie par :

$$\forall (x, y) \in E^2, d(x, y) = ||x - y||$$

En particulier, dans  $\mathbb{R}^n$ , on peut définir de plusieurs manières la distance entre deux points, bien qu'elle soit généralement donnée par la distance euclidienne (ou 2-distance). Soit deux points de  $\mathbb{E}$ ,  $(x_1, x_2, \dots, x_n)$  et  $(y_1, y_2, \dots, y_n)$ , on exprime les différentes distances ainsi :

Nom	Paramètre	Fonction
distance de Manhattan	1-distance	$\sum_{i=1}^n  x_i - y_i $
distance euclidienne	2-distance	$\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
distance de Minkowski	$p$ -distance	$\sqrt[p]{\sum_{i=1}^n  x_i - y_i ^p}$
distance de Tchebychev	$\infty$ -distance	$\lim_{p \rightarrow \infty} \sqrt[p]{\sum_{i=1}^n  x_i - y_i ^p} = \sup_{1 \leq i \leq n}  x_i - y_i $

Figure 1: distance dans  $\mathbb{R}^n$

## Principe

L'algorithme **K-NN (K-nearest neighbors)** est une méthode **d'apprentissage supervisé**. Il peut être utilisé aussi bien pour la **régression** que pour la **classification**.

### Principe

*Principe de K-NN : dis moi qui sont tes voisins, je te dirais qui tu es !*

# Comment K-NN effectue une prédiction ?

Pour effectuer la prédiction d'une entrée  $x_{test}$  dans le test set ( $\mathcal{D}_{test}$ ) va suivre les étapes suivantes :

1. l'algorithme K-NN va utiliser tout le training set ( $\mathcal{D}_{train}$ )
2. l'algorithme va chercher les  $K$  entrées du  $\mathcal{D}_{train}$  les plus proches de notre entrée  $x_{test}$  qu'on veut prédire
3. Ensuite pour ces  $K$  voisins dans  $\mathcal{D}_{train}$ , l'algorithme se basera sur leurs variables de sortie (output variable)  $y$  pour calculer la valeur de la variable  $y$  de l'observation qu'on souhaite prédire.



## Comment K-NN effectue une prédiction ?

Que l'on se trouve dans le cas de la régression ou de la classification le traitement de la variable  $y$  est différente :

- ▶ **Régression**: c'est la moyenne (ou la médiane) des variables  $y$  des  $K$  plus proches observations qui servira pour la prédiction
- ▶ **Classification**: C'est le mode des variables  $y$  des  $K$  plus proches observations qui servira pour la prédiction

# Pseudo code

---

**Algorithm 2:** Algorithme des  $K$ -plus proches voisins
 

---

**Input:** Données d'apprentissage;  $\mathbf{X}^{\text{train}} = (\mathbf{x}_1^{\text{train}}, \dots, \mathbf{x}_n^{\text{train}})$ ; classes des données d'apprentissage  $\mathbf{z}^{\text{train}} = (z_1^{\text{train}}, \dots, z_n^{\text{train}})$ ;  $\mathbf{X}^{\text{test}} = (\mathbf{x}_1^{\text{test}}, \dots, \mathbf{x}_m^{\text{test}})$ ; nombre des ppv  $K$

Algorithme Knn :

```

for  $i \leftarrow 1$  to  $m$  do
    for  $j \leftarrow 1$  to  $n$  do
        Calculer la distance euclidienne  $d_{ij}$  entre  $\mathbf{x}_i^{\text{test}}$  et  $\mathbf{x}_j^{\text{train}}$  en utilisant l'équation (1)
         $d_j \leftarrow d_{ij}$ 
    end
    Calculer la classe  $z_i^{\text{test}}$  du  $i$ ème exemple qui vaut la classe de son ppv :

    /* trouver les K-ppv de  $\mathbf{x}_i^{\text{test}}$  */ :

    Trier les distances  $d_j$  selon un ordre croissant pour  $j = 1, \dots, n$ 
    Récupérer en même temps les indices IndVoisins avant le tri des  $d_j$ 
    Récupérer les classes des  $K$  premiers ppv à partir des indices IndVoisins et en
    trouver la classe majoritaire :

     $C_k \leftarrow 0$  ( $k = 1, \dots, K$ )
    for  $k \leftarrow 1$  to  $K$  do
         $ind\_voisin_k \leftarrow IndVoisins_k$ 
         $h \leftarrow z_{ind\_voisin_k}^{\text{train}}$ 
         $C_h = z_h + 1$ 
    end

    /* trouver la classe du ppv de  $\mathbf{x}_i^{\text{test}}$  :
    (la classe majoritaire de celles de ses  $K$ -ppv) */ :

     $z_i^{\text{test}} = \arg \max_{k=1}^K C_k$ 
end
    
```

**Result:** classes des données de test  $\mathbf{z}^{\text{test}} = (z_1^{\text{test}}, \dots, z_n^{\text{test}})$

---

# La distance Comme mesure de similarité

## Mesure de Similarité

Comme on vient de le voir dans notre écriture algorithme, K-NN a besoin d'une fonction de calcul de distance entre deux observations. Plus deux points sont proches l'un de l'autre, plus ils sont similaires et vice versa.

## Choisir la bonne valeur pour $k$

Pour sélectionner la valeur de  $k$  qui convient à vos données, nous exécutons plusieurs fois l'algorithme KNN avec différentes valeurs de  $k$ . Puis nous choisissons le  $k$  qui réduit le nombre d'erreurs rencontrées tout en maintenant la capacité de l'algorithme à effectuer des prédictions avec précision lorsqu'il reçoit des données nouvelles .

- Lorsque nous diminuons la valeur de  $k = 1$ , nos prédictions deviennent moins stables.

## Avantages

- ▶ Facile à comprendre
- ▶ Apprentissage rapide

## Inconvénients

- ▶ Pas efficace pour des jeux de données larges.
- ▶ L'estimation de ce modèle devient de mauvaise qualité quand le nombre de variables explicatives est grand.

# References I

- ▶ **Hugo Larochelle**, Professeur associé, Université de Montréal, Google
- ▶ **Pierre-Marc Jodoin**, Professeur titulaire Université Sherbrooke
- ▶ Bayesian Reasoning and Machine Learning de David Barber
- ▶ The Elements of Statistical Learning de Trevor Hastie,
- ▶ Robert Tibshirani et Jerome Friedman
- ▶ Information Theory, Inference, and Learning Algorithms de David J.C. MacKay
- ▶ Convex Optimization de Stephen Boyd et Lieven Vandenberghe
- ▶ Natural Image Statistics de Aapo Hyvärinen, Jarmo Hurri et Patrik O. Hoyer
- ▶ The Quest for Artificial Intelligence - A History of Ideas and Achievements de Nils J. Nilsson
- ▶ Gaussian Processes for Machine Learning de Carl Edward Rasmussen et Christopher K. I. Williams
- ▶ Introduction to Information Retrieval de Christopher D. Manning, Prabhakar Raghavan et Hinrich Schütze