

1 Apprentissage et généralisation

2 Méthodes de rééchantillonnage

- Validation
- Validation croisée
- Bootstrap

- données non contradictoires \Rightarrow un modèle parfait existe
- contradictoire : $\mathbf{x}_i = \mathbf{x}_j, y_i \neq y_j$ et $i \neq j$
- construction d'un modèle parfait :
 - données non contradictoires \Rightarrow il existe σ tel que

$$\exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) < \frac{\epsilon}{N-1},$$

- pour $i \neq j$
- classifieur

$$\psi(\mathbf{x}) = \text{signe}\left(\sum_{i=1}^N y_i \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{2\sigma^2}\right)\right)$$

- $\left|\sum_{i=1}^N y_i \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) - y_j\right| \leq \epsilon$ et donc $\psi(\mathbf{x}_j) = y_j$

Généralisation

- on observe un ensemble d'*apprentissage* $\mathcal{D} = (\mathbf{x}_i, y_i)_{1 \leq i \leq N}$
- erreur (risque) *empirique* d'un classifieur ψ :

$$\widehat{L}(\psi, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^N \mathbb{I}_{\{\psi(\mathbf{x}_i) \neq y_i\}}$$

- on peut généralement trouver ψ tel que $\widehat{L}(\psi, \mathcal{D}) = 0$
- mais on s'intéresse à la **généralisation** :
 - capacité à bien classer de nouvelles observations
 - liée à $\widehat{L}(\psi, \mathcal{D}')$ pour un nouvel ensemble de données $\mathcal{D}' = (\mathbf{x}'_i, y'_i)_{1 \leq i \leq N'}$, l'ensemble de *test*
- question fondamentale : lien entre $\widehat{L}(\psi, \mathcal{D})$ et $\widehat{L}(\psi, \mathcal{D}')$

Modélisation mathématique

- observations i.i.d. (indépendantes, identiquement distribuées) selon (X, Y)
- erreur *théorique* (« réelle ») d'un classifieur ψ :

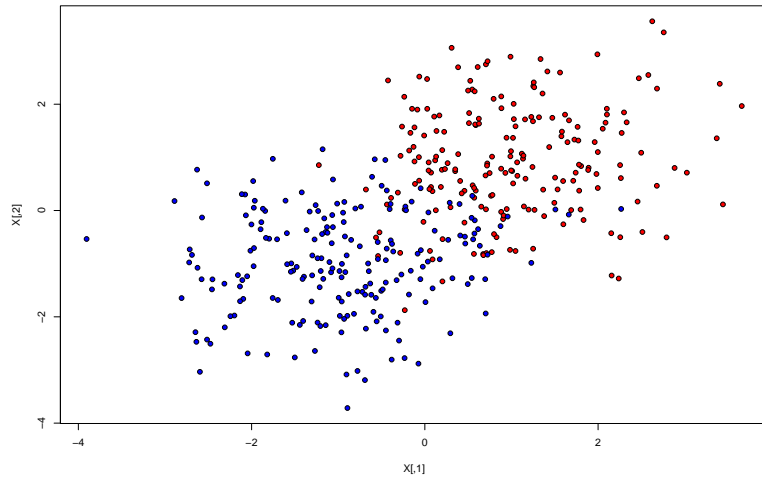
$$L(\psi) = \mathbb{P}(\psi(X) \neq Y)$$

- risque bayésien (plus petite erreur possible) :

$$L^* = \inf_{\psi: \mathbb{R}^p \rightarrow \{-1, 1\}} L(\psi)$$

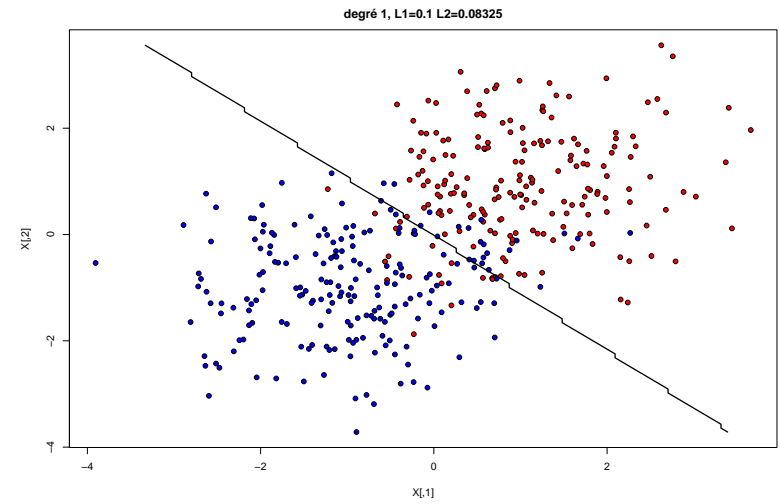
- trois questions :
 - 1 **généralisation** : peut on prévoir $\widehat{L}(\psi, \mathcal{D}')$ connaissant seulement $\widehat{L}(\psi, \mathcal{D})$ (si \mathcal{D} et \mathcal{D}' sont issus de (X, Y)) ?
 - 2 **performances réelles** : peut on estimer $L(\psi)$ à partir de $\widehat{L}(\psi, \mathcal{D})$ (en général $\widehat{L}(\psi, \mathcal{D}) < L(\psi)$)
 - 3 **consistance** : peut on trouver ψ^* d'erreur théorique L^* ?

Exemple de surapprentissage



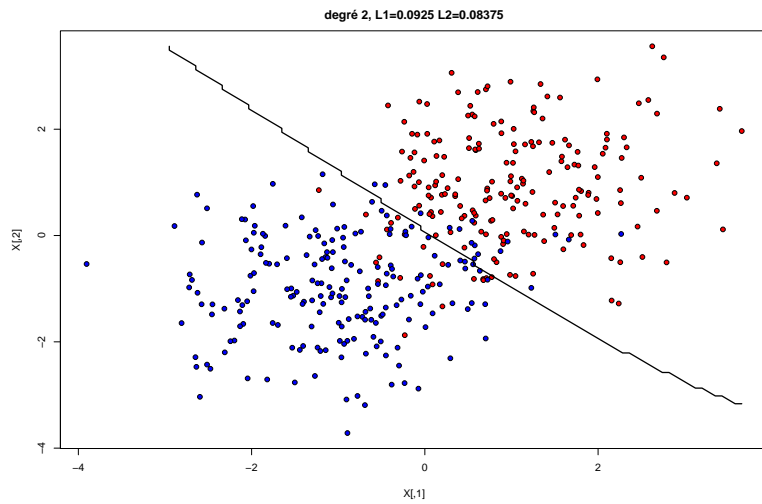
Navigation icons: back, forward, search, etc.

Exemple de surapprentissage



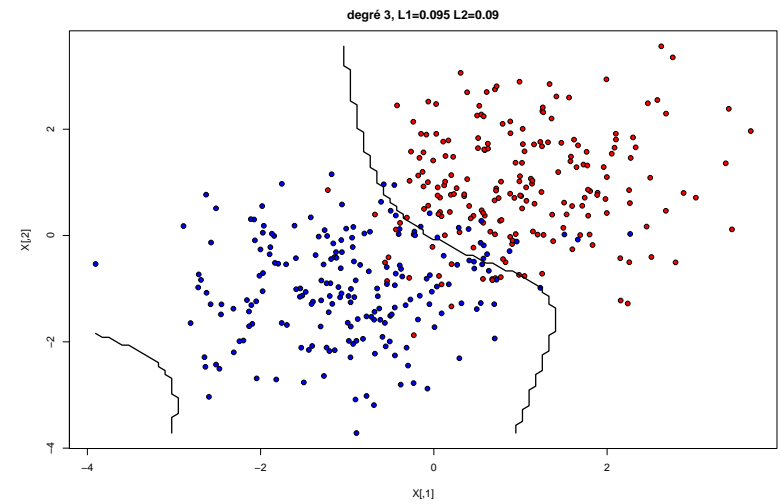
Navigation icons: back, forward, search, etc.

Exemple de surapprentissage



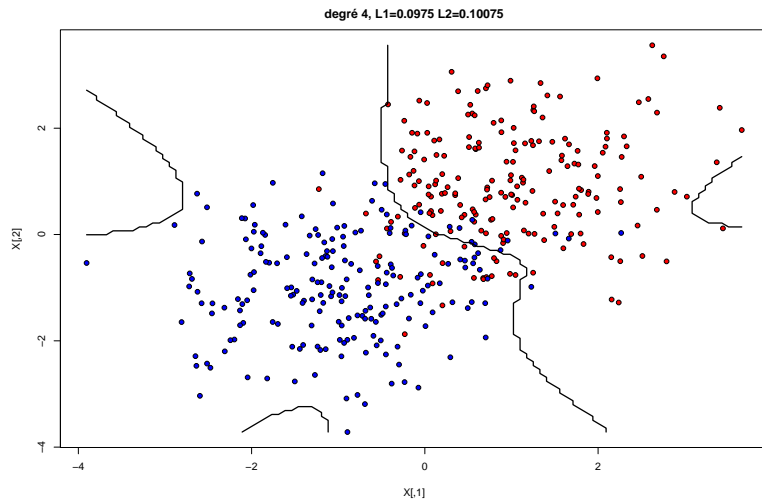
Navigation icons: back, forward, search, etc.

Exemple de surapprentissage

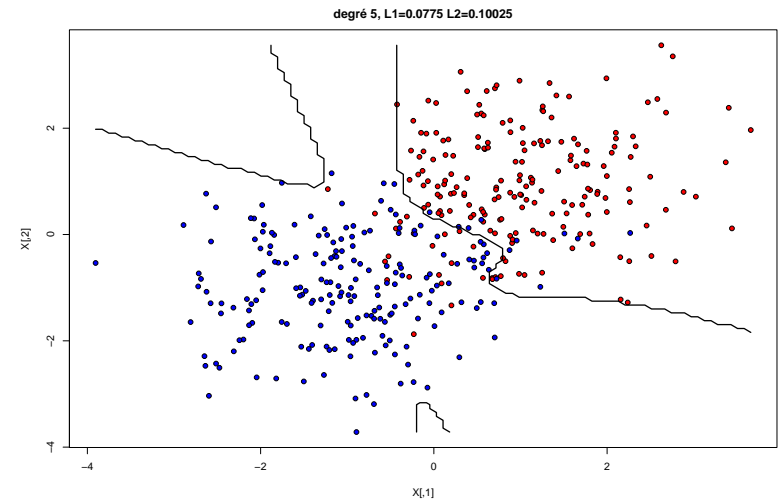


Navigation icons: back, forward, search, etc.

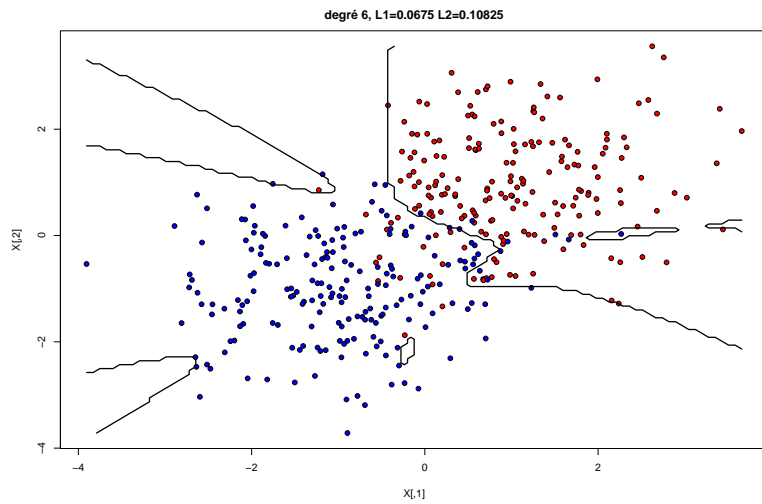
Exemple de surapprentissage



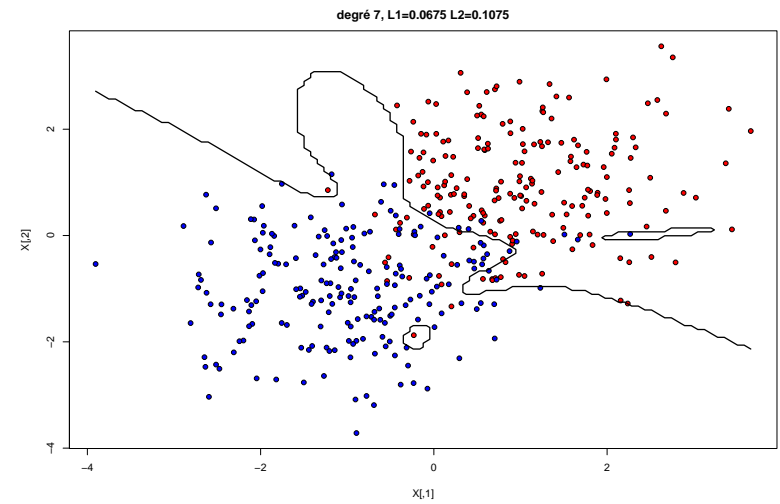
Exemple de surapprentissage



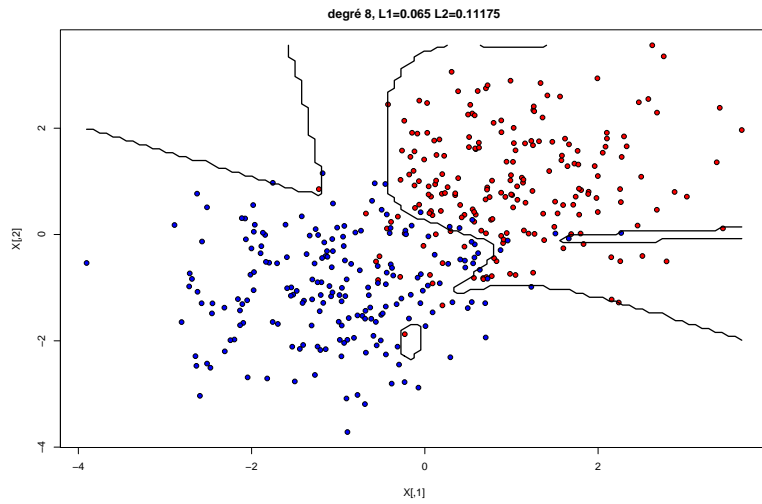
Exemple de surapprentissage



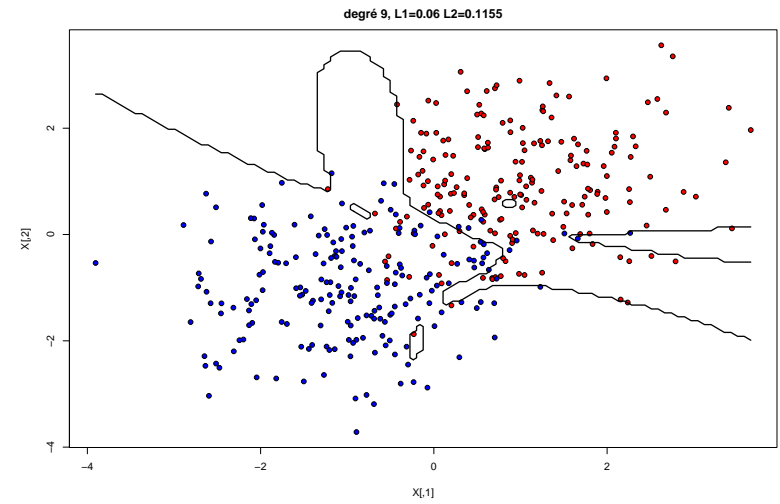
Exemple de surapprentissage



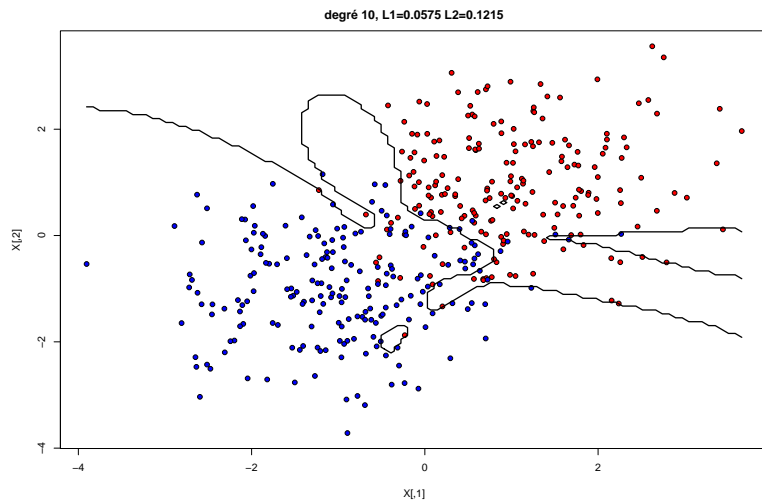
Exemple de surapprentissage



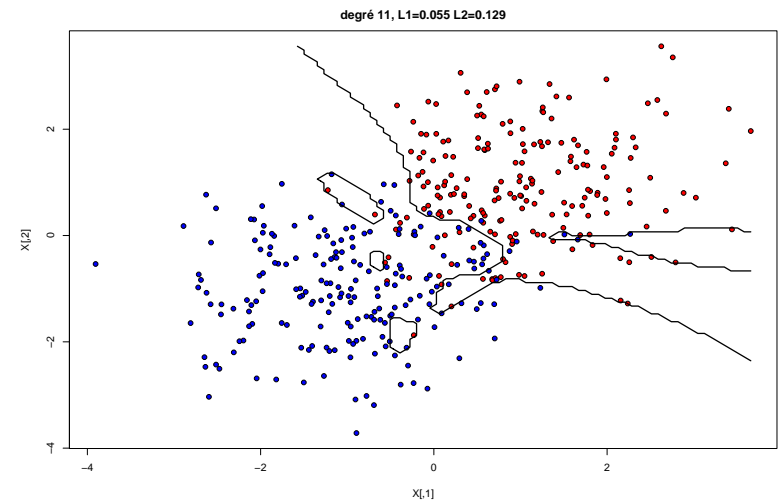
Exemple de surapprentissage



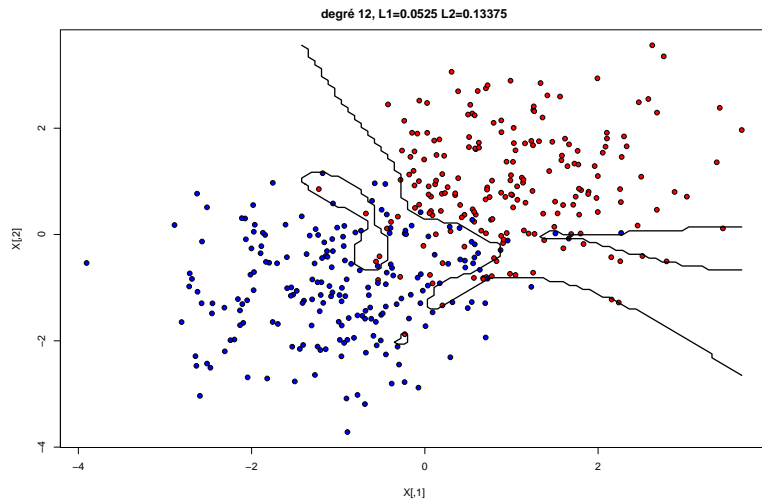
Exemple de surapprentissage



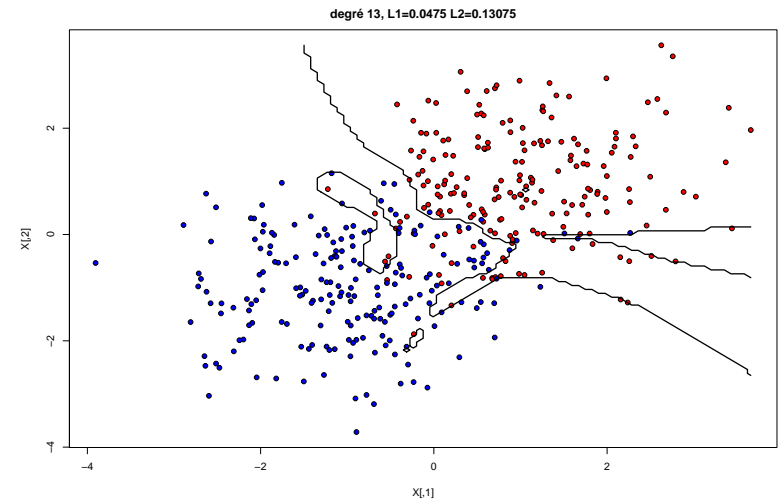
Exemple de surapprentissage



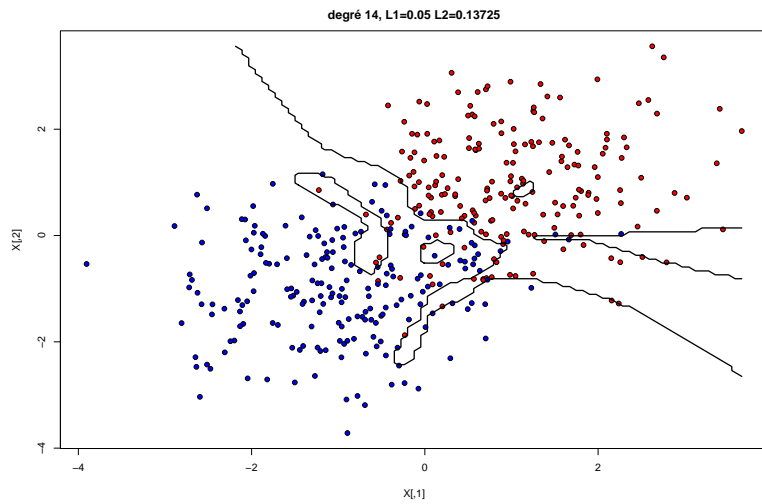
Exemple de surapprentissage



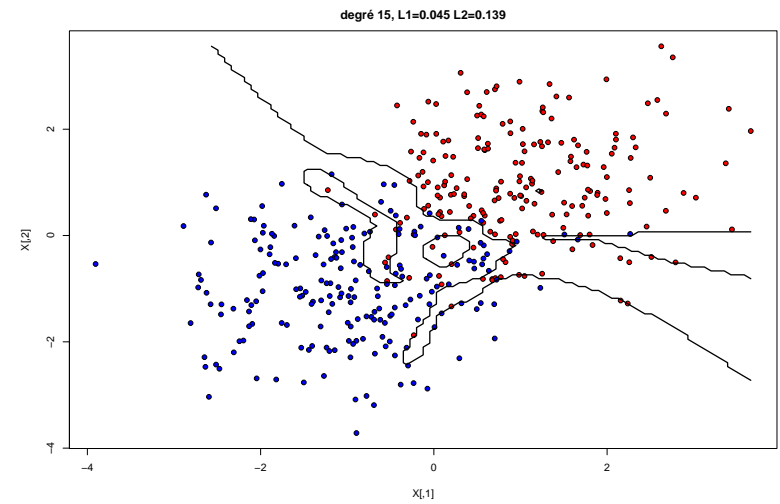
Exemple de surapprentissage



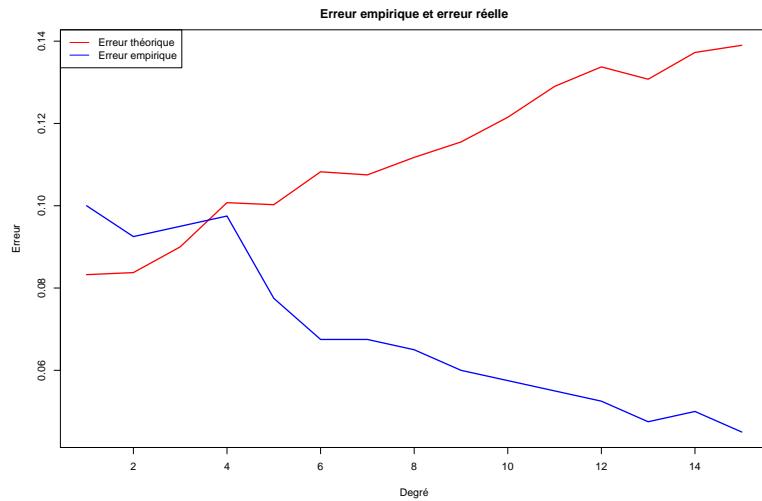
Exemple de surapprentissage



Exemple de surapprentissage



Erreur empirique et erreur réelle

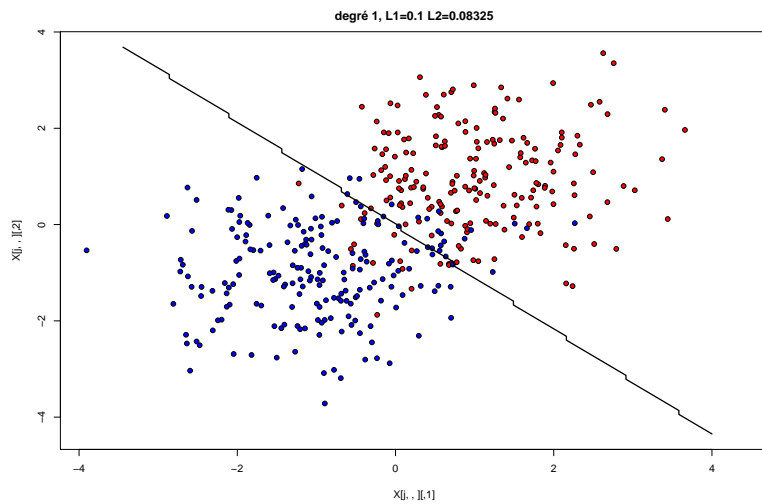


Dilemme biais/variance

- biais : ampleur de l'erreur théorique minimale
- variance : sensibilité de l'erreur empirique aux données
- modèle peu puissant (e.g., linéaire en dimension faible) :
 - biais important : $L \gg L^*$
 - variance faible : $\hat{L}(\psi, \mathcal{D}) \simeq \hat{L}(\psi, \mathcal{D}')$
- modèle puissant (e.g., polynôme de degré élevé) :
 - biais faible : $L \simeq L^* \simeq \hat{L}(\psi, \mathcal{D})$
 - variance importante : $|\hat{L}(\psi, \mathcal{D}) - \hat{L}(\psi, \mathcal{D}')|$ grand
- on cherche à minimiser L en utilisant seulement des $\hat{L}(\psi, \mathcal{D})$:
 - modèle peu puissant : résultats mauvais mais bien évalués
 - modèle puissant : résultats potentiellement bons, mais difficiles à évaluer

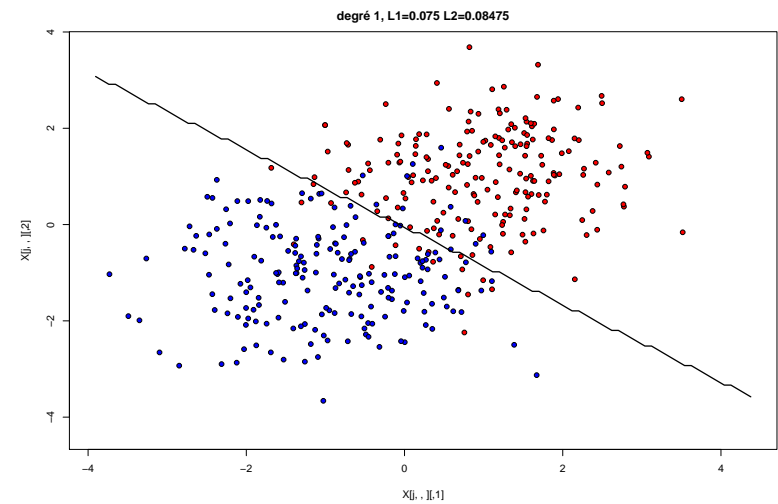
Exemple

MVS linéaire



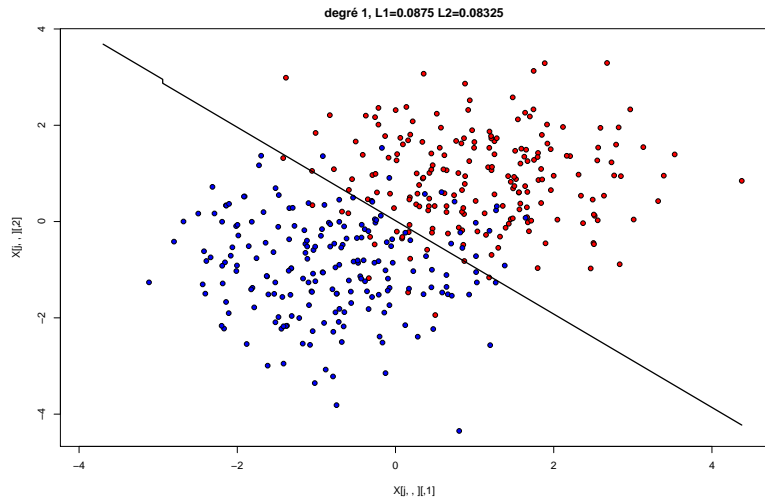
Exemple

MVS linéaire



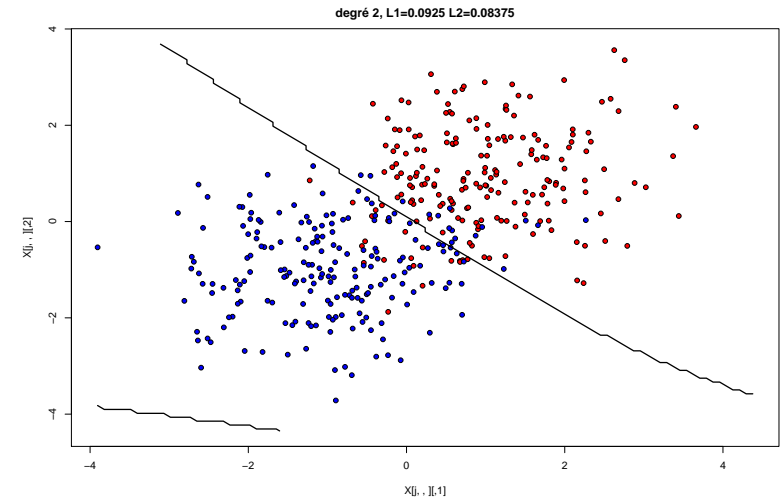
Exemple

MVS linéaire



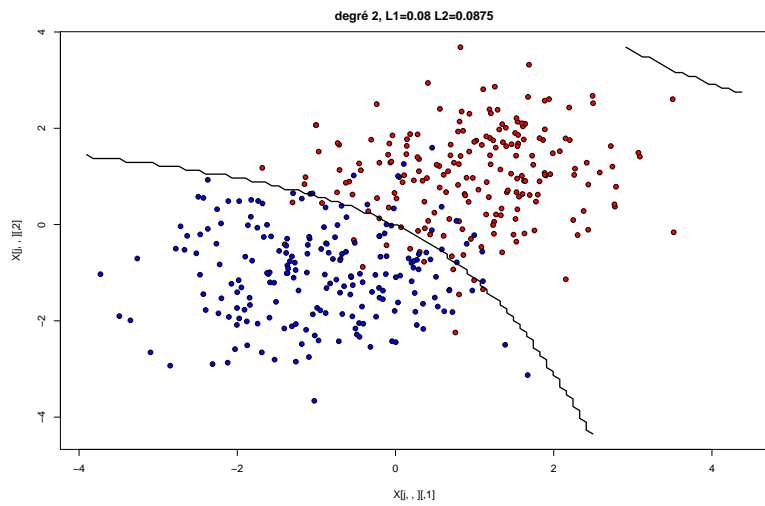
Exemple

MVS quadratique



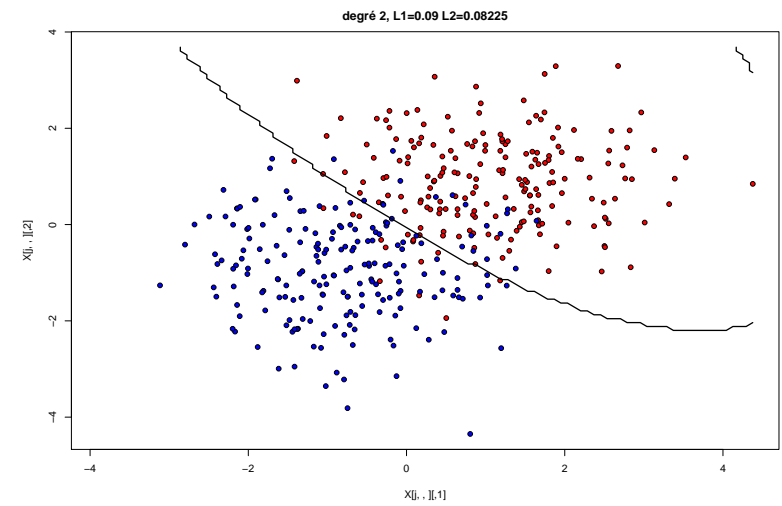
Exemple

MVS quadratique



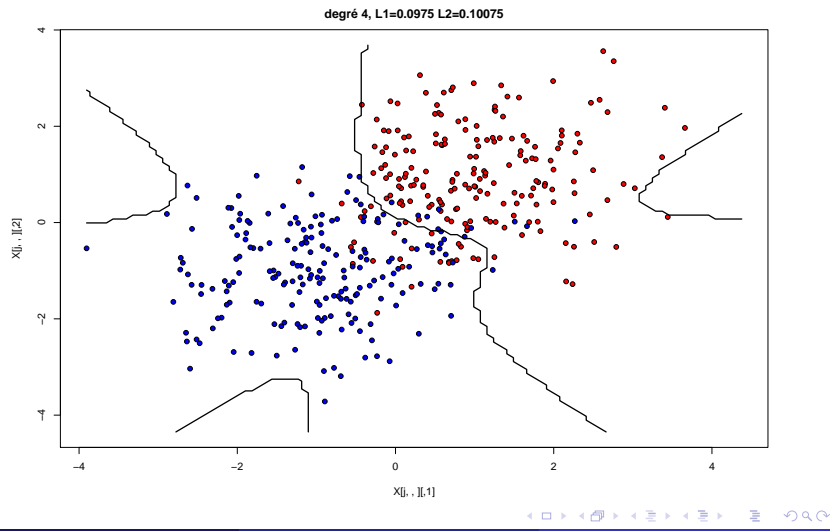
Exemple

MVS quadratique



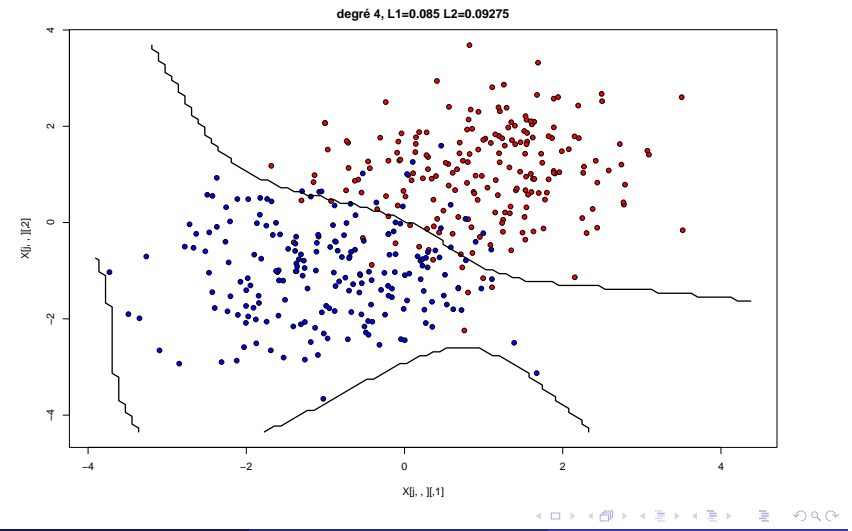
Exemple

Degré 4



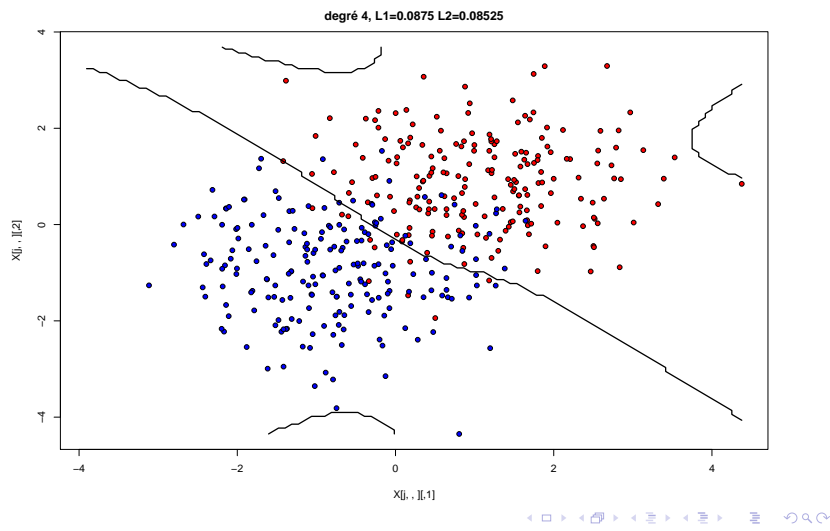
Exemple

Degré 4



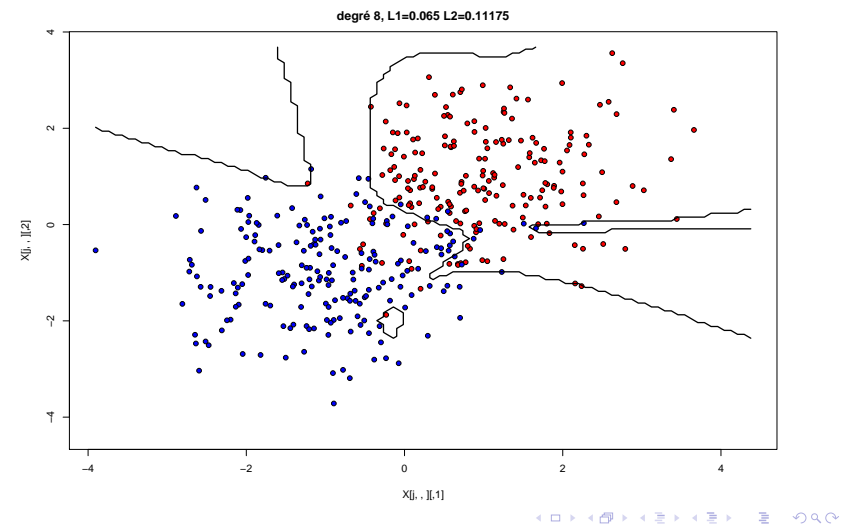
Exemple

Degré 4



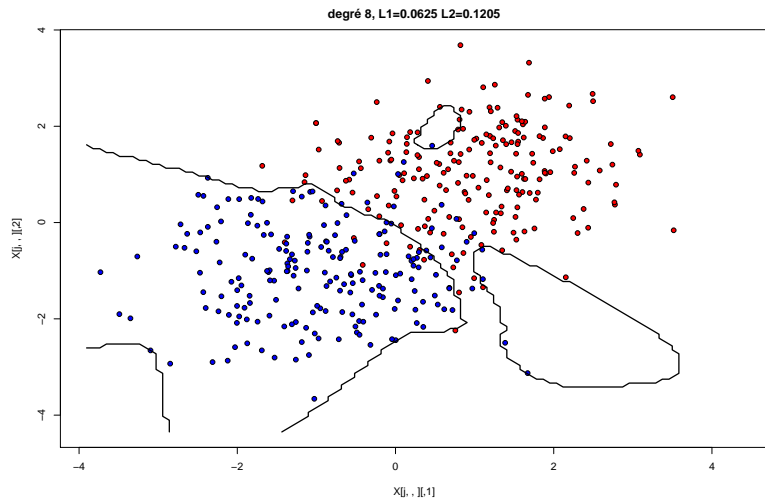
Exemple

Degré 8



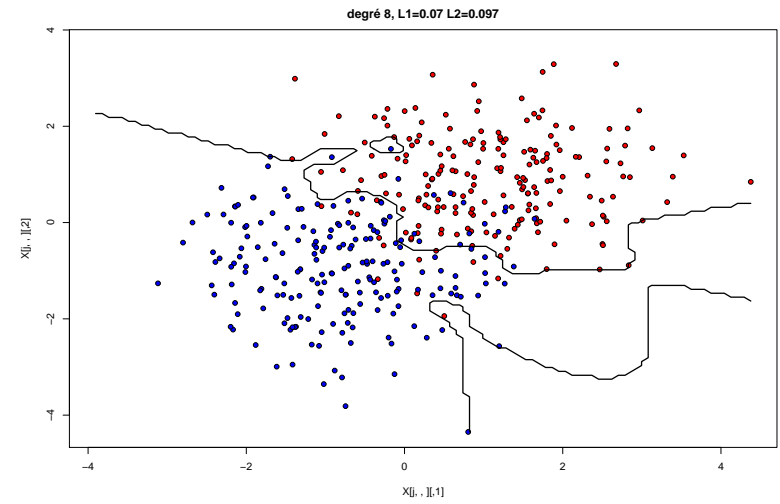
Exemple

Degré 8



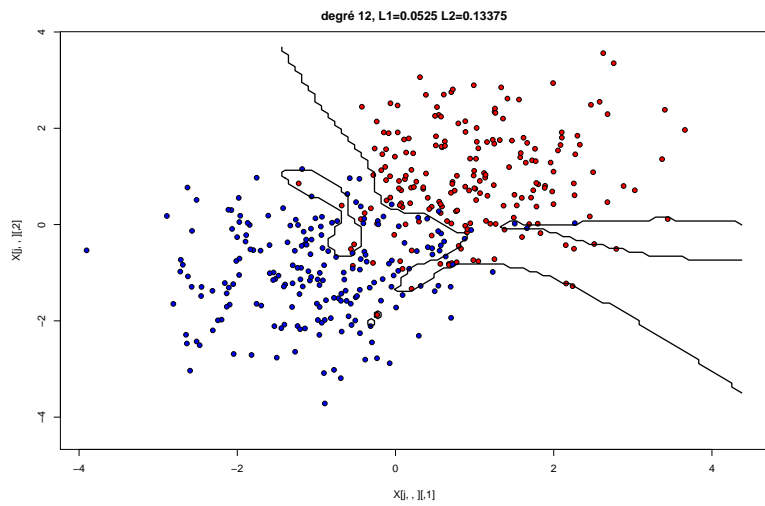
Exemple

Degré 8



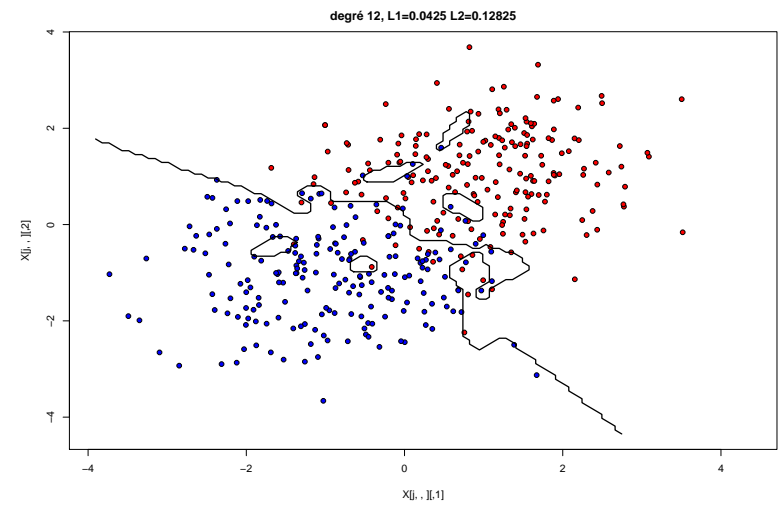
Exemple

Degré 12



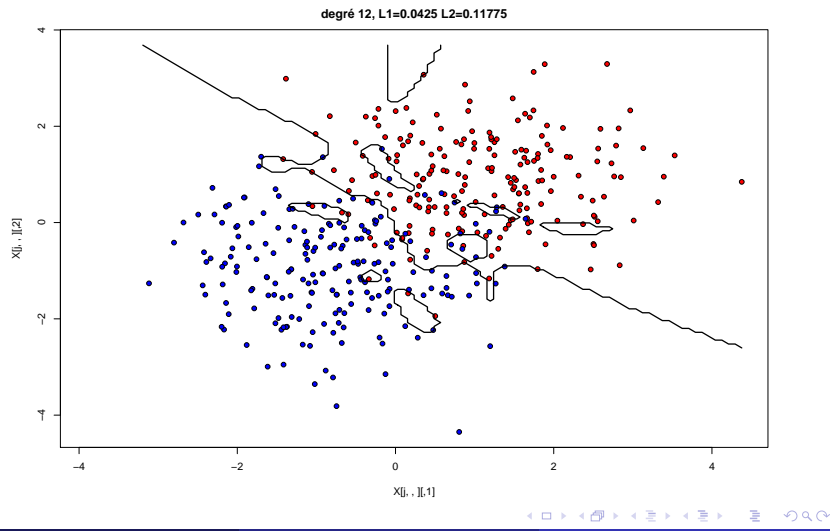
Exemple

Degré 12



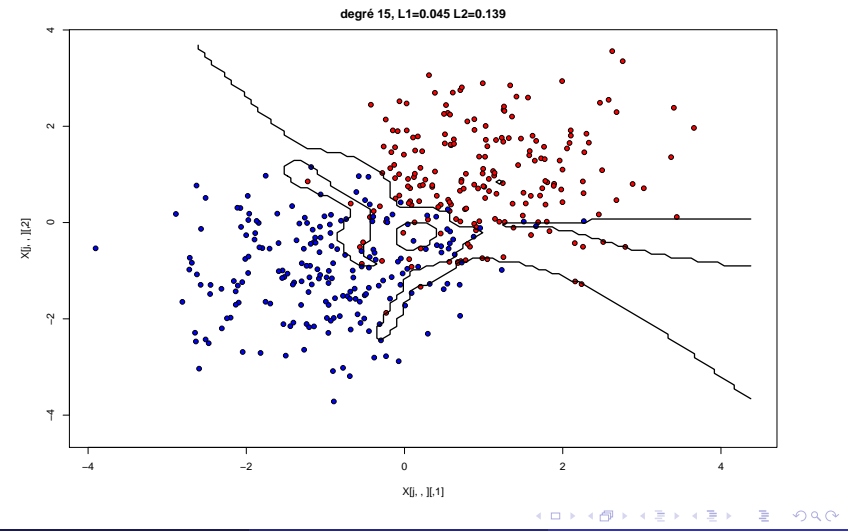
Exemple

Degré 12



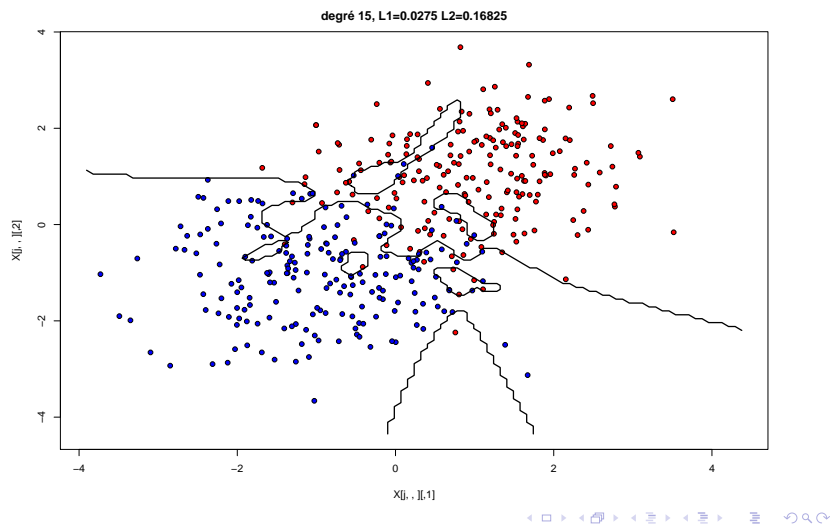
Exemple

Degré 15



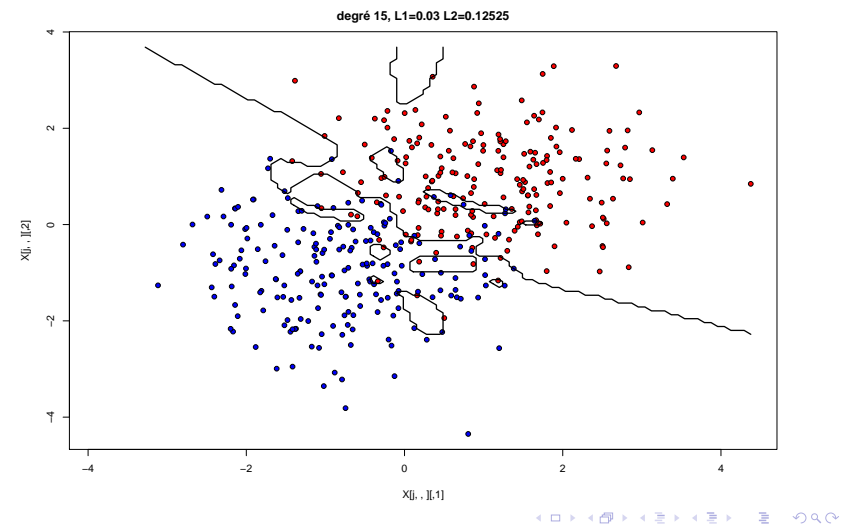
Exemple

Degré 15



Exemple

Degré 15



1 Apprentissage et généralisation

2 Méthodes de rééchantillonnage

- Validation
- Validation croisée
- Bootstrap

- problème mathématique : estimer une probabilité à partir d'une fréquence
- loi des grands nombres :

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \mathbb{I}_{\{\psi(\mathbf{x}_i) \neq y_i\}} = \mathbb{P}(\psi(X) \neq Y)$$

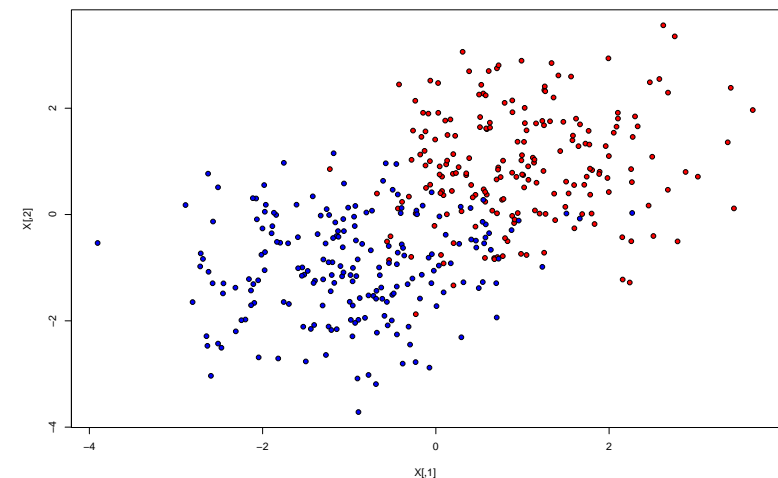
- inégalité de Hoeffding :

$$\mathbb{P}\left(\left|\widehat{L}(\psi, \mathcal{D}') - L(\psi)\right| > \epsilon\right) \leq 2^{-2|\mathcal{D}'|\epsilon^2},$$

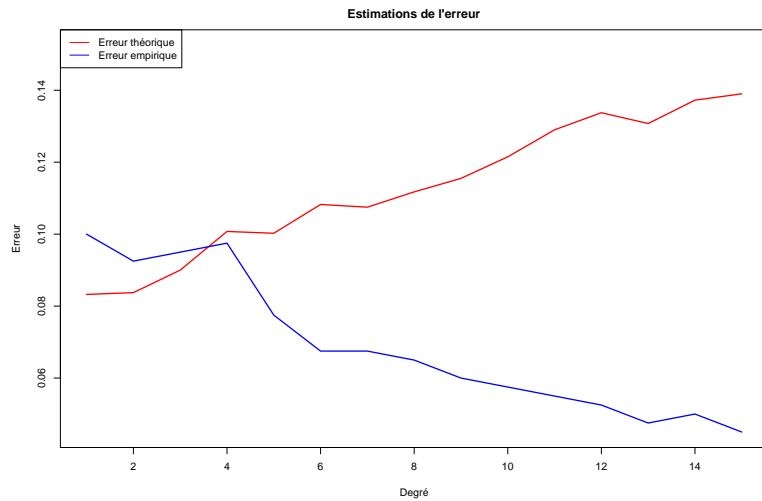
valable seulement si ψ est indépendant de \mathcal{D}'

- idée de base : construire un classifieur sur un ensemble d'**apprentissage** et l'évaluer sur un ensemble de **validation**

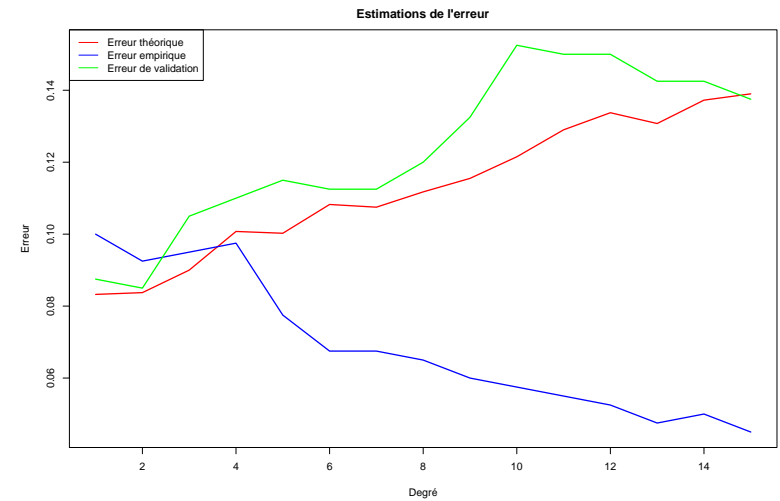
- solution la plus simple :
 - deux ensembles
 - $\mathcal{D} = (\mathbf{x}_i, y_i)_{1 \leq i \leq N} \Rightarrow$ construction du classifieur (apprentissage)
 - $\mathcal{D}' = (\mathbf{x}_i, y_i)_{N+1 \leq i \leq N+M} \Rightarrow$ évaluation du classifieur (validation)
 - en pratique : on coupe en deux (ou trois) l'ensemble des données
- applications :
 - évaluation des performances
 - sélection de modèle (choix des paramètres, par exemple) :
 - on construit deux modèles (ou plus) à partir de \mathcal{D}
 - on garde celui qui donne les meilleurs résultats sur \mathcal{D}'
 - **Attention** : applications incompatibles, car le modèle sélectionné dépend des **deux** ensembles \Rightarrow il faut un troisième ensemble pour évaluer ses performances (**ensemble de test**).



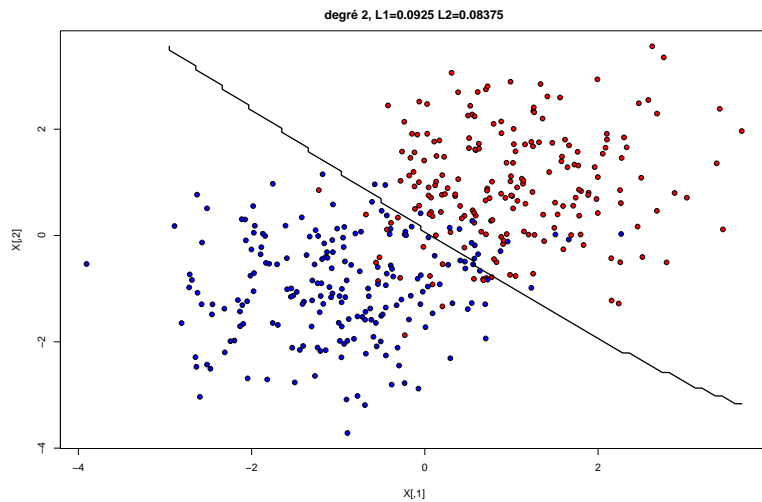
Exemple



Exemple



Exemple

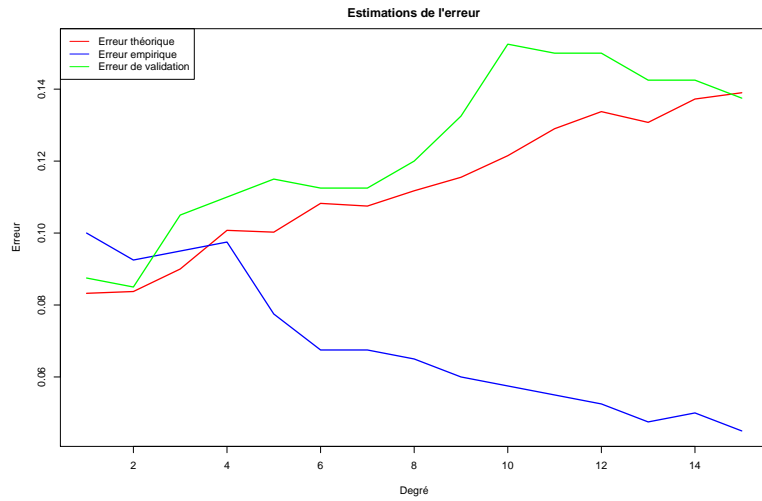


Validation

- Avantages :
 - facile à mettre en œuvre
 - temps de calcul raisonnable
- Inconvénients :
 - nécessite beaucoup de données (découpage en 2 ou 3 des données)
 - sensible au découpage
 - réduit les données utilisées pour construire le modèle : résultats moins robustes

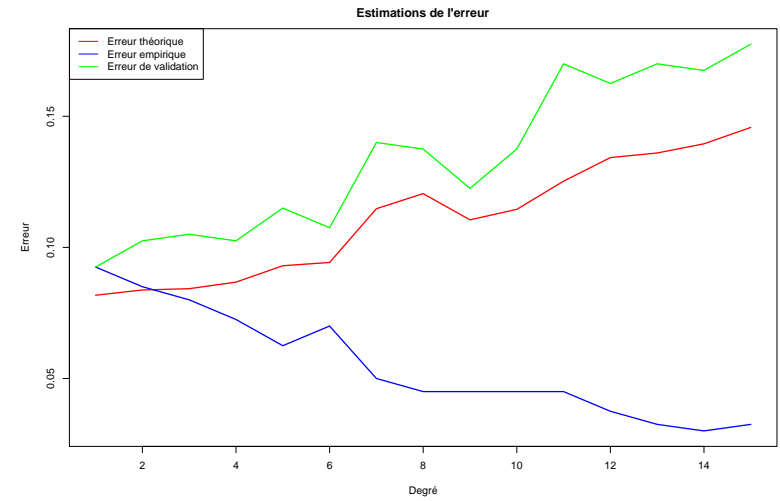
Sensibilité au découpage

Deux découpages différents



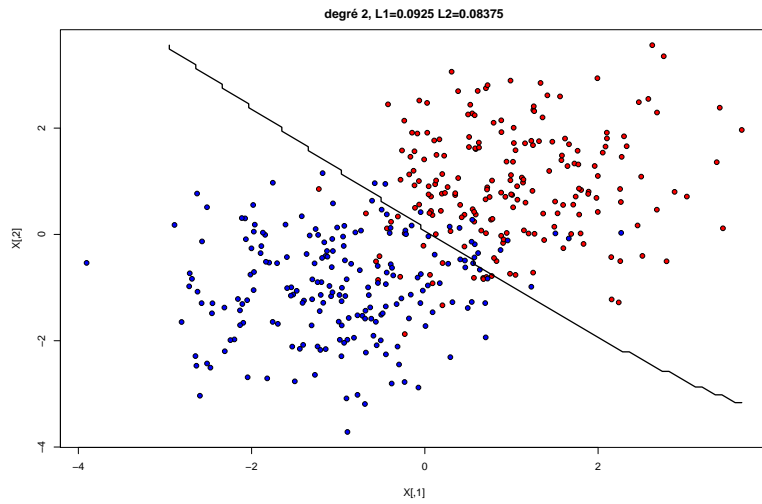
Sensibilité au découpage

Deux découpages différents



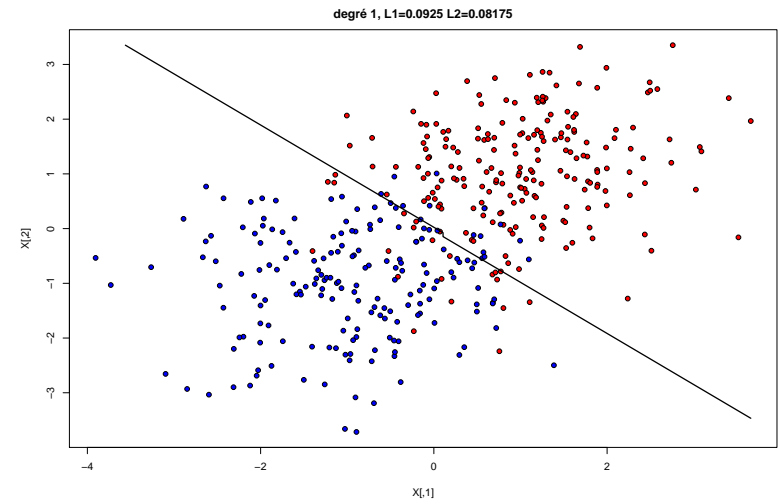
Sensibilité au découpage

Deux découpages différents

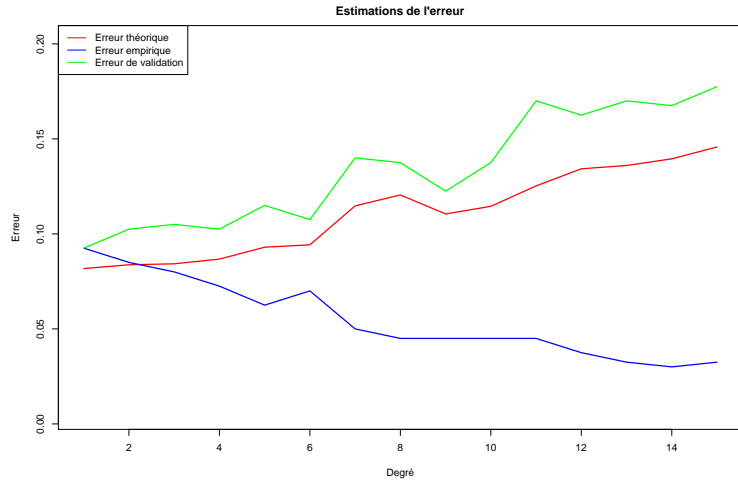


Sensibilité au découpage

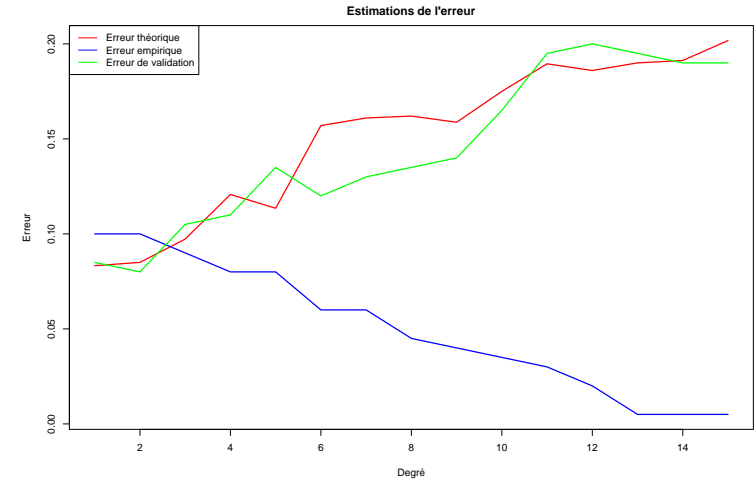
Deux découpages différents



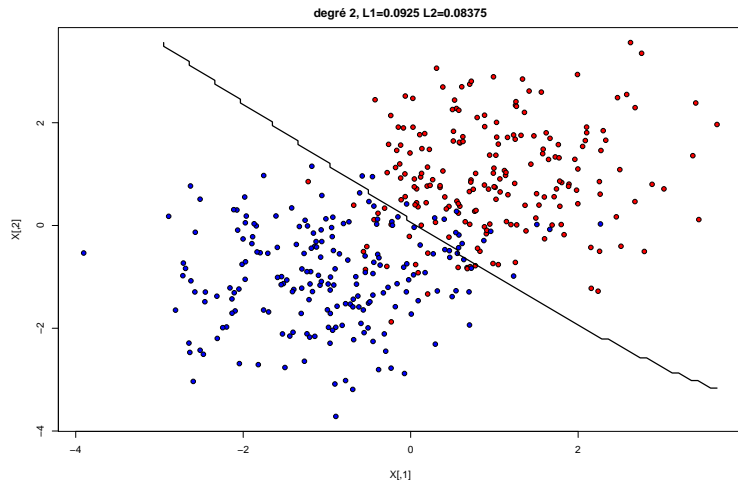
Effet de la taille



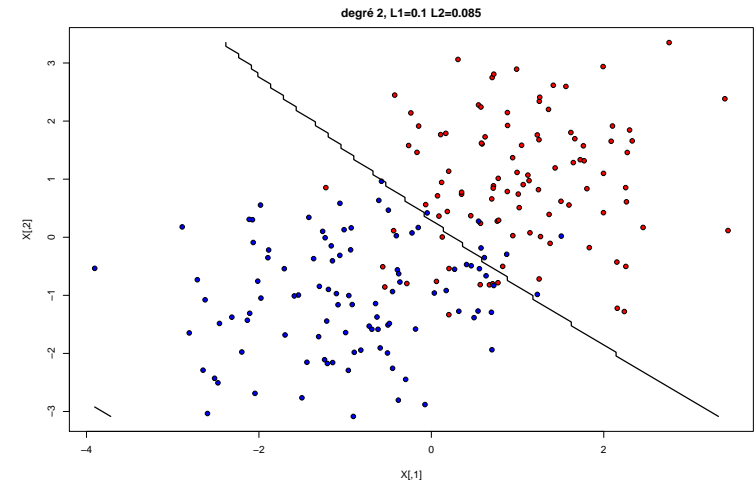
Effet de la taille



Effet de la taille



Effet de la taille

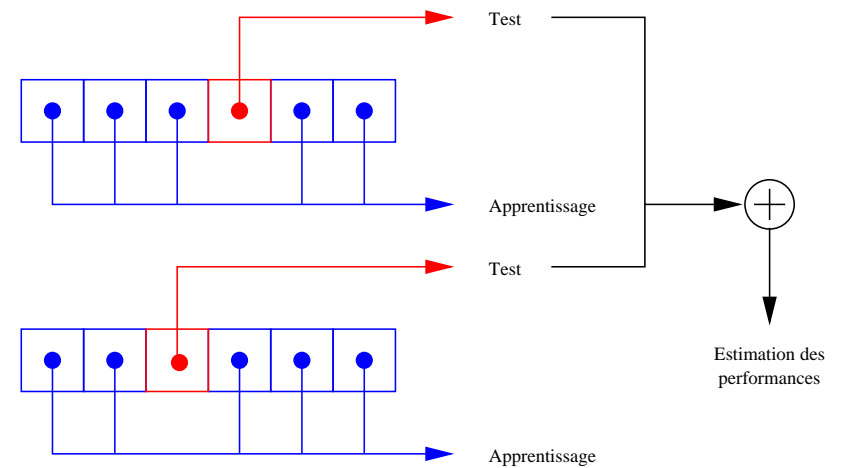


Validation croisée

- idée principale
 - échanger les ensembles d'apprentissage et de validation
 - apprendre un modèle sur $\mathcal{D} = (\mathbf{x}_i, y_i)_{1 \leq i \leq N}$ et l'évaluer sur $\mathcal{D}' = (\mathbf{x}_i, y_i)_{N+1 \leq i \leq N+M} \dots$
 - puis apprendre un modèle sur \mathcal{D}' et l'évaluer sur $\mathcal{D} \dots$
 - et enfin combiner les évaluations
- solution générale :
 - 1 découpage des données en k sous-ensembles $\mathcal{D}_1, \dots, \mathcal{D}_k$
 - 2 pour tout i :
 - 1 apprentissage sur l'union des \mathcal{D}_j avec $j \neq i$
 - 2 évaluation sur \mathcal{D}_i
 - 3 combinaison des évaluations
- si $k = N$ on parle de *leave-one-out*.



Validation croisée

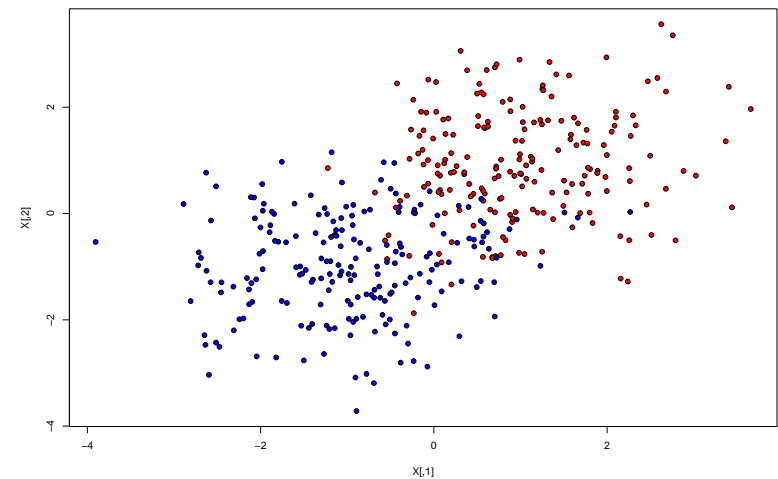


Validation croisée

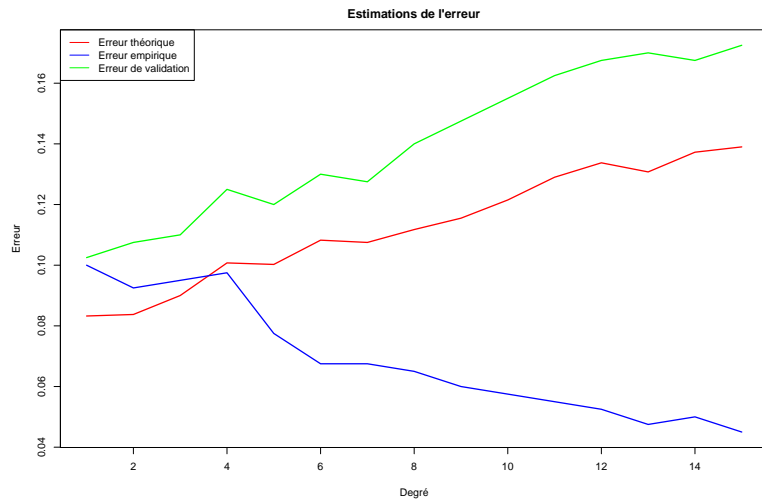
- procédure détaillée :
 - apprentissage sur $\bigcup_{j \neq i} \mathcal{D}_j \Rightarrow \psi_i$
 - prédictions sur \mathcal{D}_i , $y_i^{(i)} = \psi_i(\mathbf{x}_i)$ pour $\mathbf{x}_i \in \mathcal{D}_i$
 - donc pour tout $\mathbf{x}_i \in \mathcal{D}$, on a une prédiction $y_i^{(i)}$ (pour un certain i)
 - évaluation : $\frac{1}{N} \sum_{i=1}^N \mathbb{I}_{\{y_i^{(i)} \neq y_i\}}$
- pas de classifieur unique !
- applications :
 - évaluation de performances
 - sélection de modèle :
 - évaluation des performances pour chaque configuration choisie (degré du polynôme, etc.)
 - choix de la meilleure configuration
 - construction d'un classifieur sur l'ensemble des données



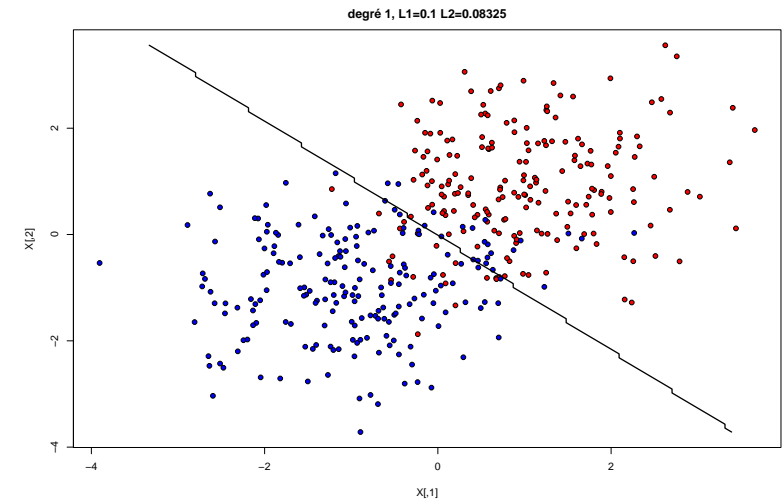
Exemple



Exemple



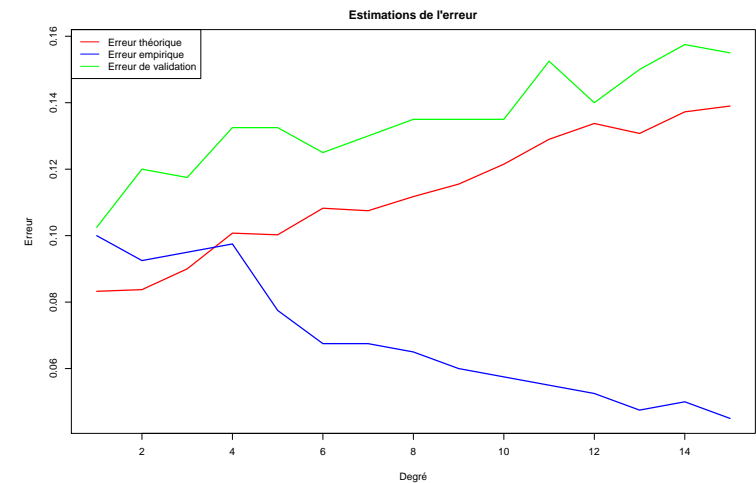
Exemple



Validation croisée

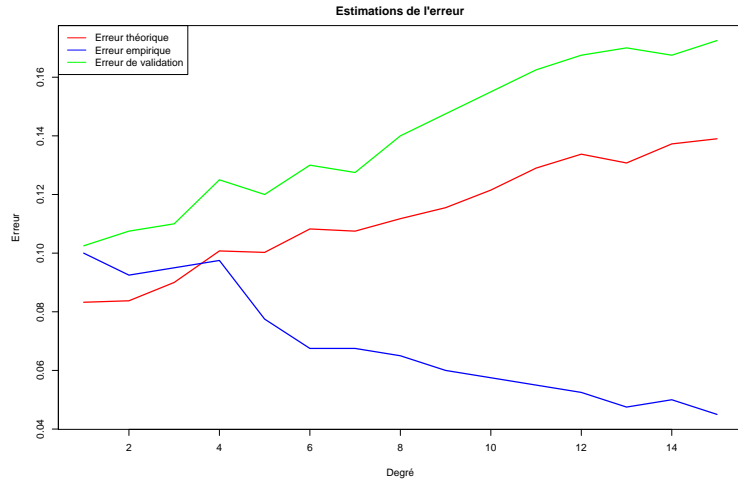
- Avantages :
 - facile à mettre en œuvre
 - utilise toutes les données pour évaluer le modèle
- Inconvénients :
 - sensible au découpage et au nombre de blocs
 - temps de calcul élevé
 - ne donne pas directement un modèle

Effet du nombre de blocs



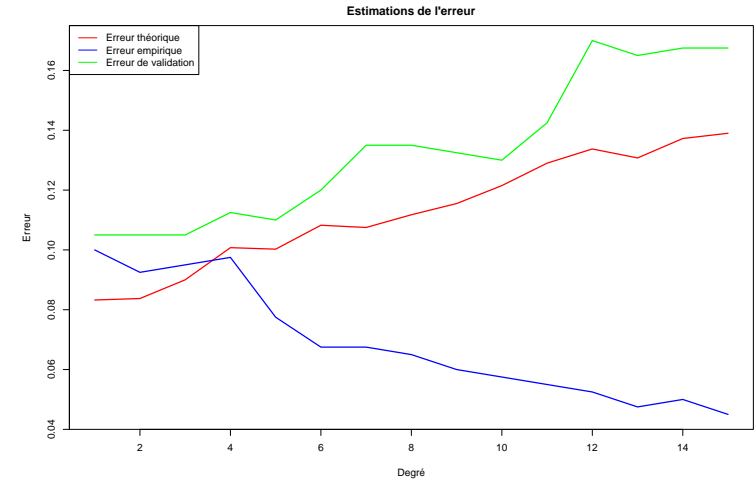
3 blocs

Effet du nombre de blocs



5 blocs

Effet du nombre de blocs



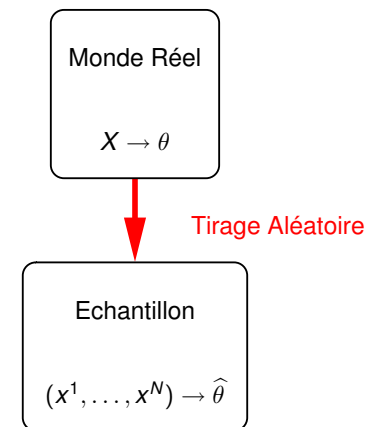
10 blocs

Le Bootstrap

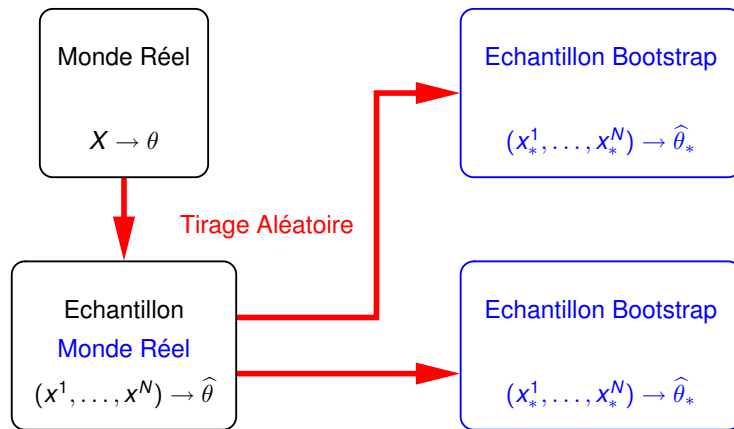
Méthode générale d'estimation de la qualité d'un estimateur, basée sur un ré-échantillonnage :

- on cherche à estimer θ , une statistique sur les observations (les x^i)
- on se donne $\hat{\theta}(x^1, \dots, x^N)$ un estimateur de θ
- on cherche à déterminer :
 - le biais de $\hat{\theta}$
 - la variance de $\hat{\theta}$
- solution :
 - fabriquer des échantillons *bootstrap*
 - un échantillon *bootstrap* : (x_*^1, \dots, x_*^N) obtenu par **tirage aléatoire uniforme avec remise** dans l'échantillon d'origine (x^1, \dots, x^N)
 - simule des nouveaux tirages pour les (x^1, \dots, x^N)

Principe



Principe



Estimation du biais

Algorithme :

- pour b allant de 1 à n
 - engendrer un échantillon bootstrap $(x_{*b}^1, \dots, x_{*b}^N)$
 - calculer $\hat{\theta}_{*b} = \hat{\theta}(x_{*b}^1, \dots, x_{*b}^N)$
- l'estimation du biais est

$$\frac{1}{n} \sum_{b=1}^n \hat{\theta}_{*b} - \hat{\theta}(x^1, \dots, x^N)$$

Idee, remplacer le monde réel par l'échantillon :

- le premier terme estime l'espérance de l'estimateur
- le second terme estime l'estimateur

Estimation de la variance

Algorithme :

- pour b allant de 1 à n
 - engendrer un échantillon bootstrap $(x_{*b}^1, \dots, x_{*b}^N)$
 - calculer $\hat{\theta}_{*b} = \hat{\theta}(x_{*b}^1, \dots, x_{*b}^N)$
- calculer

$$\hat{\theta}_* = \frac{1}{b} \sum_{b=1}^n \hat{\theta}_{*b}$$

- l'estimation de la variance est

$$\frac{1}{n-1} \sum_{b=1}^n (\hat{\theta}_{*b} - \hat{\theta}_*)^2$$

Application à l'évaluation d'un modèle

Raisonnement :

- l'évaluation d'un modèle consiste à estimer ses performances
- l'erreur résiduelle sur l'ensemble d'apprentissage sous-estime l'erreur réelle
- idée, estimer l'ampleur de la sous-estimation par *bootstrap* :
 - calculer la sous-estimation pour un échantillon *bootstrap*
 - moyenner les sous-estimations pour beaucoup d'échantillons *bootstrap*
 - corriger l'erreur résiduelle en ajoutant la moyenne

Algorithme :

- 1 pour b allant de 1 à n
 - 1 engendrer un échantillon bootstrap $(x_{*b}^1, \dots, x_{*b}^N)$ (à partir de l'ensemble d'apprentissage)
 - 2 estimer le modèle optimal pour l'échantillon **bootstrap**
 - 3 calculer $\hat{\mathcal{B}}_{*b}$ comme la différence entre l'erreur résiduelle du modèle sur l'échantillon d'**apprentissage** et l'erreur résiduelle du modèle sur l'échantillon **bootstrap**
- 2 estimer l'erreur résiduelle $\hat{\mathcal{E}}$ du modèle optimal sur l'ensemble d'apprentissage
- 3 corriger cette erreur en lui ajoutant $\frac{1}{n} \sum_{b=1}^n \hat{\mathcal{B}}_{*b}$

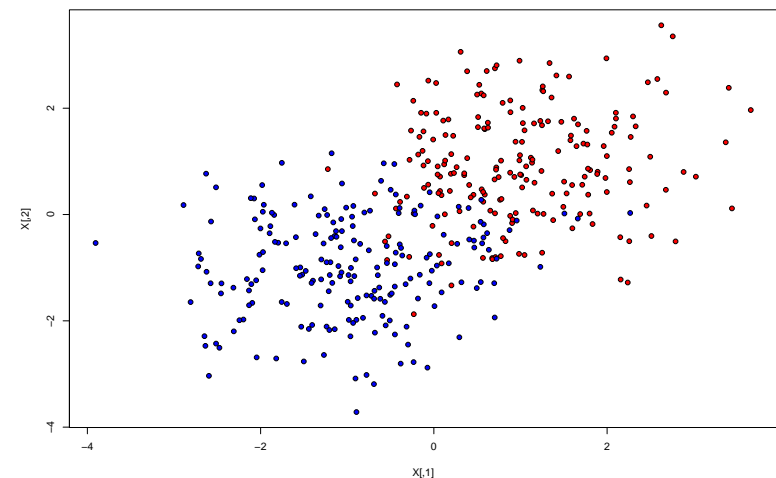
Estimation **directe** de l'erreur du modèle optimal

- moyenne empirique de l'erreur commise sur l'ensemble d'apprentissage par le modèle construit sur l'échantillon *bootstrap* ($\hat{\mathcal{E}}_B$)
- moyenne empirique de l'erreur commise sur le complémentaire de l'échantillon *bootstrap* par le modèle construit sur l'échantillon (*bootstrap out-of-bag*, $\hat{\mathcal{E}}_{oob}$)
- *bootstrap 632* : combinaison de l'estimation *out-of-bag* et de l'estimation naïve (sur l'ensemble d'apprentissage)

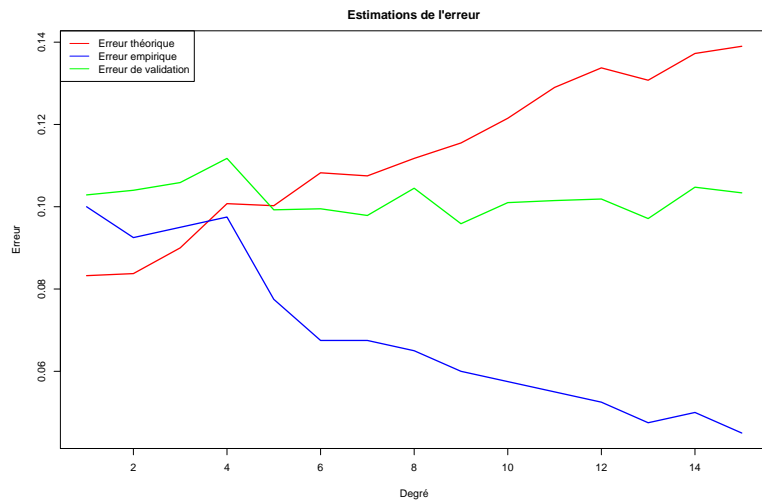
$$\hat{\mathcal{E}}_{632} = 0.632 \hat{\mathcal{E}}_{oob} + 0.368 \hat{\mathcal{E}}$$

Probabilité qu'une observation de l'ensemble d'apprentissage soit dans un échantillon *bootstrap* : 0.632

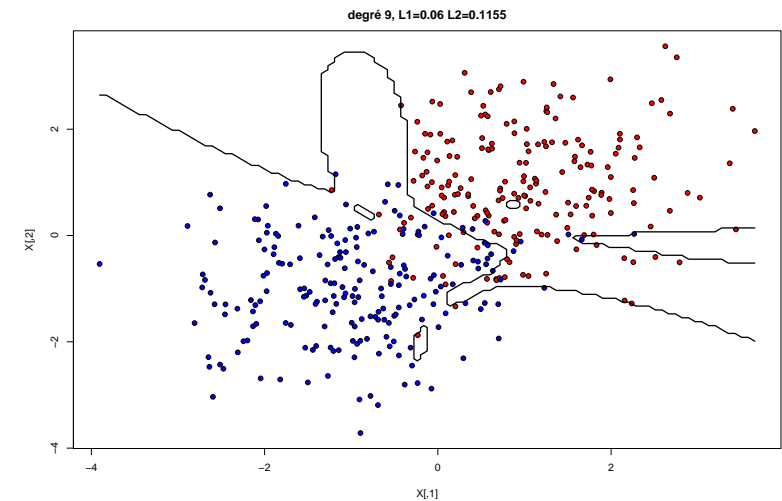
- Points positifs :
 - facile à mettre en œuvre
 - utilise toutes les données
 - donne des intervalles de confiance
- Points négatifs :
 - temps de calcul très élevé
 - nombreuses variantes
 - ne donne pas directement un modèle



Exemple



Exemple

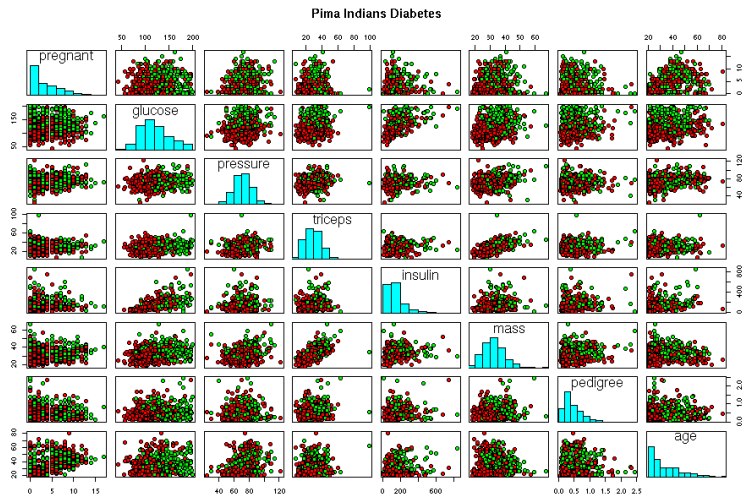


Résumé

- l'erreur empirique ne donne pas une bonne idée des performances en généralisation
- il faut **toujours** utiliser une méthode valide pour estimer les performances
- découpage et rééchantillonnage :
 - méthodes classiques et éprouvées
 - rééchantillonnage (validation croisée et *bootstrap*) : lent mais utilise toutes les données
 - validation (découpage) : rapide mais nécessite beaucoup de données

Exemple réaliste

- diagnostic du diabète chez des femmes d'ascendance Pima (tribu indienne)
- deux classes : diabétique ou non
- huit variables : nombre de grossesses, âge, indice de masse corporelle, etc.
- 768 sujets \Rightarrow non linéairement séparable !
- 500 sujets sains, 268 diabétiques
- beaucoup de données manquantes (valeurs inconnues pour les variables)
- cf <http://www.ics.uci.edu/~mllearn/MLSummary.html>



- données manquantes :
 - solution basique
 - donnée manquante remplacée par la moyenne de la variable
- évaluation des performances :
 - validation croisée *externe*
 - 4 morceaux : 576 apprentissage, 192 validation
 - proportions respectées dans les morceaux (125/67)
- pour chaque bloc :
 - sélection de modèle (σ et C pour la MVS)
 - validation croisée *interne* à 5 blocs équilibrés
- équilibrage des classes :
 - pondération
 - poids de la classe i : $\frac{N}{2|C_i|}$

- temps de calcul : quelques minutes pour 25 couples (σ, C)
- matrice de confusion :

$$\begin{pmatrix} 373 & 80 \\ 127 & 188 \end{pmatrix}$$

- environ 27% d'erreur
- amélioration de la procédure :
 - test de plus de couples pour (σ, C)
 - évaluation des performances par découpage aléatoire :
 - remplace la validation croisée *externe*
 - on découpe l'ensemble des données en deux blocs, apprentissage et test
 - on recommence autant de fois que possible l'opération