

Éléments de théorie de l'apprentissage

S. Herbin

stephane.herbin@onera.fr

Cours précédents

- ▶ Principes généraux d'apprentissage : données apprentissage/validation/test, optimisation, évaluation.
- ▶ Plusieurs algorithmes de *classification supervisée* : plus proche voisin, classifieur Bayésien, arbres de décision.

Objectifs de ce cours

- ▶ Pourquoi ça marche : justifications théoriques, comment ça s'exprime.
- ▶ Domaine : « Statistical Machine Learning » ou « Computational Learning Theory »

Généralisation

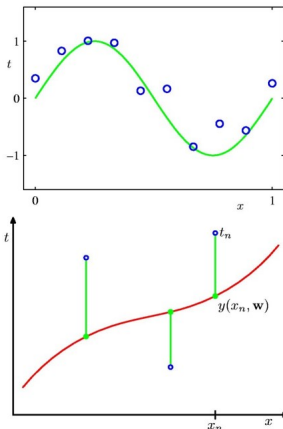
Régression

- ▶ La courbe verte est la véritable fonction $f(x)$ à estimer – mais inconnue.
- ▶ Les données $D_n = \{(x_i, y_i)\}_{i=1}^n$ sont considérées comme échantillonnées en x et bruitées en y par ϵ :

$$y = f(x) + \epsilon$$

- ▶ On cherche un prédicteur $f(x; \mathbf{w})$ paramétré par \mathbf{w} qui minimise l'erreur de régression :

$$E_{\text{RMS}} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - f(x_i; \mathbf{w}))^2}$$



Modèle linéaire généralisé

- ▶ On va rechercher le prédicteur comme combinaison linéaire de fonctions de base $\phi_k(x)$:

$$\begin{aligned}f(x; \mathbf{w}) &= w_0 \cdot \phi_0(x) + w_1 \cdot \phi_1(x) + \dots + w_M \cdot \phi_M(x) \\&= \mathbf{w}^t \cdot \Phi(x)\end{aligned}$$

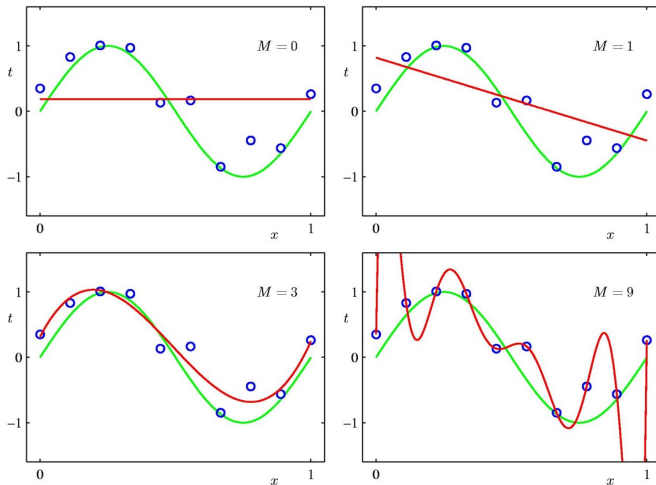
où $\mathbf{w} = [w_0, w_1, \dots, w_M]^t$ et
 $\Phi(x) = [\phi_0(x), \phi_1(x), \dots, \phi_M(x)]^t$.

- ▶ Si les fonctions de base sont $\phi_k(x) = x^k$, les prédicteurs sont à chercher dans la famille des polynômes de degré M .
- ▶ La minimisation de E_{RMS} donne la solution aux moindres carrés :

$$\mathbf{w}_{\text{RMS}} = (\Phi^t \Phi)^{-1} \Phi^t \mathbf{y}$$

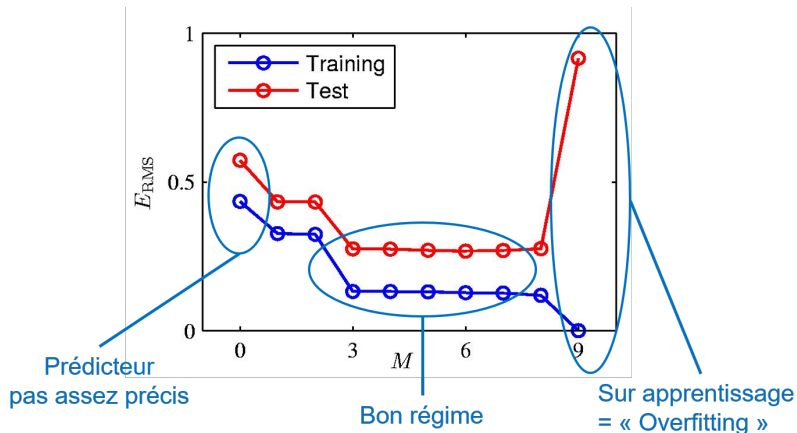
où Φ est la matrice de taille $n \times M + 1$ définie par
 $\Phi_{i,k} = \phi_k(x_i)$ et $\mathbf{y} = [y_1, y_2 \dots y_n]^t$.

Comportement des prédicteurs



Que valent-ils ?

Evaluation des prédicteurs



Comparaison des erreurs de prédiction entre données d'apprentissage D_n et de test.

Généralisation

Train & Test

- ▶ L'**erreur de généralisation** est l'erreur commise sur les données nouvelles (« non vues »).
- ▶ Elle est en général estimée par les données de **test**.
- ▶ Les données d'**apprentissage** sont utilisées comme moyen de modélisation dans le critère à optimiser.

Deux situations à contrôler (ou éviter)

- ▶ **Simplisme** : modélisation trop grossière pour rendre compte de la variété des données
Erreurs d'apprentissage et de test importantes
- ▶ **Sur-apprentissage (« Overfitting »)** : modèle trop complexe se spécialisant sur les données d'apprentissage
Écart important entre erreur d'apprentissage et erreur de test

Biais et variance

Biais et variance

Exemple de la régression : $y = f(\mathbf{x}) + \epsilon$

Il y a deux sources d'aléatoire :

- ▶ Le bruit : ϵ (un même \mathbf{x} peut produire différents y)
- ▶ L'échantillonnage des données d'apprentissage : D_n

On définit pour un prédicteur appris $\hat{f}_{D_n}(\mathbf{x})$:

Erreur écart quadratique moyen entre prédiction et valeur idéale

Biais erreur de la prédiction moyenne par rapport à la valeur idéale

Variance écart quadratique moyen entre prédiction et prédiction moyenne

Biais et variance

Compromis biais variance

L'erreur pour un \mathbf{x} donné peut se décomposer en :

$$\begin{aligned}\text{Err}^2 &= E_{D_n}[(y - \hat{f}_{D_n}(\mathbf{x}))^2] \\ &= \underbrace{\epsilon^2}_{\text{bruit}^2} + \underbrace{(E_{D_n}[\hat{f}_{D_n}(\mathbf{x})] - y)^2}_{\text{biais}^2} + \underbrace{E_{D_n}[(E_{D_n}[\hat{f}_{D_n}(\mathbf{x})] - \hat{f}_{D_n}(\mathbf{x}))^2]}_{\text{variance}}\end{aligned}$$

L'origine de l'erreur de généralisation est double, mais les deux termes sont difficiles à contrôler individuellement.

Rem : pour la classification, une telle décomposition est plus difficile à obtenir, mais les comportements sont comparables.

Biais et variance

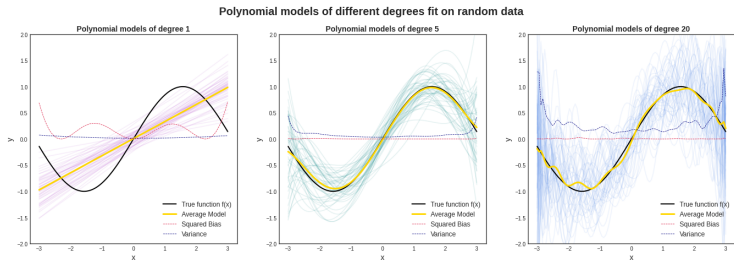


Figure 1 – Simulation d'une régression pour 50 échantillons et polynômes de degrés 1,5,20.

- ▶ Degré 1 : variance faible, mais biais important
- ▶ Degré 5 : variance et biais faibles
- ▶ Degré 20 : variance importante et biais très faible

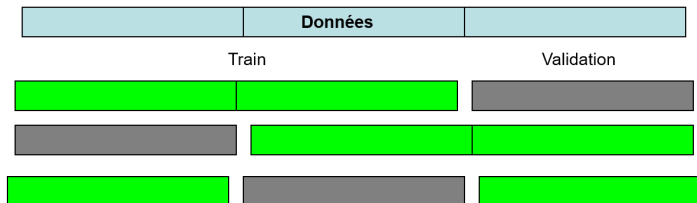
Validation croisée

Validation croisée : un outil pratique

- ▶ Permet d'estimer l'erreur de généralisation à partir des données d'apprentissage (« astuce »)
- ▶ Principe :
 1. Division des données en k sous ensembles (« fold »)
 2. Choix d'une partie comme ensemble de validation (= ensemble de test fictif), les autres comme train
 3. Apprentissage sur l'ensemble train
 4. Estimation des erreurs sur validation
 5. On fait tourner l'ensemble de validation sur chacun des fold
- ▶ L'erreur de généralisation estimée est la moyenne des erreurs sur chaque ensemble de validation

Principe de la Validation Croisée

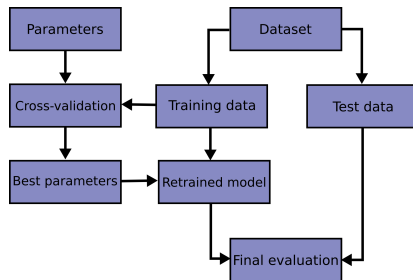
« k-fold »



« Leave-one-out »



Utilisation de la VC



Utilisation principale : réglage des **hyper-paramètres** :

- ▶ ex : Nombre de voisins d'un k-NN, Complexité d'un prédicteur (degré du polynôme), Profondeur des arbres, Contrainte de régularisation (prochain cours), etc.
- ▶ Stratégie d'optimisation : « grid search » + dichotomie.

Minimisation du risque empirique

Rappels

f une fonction de prédiction $\mathcal{X} \rightarrow \mathcal{Y} : y=f(x)$

Un ensemble de données $D_n = \{(x_i, y_i)\}_{1 \leq i \leq n}$ issues d'une distribution P

Plusieurs fonctions d'erreur

- ▶ Coût : $l(y, y') \in [0, +\infty[$, par exemple $l_{01}(y, y') = \mathbb{1}_{\{y \neq y'\}}$
- ▶ Risque vrai ou réel (théorique) :

$$L(f) = E[l(f(X), Y)] = \int l(f(x), y) dP(x, y)$$

- ▶ Risque empirique (calculable) :

$$L_n(f) = \frac{1}{n} \sum_{i=1}^n l(f(X_i), Y_i)$$

Minimisation du risque empirique (MRE)

Principe

- ▶ On se donne une famille de prédicteurs \mathcal{F} (on parle parfois d'hypothèses)
- ▶ Algorithme = Trouver dans cette famille celui qui minimise le risque empirique :

$$\hat{f}_n = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n l(f(X_i), Y_i)$$

Erreur de généralisation

- ▶ Le risque de ce prédicteur est potentiellement supérieur au risque réel (erreur de Bayes)

$$L(\hat{f}_n) \gg L^*$$

où $L^* = \inf_f L$ est le risque réel (idéal)

Autre décomposition biais/variance

$$L(\hat{f}_n) - L^* = \underbrace{L(\hat{f}_n) - L(f^*)}_{\text{estimation, stochastique}} + \underbrace{L(f^*) - L^*}_{\text{approximation, déterministe}}$$

où $f^* = \arg \min_{f \in \mathcal{F}} L(f)$ est le meilleur prédicteur possible de la famille \mathcal{F} .

Deux sources d'erreur :

- ▶ Le nombre limité de données
- ▶ La famille des prédicteurs (arbres, réseaux de neurones...)
- ▶ L'erreur d'approximation est liée à la modélisation du problème
- ▶ L'erreur d'estimation est liée à l'apprentissage : c'est elle que l'on peut essayer de contrôler.

Que veut-dire apprendre ?

Questions

1. Comment garantir que mon algorithme d'apprentissage se comporte bien : $L(\hat{f}_n) \xrightarrow{n \rightarrow \infty} L(f^*)$?
2. Combien de données pour garantir une erreur de généralisation minimale ?
3. Comment contrôler ou caractériser l'espace des fonctions de prédiction \mathcal{F} ?

PAC learning

Comment qualifier un algorithme d'apprentissage ?

Un premier cas simplifié

- ▶ Classification binaire : $\mathcal{Y} = \{0, 1\}$
- ▶ $|\mathcal{F}|$ fini
- ▶ \mathcal{F} contient un prédicteur parfait ($L(f^*) = 0$) : on parle de problème « réalisable »

Quelques conséquences

- ▶ Pour tout échantillon D_n : $L_n(f^*) = 0$ et $L_n(\hat{f}_n) = 0$ car $l(y, y') \geq 0$
- ▶ Mais en général $L(\hat{f}_n) > 0$ (erreur de généralisation)

Objectif

- ▶ Majorer la probabilité de faire des erreurs d'au plus ϵ : $P[L(\hat{f}_n) > \epsilon]$ par une fonction de ϵ et n .

Démonstration

On va prouver que

$$P[L(\hat{f}_n) > \epsilon] \leq |\mathcal{F}|e^{-n\epsilon}$$

Étapes

1. On s'intéresse aux ensembles de données D_n *trompeurs*, c-à-d qui estiment un prédicteur \hat{f}_n tel que $L(\hat{f}_n) > \epsilon$ (erreur réelle) mais avec $L_n(\hat{f}_n) = 0$ (pas d'erreur empirique)
2. Ces ensembles sont les seules sources d'erreur !
3. On va repérer ces ensembles à partir des prédicteurs erronés (ceux pour lesquels $L(f) > \epsilon$)
4. Puis on va calculer pour chaque prédicteur erroné la probabilité de tomber sur un ensemble trompeur

Détails I

1. On veut majorer la probabilité qu'un ensemble de données soit source d'erreur : $P\{D_n : L(\hat{f}_n) > \epsilon\}$
2. On repère les prédicteurs erronés : $\mathcal{F}_\epsilon = \{f : L(f) > \epsilon\}$
3. Les données source d'erreur sont trompeuses :

$$\{D_n : L(\hat{f}_n) > \epsilon\} \subset \bigcup_{f \in \mathcal{F}_\epsilon} \{D_n : L_n(f) = 0\}$$

4. On passe aux probabilités et on applique l'inégalité de Boole (« Union Bound ») :

$$P[L(\hat{f}_n) > \epsilon] \leq P\left[\bigvee_{f \in \mathcal{F}_\epsilon} \{L_n(f) = 0\}\right] \leq \sum_{f \in \mathcal{F}_\epsilon} P[L_n(f) = 0]$$

Détails II

5. On calcule la probabilité d'être erroné pour un ensemble $D_n = \{(x_i, y_i)\}_{1 \leq i \leq n}$:

$$\begin{aligned} P[L_n(f) = 0] &= P[\forall i \ f(x_i) = f^*(x_i)] \\ &= \prod_{1 \leq i \leq n} P[f(x_i) = f^*(x_i)] \\ &\leq \prod_{1 \leq i \leq n} (1 - \epsilon) = (1 - \epsilon)^n \leq e^{-n\epsilon} \end{aligned}$$

car $f \in \mathcal{F}_\epsilon$

6. Et finalement

$$\begin{aligned} P[L(\hat{f}_n) > \epsilon] &\leq \sum_{f \in \mathcal{F}_\epsilon} e^{-n\epsilon} \\ &\leq |\mathcal{F}_\epsilon| e^{-n\epsilon} \leq |\mathcal{F}| e^{-n\epsilon} \end{aligned}$$

Interprétation de $P[L(\hat{f}_n) > \epsilon] \leq |\mathcal{F}|e^{-n\epsilon}$

- ▶ Cette inégalité est valable pour l'utilisation de l'algorithme MRE et pour toute famille finie de prédicteurs \mathcal{F} qui contient un prédicteur sans erreur ($L(f^*) = 0$).
- ▶ Elle indique que l'on peut obtenir un prédicteur par MRE avec une probabilité d'erreur bornée.
- ▶ On peut reformuler le résultat : Si $n \geq \frac{\log(|\mathcal{F}|/\delta)}{\epsilon} = m(\epsilon, \delta)$ alors on aura un risque réel inférieur à ϵ avec une probabilité $1 - \delta$.
- ▶ La valeur de $m(\epsilon, \delta)$ ne dépend pas de P (la distribution de données) !

« Probably Approximately Correct learnable »

Définition

Une famille de prédicteurs \mathcal{F} est PAC apprenable s'il existe une fonction $m :]0, 1]^2 \rightarrow \mathbb{N}$ et un algorithme d'apprentissage tels que : pour tout $\epsilon > 0$ et $\delta > 0$, en appliquant l'algorithme d'apprentissage sur un échantillon de taille $m(\epsilon, \delta)$, on obtienne un prédicteur de risque inférieur à ϵ avec une probabilité de $1 - \delta$.

$$n \geq m(\epsilon, \delta) \Rightarrow P[L(\hat{f}_n) \leq \epsilon] \geq 1 - \delta$$

Relâcher la contrainte de réalisabilité

- ▶ Il est difficile de garantir : $\min_{f \in \mathcal{F}} L(f) = 0$ et donc de garantir une borne absolue sur le risque
- ▶ Lorsque $\min_{f \in \mathcal{F}} L(f) > 0$, on cherche plutôt à borner l'erreur d'estimation du risque réel : $L(\hat{f}_n) - L(f^*)$
- ▶ Cela conduit à une version dérivée de la notion de « PAC apprenable » dite « agnostique » = on ne sait pas si le prédicteur idéal est dans \mathcal{F} .
- ▶ On peut montrer, avec probabilité $1 - \delta$:

$$L(\hat{f}_n) - L(f^*) \leq \sqrt{\frac{2 \log(2|\mathcal{F}|/\delta)}{n}}$$

- ▶ Ce qui donne comme indicateur de complexité PAC :

$$m_{\text{agnostique}}(\epsilon, \delta) = \frac{2 \log(2|\mathcal{F}|/\delta)}{\epsilon^2}$$

Inégalité de Hoeffding

- ▶ Si $Z_1, Z_2 \dots Z_n$ sont des variables i.i.d. à valeur dans $[0, 1]$.
- ▶ Alors pour tout $\epsilon > 0$:

$$P \left[\left| \frac{1}{n} \sum_{i=1}^n Z_i - E(Z_1) \right| > \epsilon \right] \leq 2 \exp(-2n\epsilon^2)$$

Convergence uniforme

- ▶ On s'intéresse au comportement de tous les prédicteurs et on utilise l'inégalité :

$$L(\hat{f}_n) - L(f^*) \leq 2 \sup_{f \in \mathcal{F}} |L(f) - L_n(f)|$$

Des résultats généraux

- ▶ Permet de garantir que l'apprentissage MRE fonctionne pour des familles finies de prédicteurs
- ▶ Donne des bornes sur le nombre de données utiles (indépendamment de la distribution sous-jacente !)

Mais des limitations

- ▶ Ne donne de résultats que pour les familles finies de prédicteurs
- ▶ Ne dit rien sur la nature des prédicteurs
- ▶ Ne dit rien sur l'erreur d'approximation $L(f^*) - L^*$
- ▶ Calculer le MRE peut être complexe si $|\mathcal{F}|$ est grand
- ▶ Les bornes sont grossières (ne dépendent ni des données, ni de la nature des prédicteurs)

Caractériser les familles de prédicteurs

Des résultats plus fins ?

Que faire lorsque $|\mathcal{F}| = \infty$?

- ▶ Beaucoup de familles de prédicteurs sont dans ce cas (arbres, réseaux de neurones, bayésien naïf, ...)
- ▶ Ils ont une expressivité variable et contrôlable (profondeur des arbres, couches des RN)
- ▶ Peut-on produire des résultats comparables au cas fini ?

Comment particulariser le « PAC learning » ?

- ▶ Les résultats précédents ne dépendent pas de la nature de \mathcal{F}
- ▶ Comparer différentes familles de prédicteurs ?

⇒ Théorie de Vapnik Chervonenkis [Vapnik, 2013]

Complexité des familles de prédicteurs

Intuitions

- ▶ Expressivité = Capacité à discriminer un grand *nombre* de données.
- ▶ Famille de prédicteurs décrites par un nombre fini de paramètres : complexité = # paramètres ? ... pas tout à fait.

Un problème combinatoire

- ▶ Idée = on compte le nombre maximal de prédicteurs capables de discriminer un jeu de données quelconque de taille m .
- ▶ On regarde la forme de la fonction de croissance :

$$G_{\mathcal{F}}(n) = \max_{x_1, x_2, \dots, x_n \in \mathcal{X}} |\{(f(x_1), f(x_2), \dots, f(x_n)) : f \in \mathcal{F}\}|$$

- ▶ On a nécessairement : $G_{\mathcal{F}}(n) \leq 2^n$

Fonction de croissance

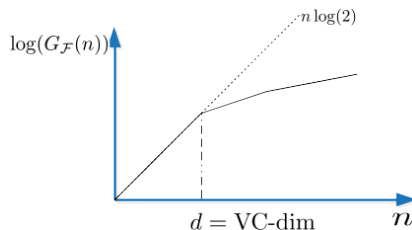


Figure 2 – Allure de la fonction de croissance.

Allure

- ▶ La fonction croît comme 2^n jusqu'à une certaine valeur (h sur Figure 2) puis moins vite
- ▶ Cette valeur est la dimension de Vapnik-Chervonenkis
- ▶ !!! Elle peut être infinie! = famille de prédicteurs très complexe \Rightarrow sur-apprentissage (en fait, aucune garantie d'apprentissage)

Dimension de Vapnik-Chervonenkis

Définition

- ▶ Plus grande taille n de données $X_n = (x_1, x_2, \dots, x_n)$ étiquetée de manière quelconque qui puisse être discriminée par un élément de \mathcal{F}
- ▶ On dit que \mathcal{F} pulvérise X_n
- ▶ Conséquence : il suffit de trouver une configuration de n points pulvérisée pour avoir $\text{VC-dim}(\mathcal{F}) \geq n$

Propriétés

- ▶ Lien avec fonction de croissance :
$$\text{VC-dim}(\mathcal{F}) = \max_n \{n : G_{\mathcal{F}}(n) = 2^n\}$$
- ▶ Lemme de Sauer : si $\text{VC-dim}(\mathcal{F}) = d < \infty$ alors
$$G_{\mathcal{F}}(n) \leq \sum_{i=0}^d \binom{n}{i}.$$

En particulier, si $n > d + 1$, $G_{\mathcal{F}}(n) \leq (e \cdot n/d)^d$ (croissance polynomiale $<$ exponentielle)

Un exemple 2D : hyperplans dans \mathbb{R}^2

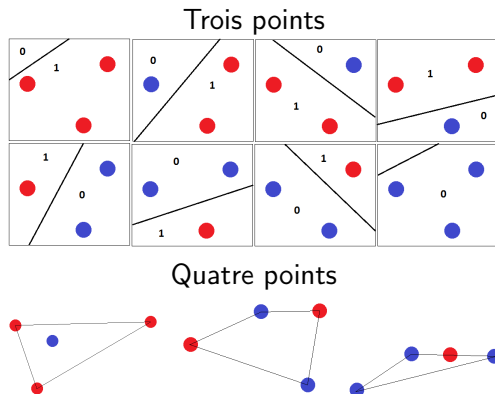


Figure 3 – Pulvérisation par hyperplan.

Les configurations de 4 points ne sont pas linéairement séparables :
 $\Rightarrow \text{VC-dim}(\text{« Hyperplans »}) = 3$

Un exemple 2D : rectangles dans \mathbb{R}^2

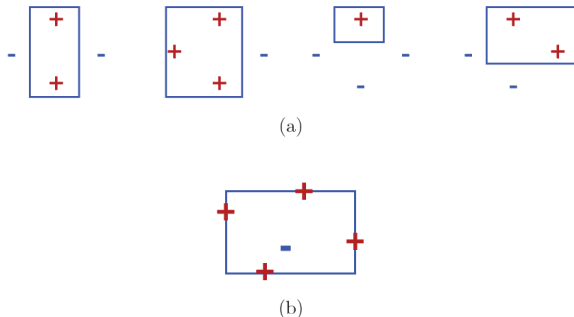


Figure 4 – Pulvérisation par fonction rectangle.

Pas moyen de séparer l'étiquetage (b) avec des rectangles :

$$\Rightarrow \text{VC-dim}(\text{« Rectangles »}) = 4$$

Un autre exemple 1D : fonction caractéristique sinusoïdale

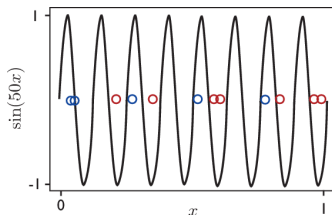


Figure 5 – Pulvérisation par fonction sinusoïdale.

- ▶ $f(x) = \text{signe}(\sin(\omega x))$ où $\omega \in [0, 2\pi)$.
- ▶ On peut trouver un ω qui sépare un ensemble de points de taille n quelconque

- ▶ $x_j = 2\pi 10^{-j}$
- ▶ $w = \frac{1}{2} \left(1 + \sum_{i=1}^n \frac{1-y_i}{2} 10^i \right)$

⇒ $\text{VC-dim}(\text{« Sinusoides »}) = \infty$

Complexité et erreur d'estimation

On peut montrer, si $\text{VC-dim}(\mathcal{F}) = d$, avec probabilité $1 - \delta$:

$$L(\hat{f}_n) \leq \inf_{f \in \mathcal{F}} L(f) + \sqrt{\frac{2d(1 + \log(n/d))}{n}} + \sqrt{\frac{\log(1/\delta)}{2n}}$$

Interprétation

- ▶ Si la famille de prédicteurs est de dimension VC fini, alors elle est PAC apprenable
- ▶ On peut borner l'erreur sur le risque par un $O\left(\sqrt{\frac{\log(n/d)}{n/d}}\right)$
- ▶ On peut aussi montrer que si la dimension VC de \mathcal{F} est infinie, elle n'est pas PAC apprenable.

Ce *théorème fondamental de l'apprentissage statistique* implique qu'il y a **équivalence entre PAC apprenable et avoir une dimension VC finie** pour une famille de prédicteurs.

Exemples de dimensions VC

- ▶ Hyperplans dans \mathbb{R}^d : $d + 1$
- ▶ Rectangles alignés sur les axes dans \mathbb{R}^2 : 4
- ▶ Rectangles quelconques dans \mathbb{R}^2 : 7
- ▶ Triangles dans \mathbb{R}^2 : 7
- ▶ Polygones convexes dans \mathbb{R}^2 : ∞
- ▶ Réseaux de neurones avec RELU (W paramètres et L couches)[Bartlett et al., 2019] : $\geq c \cdot WL \log(W/L)$ et $\leq C \cdot WL \log W$

D'autres inégalités I

Fonction de croissance

- ▶ On peut également montrer, avec probabilité $1 - \delta$, $\forall f \in \mathcal{F}$:

$$L(f) \leq L_n(f) + \sqrt{\frac{2G_{\mathcal{F}}(n)}{n}} + \sqrt{\frac{\log(1/\delta)}{2n}}$$

- ▶ ou de manière plus générale

$$P[|L(f) - L_n(f)| > \epsilon] \leq 4 \cdot G_{\mathcal{F}}(2n) \exp(-n\epsilon^2/8)$$

- ▶ La difficulté est d'estimer la fonction de croissance (la dimension VC est une simplification)

D'autres inégalités II

Complexité de Rademacher

- Définition : espérance de la « pire mauvaise classification »

$$R_n(\mathcal{F}, D_n) = E_\sigma \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \sigma_i h(x_i) \right] \text{ et } \bar{R}_n(\mathcal{F}) = E_P[R_n(\mathcal{F}, D_n)]$$

où σ_i est une variable aléatoire uniforme i.i.d. sur $\{-1, 1\}$

- C'est une quantité qui dépend de la distribution
⇔ bornes plus fines

D'autres inégalités III

Complexité de Rademacher

- Lien avec fonction de croissance

$$\bar{R}_n(\mathcal{F}) \leq \sqrt{\frac{2 \log(G_{\mathcal{F}}(n))}{n}}$$

- Lien avec erreur d'estimation

$$L(f) \leq L_n(f) + \bar{R}_n(\mathcal{F}) + \sqrt{\frac{\log(1/\delta)}{2n}}$$

Les deux combinés redonnent la borne utilisant la VC-dimension.

MRE : bornes sur le nombre de données

Hypothèses

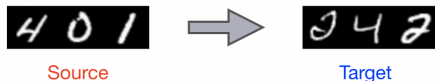
- ▶ \mathcal{F} est fini ou sa dimension $\text{VC-dim}(\mathcal{F}) = d$ est finie.
- ▶ Fonction de coût 0 – 1 ($l(y, y') = \mathbb{1}_{\{y \neq y'\}}$).
- ▶ Inégalité à probabilité $1 - \delta$.

$ \mathcal{F} $	Réalisable ($\inf_{f \in \mathcal{F}} = 0$) $P[L(\hat{f}_n) \leq \epsilon]$	Agnostique ($\inf_{f \in \mathcal{F}} > 0$) $P[L(\hat{f}_n) - \inf_{f \in \mathcal{F}} \leq \epsilon]$
$< \infty$	$n \geq \frac{\log(\mathcal{F} /\delta)}{\epsilon}$	$n \geq \frac{2 \log(2 \mathcal{F} /\delta)}{\epsilon^2}$
$= \infty$	$n = O\left(\frac{d \log(1/\epsilon) + \log(1/\delta)}{\epsilon}\right)$	$n = O\left(\frac{d + \log(1/\delta)}{\epsilon^2}\right)$

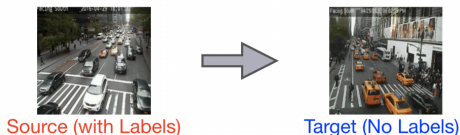
Domaine d'utilisation

Domaine d'utilisation et apprentissage I

Generalization: Source (Train) = Target (Test)



Domain adaptation: Source (Train) \neq Target (Test)



- ▶ En pratique, la distribution des données d'apprentissage (« source domain ») n'est souvent pas identique à celles qui seront traitées en conditions réelles (« target domain »).
- ▶ Rupture de l'hypothèse que le passé (apprentissage) = futur (inférence).
- ▶ C'est une autre source d'erreur que la généralisation.

Domaine d'utilisation et apprentissage II

Ce phénomène conduit à d'autres types de problèmes :

- ▶ décalage de domaine (« shift »)
- ▶ généralisation de domaine
- ▶ adaptation/transfert de domaine
- ▶ dérive de domaine (« drift »)
- ▶ ...

Théorie de l'apprentissage limitée en résultats : dépendance au contexte d'utilisation et à la nature des données difficile à formaliser.

Et le deep learning ?

Erreur d'optimisation

- ▶ Dans la théorie PAC, on suppose que l'on peut minimiser le risque empirique.
- ▶ Mais : l'optimisation peut aussi générer des erreurs.
Une autre décomposition de l'erreur de généralisation

$$\begin{aligned} L(\hat{f}_n) - L(f^*) = \\ \underbrace{L(\hat{f}_n) - L_n(\hat{f}_n)}_{\text{estimation, stochastique}} + \underbrace{L_n(\hat{f}_n) - L_n(f^*)}_{\text{optimisation, stochastique}} + \underbrace{L_n(f^*) - L(f^*)}_{\text{estimation, stochastique}} \end{aligned}$$

où f^* est le prédicteur optimal du risque L et \hat{f}_n est le prédicteur appris à partir du jeu de données de taille n et issu d'une optimisation (gradient stochastique).

- ▶ Il faut encore rajouter l'erreur d'approximation (famille de prédicteurs).

Régularisation implicite du SGD

- ▶ Régularisation = une manière de limiter les espaces de prédicteurs. (prochain cours)
- ▶ La descente de gradient permet d'introduire une forme de régularisation. [Neyshabur et al., 2014, Dauber et al., 2020, Soudry et al., 2018]

La « double descente »

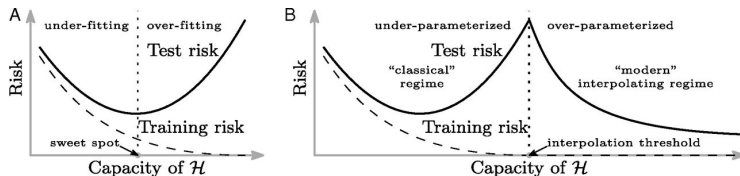


Figure 6 – Ecart apprentissage/test pour différentes complexités de prédicteurs.

- Les réseaux sur-paramétrés ont (parfois) des erreurs de généralisation plus faible : phénomène contre-intuitif.

[Belkin et al., 2019, Dar et al., 2021, Nakkiran et al., 2019]

Il y a encore bien d'autres questions...

- ▶ Autres algorithmes que MRE : arbres, ensembles, SVM, RN ?
- ▶ Comment introduire l'optimisation ($\arg \min$) dans les bornes
- ▶ Bornes inférieures (conditions nécessaires)
- ▶ Bornes dépendant des algorithmes, des distributions
- ▶ Contrôler la variance des écarts plutôt que \sup
- ▶ Utiliser les bornes en pratique pour contrôler les algorithmes
- ▶ Quel algorithme/stratégie utiliser avec des familles de dimension VC infinies ?
- ▶ Les algorithmes itératifs/séquentiels (renforcement, bandit...) : quelles garanties de convergence ?
- ▶ Pourquoi (et comment) les réseaux profonds qui contiennent plus de paramètres que de données généralisent-ils ?
- ▶ Pourquoi existe-t-il des exemples adversariaux ? Comment les contrer ?
- ▶ ...

Minimisation du risque empirique : Résumé

Résultats

- + Bornes théoriques
- + Justification de la faisabilité de l'apprentissage
- + Bornes indépendantes des distributions
 - Résultats en probabilité $(1 - \delta)$
 - Il y a d'autres algorithmes que MRE
 - Pas applicable au « Deep Learning »

Utilisations

- + Garantie
 - Bornes trop lâches ou trop générales (convergence uniforme)
 - Complexités difficiles à calculer (VC ou Rademacher)

Références I



Bach, F. (2021).
Learning theory from first principles.



Bartlett, P. L., Harvey, N., Liaw, C., and Mehrabian, A. (2019).
Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks.
Journal of Machine Learning Research, 20(63) :1–17.



Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019).
Reconciling modern machine-learning practice and the classical bias–variance trade-off.
Proceedings of the National Academy of Sciences, 116(32) :15849–15854.
Publisher : National Academy of Sciences Section : Physical Sciences.



Dar, Y., Muthukumar, V., and Baraniuk, R. G. (2021).
A Farewell to the Bias-Variance Tradeoff? An Overview of the Theory of Overparameterized Machine Learning.
arXiv :2109.02355 [cs, stat].
arXiv : 2109.02355.



Dauber, A., Feder, M., Koren, T., and Livni, R. (2020).
Can implicit bias explain generalization ? stochastic convex optimization as a case study.
Advances in Neural Information Processing Systems, 33 :7743–7753.



Devroye, L., Györfi, L., and Lugosi, G. (2013).
A probabilistic theory of pattern recognition, volume 31.
Springer Science & Business Media.



Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018).
Foundations of machine learning.
MIT press.

