Université de Rennes

ENSAI École nationale de la statistique et de l'analyse de l'information

Chapitre 2. Apprentissage supervisé

Claude Petit, Insee et université de Rennes - claude.petit@univ-rennes.fr

Oct. 2026

Overview

1. Théorie de la décision statistique

2. Minimisation du risque empirique

1. Théorie de la décision statistique

Prédicteur de Bayes en classification binaire 1

- $\mathcal{D}_n = \{Z_1, ..., Z_n\}$ n-échantillon de v.a.i.i.d. $Z_i = (X_i, Y_i)$.
- X_i observations issues d'une v.a. X : données, variables explicatives.
- Y_i issues d'une v.a. Y, catégories des X_i : étiquettes ou labels.
- $X \in \mathbb{X}$, $Y \in \mathbb{Y}$.
- $\mathbb{P}$ proba sur $\mathcal{E} = \mathbb{X} \times \mathbb{Y}$: loi (inconnue) de (X, Y) et des (X_i, Y_i).

Objectif de l'apprentissage supervisé : déterminer Y sachant X, à partir des seules observations de $Z_1, ...Z_n$.

- $g \in \mathcal{F} = \mathcal{F}(\mathbb{X}, \mathbb{Y})$ fonction de prédiction : $g(x) = y$.
- $l : \mathbb{Y}^2 \longrightarrow \mathbb{R}_+$ fonction de perte pour mesurer la qualité de g.
- $R(g) = R_{\mathbb{P}}(g) = \mathbb{E}\left[l(Y, g(X))\right]$ risque de g : valeur moyenne de la perte sur toutes les réalisations possibles.

Prédicteur de Bayes en classification binaire 2

Problème de classification si $\mathbb{Y}$ fini, de régression si $\mathbb{Y} = \mathbb{R}$.

En classification binaire, $Y = 0$ ou 1. La fonction de perte associée est

$$l(Y, Y') = \mathbb{1}_{[Y \neq Y']} \tag{1}$$

On note

$$\eta(x) = \mathbb{P}[Y = 1 | X = x] = \mathbb{E}[Y | X = x] \tag{2}$$

et

$$g^\star(x) = \mathbb{1}_{[\eta(x) > 1/2]} \tag{3}$$

Alors $g^\star$ est le classifieur optimal de Bayes. Il minimise l'erreur de classification binaire (démonstration en exercice), mais c'est une fonction oracle inconnue.

Prédicteur optimal en régression

Toute fonction $g^\star$, si elle existe, minimisant $R(g)$ est appelée oracle.

Minimiseur du risque de régression

Si pour tout $x \in \mathbb{X}$ la borne inférieure sur $y \in \mathbb{Y}$ de $\mathbb{E}[l(Y, y)|X = x]$ est atteinte, alors toute fonction $g^\star$ la minimisant est fonction oracle:

$$\forall x \in \mathbb{X}, g^\star \in \arg\min_{y \in \mathbb{Y}} \mathbb{E}[l(Y, y)|X = x] \Rightarrow g^\star \in \arg\min_{g \in \mathcal{F}} R(g) \quad (4)$$

Fonction oracle pour la régression

En régression au sens des moindres carrées, la fonction oracle est

$$\eta^\star(x) = \mathbb{E}[Y|X = x] \quad (5)$$

et vérifie

$$\forall \eta : \mathbb{X} \longrightarrow \mathbb{R}, \; R(\eta) = R(\eta^\star) + \mathbb{E}\left[(\eta(X) - \eta^\star(X))^2\right] \quad (6)$$

Algorithme d'apprentissage

Algorithme d'apprentissage : fonction

$$g : \bigcup_{n=1}^{+\infty} (\mathbb{X} \times \mathbb{Y})^n \longrightarrow \mathcal{F}(\mathbb{X}, \mathbb{Y}) \tag{7}$$

qui à $\mathcal{D}_n \longrightarrow g_n$. Estimateur de la meilleure fonction de prédiction. $g_n(x) = g_n(x, \mathcal{D}_n)$ dépend de l'échantillon !

$$g_n = g_n(., \mathcal{D}_n) \in \mathcal{F}(\mathbb{X}, \mathbb{Y}) \tag{8}$$

$\mathcal{D}_n$ fonction des $(X_i, Y_i) \Rightarrow$ aléatoire, $g_n(x) = g_n(x, \mathcal{D}_n)$ également aléatoire, tout comme :

$$R_{\mathbb{P}}(g_n) = \mathbb{E}[l(Y, g_n(X))] = \int_{\mathbb{X}\times\mathbb{Y}} l(y, g_n(x))\mathbb{P}(dx, dy) \tag{9}$$

Consistance d'un algorithme d'apprentissage

Algorithme consistant par rapport à $\mathbb{P}$, si :

$$\lim_{n \to +\infty} \mathbb{E}\left[R\left(g_n\right)\right] = R\left[g^\star\right] \tag{10}$$

car $R(g_n)$ est une variable aléatoire.

g_n est doublement aléatoire : au travers de la loi de (X, Y), et au travers de $\mathcal{D}_n$ qui intervient dans sa construction.

Algorithme universellement consistant si consistant par rapport à toute loi de probabilité $\mathbb{P}$ sur l'ensemble $\mathcal{P}$ des mesures de proba.
Universellement et uniformément consistant si uniformément consistant par rapport à toute loi $\mathbb{P}$:

$$\sup_{\mathbb{P} \in \mathcal{P}} \lim_{n \to +\infty} \left(\mathbb{E}\left[R\left(g_n\right)\right] - R\left[g^\star\right]\right) = 0 \tag{11}$$

« No Free Lunch » Theorem

NFL Theorem, David Wolpert 1997

Si $\text{card}(\mathbb{X}) = +\infty$, il n'existe pas d'algorithme d'apprentissage uniformément et universellement consistant.

Objectif du ML : construire un algorithme ayant consistance universelle sur une classe de proba. pertinente pour le problème et une famille de fonctions de prédiction assez grande.

$\mathcal{P}$ et $\mathcal{D}_n$ étant donnés, on cherche un algo. d'apprentissage g_n tel que

$$\lim_{n \to +\infty} \sup_{\mathbb{P} \in \mathcal{P}} \left(\mathbb{E}\left[R\left(g_n \right) \right] - R\left[g^\star \right] \right) = 0, \;\; g_n, g^\star \in \mathcal{G} \tag{12}$$

Doit décroître vite vers 0 pour que peu de données soient nécessaires à l'algo. pour bien prédire. $\mathcal{P}$ modélise notre *a priori* et entraîne un *a priori* sur la fonction cible.

Risque empirique

$R_{\mathbb{P}}[l(Y, g(X)]$ inconnu. L'algo d'apprentissage doit trouver g de risque le plus petit possible. On estime R par l'estimateur plug-in :

$$R_n(g) = \frac{1}{n} \sum_{i=1}^{n} l(Y_i, g(X_i)) \tag{13}$$

R_n défini pour une famille de fonctions de prédiction (sous-ensemble $\mathcal{G}$ de $\mathcal{F}$). La LFGN et le TLC donnent :

$$\lim_{n \to +\infty} R_n(g) = R(g) \text{ p.s.} \tag{14}$$

$$\sqrt{n}\,(R_n(g) - R(g)) \rightsquigarrow \mathcal{N}(0, \sigma^2) \tag{15}$$

si $\sigma^2 = \mathbb{V}\,(l(Y, g(X))) < \infty$

$\Rightarrow$ $R_n(g)$ bonne approximation du de $R(g)$ si n suffisamment grand.

2. Minimisation du risque empirique

Minimisation du risque empirique 1 : capacité du modèle

$\mathcal{G}$ étant fixé, on choisit comme fonction de prédiction un minimiseur $\widehat{g}_{n,\mathcal{G}}$ du risque empirique défini par:

$$\widehat{g}_{n,\mathcal{G}} = \widehat{g}_n(\mathcal{D}_n, \mathcal{G}) = \arg\min_{g \in \mathcal{G}} R_n(g) \tag{16}$$

où $\mathcal{G} \subset \mathcal{F}$ est un sous-ensemble de toutes les fonctions de $\mathbb{X}$ dans $\mathbb{Y}$.
C'est la classe de fonctions à laquelle on se restreint pour déterminer g_n.

$\mathcal{G} = \mathcal{F} \Rightarrow$ mauvaise idée :

- Souvent infinité de fonctions minimisantes.
- Très loin d'être universellement consistant.
- Sur-apprentissage assuré.

Taille de $\mathcal{G}$ = capacité ou complexité du modèle.

On doit prendre $\mathcal{G}$ assez grand pour bien approcher toute fonction, mais pas trop pour éviter le sur-apprentissage.

Minimisation du risque empirique 2 : surapprentissage

Soit $M = |\mathcal{G}|$ la complexité du modèle.

Quand M augmente, $R_n(g)$ diminue tandis que - souvent - $R(g)$ diminue d'abord, puis augmente à nouveau avec M (ce n'est pas toujours vrai).

À n fixé, si $\mathcal{G}$ est suffisamment riche, on peut toujours trouver un prédicteur $g_{n,\mathcal{G}}$ avec $R_n(g_{n,\mathcal{G}})$ très faible, même si $\mathcal{D}_n$ très grand, mais dont le risque moyen de prédiction $R(g)$ est grand.

Ex : g défini par $g(x_i) = y_i$ (et n'importe quelle valeur sinon). Apprend par coeur la base d'apprentissage. $R_n(g) = 0$, mais risque de prédiction très grand : sa capacité de généralisation est faible. On retient que :

$$R_n(\widehat{g}_n(\mathcal{D}_n, \mathcal{G}_M)) \geq R_n(\widehat{g}_n(\mathcal{D}_n, \mathcal{G}_{M'})) \text{ si } M < M' \tag{17}$$

Si $\mathcal{G}_M \subset \mathcal{G}_{M'}$.

Minimisation du risque empirique 3 : démarche

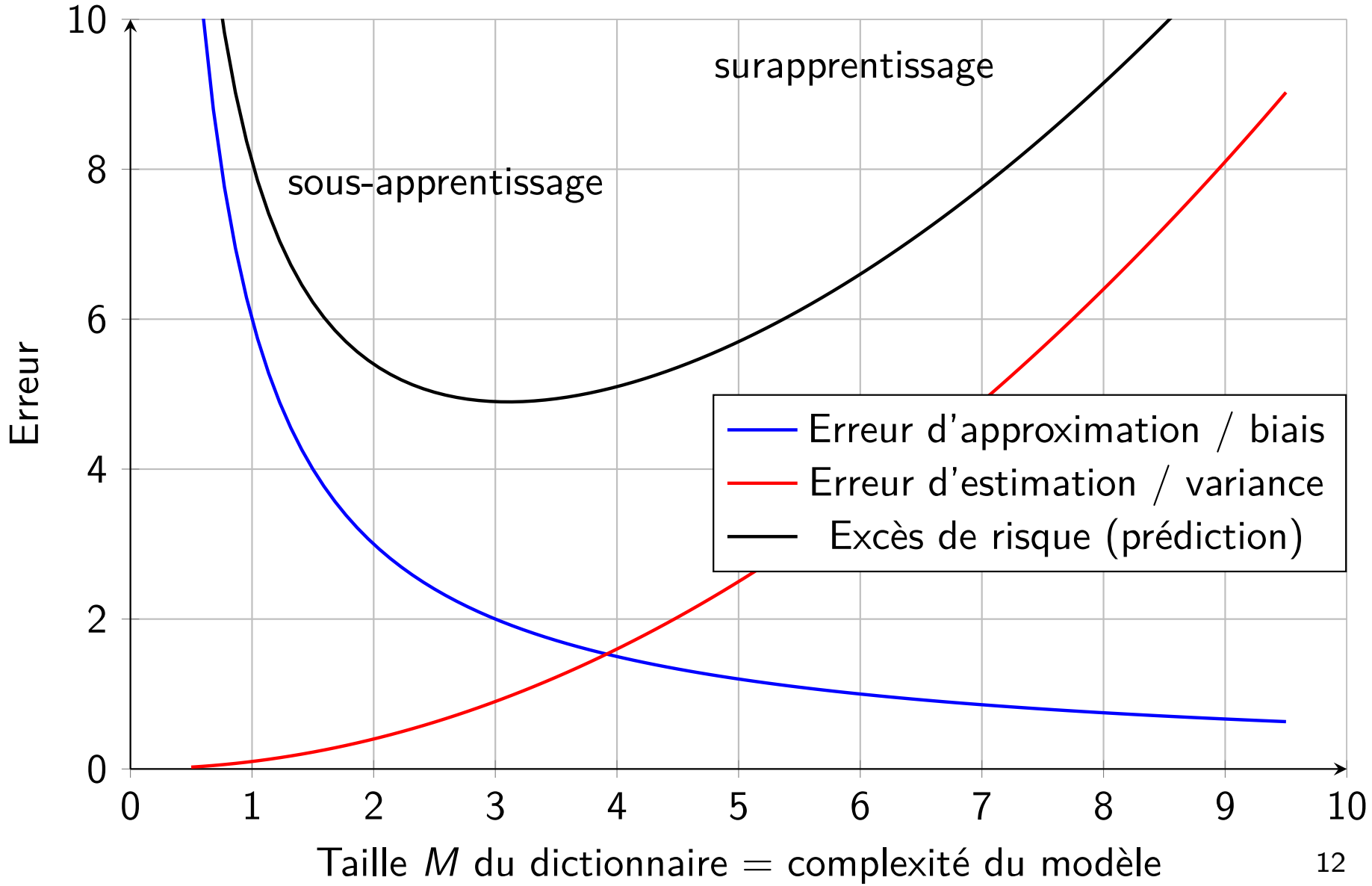
Pour approcher $R(g^\star)$, on utilise le risque empirique R_n, mais limité à une fonction $g \in \mathcal{G}$. Dans cet excès de risque, on fait donc deux erreurs :

- stochastique (ou d'estimation ou variance) : $R(\widehat{g}_{n,\mathcal{G}}) - R(g^\star_{\mathcal{G}})$
- systématique (ou d'approximation ou biais) : $R(g^\star_{\mathcal{G}}) - R(g^\star)$

$$R(\widehat{g}_{n,\mathcal{G}}) - R(g^\star) = \left[(R(\widehat{g}_{n,\mathcal{G}}) - R(g^\star_{\mathcal{G}})\right] + \left[R(g^\star_{\mathcal{G}}) - R(g^\star)\right]$$

Quand $\mathcal{G}$ grand, l'erreur d'approximation est petite mais l'erreur d'estimation grande. Il y a un compromis à trouver : c'est le dilemne biais / variance.

Minimisation risque empirique 4 : dilemne biais / variance



Minimisation du risque empirique 5 : erreur stochastique

• $g^\star_{\mathcal{G}}$ prédicteur minimisant $R(g)$ sur $\mathcal{G}$: coût moyen des erreurs de prédiction sur toutes les observations étiquetées. Mais $g^\star$ inconnu.

• On se contente de $\widehat{g}_{n,\mathcal{G}}$ minimisant R_n sur $\mathcal{D}_n$ pour $g \in \mathcal{G}$.

Bornes sur l'erreur stochastique

$$0 \leq R(\widehat{g}_{n,\mathcal{G}}) - R(g^\star_{\mathcal{G}}) \leq 2 \max_{g \in \mathcal{G}} |R(g) - R_n(g)| \tag{18}$$

Démonstration en exercice.

• Le max dans l'inégalité précédente mesure les fluctuations entre l'écart empirique et l'écart théorique sur le dictionnaire $\mathcal{G}$.

Remarques sur le sur-apprentissage et sous-apprentissage

On cherche un biais faible et un risque empirique proche du risque théorique, pour une bonne capacité de généralisatiion du prédicteur $\widehat{g}_n$: antagonisme appelé dilemne biais-variance :

- Pour minimiser $R(g)$ il faut avoir un modèle riche, c'est à dire un dictionnaire de bonne taille.
- La fluctuation aléatoire augmente avec la taille du dictionnaire.

Mais le risque empirique est un estimateur sans biais et consistant de $R(g)$: on a dans tous les cas intérêt à utiliser une grande base d'apprentissage (n grand) pour diminuer la fluctuation sur $\mathcal{G}$.

$\Rightarrow$ Lorsqu'on augmente M, on doit augmenter également n.

Inégalité oracle pour un dictionnaire fini

Inégalité oracle pour un dictionnaire fini

Pour une fonction de perte l à valeurs dans $[0,1]$ et un dictionnaire fini $\mathcal{G}$ contenant M fonctions, alors pour tout $\delta \in]0,1[$, avec probabilité supérieure à $1-\delta$,

$$R(\widehat{g}_{n,\mathcal{G}}) - R(g^{\star}_{\mathcal{G}}) \leq \sqrt{\frac{2}{n} \ln\left(\frac{2M}{\delta}\right)} \tag{19}$$

Démonstration en exercice.

Quand $n \to +\infty$, $\not\to 0$. $\uparrow n \Rightarrow \uparrow$ la qualité du prédicteur empirique. Mais en même temps, $|\mathcal{G}| \uparrow \Rightarrow$ le terme de droite augmente. M petit $\Rightarrow$ risque théorique est grand. À rapprocher du dilemne biais variance: quand $|\mathcal{G}|$ augmente, le biais diminue (erreur de modélisation due au modèle) mais la variance augmente (erreur statistique due à l'aléa des données). Augmenter M augmente le risque de sur-apprentissage.

Utilisation de $R_n(g_n)$ pour estimer $R(g_n)$

$$R_n(g_n) = \frac{1}{n} \sum_{i=1}^{n} l(Y_i, g_n(X_i)) \tag{20}$$

g_n dépend de $\mathcal{D}_n$ donc de tous les (X_i, Y_i) en même temps : les va $l(Y_i, g_n(X_i))$ ne sont pas indépendantes et on ne peut pas appliquer la LFGN pour poruver une convergence.

$R_n(g_n)$ conduit à sous-estimer $R(g_n)$.

Pour éviter cela, on utilise la validation croisée ou du bootstrap.

Le phénomène de double-descente - 1/2

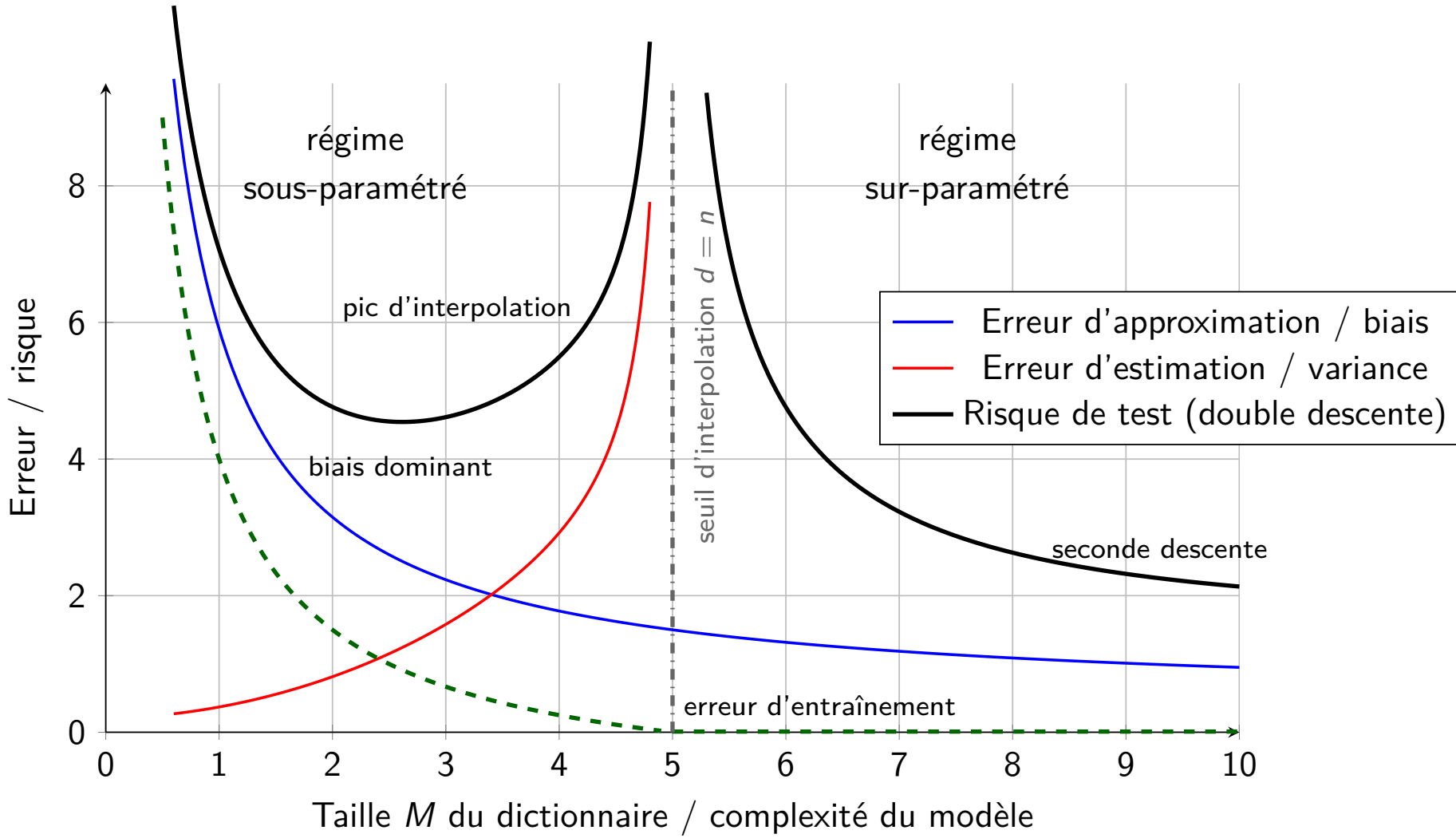


Figure 2: Régime sous-paramétré: compromis biais–variance classique. Pic au seuil d'interpolation ($d = M = n$), redescend dans le régime sur-paramétré.

Le phénomène de double-descente - 2/2

- Récent : redécouvert aux alentours de 2018.
- S'intéresse aux modèles sur-paramétrés, lorsque le nombre de paramètres et/ou la complexité du modèle est plus grand que la taille de l'échantillon d'entraînement.
- Typiquement le cas dans les méthodes de réseaux de neurones.
- Le dilemne biais-variance laisse place à une seconde descente qui permet parfois d'atteindre des niveaux d'erreur inférieurs à ceux de la zone sous-paramétrée.
- Le phénomène est dû à la capacité de généralisation d'un modèle, qui est fondamentale en apprentissage statistique, mais très mal connue.
- Voir l'ouvrage de Francis Bach Learning from First Principles.

Dilemne biais/variance

La complexité des modèles et la présentation du dilemme biais-variance est exposée par Stéphane Mallat dans son cours du Collège de France de 2018, dont les vidéos sont disponibles en ligne en suivant ce lien :

https://www.college-de-france.fr/site/stephane-mallat/course-2017-2018.htm

Quelques exemples de dictionnaires / classes

- Régression linéaire : $\mathcal{G} = \{h : x \mapsto h(x) = b + w^\top x\}$.
- Régression logistique : $\mathcal{G} = \{h : x \mapsto h(x) = \Lambda(b + w^\top x)\}$.
- Classifieur bayésien naïf binaire :
  $\mathcal{G} = \{h : x \mapsto h(x) = \mathbb{1}_{\left[\mathbb{P}[Y=1|X=x] > \frac{1}{2}\right]}\}$.
- Classifieur bayésien naïf multinomial :
  $\mathcal{G} = \{h : x \mapsto h(x) = \mathbb{1}_{[\arg\max_k \mathbb{P}[Y=y_k|X=x]]}\}$.
- Réseaux de neurones : $\mathcal{G} = \{h : x \mapsto h(x)\}$ avec h tel que

$$h(x) = \bigodot_{i=1}^{L} \sigma(w_i^T x + b_i) = \sigma\left(w_L \sigma\left(....\sigma\left(w_1^T x + b_1\right)...\right)\right)$$

Validation croisée et sélection de modèles -1-

• L'estimateur minimisant le risque empirique sur $\mathcal{G}$ est $\widehat{g}_n$ (dépend évidemment de $\mathcal{G}$).

• $R(\widehat{g}_n)$ possiblement éloigné de $R_n(\widehat{g}_n)$ car non indépendants : dépendent de $\mathcal{D}_n$ au travers de $\widehat{g}_n$.

• Ce sont donc des v.a. fonctions de (X_i, Y_i). On ne peut donc pas utiliser R_n pour évaluer un estimateur qui a été construit à partir du même échantillon.

• Pour contourner ce problème, 3 méthodes : découpage simple des données en un jeu d'entraînement et un jeu de test, validation « Hold-out » ou méthode de validation croisée.

Validation croisée et sélection de modèles -2-

• Découpage basique de $\mathcal{D}_n$ en un jeu d'entraînement et de test : on entraîne le modèle sur le sous-échantillon d'entraînement et on évalue ses performances sur l'échantillon de test, indépendant de l'estimateur.

• Méthode de validation « Hold-out » : utile si sélection de modèles avec plusieurs jeux d'hyperparamètres.

• Découpage en 2 ne suffit plus : les deux sous-échantillons sont dépendants ; il est nécessaire de conserver une troisième partie des données indépendante pour valider l'ensemble des modèles.

• $\mathcal{D}_n^E$, $\mathcal{D}_n^T$, $\mathcal{D}_n^V$ respectivement sous-échantillon d'entraînement, de test et de validation.

Validation « Hold-out »

1 • Pour chaque jeu d'hyperparamètres, on entraîne un estimateur $g_1, ..., g_m$ sur $\mathcal{D}_n^E$ avec $g_i = g_i(., \mathcal{D}_n^E)$.

2 • Pour chaque estimateur, on évalue le risque empirique $R_{n,\mathcal{D}_n^V}(g_1), ..., R_{n,\mathcal{D}_n^V}(g_m)$ sur le sous-échantillon $\mathcal{D}_n^V$.

3 • On sélectionne celui qui minimise le risque empirique :

$$\widehat{g}_n = \underset{i=1,...,m}{\arg\min}\, R_{n,\mathcal{D}_n^V}\left(g_i(\mathcal{D}_n^E, .)\right) \tag{21}$$

4 • On évalue les performances finales du modèle avec le risque calculé à l'aide du sous-échantillon de test :

$$\widehat{R}_n = R_{n,\mathcal{D}_n^T}\left(\widehat{g}_n\right). \tag{22}$$

Validation croisée k-Fold -1-

• 2 inconvénients à cette méthode : découpage réduit les données disponibles pour l'entraînement et performances fortement dépendantes du choix du sous-échantillon.

• La validation croisée corrige ce problème. Peut être utilisée dans tous les cas de figure (estimation unique ou sélection de modèles). Méthode à privilégier.

• On découpe en 3 l'échantillon $\mathcal{D}_n$: un sous-échantillon de test, un 2e de validation et un 3e d'entraînement.

• Sous-échantillon de test noté $\mathcal{D}_n^T$ représente traditionnellement 20% des données. Il restera constant tout au long du traitement des données.

• Les deux autres sous-échantillons forment une partition du reste de l'échantillon initial et vont varier durant le processus de validation croisée.

Validation croisée k-Fold -2-

Principe : diviser les données (hors échantillon de test) en sous-ensembles (les plis, ou « folds ») et permuter leurs rôles entre entraînement et validation.

1 • Après sélection du sous-échantillon de test, on divise les données restantes en k sous-échantillons disjoints de taille égale : $\mathcal{D}_{n,1}, ..., \mathcal{D}_{n,k}$.

2 • Pour $i = 1, ..., k$, $\mathcal{D}_{n,i}$ = ensemble de validation et les $k-1$ autres sous-échantillons $\mathcal{D}_{\bullet i} = \cup_{j \neq i} \mathcal{D}_{n,j}$ = base d'entraînement.

3 • On sélectionne l'estimateur qui minimise le risque empirique :

$$\widehat{g}_n = \underset{i=1,...,k}{\arg\min} \, R_{n,\mathcal{D}_{\bullet i}} \left(g_i(\mathcal{D}_{n,i}, .) \right) \tag{23}$$

Validation croisée k-Fold -3-

• On évalue la performance globale par moyenne des scores à chaque itération. Scores = risques empiriques de chaque estimateurs, calculés avec le sous-échantillon de test.

$$\widehat{R}_n = \frac{1}{k} \sum_{i=1}^{k} R_{n,\mathcal{D}_n^T} \left(g_i(\mathcal{D}_{n,i}, .) \right) \tag{24}$$

En pratique, on effectue souvent une re-calibration en ré-entraînant le modèle correspondant au meilleur jeu d'hyperparamètres sur la totalité des données (hors données de test).

Validation croisée k-Fold -4-

