K-plus proche Voisin (KNN/KPP)

L'apprentissage automatique (2016-2017)

UFR MIME

Université Lille 3

7 décembre 2016

Sommaire

Pourquoi Faire l'Apprentissage

Apprentissage

Au lieu de programmer les regles manuellement dans un programme, donner une ordinateur une moyenne de extraires les regles automatiquement.

Pourquoi

- Problemes trop complexe
- Travail manuelle trop couteux
- Tres grandes quantite des donnees
- Pour devenir plus feignant/efficace avec ces taches
- C'est rigolo :)

Classification

Apprentissage

Au lieu de programmer les règles manuellement dans un programme, donner une ordinateur une moyenne de extraire les règles automatiquement.

Apprentissage Supervisé : Classification

- La classification consiste a prédire **une catégorie**
- On va se limiter a un sous-classe de problème : **classification binaire**
- Fournir au système en entrée un ensemble de d'exemples étiquetés $\mathbf{x}_i, y_i$ d'apprentissage.
- $\mathbf{x}_i \in \mathbb{R}^d$ et $y_i \in \{C_1, C_2, ..., C_q\}$ avec q classes
- Pour classification binaire y_i est un catégorie soit positif / négatif dans $\{+1, -1\}$

Classification

Classification Binaire

- Tache pour nous : Retrouver une mapping (correspondance) $f_\theta : \mathbb{R}^d \rightarrow \{-1, +1\}$
- Algo permet ordinateur de se programmer **lui-même**.
- Ici L'algorithme du coup consiste a retrouver les paramétrés optimaux θ
- Le classifier finale vas nous aider a prédire les taches comme si :
  - **Oui(+1)**/**Non(-1)** Une visage apparaît dans une image ?
    Entree : ensemble des pixels d'image
  - **Oui(+1)**/**Non(-1)** Un document parle de sport ?
    Entree : Les suites des characters de texte
  - **Oui(+1)**/**Non(-1)** Un client risque de quitter ma banque ?
    Entree : Attributs de client (age, salaire, épargne, type maison, type contrat etc)

K-plus proche Voisin (KPPV) K-Nearest neighbours (KNN)

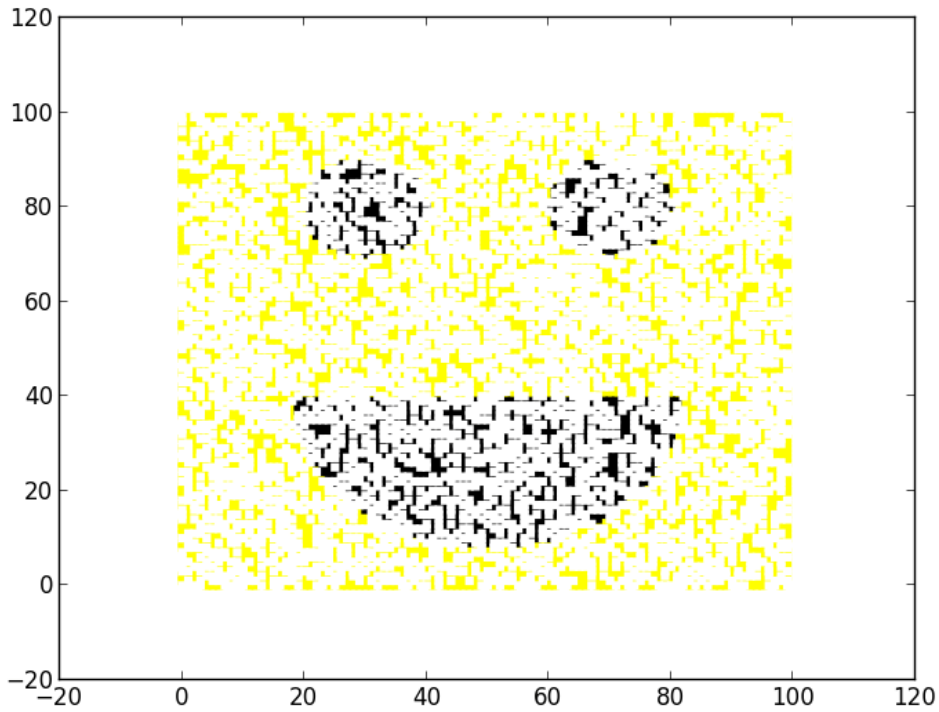


FIGURE – Comment prédire les pixels manquant ?

K-plus proche Voisin K-Nearest neighbours (KNN)

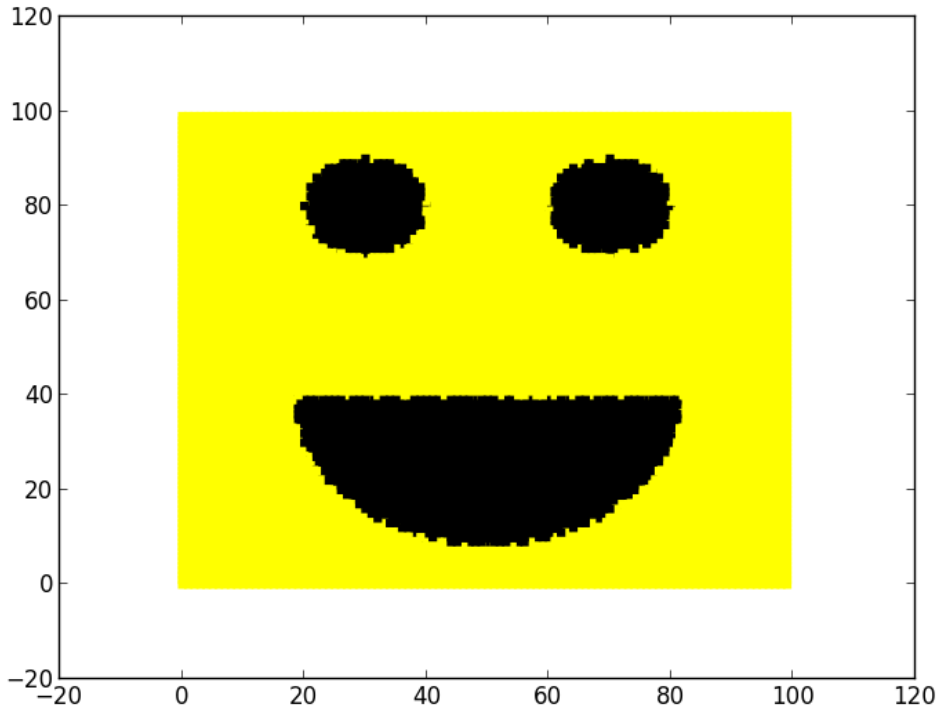
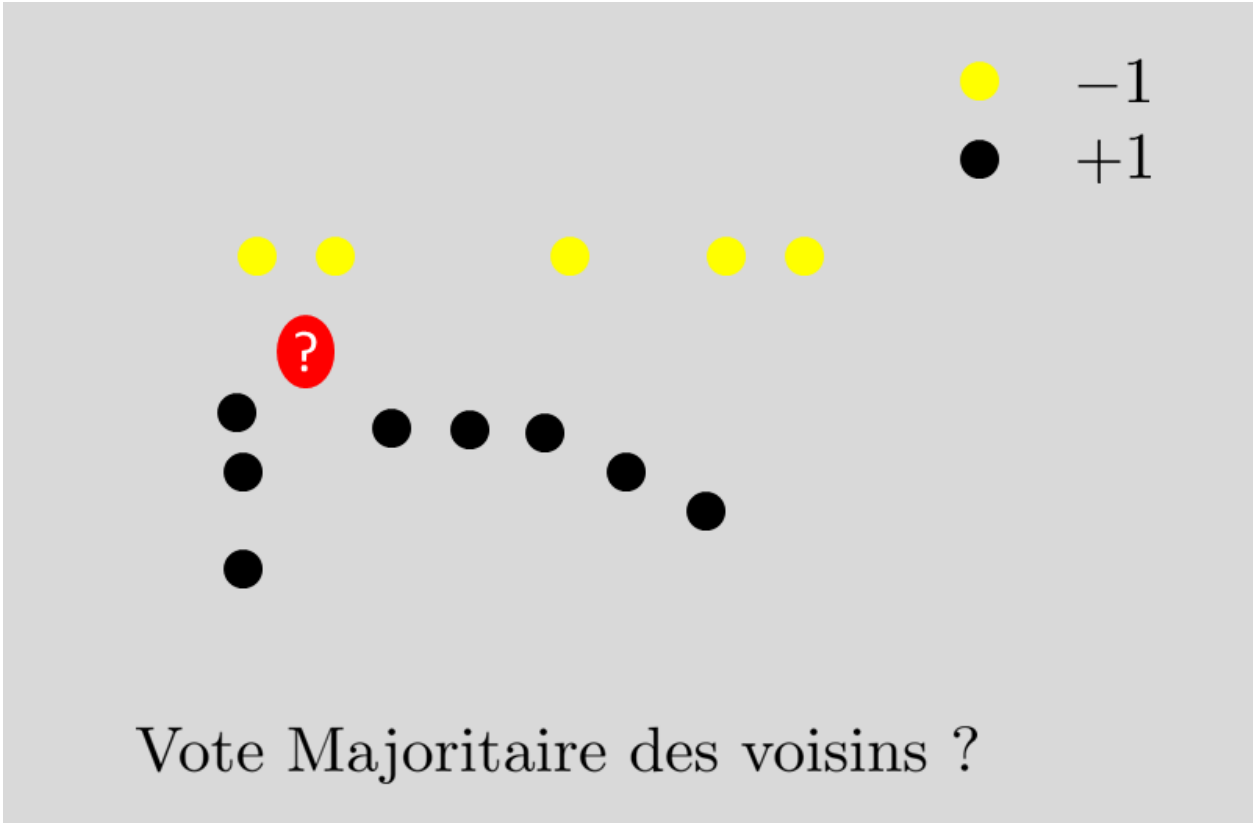
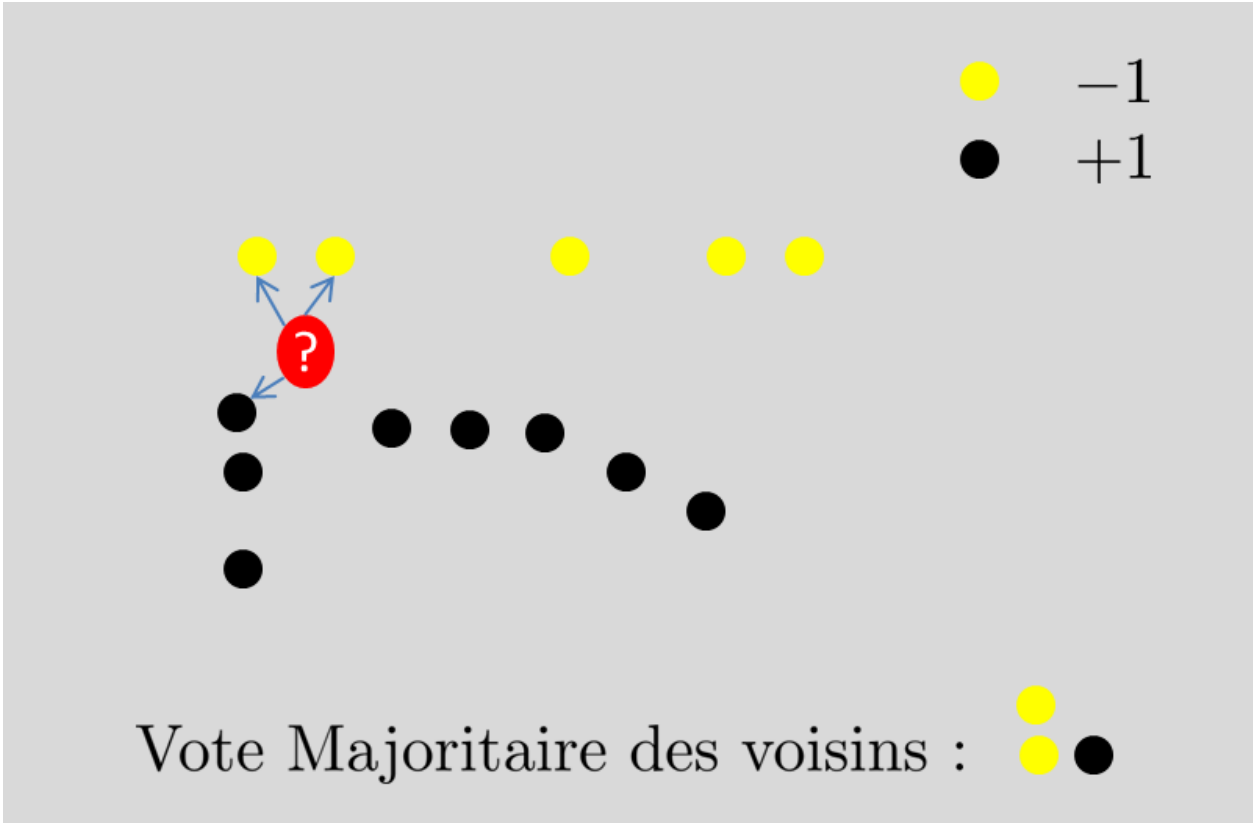


FIGURE – Utiliser les voisins de pixel a remplier pour décider la libelle (noir(+1)/jaune(-1)).

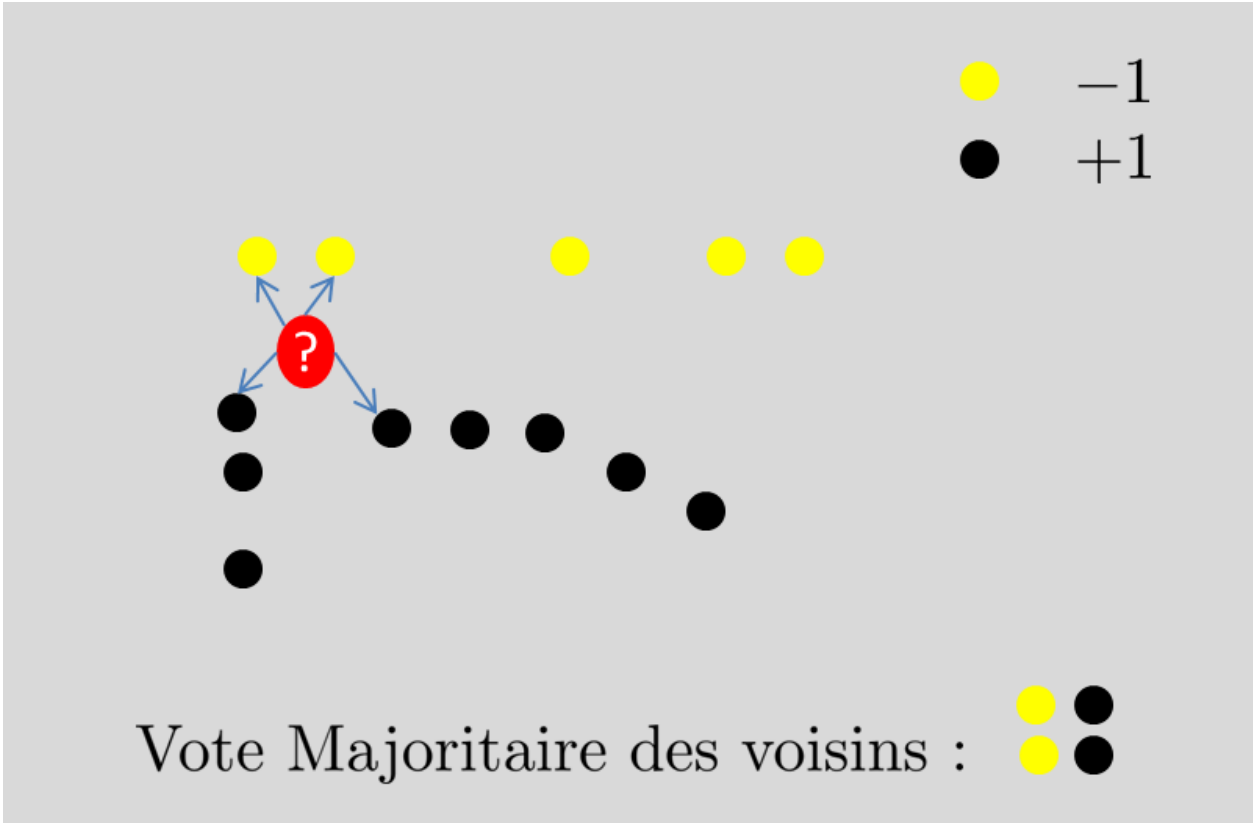
Exemple : dimension d = 2



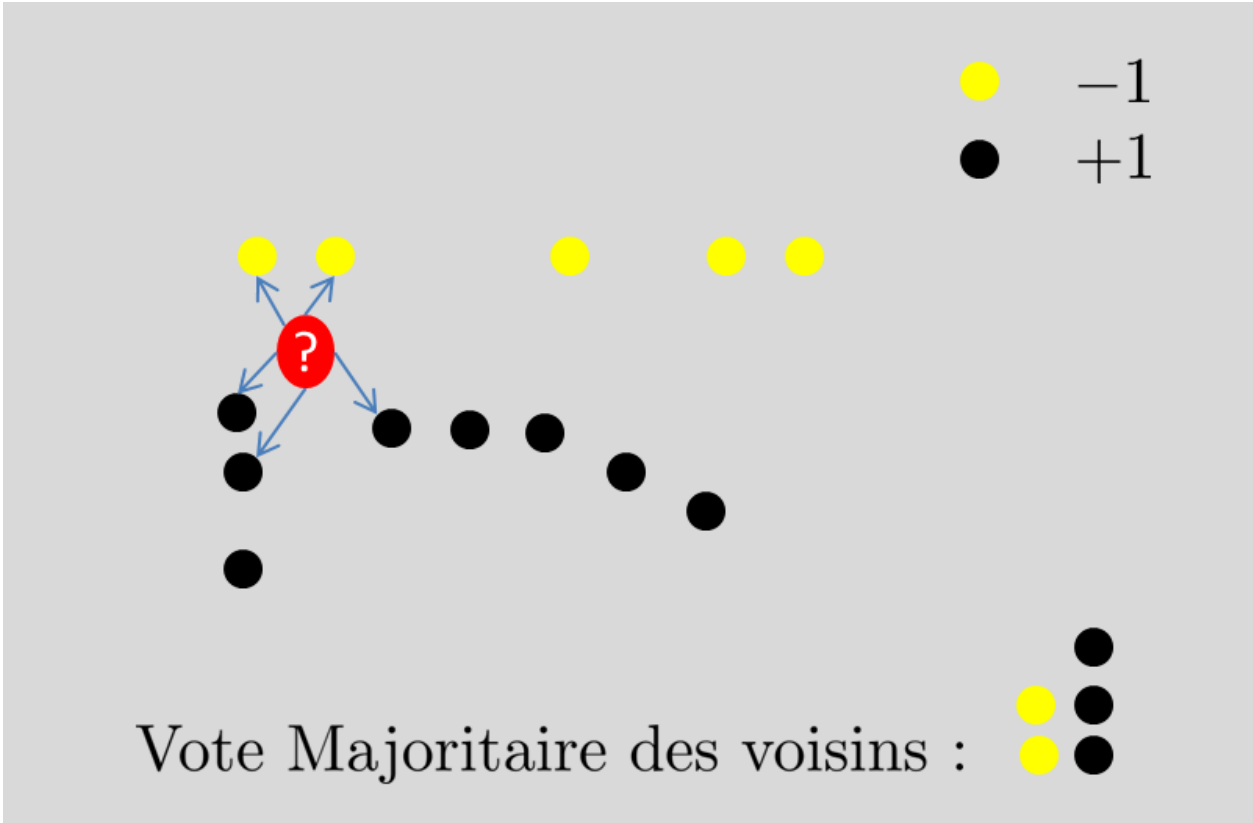
Exemple : dimension d = 2



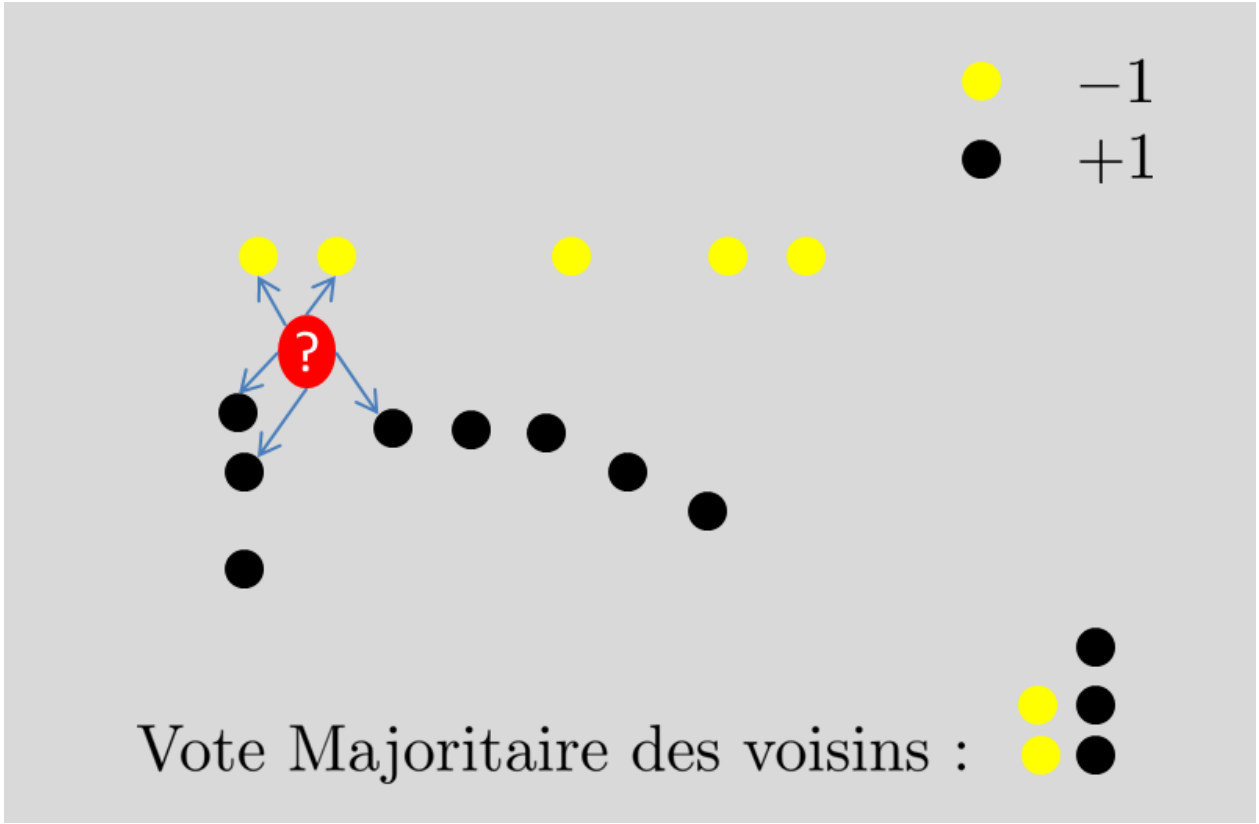
Exemple : dimension d = 2



Exemple : dimension d = 2



Exemple : dimension = 2



Voisinage

En réglant la **distance** → *nombredevoisin* on arrive a recouvrir plus des voisins qui change le résultat de **classification**.

KNN

Algorithme

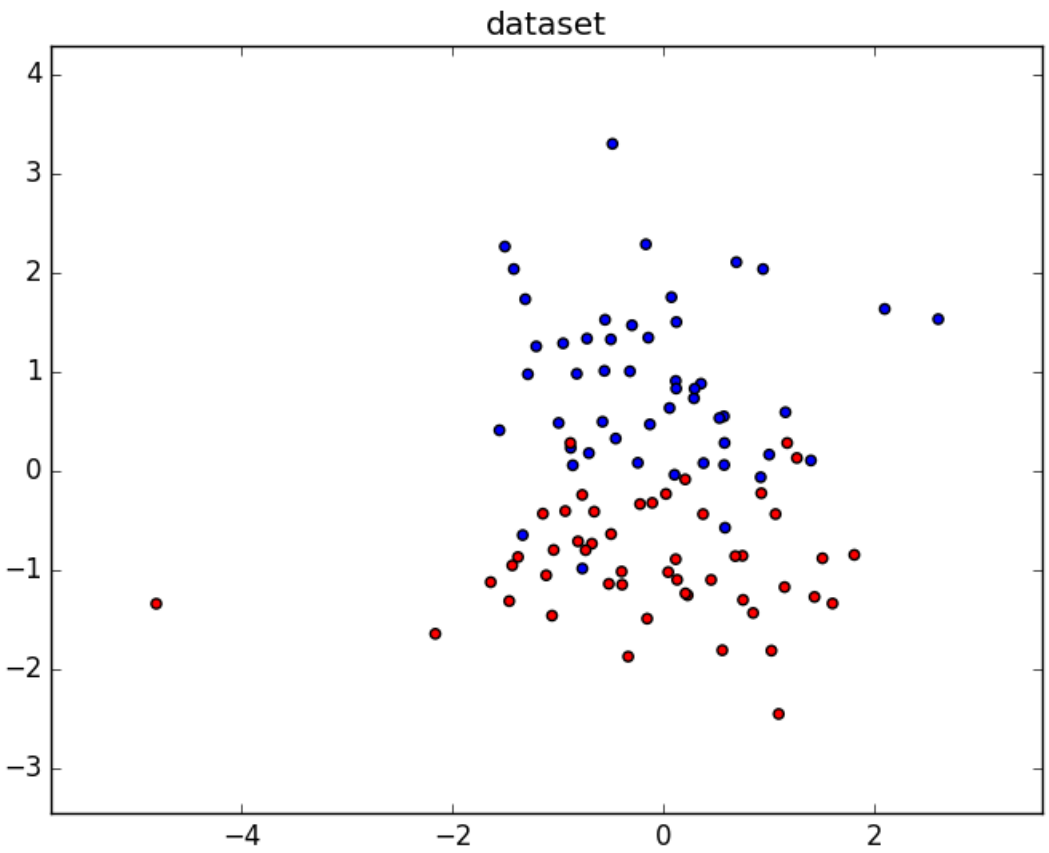
- Ensemble d'apprentissage (ou Training set) : $X_{\text{Tr}} := \{(\mathbf{x}_1, y_1), \mathbf{x}_2, y_2), \ldots, \mathbf{x}_N, y_N)\}$
- $\mathbf{x}_i \in \mathbb{R}^d$ et $y_i \in \{C_q\}_{q=1}^{Q}$
- Pour la classification binaire $y_i \in \{-1, +1\}$
- Pour une nouvelle entrée $\mathbf{z}$ (dans le ensemble test)
- $f_\theta(\mathbf{z}) = \text{VoteMajoritaire}\{y_i | i \in \text{k-plus proche voisin}(z)\}$ dans X_{Tr}
- Pour calculer la k-plus proche voisin de $\mathbf{x}$:
  - Pour calculer distance euclidien entres deux point :
  - $d(\mathbf{x}, \mathbf{x}') = \sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2 + \cdots + (x_d - x_d')^2}$
- $P(y = +1) =$ Proportion des pts de classe $+1$ dans le voisinage $N_k(\mathbf{z})$

Matrice de distance

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	$\mathbf{x}_{15}$	$\mathbf{x}_{16}$	$\mathbf{x}_{17}$	$\mathbf{x}_{18}$
$\mathbf{x}_2$	1.5																	
$\mathbf{x}_3$	1.4	1.6																
$\mathbf{x}_4$	1.6	1.4	1.3															
$\mathbf{x}_5$	1.7	1.4	1.5	1.5														
$\mathbf{x}_6$	1.3	1.4	1.4	1.5	1.4													
$\mathbf{x}_7$	1.6	1.3	1.4	1.4	1.5	1.8												
$\mathbf{x}_8$	1.5	1.4	1.6	1.3	1.7	1.6	1.4											
$\mathbf{x}_9$	1.4	1.3	1.4	1.5	1.2	1.4	1.3	1.5										
$\mathbf{x}_{10}$	2.3	2.4	2.5	2.3	2.6	2.7	2.8	2.7	3.1									
$\mathbf{x}_{11}$	2.9	2.8	2.9	3.0	2.9	3.1	2.9	3.1	3.0	1.5								
$\mathbf{x}_{12}$	3.2	3.3	3.2	3.1	3.3	3.4	3.3	3.4	3.5	3.3	1.6							
$\mathbf{x}_{13}$	3.3	3.4	3.2	3.2	3.3	3.4	3.2	3.3	3.5	3.6	1.4	1.7						
$\mathbf{x}_{14}$	3.4	3.2	3.5	3.4	3.7	3.5	3.6	3.3	3.5	3.6	1.5	1.8	0.5					
$\mathbf{x}_{15}$	4.2	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	1.7	1.6	0.3	0.5				
$\mathbf{x}_{16}$	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	1.6	1.5	0.4	0.5	0.4			
$\mathbf{x}_{17}$	5.9	6.2	6.2	5.8	6.1	6.0	6.1	5.9	5.8	6.0	2.3	2.3	2.5	2.3	2.4	2.5		
$\mathbf{x}_{18}$	6.1	6.3	6.2	5.8	6.1	6.0	6.1	5.9	5.8	6.0	3.1	2.7	2.6	2.3	2.5	2.6	3.0	
$\mathbf{x}_{19}$	6.0	6.1	6.2	5.8	6.1	6.0	6.1	5.9	5.8	6.0	3.0	2.9	2.7	2.4	2.5	2.8	3.1	0.4

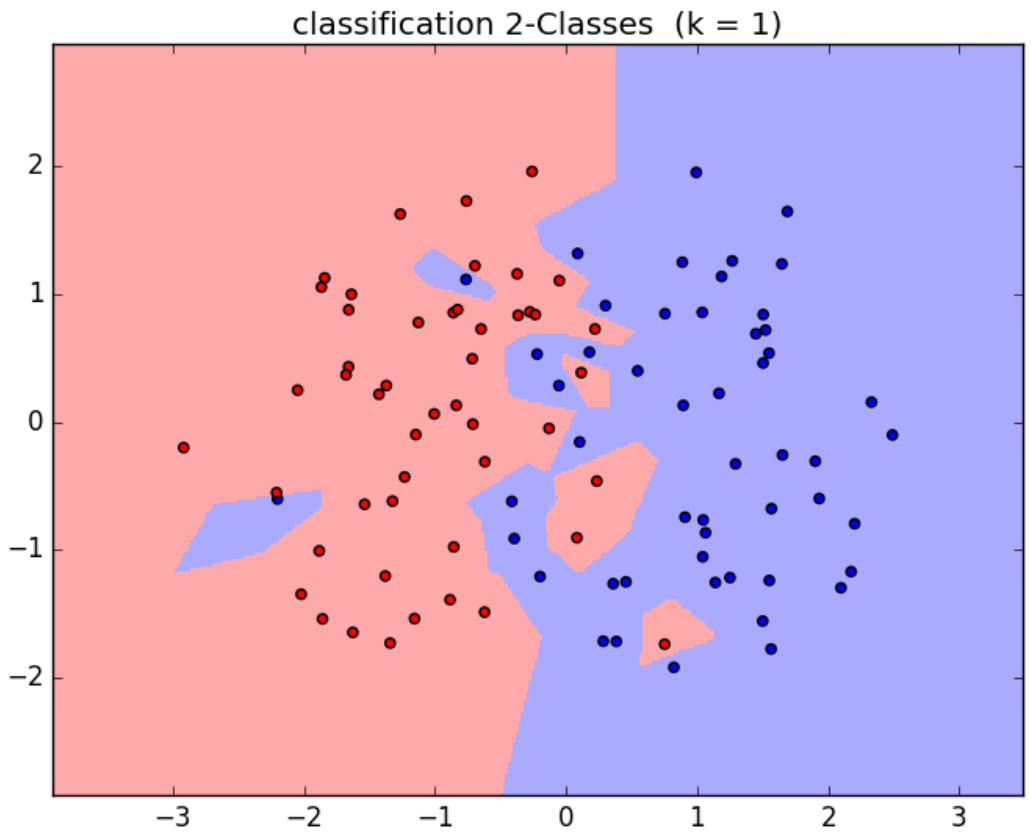
FIGURE – Pour $k = 4$ on cherche les plus proche voisin pour le vecteur $\mathbf{x}_{11}$. On retrouve $\mathbf{x}_{10}$ (classe bleu), $\mathbf{x}_{12}, \mathbf{x}_{13}, \mathbf{x}_{14}$ (classe rouge)

Jeu de données synthétique avec deux classes



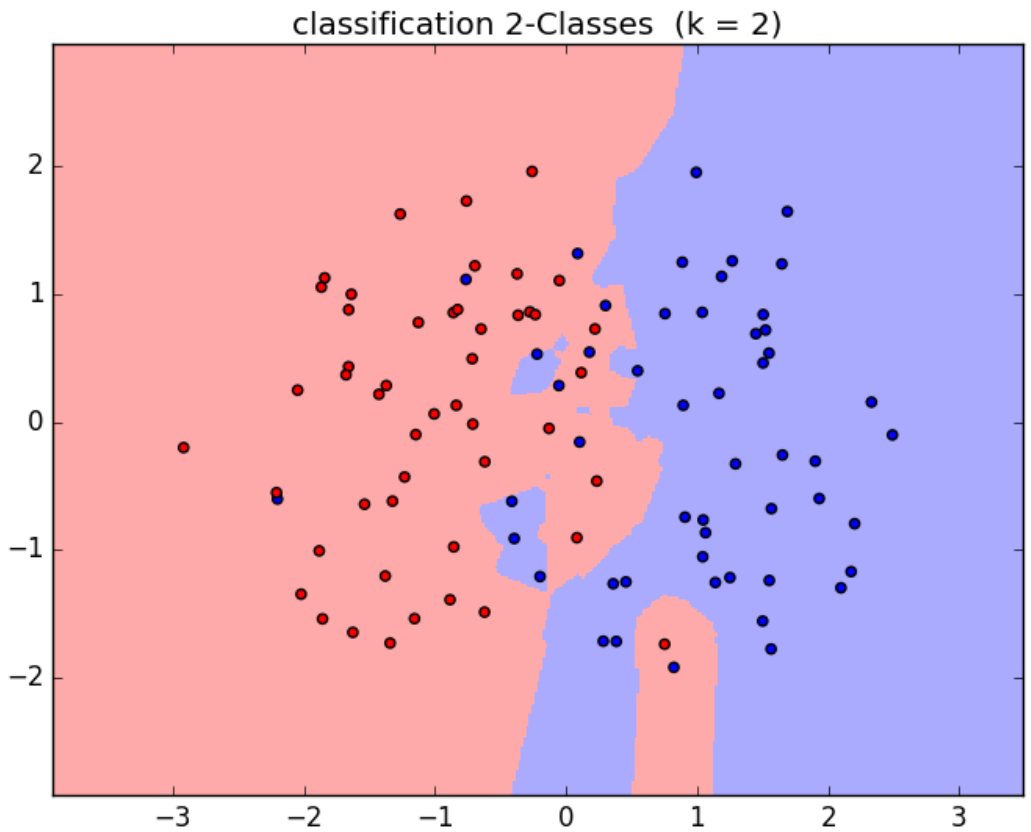
Exemple avec frontières de décision

k=1, trop compliqué



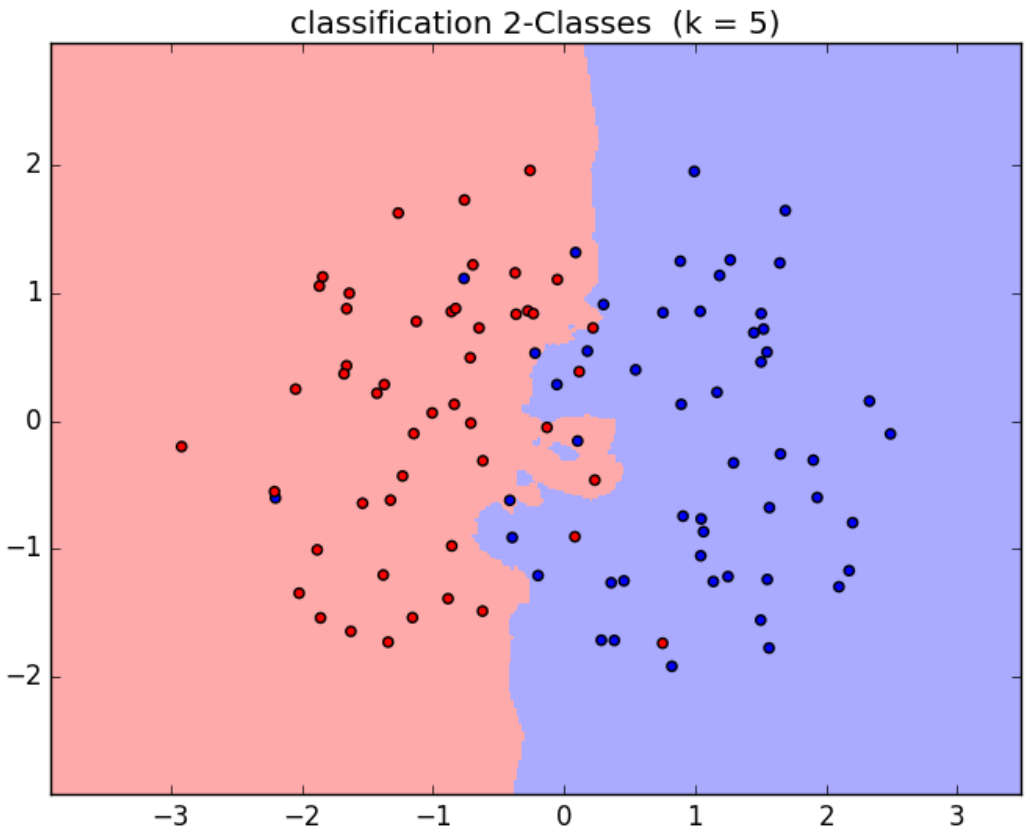
Exemple avec frontières de décision

k=2, compliqué



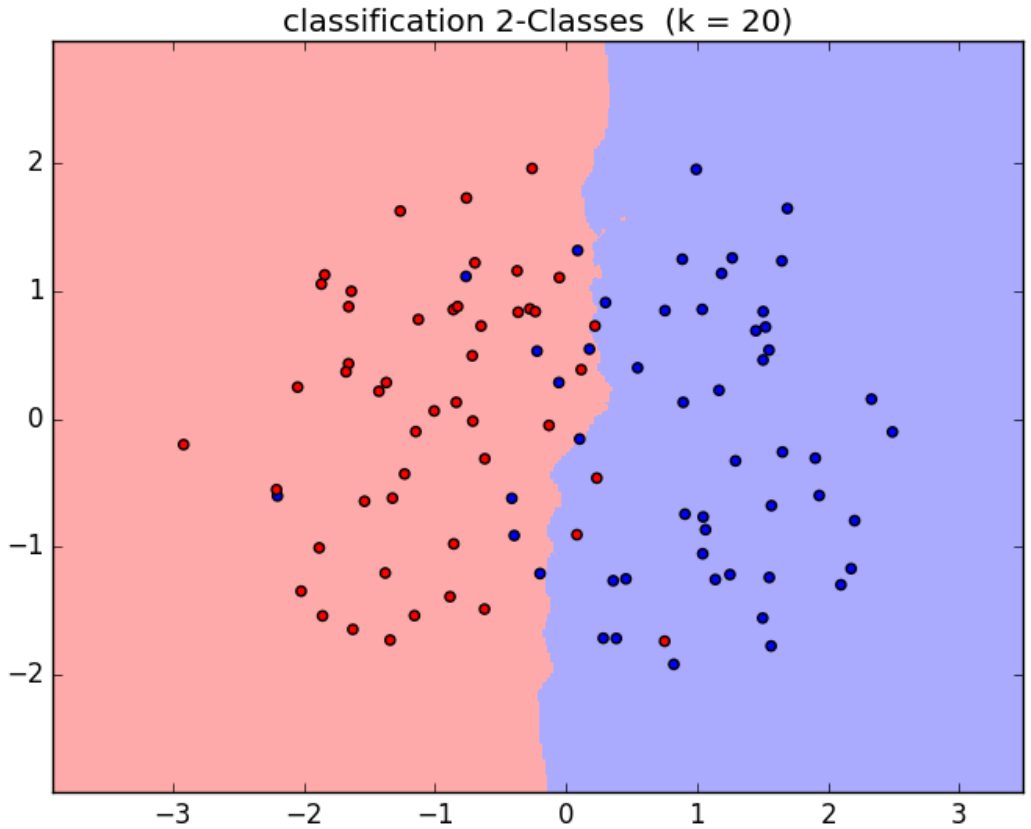
Exemple avec frontières de décision

k=5, OK !



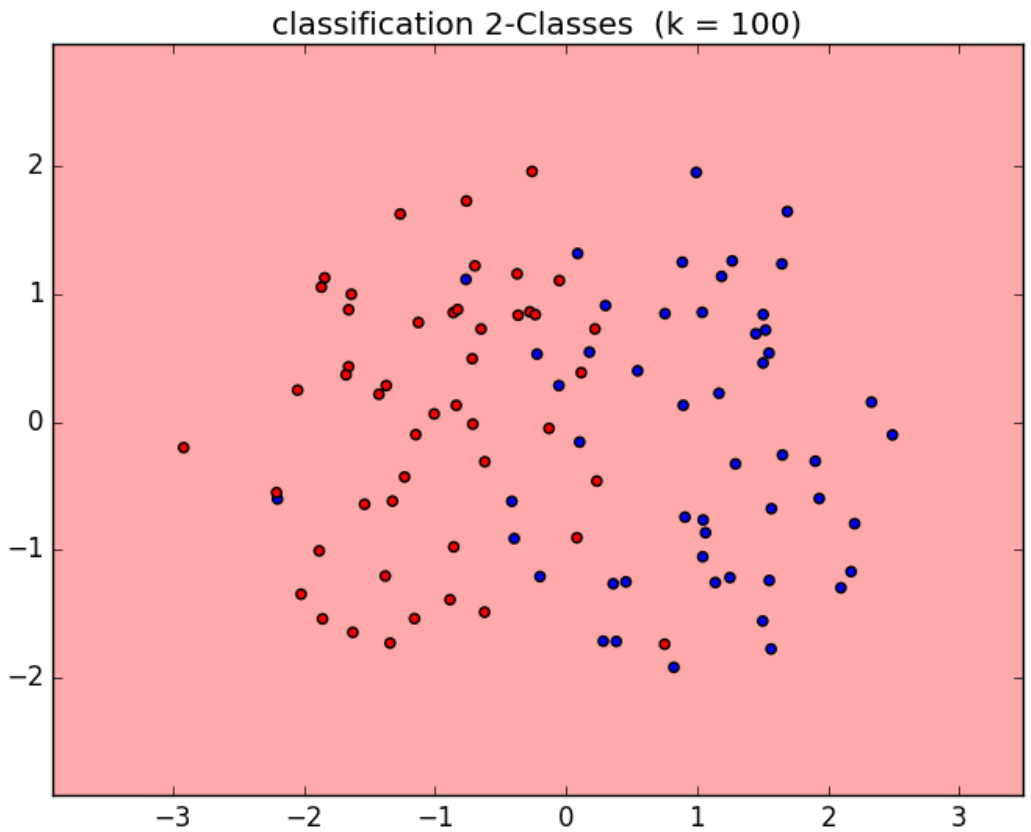
Exemple avec frontières de décision

k=20, pas mal



Exemple avec frontières de décision

k=100, trop simplifié



Généralisation

Sur-Sous Apprentissage

- f_θ est trop compliqué comme frontière des décision : sur-apprentissage
  - Il prédit très bien sur donnée d'apprentissage mais pas très bien sur les nouvelles données
- f_θ est trop simple comme frontière des décision : sous-apprentissage
  - Il prédit ni bien sur donnée d'apprentissage ni sur les nouvelles données
  - Ceci est possible si k est très grandes.
  - Il prédit le vote majoritaire "globale"
- Compromis Simple-Compliqué (Biais-Variance)
  - Choix de paramétré optimale θ : ici nombre de voisins K

Évaluation

Performance

Comment savoir si notre classifier va bien généraliser ? → vas bien prédire sur les nouveau données ?

Erreur de classification

- Étant donnée un scénario d'apprentissage :
  - On peut utiliser l'ensemble d'apprentissage pour évaluation d'erreur ?
  - Il nous faut des nouveau exemples ? Pas déjà prise en compte par la classifier ? Pour quoi ?
  - Quelle est l'erreur des prédictions pour k=1 pour KPPV sur l'ensemble d'apprentissage ?

Évaluation

Split/Decoupage

- D = **Donnée apprentissage** $\cup$ **donnée test**
- **Donnée apprentissage** $\cap$ **donnée test** $= \emptyset$
- Erreur est évaluer sur l'ensemble test en comparant
  - Catégorie prédit par le classifier (classification)
  - Le vrai valeur de catégorie (vérité terrain)

TP : Codez les différentes fonctions pour KPPV

- qui calcul de distance a pair pour l'ensemble d'apprentissage
- qui calcul le vote majoritaire étant données k (classification)
- fonction qui va évaluer les erreur de classification sur ensemble d'évaluation (test)