Apprentissage automatique supervisé

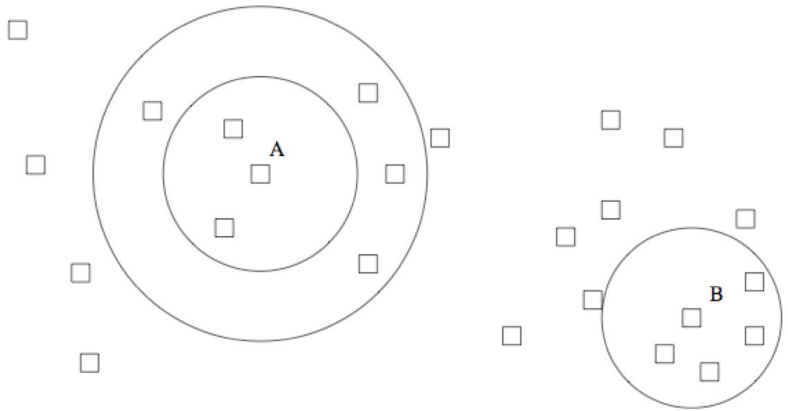
Amira Barhoumi

amira.barhoumi@univ-grenoble-alpes.fr

Année universitaire : 2025-2026

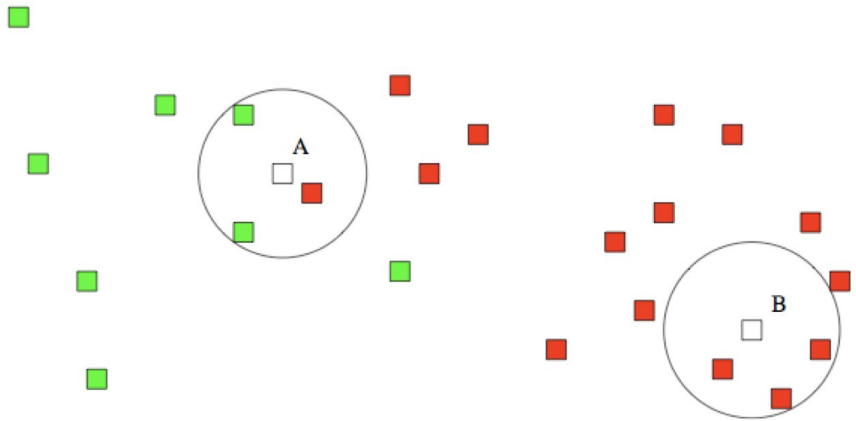
K plus proches voisins

- Méthode de K plus proches voisins (*K-nearest neighbors K-NN*)
- Méthode d'apprentissage automatique supervisé
- Objectif : prédire la classe d'un nouvel exemple
- Principe : utiliser le corpus d'apprentissage (exemples déjà connus)
- Exemple :



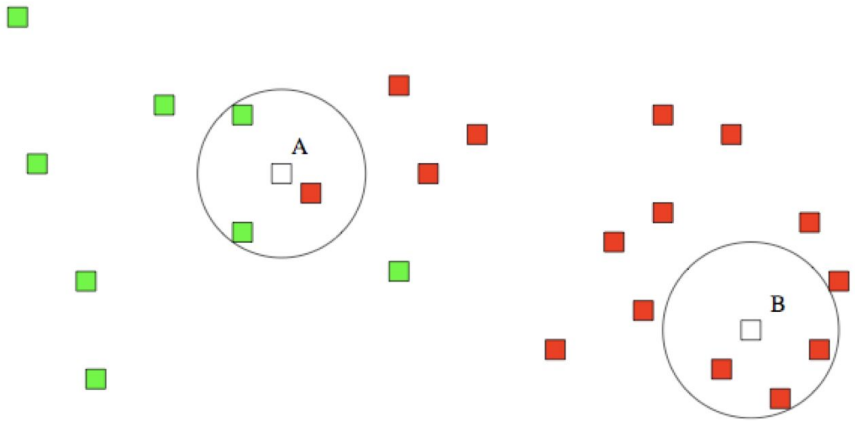
K plus proches voisins

- Méthode de K plus proches voisins (*K-nearest neighbors K-NN*)
- Méthode d'apprentissage automatique supervisé
- Objectif : prédire la classe d'un nouvel exemple
- Principe : utiliser le corpus d'apprentissage (exemples déjà connus)
- Exemple : 2 classes et k = 3



K plus proches voisins

- Principe du KNN:
  - Pour un nouvel exemple
  - Regarder la classe des K exemples les plus proches (k =1, 2, 3, …)
  - Attribuer la classe majoritaire au nouvel exemple
- Exemple : 2 classes et k = 3



K plus proches voisins

- Utilisation de cas similaires/proches pour prendre décision
- Étapes du K-NN:

    1- Corpus d'apprentissage étiqueté/annoté

    D={(x,y); x est un exemple et y sa classe correspondante}

K plus proches voisins

- Utilisation de cas similaires/proches pour prendre décision
- Étapes du K-NN:

1- Corpus d'apprentissage étiqueté/annoté

D={(x,y); x est un exemple et y sa classe correspondante}

2- Dés qu'on a une nouvelle instance x' à classer, on calcule sa distance avec tous les exemples du corpus d'apprentissage D

$$d(A,A) = 0 \; ; \; d(A,B) = d(B,A) \; ; \; d(A,C) <= d(A,B) + d(B,C)$$

K plus proches voisins

- Utilisation de cas similaires/proches pour prendre décision
- Étapes du K-NN:

1- Corpus d'apprentissage étiqueté/annoté

D={(x,y); x est un exemple et y sa classe correspondante}

2- Dés qu'on a une nouvelle instance x' à classer, on calcule sa distance avec tous les exemples du corpus d'apprentissage D

$d(A,A) = 0 ; d(A,B) = d(B,A) ; d(A,C) <= d(A,B) + d(B,C)$

3- Sélectionner ensuite les k voisins les plus proches

k = nombre d'attributs + 1 (souvent)

K plus proches voisins

- Utilisation de cas similaires/proches pour prendre décision
- Étapes du K-NN:

1- Corpus d'apprentissage étiqueté/annoté

D={(x,y); x est un exemple et y sa classe correspondante}

2- Dés qu'on a une nouvelle instance x' à classer, on calcule sa distance avec tous les exemples du corpus d'apprentissage D

$d(A,A) = 0 ; d(A,B) = d(B,A) ; d(A,C) <= d(A,B) + d(B,C)$

3- Sélectionner ensuite les k voisins les plus proches

k = nombre d'attributs + 1 (souvent)

4- Déterminer la classe de prédiction pour la nouvelle instance en fonction des classes des k plus proches voisins

classe majoritaire, avec/sans pondération

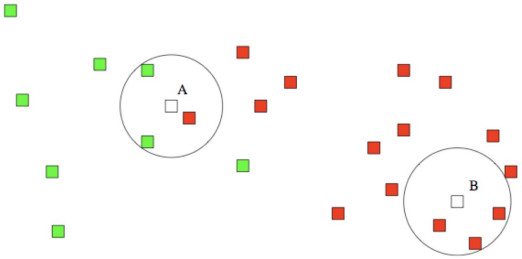
K plus proches voisins

- Pseudo-algorithme du K-NN:
  - Fixer la valeur de k
  - Définir la distance (mesure de similarité) entre 2 instances
  - Pour un nouvel exemple x'
  - Pour chaque exemple x dans le corpus d'apprentissage D

    D={(x,y); x est un exemple et y sa classe correspondante}

    - Calculer la distance entre x' et x
  - Former le groupe KNN(x')
  - Pour chaque x dans KNN(x')

    - Calculer le nombre d'occurrences de différentes classes
  - Attribuer la classe majoritaire au nouvel exemple x'



K plus proches voisins

- Fonctions de distance :
  - Distance euclidienne

$$D_e(x,y) = \sqrt{\sum_{j=1}^{n}(x_j - y_j)^2}$$

  - Distance Manhattan

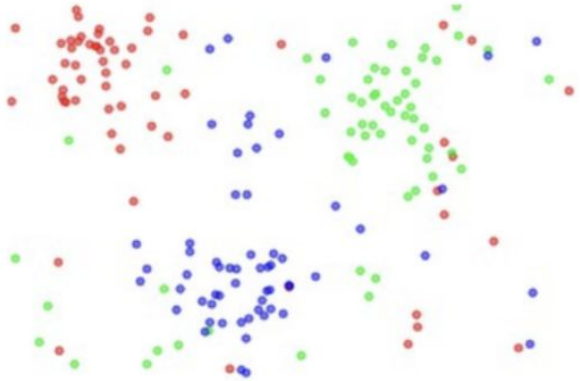
$$D_m(x,y) = \sum_{i=1}^{k} \mid x_i - y_i \mid$$

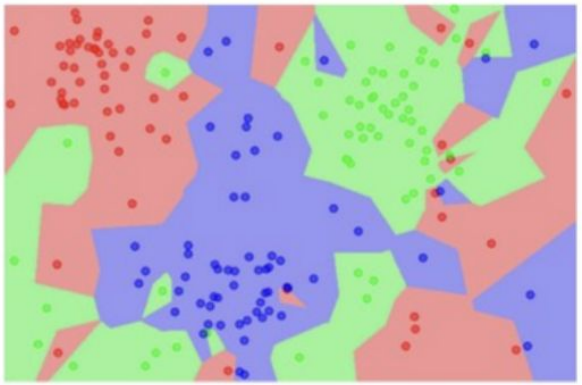
  - Distance de Hamming

$$D_h(x,y) = \#\{i \; ; \; x_i \neq y_i\}$$

K plus proches voisins

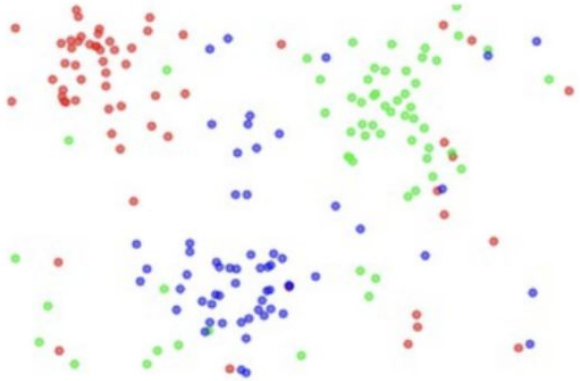
- Choix de K :
  - varie en fonction de la taille **N** du corpus d'apprentissage
  - Règle générale pour une bonne généralisation :
    - K = 1 (très petit) ⇒ risque de sous-apprentissage (*underfitting*)
    - K = **N** ⇒ risque de sur-apprentissage (*overfitting*)





K plus proches voisins

- Choix de K :
  - varie en fonction de la taille **N** du corpus d'apprentissage
  - Règle générale pour une bonne généralisation :
    - K = 1 (très petit) ⇒ risque de sous-apprentissage (*underfitting*)
    - K = **N** ⇒ risque de sur-apprentissage (*overfitting*)



sur-apprentissage

K plus proches voisins : Exemple

	TEMPS	HUMIDITE	VENT	TENNIS
Ex1	Soleil	Haute	Oui	Oui
Ex2	Soleil	Basse	Non	Non
Ex3	nuageux	Basse	Oui	Oui
Ex4	pluvieux	Haute	Oui	Non
Ex5	pluvieux	Basse	Oui	Non
Ex6	Soleil	Basse	Oui	Oui
Ex7	pluvieux	Basse	Non	Non
	Soleil	***haute***	***Non***	***?***

Va-t-on jouer s' il y a du soleil, beaucoup d' humidité et pas de vent ?

K plus proches voisins : Exemple

	TEMPS	HUMIDITE	VENT	TENNIS
Ex1	Soleil	Haute	Oui	Oui
Ex2	Soleil	Basse	Non	Non
Ex3	nuageux	Basse	Oui	Oui
Ex4	pluvieux	Haute	Oui	Non
Ex5	pluvieux	Basse	Oui	Non
Ex6	Soleil	Basse	Oui	Oui
Ex7	pluvieux	Basse	Non	Non
	Soleil	***haute***	***Non***	***?***

Va-t-on jouer s' il y a du soleil, beaucoup d' humidité et pas de vent ?

k = 4 ; distance euclidienne

K plus proches voisins : Exemple

	Temps	Humidite	Vent	Tennis
Ex1	Soleil	Haute	Oui	**Oui**
Ex2	Soleil	Basse	Non	**Non**
Ex3	nuageux	Basse	Oui	Oui
Ex4	Pluvieux	Haute	Oui	**Non**
Ex5	Pluvieux	Basse	Oui	Non
EX6	Soleil	Basse	Oui	**Oui**
EX7	Pluvieux	Basse	Non	Non

K plus proches voisins : Exemple

	Temps	Humidite	Vent	Tennis
Ex1	Soleil	Haute	Oui	**Oui**
Ex2	Soleil	Basse	Non	**Non**
Ex3	nuageux	Basse	Oui	Oui
Ex4	Pluvieux	Haute	Oui	**Non**
Ex5	Pluvieux	Basse	Oui	Non
EX6	Soleil	Basse	Oui	**Oui**
EX7	Pluvieux	Basse	Non	Non

Temps = 0 si “Soleil”
1 sinon

Humidité = 0 si “Haute”
1 sinon

Vent = 0 si “non”
1 sinon

K plus proches voisins : Exemple

	Temps	Humidite	Vent	Tennis
Ex1	Soleil	Haute	Oui	**Oui**
Ex2	Soleil	Basse	Non	**Non**
Ex3	nuageux	Basse	Oui	Oui
Ex4	Pluvieux	Haute	Oui	**Non**
Ex5	Pluvieux	Basse	Oui	Non
EX6	Soleil	Basse	Oui	**Oui**
EX7	Pluvieux	Basse	Non	Non

Temps	Humidite	Vent
0	0	1
0	1	0
1	1	1
1	0	1
1	1	1
0	1	1
1	1	0

Temps = 0 si “Soleil”
1 sinon

Humidité = 0 si “Haute”
1 sinon

Vent = 0 si “non”
1 sinon

K plus proches voisins : Exemple

	Temps	Humidite	Vent	Tennis
Ex1	Soleil	Haute	Oui	**Oui**
Ex2	Soleil	Basse	Non	**Non**
Ex3	nuageux	Basse	Oui	Oui
Ex4	Pluvieux	Haute	Oui	**Non**
Ex5	Pluvieux	Basse	Oui	Non
EX6	Soleil	Basse	Oui	**Oui**
EX7	Pluvieux	Basse	Non	Non

Temps	Humidite	Vent
0	0	1
0	1	0
1	1	1
1	0	1
1	1	1
0	1	1
1	1	0

Distances
Résultat

Distance euclidienne

K plus proches voisins : Exemple

	Temps	Humidite	Vent	Tennis
Ex1	Soleil	Haute	Oui	**Oui**
Ex2	Soleil	Basse	Non	**Non**
Ex3	nuageux	Basse	Oui	Oui
Ex4	Pluvieux	Haute	Oui	**Non**
Ex5	Pluvieux	Basse	Oui	Non
EX6	Soleil	Basse	Oui	**Oui**
EX7	Pluvieux	Basse	Non	Non

Temps	Humidite	Vent	Distances Résultat
0	0	1	**1**
0	1	0	**1**
1	1	1	1,73
1	0	1	**1,41**
1	1	1	1,73
0	1	1	**1,41**
1	1	0	1,41

Distance euclidienne

K plus proches voisins : Exemple

	Temps	Humidite	Vent	Tennis
Ex1	Soleil	Haute	Oui	**Oui**
Ex2	Soleil	Basse	Non	**Non**
Ex3	nuageux	Basse	Oui	Oui
Ex4	Pluvieux	Haute	Oui	**Non**
Ex5	Pluvieux	Basse	Oui	Non
EX6	Soleil	Basse	Oui	**Oui**
EX7	Pluvieux	Basse	Non	Non

Temps	Humidite	Vent	Distances Résultat
0	0	1	**1**
0	1	0	**1**
1	1	1	1,73
1	0	1	**1,41**
1	1	1	1,73
0	1	1	**1,41**
1	1	0	1,41

Distance euclidienne

k = 4 ; distance euclidienne
Classes du KNN(nouvel exemple) ={oui, non, oui, non}

K plus proches voisins : Exemple

	Temps	Humidite	Vent	Tennis
Ex1	Soleil	Haute	Oui	**Oui**
Ex2	Soleil	Basse	Non	**Non**
Ex3	nuageux	Basse	Oui	Oui
Ex4	Pluvieux	Haute	Oui	**Non**
Ex5	Pluvieux	Basse	Oui	Non
EX6	Soleil	Basse	Oui	**Oui**
EX7	Pluvieux	Basse	Non	Non

Temps	Humidite	Vent	Distances Résultat
0	0	1	**1**
0	1	0	**1**
1	1	1	1,73
1	0	1	**1,41**
1	1	1	1,73
0	1	1	**1,41**
1	1	0	1,41

Distance euclidienne

k = 4 ; distance euclidienne
Classes du KNN(nouvel exemple) ={oui, non, oui, non}
Solutions possibles :

- choisir une autre valeur de k (impaire!)
- Choisir aléatoirement une classe parmi celles ambiguës
- Pondération des exemples de KNN(nouvel exemple) par leurs distances au nouvel exemple

K plus proches voisins : Exemple

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	sunny	hot	high	weak	no
D2	sunny	hot	high	strong	no
D3	overcast	hot	high	weak	yes
D4	rain	mild	high	weak	yes
D5	rain	cool	normal	weak	yes
D6	rain	cool	normal	strong	no
D7	overcast	cool	normal	strong	yes
D8	sunny	mild	high	weak	no
D9	sunny	cool	normal	weak	yes
D10	rain	mild	normal	weak	yes
D11	sunny	mild	normal	strong	yes
D12	overcast	mild	high	strong	yes
D13	overcast	hot	normal	weak	yes
D14	rain	mild	high	strong	no

- Classer la nouvelle instance (sunny, cool, high, strong)

K plus proches voisins

- Avantages :
  - Simple
  - Facile à mettre en place

- Inconvénients :
  - Gourmand en calcul/mémoire
    - Mémoriser les différentes observations du corpus d'apprentissage
    - Refaire le calcul de distance pour chaque nouvelle instance à classer
  - Choix de la distance
  - Choix de K (souvent impair pour simplifier le vote)