Algorithmique des données
Classification

Charlotte Pelletier
MCF Univ. Bretagne Sud – IRISA Vannes

19 mars 2020

Plan du cours

- Partie I. Introduction
  - CM0. Introduction
  - CM1. Rappels en algèbre linéaire et probabilités
- Partie II. Apprentissage non-supervisé
  - CM2. Analyse par Composantes Principales
  - CM3. k-Means
- Partie III. Apprentissage supervisé : régression
  - CM4. Régression linéaire
  - CM5. Régression logistique
  - CM6. Compromis biais-variance et techniques de régularisation
- Partie IV. Apprentissage supervisé : classification
  - CM7. Algorithmes de classification
  - CM8. Sélection de modèles

Plan du cours

- Partie I. Introduction
  - CM0. Introduction
  - CM1. Rappels en algèbre linéaire et probabilités
- Partie II. Apprentissage non-supervisé
  - CM2. Analyse par Composantes Principales
  - CM3. k-Means
- Partie III. Apprentissage supervisé : régression
  - CM4. Régression linéaire
  - CM5. Régression logistique
  - CM6. Compromis biais-variance et techniques de régularisation
- **Partie IV. Apprentissage supervisé : classification**
  - **CM7. Algorithmes de classification**
  - CM8. Sélection de modèles

Rappel

Rappel

- **Apprentissage supervisé** : dans les données observées, on connaît la "vraie" valeur de la variable de sortie et on cherche à comprendre / prédire le lien supposé entre les variables d'entrée et de sortie
- **Variable à expliquer/prédire**, notée Y
  - quantitative : régression
  - **qualitative** : **classification binaire / multiclasses** $\mathcal{C}$
- **Variables explicatives**, notées $X^1, X^2, \cdots, X^d$?
  - **qualitatives** et/ou **quantitatives**
  - plusieurs = de quelques dizaines à plusieurs (dizaines de) milliers $\Rightarrow$ sélection de variables

Données d'apprentissage

Échantillons

- $\mathbf{x} \in \mathbb{R}^d$ est une observation de d caractéristiques réelles (d variables)
- l'ensemble d'apprentissage est défini par les observations $\{\mathbf{x}_i\}_{i=1}^m$ où m est le nombre de données d'apprentissages (observations)
- d et m définissent la dimensionnalité du problème d'apprentissage
- les données sont mises sous la forme d'une matrice $\mathbf{X} \in \mathbb{R}^{m \times d}$ définie par $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_m]^\top = [X^1, X^2, \ldots, X^d]$ contenant les exemples d'apprentissage en lignes et les variables en colonnes

Données d'apprentissage

Échantillons

- $\mathbf{x} \in \mathbb{R}^d$ est une observation de d caractéristiques réelles (d variables)
- l'ensemble d'apprentissage est défini par les observations $\{\mathbf{x}_i\}_{i=1}^m$ où m est le nombre de données d'apprentissages (observations)
- d et m définissent la dimensionnalité du problème d'apprentissage
- les données sont mises sous la forme d'une matrice $\mathbf{X} \in \mathbb{R}^{m \times d}$ définie par $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_m]^\top = [X^1, X^2, \ldots, X^d]$ contenant les exemples d'apprentissage en lignes et les variables en colonnes

Étiquettes

- à chaque observation $\mathbf{x}_i$ une valeur à prédire $y_i \in \mathcal{Y}$ est associée (étiquette)
- ses valeurs à prédire peuvent être concaténées en un vecteur $\mathbf{y} \in \mathcal{Y}^m$
- L'espace des valeurs à prédire $\mathcal{Y}$ sera :
  - $\mathcal{Y} = \{-1, 1\}$ ou $\mathcal{Y} = \{0, 1\}$ pour la classification binaire
  - $\mathcal{Y} = \{1, \ldots, \mathcal{C}\}$ pour la classification multiclasses ($\mathcal{C}$ classes)

Apprentissage supervisé

Système d'apprentissage

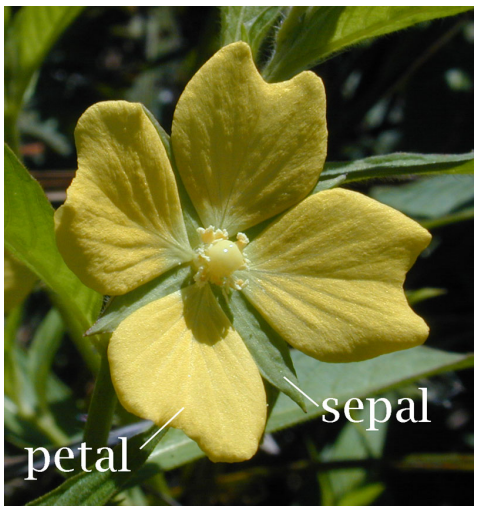
1. Phase d'apprentissage : apprendre un modèle (règle de décision)
2. Phase de prédiction : prédire la classe de nouvelles observations

Exemple du cours

Exemple

Nous cherchons à discriminer trois types d'iris (iris `virginica`, iris `versicolore` et iris `setosa` aussi appelé iris de l'Alaska) en fonction de la largeur et de la longueur des pétales et des sépales (en cm).

Extrait des données



Source : Wikipedia

Iris	Long. pétale	Larg. pétale	Long. sépale	
`setosa`	1.4	0.2	5.1	...
`setosa`	1.5	0.1	4.9	...
`setosa`	1.3	0.4	5.4	...
`versicolore`	4.7	1.4	7.0	...
`versicolore`	3.3	1.0	4.9	...
`virginica`	6.0	1.8	7.2	...
`virginica`	4.8	1.8	6.0	...
⋮	⋮	⋮	⋮	

Le jeu de données Iris est très utilisé dans le domaine de l'apprentissage automatique. Il a été pour pour la première fois utilisé par Fisher, R.A. "The use of multiple measurements in taxonomic problems" in Contributions to Mathematical Statistics (John Wiley, NY, 1950).

Visualisation des classes

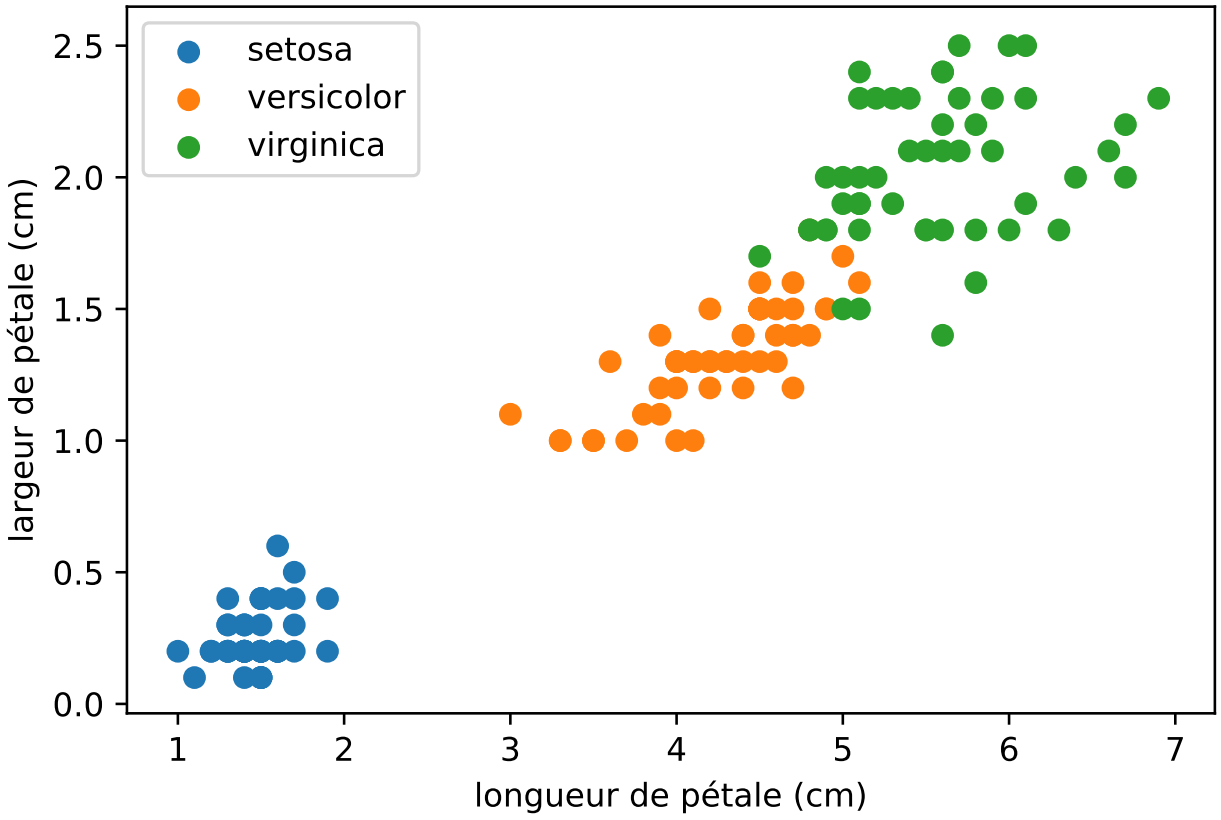
Iris setosa

Iris versicolore

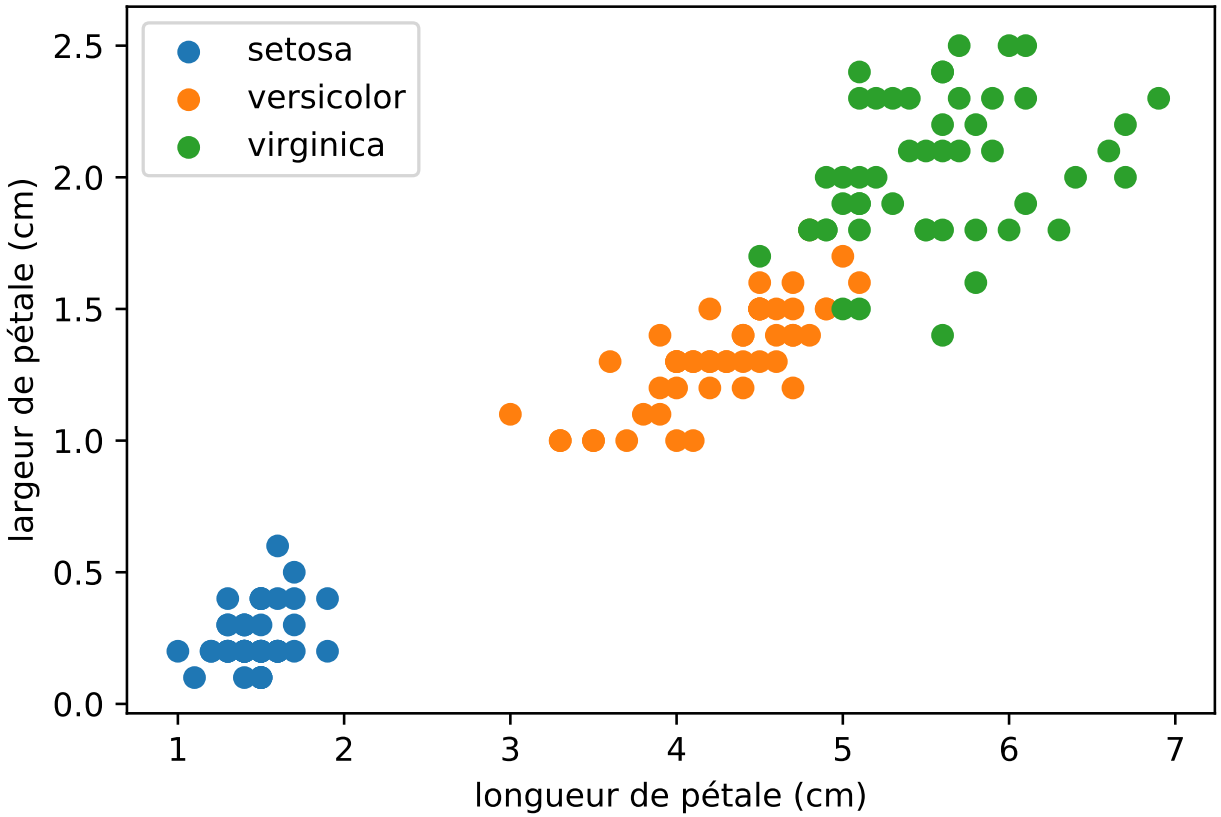
Iris virginica

Sources : Wikipedia

Visualisation des données



Visualisation des données



Analyse :

- la classe `setosa` est linéairement séparable des deux autres
- les classes `versicolor` et `virginica` ne sont pas linéairement séparables

Algorithmes génératifs

Théorème de Bayes

$$\underbrace{\mathbb{P}(Y = y|X = \mathbf{x})}_{\text{Probabilité } a \text{ posteriori}} = \frac{\overbrace{\mathbb{P}(Y = y)}^{\text{Probabilité } a \text{ priori}} \cdot \overbrace{\mathbb{P}(X = \mathbf{x}|Y = y)}^{\text{Vraisemblance}}}{\mathbb{P}(X = \mathbf{x})}$$

avec

- $\mathbf{x} \in \mathbb{R}^d$ une observation
- $y \in \mathcal{Y}$ une classe

Théorème de Bayes

$$\underbrace{\mathbb{P}(Y = y|X = \mathbf{x})}_{\text{Probabilité } a \text{ posteriori}} = \frac{\overbrace{\mathbb{P}(Y = y)}^{\text{Probabilité } a \text{ priori}} \cdot \overbrace{\mathbb{P}(X = \mathbf{x}|Y = y)}^{\text{Vraisemblance}}}{\mathbb{P}(X = \mathbf{x})}$$

avec

- $\mathbf{x} \in \mathbb{R}^d$ une observation
- $y \in \mathcal{Y}$ une classe

Dans un problème d'apprentissage supervisé, on cherche à obtenir $\mathbb{P}(Y = y|X = \mathbf{x})$: la probabilité d'observer la classe y sachant l'observation $\mathbf{x}$.

Discriminatif *versus* génératif

Algorithmes discriminatifs

- on cherche à modéliser directement $\mathbb{P}(Y = y|X = \mathbf{x})$
- exemple : la régression logistique (CM06)

Discriminatif *versus* génératif

Algorithmes discriminatifs

- on cherche à modéliser directement $\mathbb{P}(Y = y|X = \mathbf{x})$
- exemple : la régression logistique (CM06)

Algorithmes génératifs

- on cherche à modéliser $\mathbb{P}(X = \mathbf{x}|Y = y)$ et $\mathbb{P}(Y = y)$
  - $\mathbb{P}(X = \mathbf{x}|Y = y)$: comment sont générées les variables explicatives des échantillons qui appartiennent à la classe y ?
  - $\mathbb{P}(Y = y)$: quelle est la probabilité d'observer la classe y dans les données ?
- $\mathbb{P}(Y = y|X = \mathbf{x})$ est calculée en utilisant le théorème de Bayes
- la classe $\hat{y}$ d'une nouvelle observation x est déterminée en trouvant le modèle le plus probable qui aurait pu générer $\mathbf{x}$

Classifieur bayésien naïf

Classifieur bayésien naïf :

- algorithme génératif
- on cherche à estimer $\mathbb{P}(X = \mathbf{x}|Y = y)$ et $\mathbb{P}(y)$

Classifieur bayésien naïf : estimation de $\mathbb{P}(Y = y)$

Probabilité *a priori* des classes $\mathbb{P}(Y = y)$

- $\mathbb{P}(Y = y)$ est estimée en fonction de la fréquence d'apparition de la classe y dans les données d'apprentissage :

$$\mathbb{P}(Y = y) = \frac{m_y}{m}$$

avec

- m_y le nombre de données d'apprentissage qui appartiennent à la classe y
- m le nombre total de données d'apprentissage

Hypothèse du classifieur bayésien naïf (1/2)

- Chaque variable explicative est indépendante des autres variables explicatives conditionnellement à la variable réponse y.

➜ Autrement dit l'existence d'une caractéristique dans une classe est indépendante de l'existence de d'autres caractéristiques dans cette même classe.

Hypothèse du classifieur bayésien naïf (1/2)

- Chaque variable explicative est indépendante des autres variables explicatives conditionnellement à la variable réponse y.

➜ Autrement dit l'existence d'une caractéristique dans une classe est indépendante de l'existence de d'autres caractéristiques dans cette même classe.

Exemple

Soit une banque qui cherche à déterminer si elle doit accorder à un client un prêt ou non en fonction de son âge, s'il a un emploi et son salaire annuel. Le classifieur naïf bayésien fera l'hypothèse qu'avoir un emploi est indépendant de son salaire annuel ; ce qui est peu probable.

Hypothèse du classifieur bayésien naïf (1/2)

- Chaque variable explicative est indépendante des autres variables explicatives conditionnellement à la variable réponse y.

➜ Autrement dit l'existence d'une caractéristique dans une classe est indépendante de l'existence de d'autres caractéristiques dans cette même classe.

Exemple

Soit une banque qui cherche à déterminer si elle doit accorder à un client un prêt ou non en fonction de son âge, s'il a un emploi et son salaire annuel. Le classifieur naïf bayésien fera l'hypothèse qu'avoir un emploi est indépendant de son salaire annuel ; ce qui est peu probable.

➜ On parle de classifieur **naïf** car cette hypothèse **simpliste**, dite naïve, est rarement vérifiée sur des données réelles.

Classifieur bayésien naïf : estimation de $\mathbb{P}(X = \mathbf{x}|Y = y)$

Hypothèse du classifieur bayésien naïf (1/2) : indépendance des variables explicatives conditionnellement aux classes

$\Rightarrow$ Soit $X = [X^1, X^2, ...X^d]$ les variables explicatives, avec d le nombre de variables explicatives

$$\begin{aligned}
\mathbb{P}(X = \mathbf{x}|Y = y) &= \mathbb{P}(X^1 = x^1, X^2 = x^2, \cdots, X^d = x^d|Y = y) \\
&= \mathbb{P}(X^1 = x^1|Y = y) \cdot \mathbb{P}(X^2 = x^2|Y = y) \cdots \mathbb{P}(X^d = x^d|Y = y) \\
&= \prod_{j=1}^{d} \mathbb{P}(X^j = x^j|Y = y)
\end{aligned}$$

Classifieur bayésien naïf : estimation de $\mathbb{P}(X = \mathbf{x}|Y = y)$

Hypothèse du classifieur bayésien naïf (1/2) : indépendance des variables explicatives conditionnellement aux classes

$\Rightarrow$ Soit $X = [X^1, X^2, ...X^d]$ les variables explicatives, avec d le nombre de variables explicatives

$$\begin{aligned}
\mathbb{P}(X = \mathbf{x}|Y = y) &= \mathbb{P}(X^1 = x^1, X^2 = x^2, \cdots, X^d = x^d|Y = y) \\
&= \mathbb{P}(X^1 = x^1|Y = y) \cdot \mathbb{P}(X^2 = x^2|Y = y) \cdots \mathbb{P}(X^d = x^d|Y = y) \\
&= \prod_{j=1}^{d} \mathbb{P}(X^j = x^j|Y = y)
\end{aligned}$$

La seconde égalité est une conséquence de l'hypothèse du classifieur bayésien naïf.

Hypothèse du classifieur bayésien naïf (2/2) :

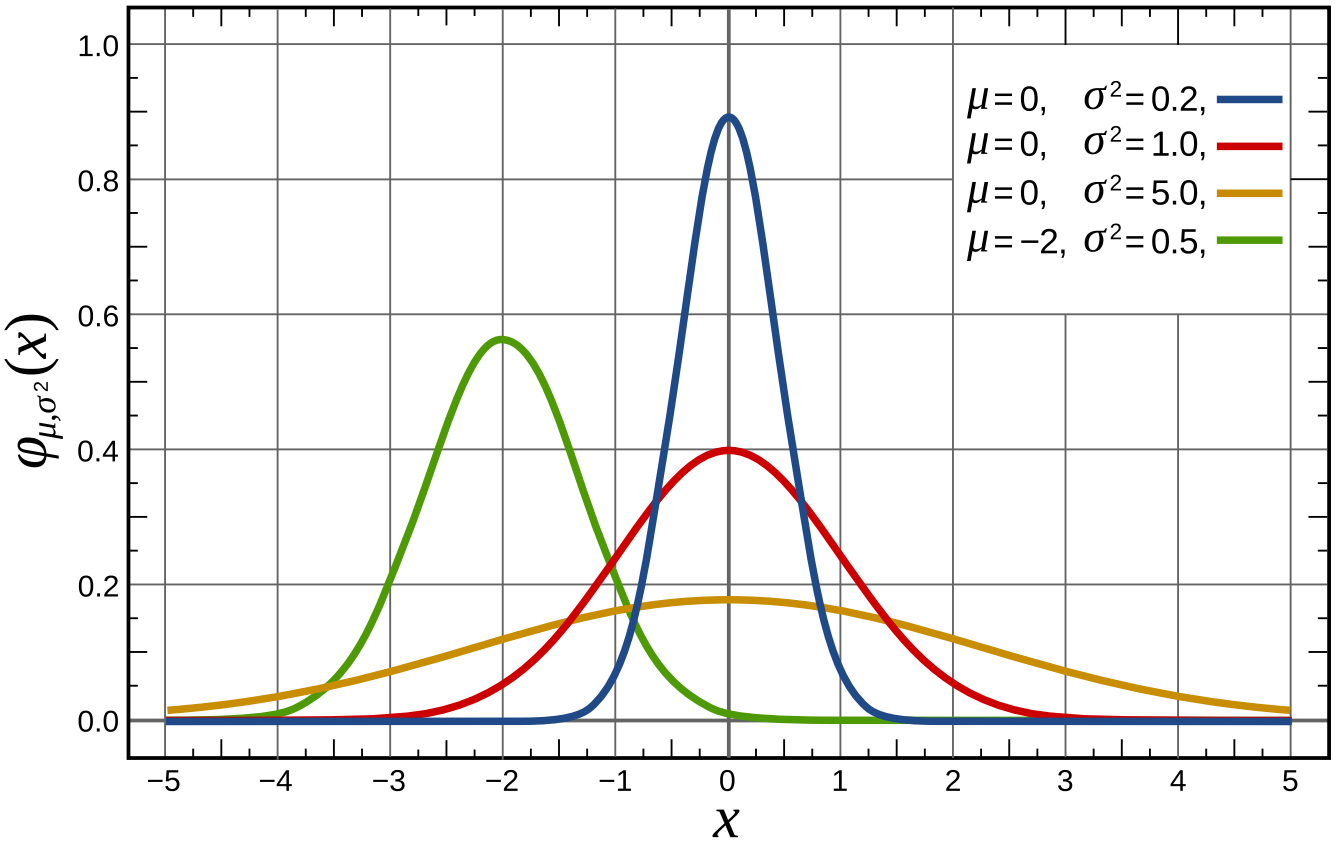
$\mathbb{P}(X^j = x^j|Y = y) \sim \mathcal{N}(\mu_y^j, {\sigma_y^j}^2)$

- la probabilité $\mathbb{P}(X^j = x^j|Y = y)$ suit une loi normale (gaussienne) de moyenne μ_y^j et de variance ${\sigma_y^j}^2$
- μ_y^j (${\sigma_y^j}^2$) est la moyenne (variance) des valeurs prises par les données d'apprentissage appartenant à la classe y pour la j-ième variable explicative

$$\mathbb{P}(X^j = x^j|Y = y) = \frac{1}{\sqrt{2\pi} \cdot \sigma_y^j} e^{-\frac{1}{2{\sigma_y^j}^2}(x^j - \mu_y^j)^2}$$

Classifieur bayésien naïf : estimation de $\mathbb{P}(X = \mathbf{x}|Y = y)$

Rappels (CM01)



Densité de probabilité de la loi normale pour différentes valeurs de μ et σ.

Source : Wikipedia

Classifieur bayésien naïf : prédiction

Pour une nouvelle observation $\mathbf{x}$, on prédit sa classe $\hat{y}$ tel que :

$$\begin{aligned} \hat{y} &= \underset{k}{\operatorname{argmax}}\, \mathbb{P}(Y = k | X = \mathbf{x}) \\ &= \underset{k}{\operatorname{argmax}}\, \mathbb{P}(Y = k) \cdot \prod_{j=1}^{d} \mathbb{P}(X^j = x^j | Y = k) \end{aligned}$$

Classifieur bayésien naïf : prédiction

Pour une nouvelle observation $\mathbf{x}$, on prédit sa classe $\hat{y}$ tel que :

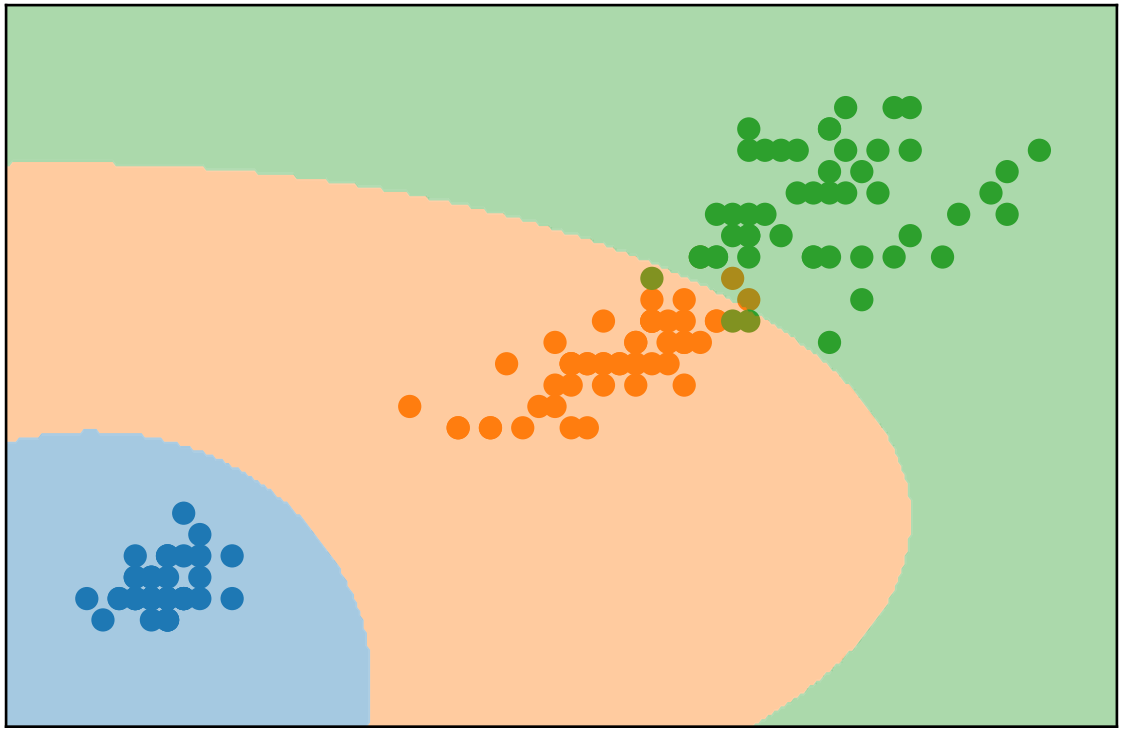
$$\begin{aligned}\hat{y} &= \underset{k}{\operatorname{argmax}}\, \mathbb{P}(Y = k | X = \mathbf{x}) \\ &= \underset{k}{\operatorname{argmax}}\, \mathbb{P}(Y = k) \cdot \prod_{j=1}^{d} \mathbb{P}(X^j = x^j | Y = k)\end{aligned}$$

Notes : La deuxième égalité vient du théorème de Bayes :
$\mathbb{P}(Y = k | X = \mathbf{x}) = \frac{\mathbb{P}(Y=k) \cdot \mathbb{P}(X=\mathbf{x}|Y=k)}{\mathbb{P}(X=\mathbf{x})} \propto \mathbb{P}(Y = k) \cdot \mathbb{P}(X = \mathbf{x}|Y = k)$.
Le dénominateur, *i.e.* la probabilité $\mathbb{P}(X = \mathbf{x})$ (appelée évidence), est identique quelque soit la classe k, et n'a donc pas besoin d'être calculée. Cette probabilité $\mathbb{P}(X = \mathbf{x})$ peut être vue comme un facteur de normalisation qui permet d'assurer que la probabilité $\mathbb{P}(Y = y|X = \mathbf{x})$ soit comprise entre 0 et 1.
On a

$$\begin{aligned}\mathbb{P}(X = \mathbf{x}) &= \sum_{c=1}^{\mathcal{C}} \Big(\mathbb{P}(Y = c) \cdot \mathbb{P}(X = \mathbf{x}|Y = c) \Big) \\ &= \sum_{c=1}^{\mathcal{C}} \Big(\mathbb{P}(Y = c) \cdot \prod_{j=1}^{d} \big(\mathbb{P}(X^j = x^j | Y = c) \big) \Big)\end{aligned}$$

avec $\mathcal{C}$ le nombre de classes.

Résultat sur le jeu de données iris



Visualisation des différentes frontières de décision.

Notes : Les points représentent les données d'apprentissage pour les trois classes du jeu de données iris (voir diapo 7).

La zone colorée orange (bleu / vert) représente la région où le classifieur bayésien naïf prédira la classe orange `versicolor` (bleue `setosa` / verte `virginica`).

Classifieur bayésien naïf

Avantages

- il nécessite peu d'échantillons d'apprentissage pour estimer les paramètres des lois normales : moyenne et variance de chaque variable explicative en fonction des classes à prédire ($2 \cdot d \cdot \mathcal{C}$ paramètres à estimer)
- il permet le passage à l'échelle (*i.e.* prédiction rapide pour une grande quantité de données)
- malgré son hypothèse très simpliste, il a d'excellente performance pour certains problèmes de classification (*e.g.* la classification de texte incluant la détection de spams, la classification d'emails dans des répertoire ou la classification de produits par rapport à leur description)

Analyse discriminante linéaire

Analyse discriminante linéaire ou *Linear Discriminant Analysis* (LDA)

- est un algorithme génératif
- $\mathbb{P}(Y = y) = \frac{m_y}{m}$ comme le classifieur bayésien naïf
- $\mathbb{P}(X = \mathbf{x}|Y = y)$ est modélisé par une loi de distribution normale multivariée

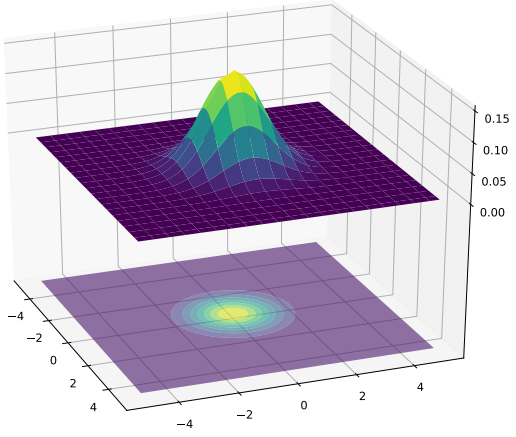
$$\begin{aligned} \mathbb{P}(X = \mathbf{x}|Y = y) &\sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}) \\ &= \frac{1}{(2\pi)^{\frac{d}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_y)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_y)} \end{aligned}$$

avec

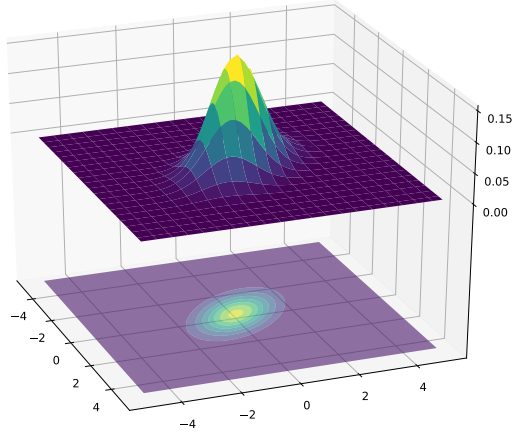
- $\boldsymbol{\mu}_y \in \mathbb{R}^d$ le vecteur moyenne pour les d variables explicatives pour les données d'apprentissage qui appartiennent à la classe y
- $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ la matrice de covariance calculée à partir de toutes les données d'apprentissage (indépendamment de leur classe)
- $|\cdot|$ le déterminant

Analyse discriminante linéaire : loi normale multivariée

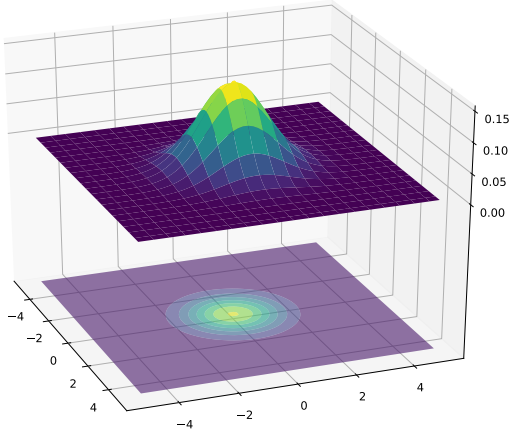
$$\boldsymbol{\mu} = [0\ 0]\ \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



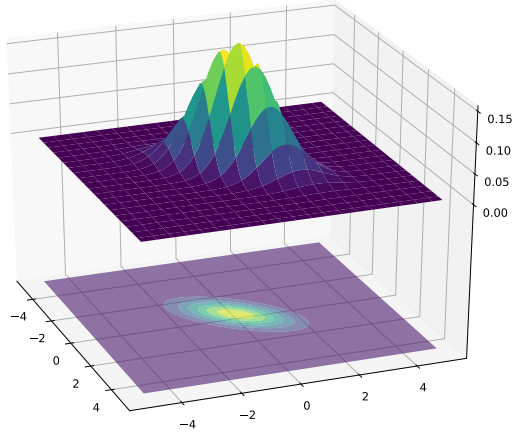
$$\boldsymbol{\mu} = [0\ 0]\ \boldsymbol{\Sigma} = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$$



$$\boldsymbol{\mu} = [0\ 0]\ \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.5 \\ -0.5 & 1 \end{bmatrix}$$

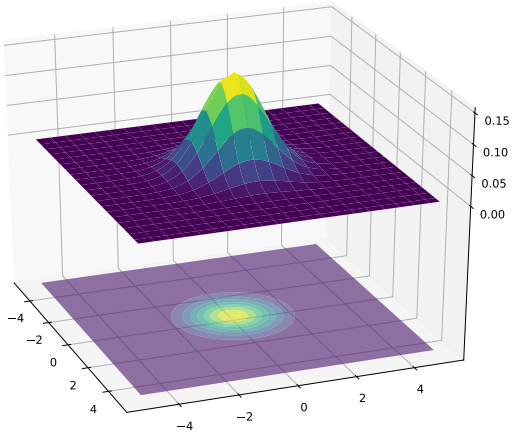


$$\boldsymbol{\mu} = [0\ 0]\ \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.5 \\ 0.8 & 1 \end{bmatrix}$$

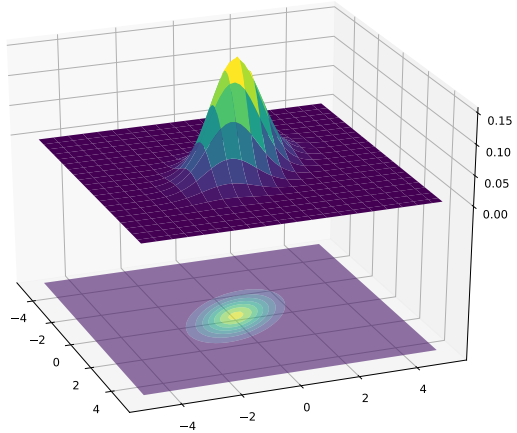


Analyse discriminante linéaire : loi normale multivariée

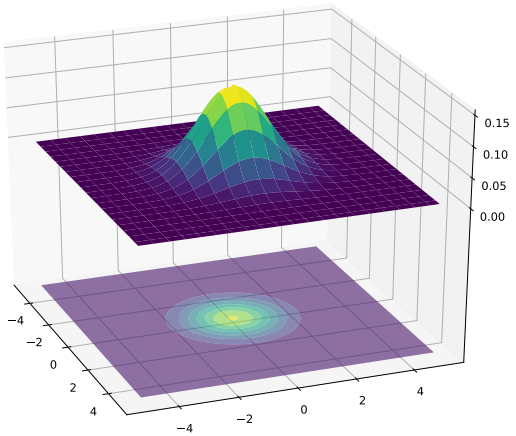
$\boldsymbol{\mu} = [0\ 0]\ \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$



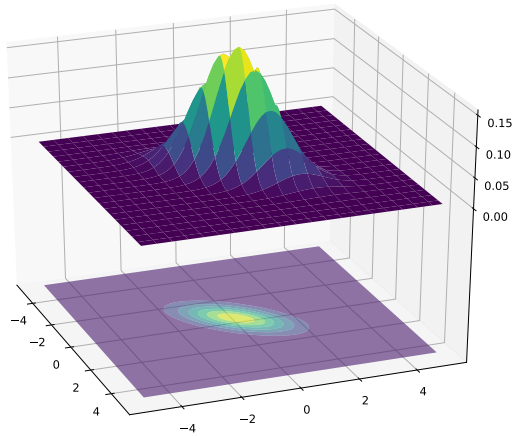
$\boldsymbol{\mu} = [0\ 0]\ \boldsymbol{\Sigma} = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$



$\boldsymbol{\mu} = [0\ 0]\ \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.5 \\ -0.5 & 1 \end{bmatrix}$



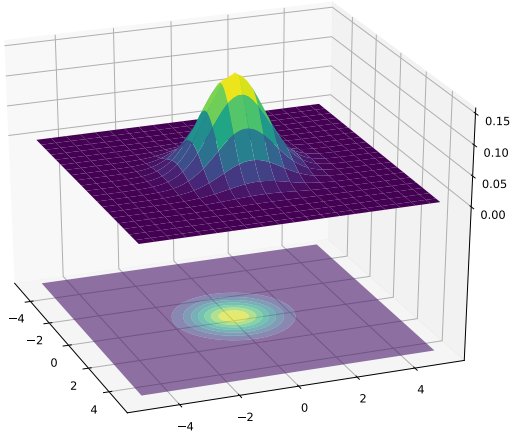
$\boldsymbol{\mu} = [0\ 0]\ \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.5 \\ 0.8 & 1 \end{bmatrix}$



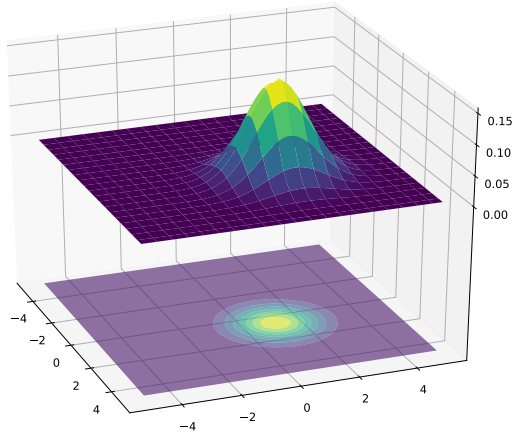
- La matrice de covariance $\boldsymbol{\Sigma}$ définit le type de relation entre les variables

Analyse discriminante linéaire : loi normale multivariée

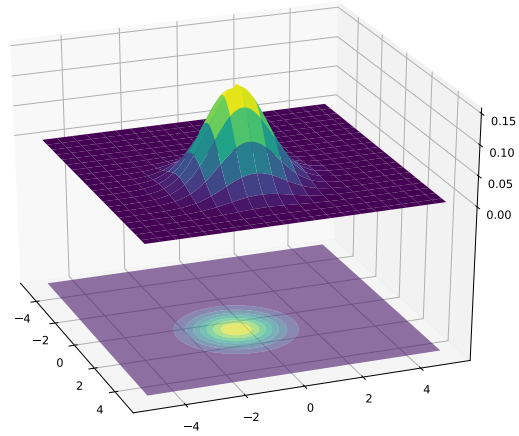
$\boldsymbol{\mu} = [0\ 0]\ \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$



$\boldsymbol{\mu} = [1\ 1]\ \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

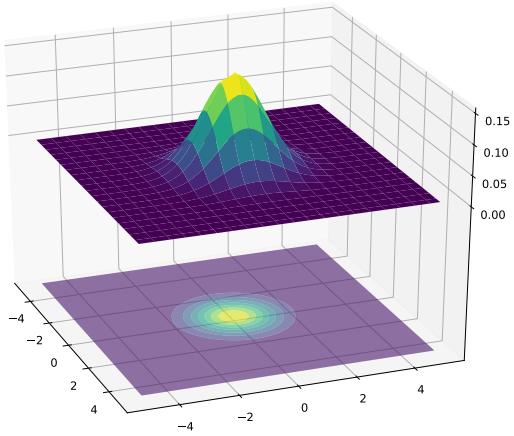


$\boldsymbol{\mu} = [1\ -0.5]\ \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

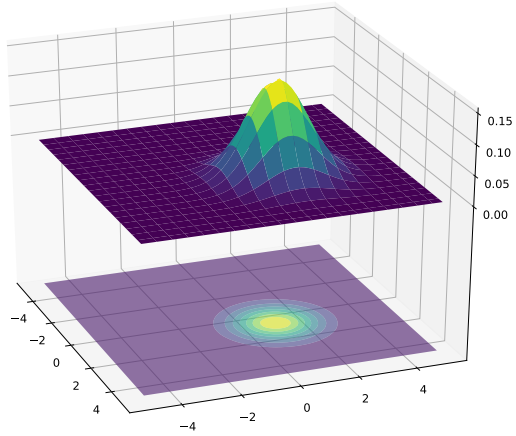


Analyse discriminante linéaire : loi normale multivariée

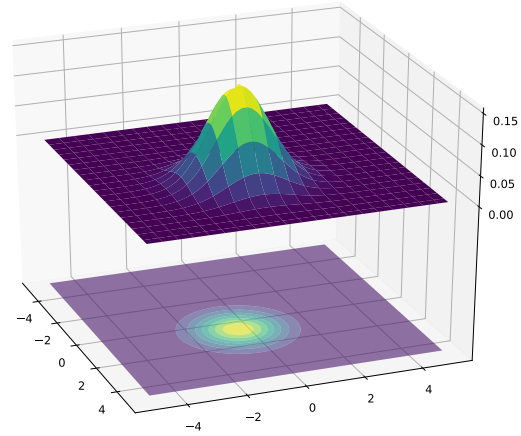
$\boldsymbol{\mu} = [0\ 0]\ \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$



$\boldsymbol{\mu} = [1\ 1]\ \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$



$\boldsymbol{\mu} = [1\ -0.5]\ \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$



- La moyenne $\boldsymbol{\mu}$ définit la “position” de la loi normale multivariée

Analyse discriminante quadratique

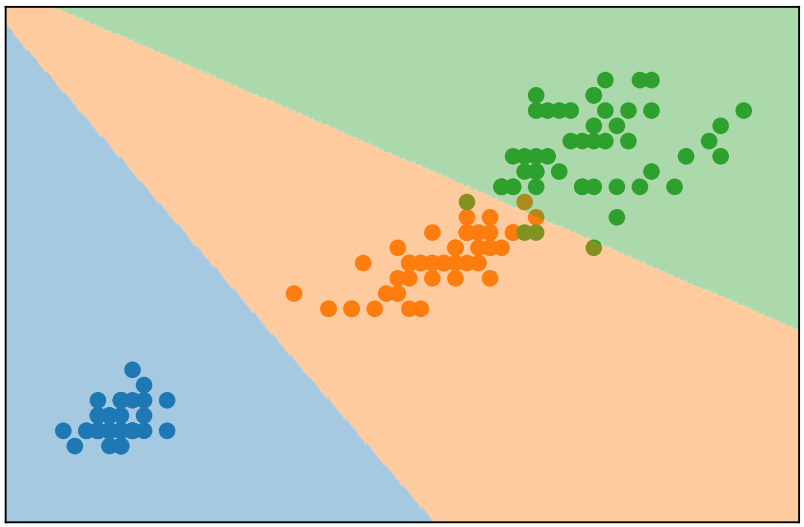
Analyse discriminante quadratique ou *Quadratic Discriminant Analysis* (QDA)

- similaire à l'Analyse Discriminante Linéaire
- **mais** une matrice de covariance $\mathbf{\Sigma}_y$ est calculée pour chaque classe y

$$\begin{aligned} \mathbb{P}(X = \mathbf{x}|Y = y) &\sim \mathcal{N}(\boldsymbol{\mu}_y, \mathbf{\Sigma}_y) \\ &= \frac{1}{(2\pi)^{\frac{d}{2}} |\mathbf{\Sigma}_y|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_y)^T \mathbf{\Sigma}_y^{-1}(\mathbf{x}-\boldsymbol{\mu}_y)} \end{aligned}$$

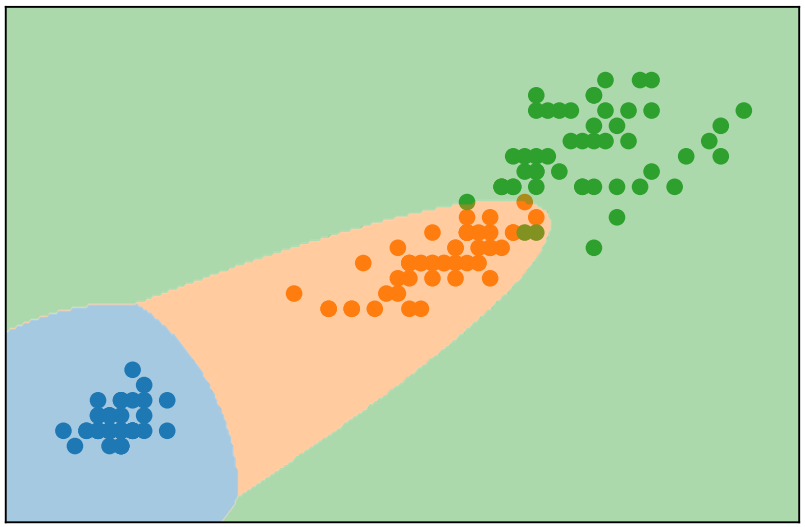
Comparaison des algorithmes

Résultat sur le jeu de données iris



LDA

frontières de décision linéaires



QDA

Visualisation des différentes frontières de décision

Comparaison des algorithmes

Comment savoir s'il est préférable de calculer une matrice de covariance sur l'ensemble des données ou par classe ?

Comparaison des algorithmes

Comment savoir s'il est préférable de calculer une matrice de covariance sur l'ensemble des données ou par classe ?

Autrement dit, faut-il préférer LDA ou QDA ?

Comparaison des algorithmes

Comment savoir s'il est préférable de calculer une matrice de covariance sur l'ensemble des données ou par classe ?
Autrement dit, faut-il préférer LDA ou QDA ?

➜ Compromis biais-variance (CM06)

- en faisant l'hypothèse que les données aient la même matrice de covariance (quelque soit leur classe d'appartenance y), LDA est un algorithme peu flexible (frontière de décision linéaire)
  $\Rightarrow$ plus fort biais, mais potentiellement une variance plus faible
- à l'inverse QDA va avoir une variance plus forte, mais potentiellement un plus faible biais

Comparaison des algorithmes

Comment savoir s'il est préférable de calculer une matrice de covariance sur l'ensemble des données ou par classe ?
Autrement dit, faut-il préférer LDA ou QDA ?

➜ Compromis biais-variance (CM06)

- en faisant l'hypothèse que les données aient la même matrice de covariance (quelque soit leur classe d'appartenance y), LDA est un algorithme peu flexible (frontière de décision linéaire)
  $\Rightarrow$ plus fort biais, mais potentiellement une variance plus faible
- à l'inverse QDA va avoir une variance plus forte, mais potentiellement un plus faible biais

➜ Compromis calculatoire

- calculer une matrice de covariance (matrice symétrique) pour d variables explicatives nécessite $d \cdot (d+1)/2$ opérations
- QDA nécessite donc $\mathcal{C} \cdot d \cdot (d+1)/2$ opérations (calcul d'une matrice de covariance par classe)

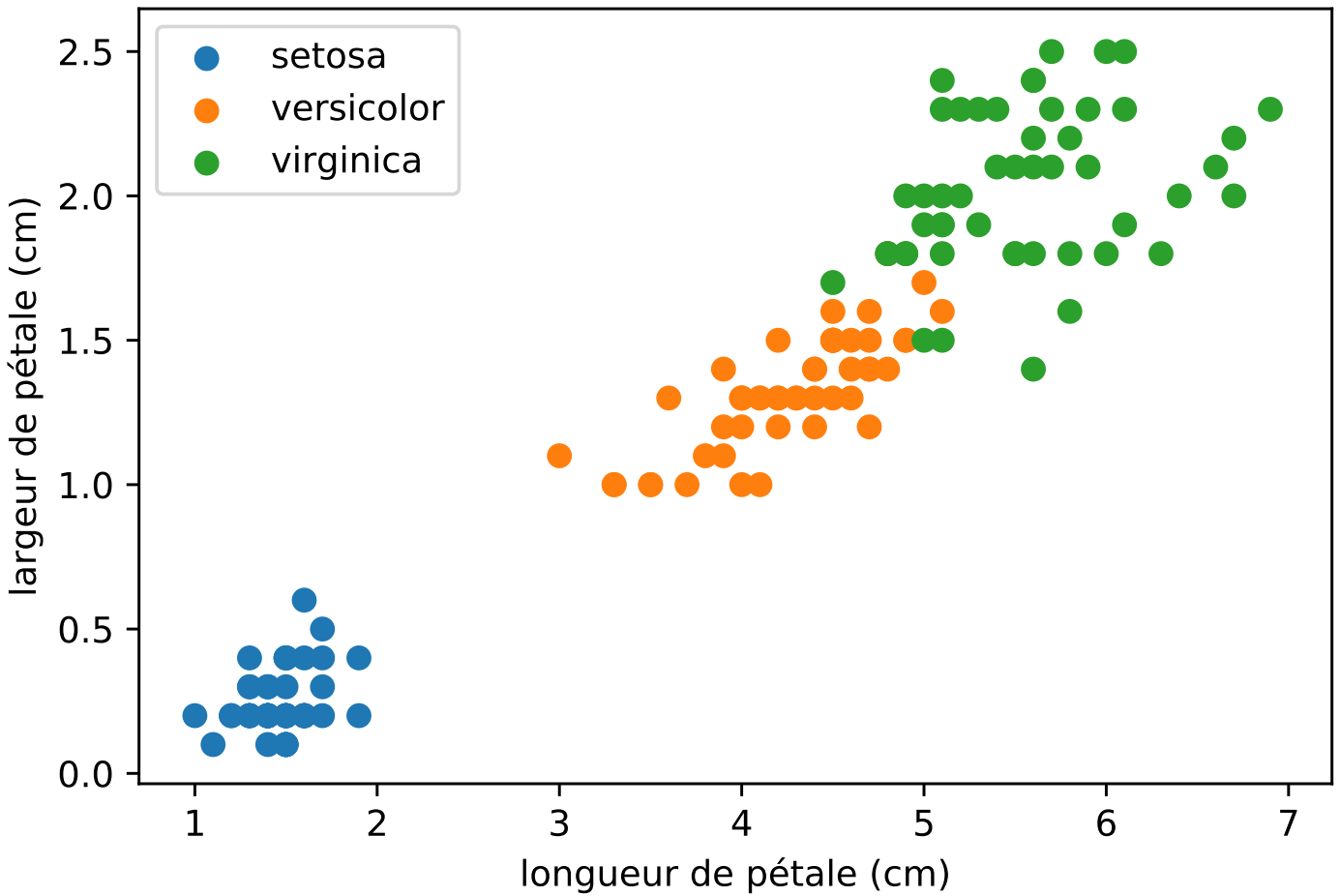
Comparaison des algorithmes

Conclusions

- préférer LDA lorsque le nombre de données d'apprentissage est petit
- préférer QDA lorsque le nombre de données d'apprentissage est suffisamment grand, et donc la variance du modèle n'est plus un problème

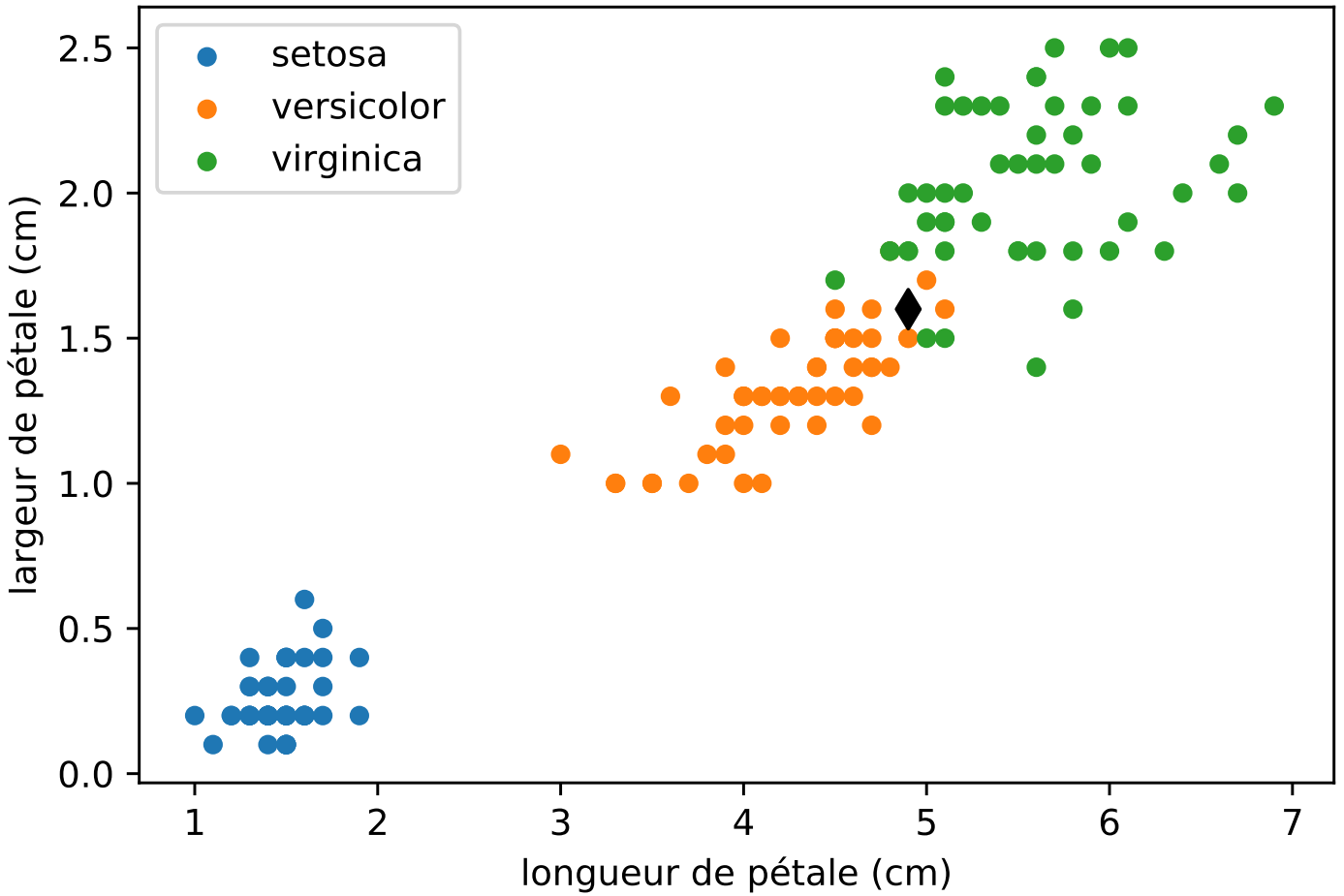
k-Plus Proches Voisins

Données d'apprentissage : $m = 150$, $d = 2$ et $\mathcal{C} = 3$



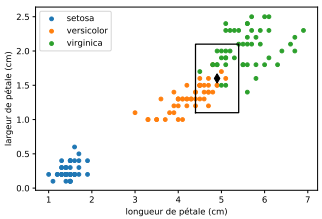
Problématique

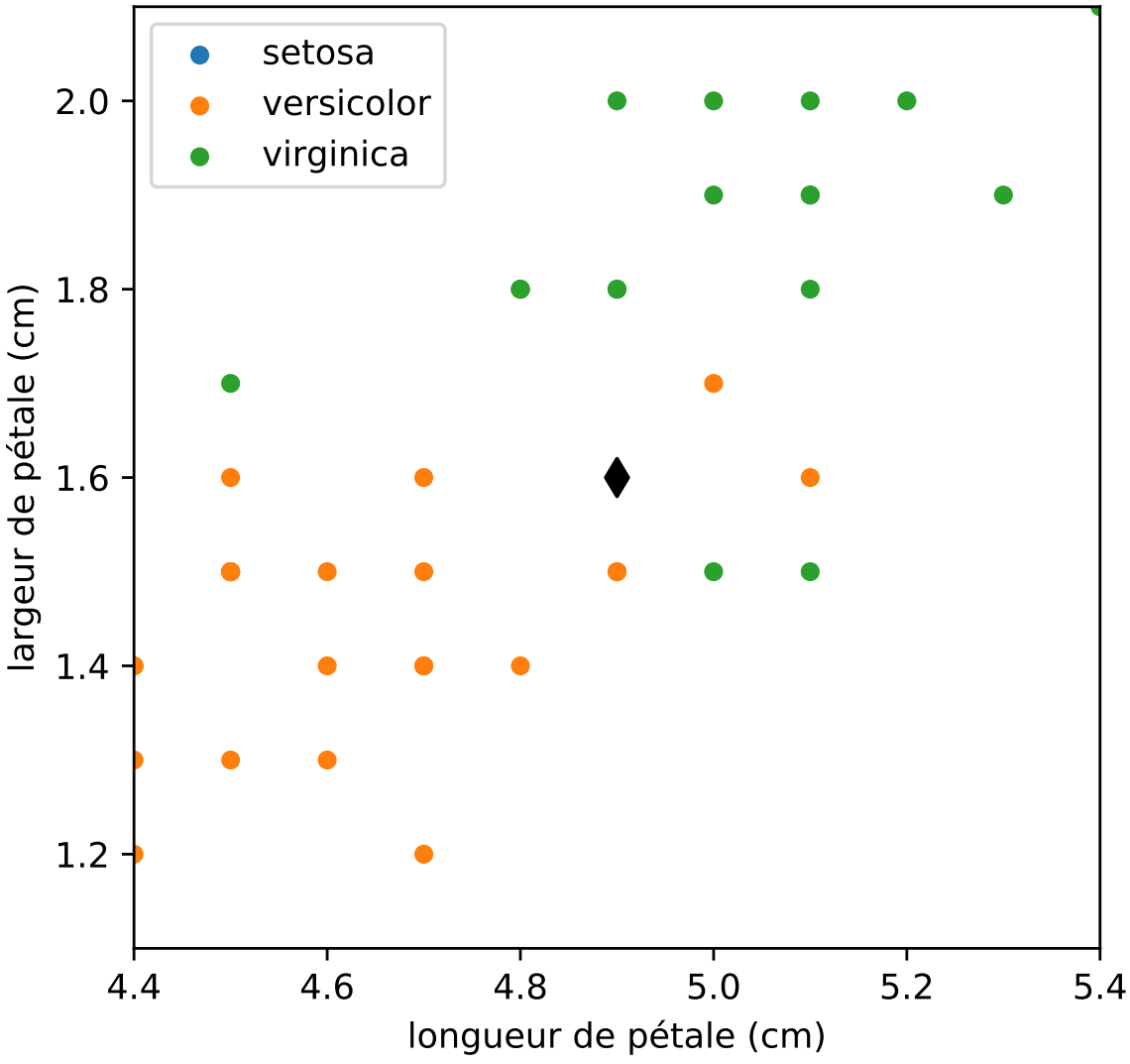
On souhaite déterminer la classe de la nouvelle observation (⧫)



Problématique

On souhaite déterminer la classe de la nouvelle observation (⧫)

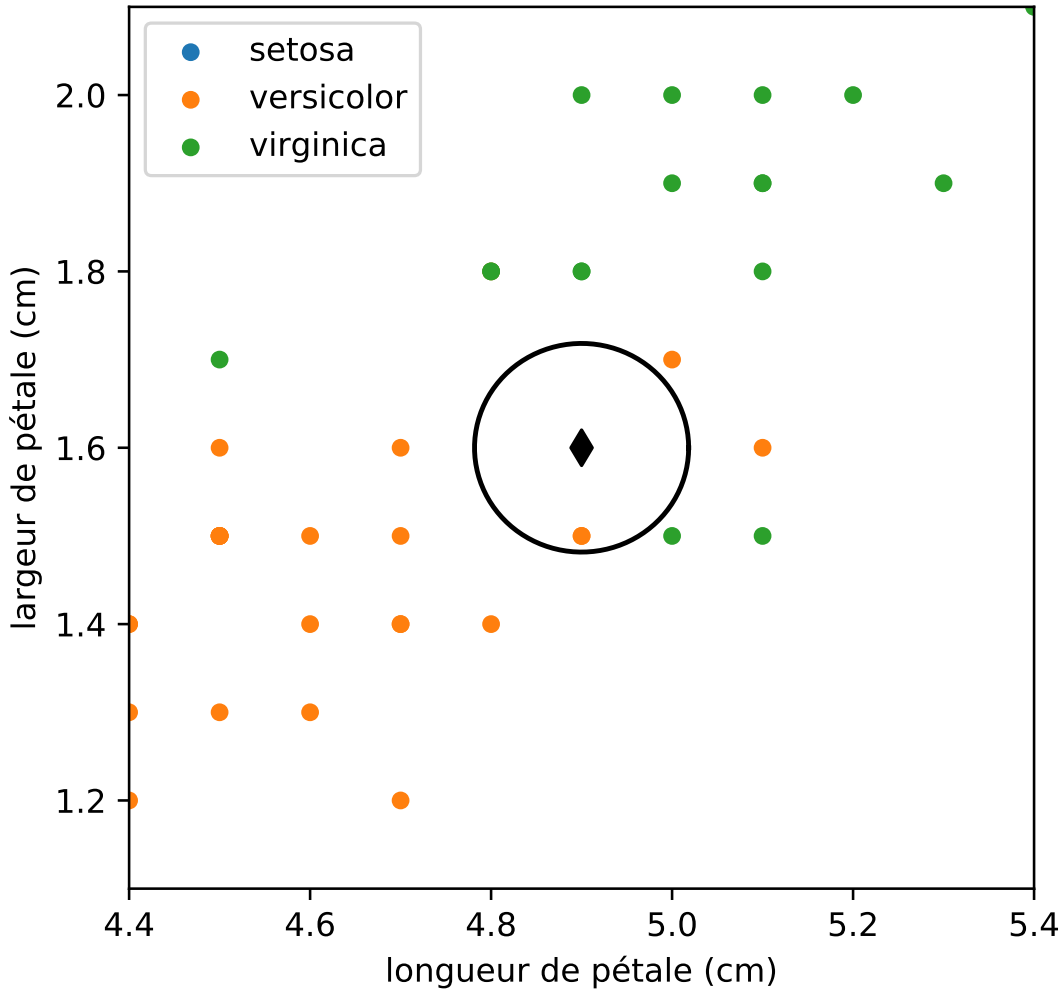




Zoom

Principe

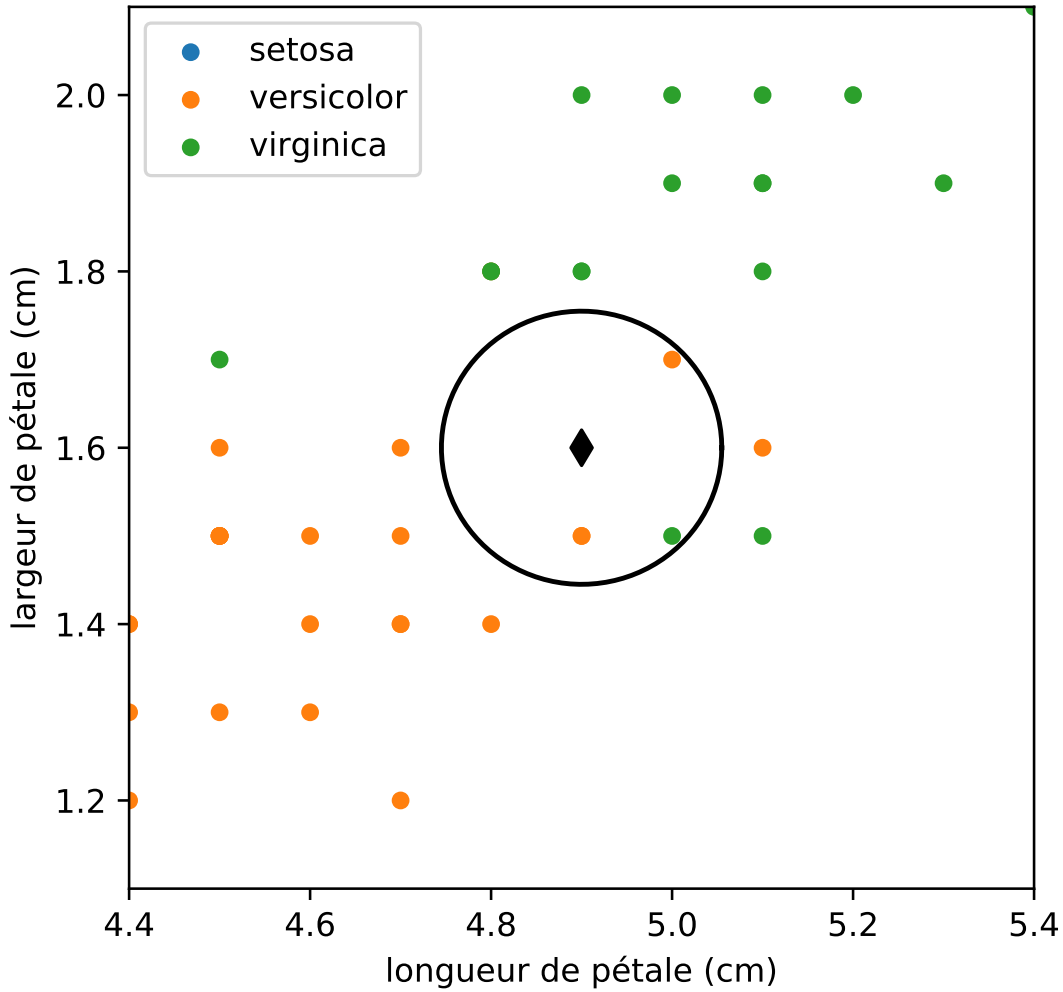
Idée générale : on regarde la classe des échantillons les plus proches et on attribue la classe majoritaire à la nouvelle observation



1-Plus Proche Voisin $\Rightarrow$ versicolor

Principe

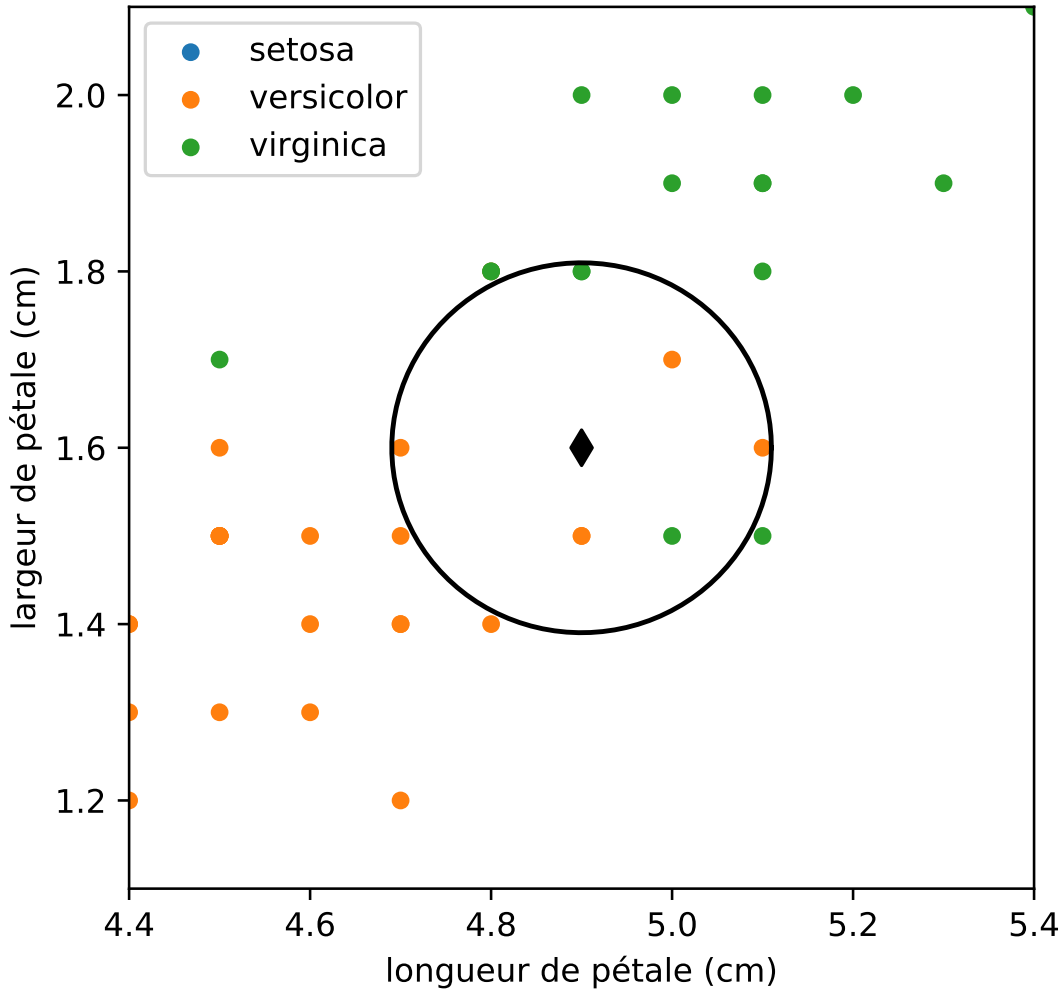
Idée générale : on regarde la classe des échantillons les plus proches et on attribue la classe majoritaire à la nouvelle observation



3-Plus Proches Voisins $\Rightarrow$ versicolor

Principe

Idée générale : on regarde la classe des échantillons les plus proches et on attribue la classe majoritaire à la nouvelle observation



6-Plus Proches Voisins $\Rightarrow$ versicolor

Algorithme

Algorithme : soit $\mathbf{x}$ une nouvelle observation à étiquetter

- pour chaque donnée d'apprentissage $\{\mathbf{x}_i\}_{i=1}^m$, calculer $d(\mathbf{x}, \mathbf{x}_i)$
- trier par ordre croissant les $d(\mathbf{x}, \mathbf{x}_i)$
- associer à $\mathbf{x}$ la classe $\hat{y}$ qui correspond à la classe majoritaire parmi les k plus petite distance $d(\mathbf{x}, \mathbf{x}_i)$ *

Algorithme

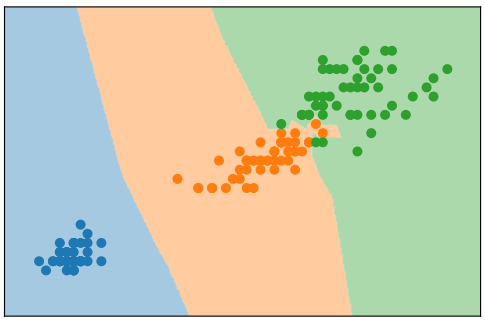
Algorithme : soit $\mathbf{x}$ une nouvelle observation à étiquetter

- pour chaque donnée d'apprentissage $\{\mathbf{x}_i\}_{i=1}^m$, calculer $d(\mathbf{x}, \mathbf{x}_i)$
- trier par ordre croissant les $d(\mathbf{x}, \mathbf{x}_i)$
- associer à $\mathbf{x}$ la classe $\hat{y}$ qui correspond à la classe majoritaire parmi les k plus petite distance $d(\mathbf{x}, \mathbf{x}_i)$ *

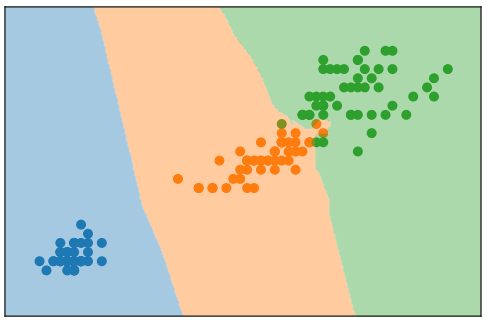
* En cas d'égalité, on tire au sort $\hat{y}$ parmi toutes les classes majoritaires.
Il existe également deux autres possibilités : (1) augmenter k de 1 (mais le problème d'égalité peut persister), et (2) utiliser la distance des données d'apprentissage à l'observation $\mathbf{x}$ pour pondérer le calcul de la classe majoritaire (les données les plus proches auront un poids plus grand).

Influence de l'hyperparamètre k

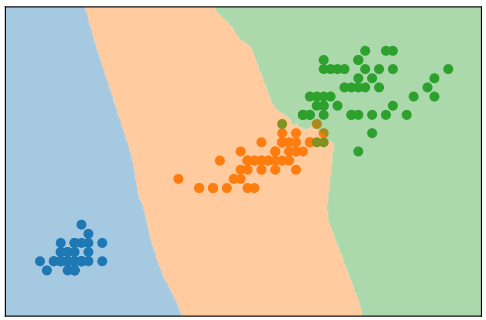
Comment choisir la valeur de k ?



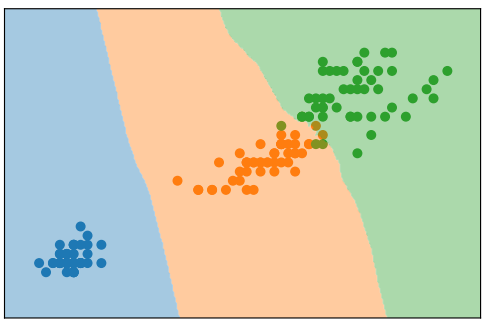
$k = 1$



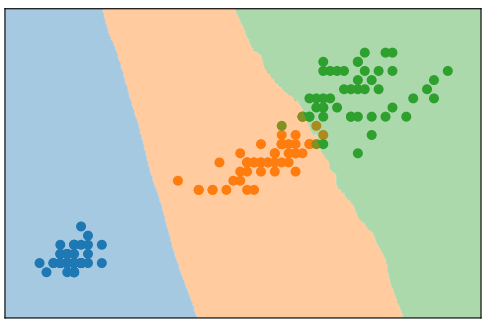
$k = 3$



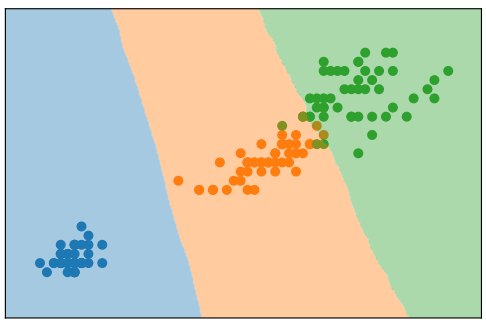
$k = 5$



$k = 10$



$k = 30$



$k = 50$

Influence de l'hyperparamètre k

Comment choisir la valeur de k ?

- k petit
  - décision locale
  - modèle complexe $\Rightarrow$ forte variance (sur-apprentissage)
- k grand
  - décision globale
  - modèle plus simple $\Rightarrow$ fort biais

Évaluation des algorithmes de classification

Évaluation

Comment évaluer les performances des algorithmes de classification ?

La matrice de confusion : $C = \{c_{ij}\}_{i,j=1}^{\mathcal{C}}$ pour $\mathcal{C}$ classes

Réelle \ Prédite	1	2	$\cdots$	$\cdots$	j	$\cdots$	$\mathcal{C}$
1	c_{11}	c_{12}	$\cdots$	$\cdots$	c_{1j}	$\cdots$	$c_{1\mathcal{C}}$
2	c_{21}	c_{22}	$\cdots$	$\cdots$	c_{2j}	$\cdots$	$c_{2\mathcal{C}}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	c_{i1}	c_{i2}	$\cdots$	$\cdots$	c_{ij}	$\cdots$	$c_{i\mathcal{C}}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathcal{C}$	$c_{\mathcal{C}1}$	$c_{\mathcal{C}2}$	$\cdots$	$\cdots$	$c_{\mathcal{C}j}$	$\cdots$	$c_{\mathcal{C}\mathcal{C}}$

Évaluation

Comment évaluer les performances des algorithmes de classification ?

La matrice de confusion : $C = \{c_{ij}\}_{i,j=1}^{\mathcal{C}}$ pour $\mathcal{C}$ classes

Réelle \ Prédite	1	2	$\cdots$	$\cdots$	j	$\cdots$	$\mathcal{C}$
1	c_{11}	c_{12}	$\cdots$	$\cdots$	c_{1j}	$\cdots$	$c_{1\mathcal{C}}$
2	c_{21}	c_{22}	$\cdots$	$\cdots$	c_{2j}	$\cdots$	$c_{2\mathcal{C}}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	c_{i1}	c_{i2}	$\cdots$	$\cdots$	c_{ij}	$\cdots$	$c_{i\mathcal{C}}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathcal{C}$	$c_{\mathcal{C}1}$	$c_{\mathcal{C}2}$	$\cdots$	$\cdots$	$c_{\mathcal{C}j}$	$\cdots$	$c_{\mathcal{C}\mathcal{C}}$

- c_{ij} corresponds au nombre d'échantillons qui appartiennent à la classe i et pour lequel l'algorithme de classification a prédit la classe j
- les éléments diagonaux $\{c_{ii}\}_{i=1}^{j}$ correspondent donc aux échantillons dont la classe a correctement été prédite par l'algorithme
- $\sum_{i=1}^{\mathcal{C}} \sum_{j=1}^{\mathcal{C}} c_{ij} = N$ avec N le nombre d'observations (test)

Mesures d'évaluation

- Taux de bonne classification (en anglais *Overall Accuracy*) :

$$OA = \frac{\sum_{i=1}^{C} c_{ii}}{N}$$

- $0\ \% \leq OA \leq 100\ \%$ on cherche à maximiser le taux de bonne classification ($100\ \%$)

Mesure d'évaluation

- Le coefficient Kappa :

$$\text{Kappa} = \frac{OA - p_h}{1 - p_h}$$

avec $p_h = \frac{1}{N^2} \sum_{i=1}^{C} \left(\sum_{j=1}^{C} c_{ij} \right) \left(\sum_{j=1}^{C} c_{ji} \right)$ le pourcentage de bonnes classifications attribué au hasard

- Le coefficient Kappa permet de s'affranchir du taux de bonne classification dû à l'aléatoire
- Référentiel de Landis et Koch pour interpréter la valeur de Kappa.

Interprétation	**Valeur de Kappa**
Excellente	1.00 – 0.81
Bonne	0.80 – 0.61
Faible	0.60 – 0.41
Négligeable	0.20 – 0.00
Mauvaise	< 0.00

Source : J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*. (1) :159 ?174, 1977.

Cas particulier de la classification binaire : $\mathcal{C} = 2$

Réelle \ Prédite	Positive	Négative
Positive	Vrais Positifs (TP)	Faux Negatifs (FN)
Negative	Faux Positifs (FP)	Vrais Négatifs (TN)

- Taux de bonne classification : $OA = \frac{TP+TN}{TP+FN+FP+TN}$
- Taux de faux positifs : $FPR = \frac{FP}{FP+TN}$
- Taux de faux négatifs : $FNR = \frac{FN}{FN+TP}$

Exemple

Soit un hôpital qui cherche à déterminer les patients malades parmi un échantillon de 100 patients. Imaginons que seulement 5 patients soient malades (classe positive), et donc 95 patients soient sains (classe négative). Un algorithme de classification qui prédit que tous les patients sont sains aura un OA de 95 %. Cependant, il sera incapable de trouver les patients malade : taux de faux positifs $FPR = 100$ %.