# Théorie de l'apprentissage

Cours 7
ARF  Master DAC

Nicolas Baskiotis
nicolas.baskiotis@lip6.fr
http://webia.lip6.fr/~baskiotisn

équipe MLIA, Laboratoire d'Informatique de Paris 6 (LIP6)
Université Pierre et Marie Curie (UPMC)

S2 (2016-2017)

# Plan

# Apprentissage supervisé et risque

## Problèmatique de l'apprentissage supervisé

- un ensemble d'apprentissage $E = \{(\mathbf{x}^i, y^i)\} \in \mathcal{X} \times \mathcal{Y}$
- un ensemble de fonctions $\mathcal{F}$
- un coût $\ell(\hat{y}, y) : Y \times Y \to \mathbb{R}^+$
- trouver $f = argmin_F \sum_i \ell(f(\mathbf{x}^i), y^i)$

## Minimisation du risque et risque bayésien

- Minimisation du risque
$$R_{\ell,P}(f) = \int_{\mathcal{X}\times\mathcal{Y}} \ell(f(\mathbf{x}), y) dP(\mathbf{x}, y) = \int_{\mathcal{X}\times\mathcal{Y}} \ell(f(\mathbf{x}), y) p(\mathbf{x}, y) dxdy$$
$$= \mathbb{E}_{\mathbf{x},y}[\ell(Y, f(X)]$$
- Risque empirique sur les données $|E| = n$ : $\frac{1}{n}\hat{R}_{n,\ell,P} \sum_{i=1}^{n} \ell(f(\mathbf{x}^i), y^i)$
- Risque bayésien : $R^\star_{\ell,P}(f) = inf_{f:\mathcal{X}\to\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \ell(f(\mathbf{x}), y) dP(\mathbf{x}, y)$
- Objectif : à partir de données $E$ trouver une fonction dont le risque est proche de l'optimal.

# Consistance d'un algorithme

## Définition

- Le risque : variable aléatoire
  $R_{\ell,P}(f) = \mathbb{E}[\ell(Y, f_E(X)|E]$
- Un algorithme est **universellement consistant** si pour toute distribution $P(X, Y)$ des données le risque de la fonction apprise converge vers le risque bayésien quand $|E| \to \infty$ : $lim_{[E]\to\infty} R_{\ell,P}(f) \to R^*_{\ell,P}$

## Théorème de Stone, 1977

- Sous certaines conditions, pour certaines fonctions de coût, les algorithmes vu dernièrement sont universellement consistants.
- Mais on dispose rarement d'une infinité de données . . .
- Et à quel rythme on converge ?

# No free lunch

## Théorème Devroy, 1982

Pour tout algorithme universellement consistant et pour tout taux de convergence $a_n$, il existe une distribution $P(X, Y)$ telle que le taux de convergence de l'algorithme soit plus lent que $a_n$.

## Autre point de vue

- Pour deux algorithmes d'apprentissage $1$ et $2$, sans a priori sur le problème, à $n$ fixé :
  - si toutes les fonctions cibles sont équiprobable, en espérance de l'erreur sur tous les problèmes les algorithmes $1$ et $2$ sont équivalents;
  - il n'y a pas d'algorithme universellement meilleur qu'un autre;
  - il existe au moins un problème tel que l'aléatoire fasse de meilleurs résultats que un algorithme donné;

Exemple dans le cas d'attributs discrets ?

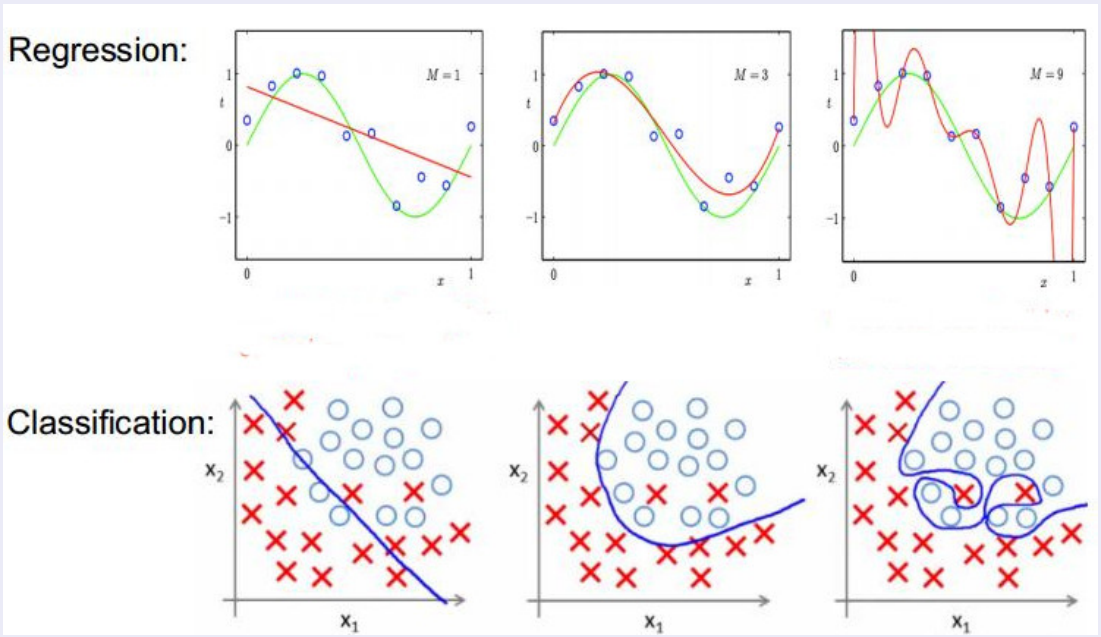
# Risque empirique

## Comment évaluer le risque ?

- Loi des grands nombres : $\hat{R}_n(f) \rightarrow_{n\rightarrow\infty} R(f)$
- Le risque empirique converge vers le risque bayésien

$\Rightarrow$ Choisir $f = inf_f \hat{R}_n(f)$

## Mais sur-apprentissage ...

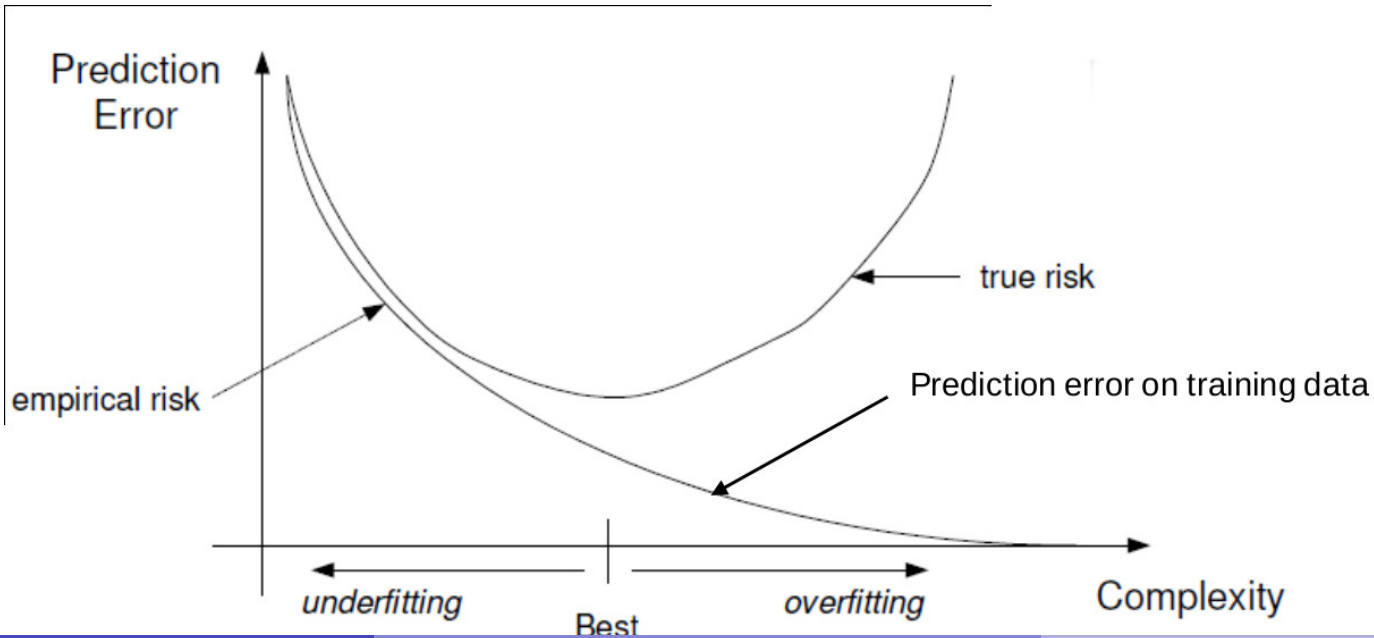



Solution : **restreindre** la famille de fonctions considérée : $f = inf_{f\in\mathcal{F}} \hat{R}_n(f)$

# Minimisation structurelle du risque

## Principe

- On ne considère que les fonctions dans une famille $\mathcal{F}$ :
  $f = argmin_{f \in \mathcal{F}} \hat{R}_n(f) = argmin_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(y^i, f(\mathbf{x}^i))$
- vrai risque sur $\mathcal{F}$ : $R^\star_{\mathcal{F}}$
- risque empirique sur $\mathcal{F}$ : $\hat{R}^\star_{n,\mathcal{F}}$
- Condition nécessaire : $R^\star_{\mathcal{F}} - R^\star \geq 0$ doit être petit !
- Minimisation structurelle : faire évoluer $\mathcal{F}$ en fonction de $n$ nombre d'exemples : $\mathcal{F}_{n+1} \supseteq \mathcal{F}_n$



# Classifieur bayésien

## Rappel

- Fonction de coût : 0-1 loss ($\ell(y, f(\mathbf{x})) = 1_{f(\mathbf{x}) \neq y}$)
- $R^\star = inf_f P(y \neq f(\mathbf{x}))$
- $f^\star = arginf_f P(y \neq f(\mathbf{x}))$
- Propriétés :
  - $P(y \neq f^\star(\mathbf{x})) \leq P(y \neq f(\mathbf{x}))$ pour tout $f$
  - $f^\star = \begin{cases} 1 & \text{si } \eta(\mathbf{x}) > 1/2 \\ 0 & \text{si } \eta(\mathbf{x}) \leq 1/2 \end{cases}$ avec $\eta(\mathbf{x}) = \mathbb{E}[Y = 1|\mathbf{x}]$

## Notations

$$R(f) = P(y \neq f(\mathbf{x})) \qquad \hat{R}(f) = \frac{1}{n} \sum_i 1_{f(\mathbf{x}^i) \neq y^i}$$

$$R^\star = R(f^\star) = inf_f R(f) \qquad R^\star_{\mathcal{F}} = R(f^\star_{\mathcal{F}}) = inf_{f \in F} R(f) \qquad \hat{R}^\star_{n,f} = inf_{f \in F} \hat{R}_n(f)$$

$$f^\star = arginf_f R(f) \qquad f^\star_{\mathcal{F}} = arginf_{f \in \mathcal{F}} R(f) \qquad f^\star_{n,\mathcal{F}} = argmin_f \hat{R}_n(f)$$

$f^\star_{n,\mathcal{F}}$ est ce que produit l'algorithme d'apprentissage.

# Erreurs d'estimation/d'approximation

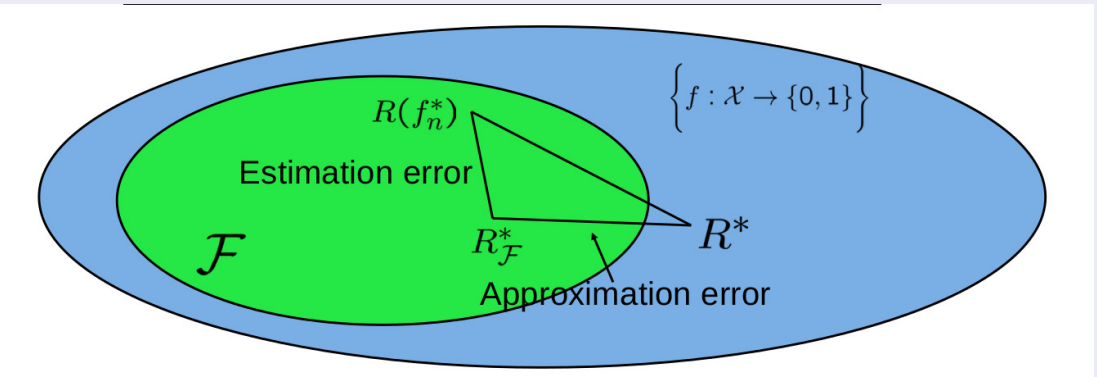
## Décomposition de l'erreur

$$R(f) - R(f^\star) = \underbrace{R(f) - inf_{f\in\mathcal{F}} R(f)}_{\text{erreur d'estimation}} + \underbrace{inf_{f\in\mathcal{F}} R(f) - R(f^\star)}_{\text{erreur d'approximation}}$$

variance biais

Biais : différence entre la fonction apprise et le vrai risque

Variance : différence dans l'approximation de la fonction cible dans $\mathcal{F}$



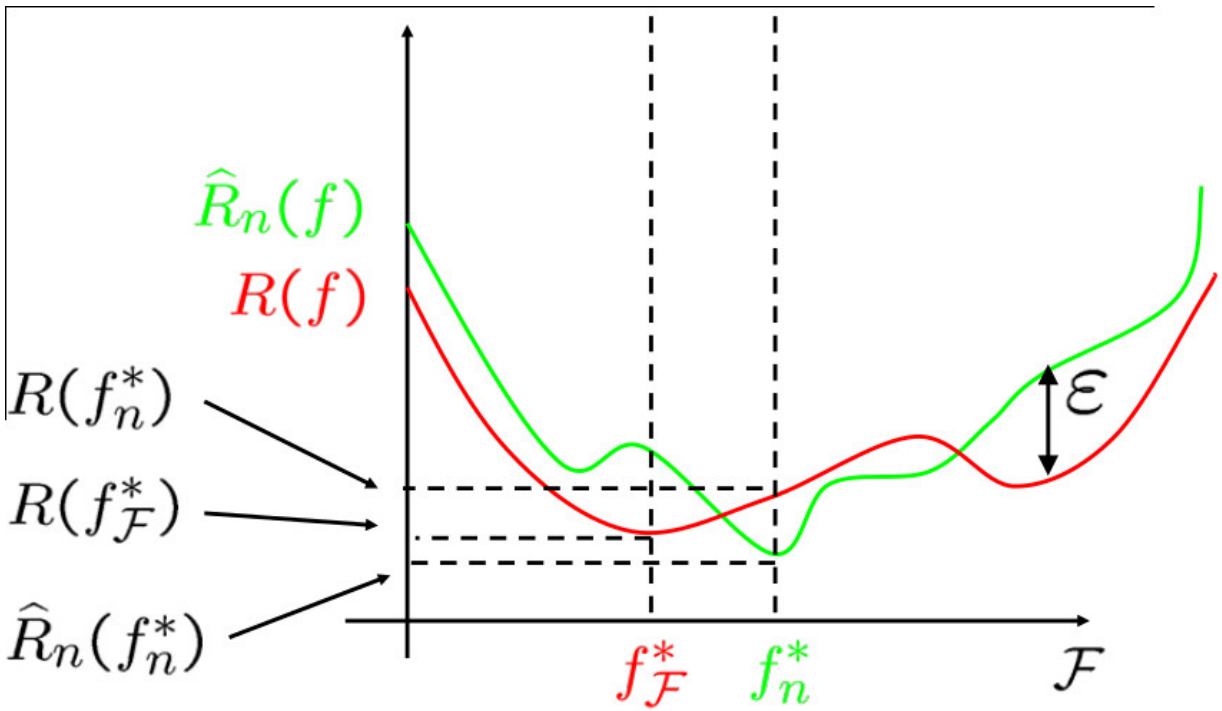

## Sur/sous-apprentissage

- Si $\mathcal{F}$ trop grand : $R^\star_{\mathcal{F}}$ petit, sur-apprentissage, erreur d'approximation petite, erreur d'estimation grande, $R(f_n^\star)$ est grand, $\hat{R}_n(f_n^\star)$ est petit
- Si $\mathcal{F}$ trop petit : $R^\star_{\mathcal{F}}$ grand, sous-apprentissage, erreur d'approximation grande, erreur d'estimation petite, $\hat{R}_n(f_n^\star)$ proche de $R(f_n^\star)$, $R(f_n^\star)$ proche de $R^\star_{\mathcal{F}}$

# Quelques théorèmes

- A combien est-on de l'optimal dans $\mathcal{F}$ :
  $|R(f^{\star}_{n,\mathcal{F}}) - R^{\star}_{\mathcal{F}}| \leq 2sup_{f\in\mathcal{F}}|\hat{R}_n(f) - R(f)|$
- A combien le risque empirique est éloigné du vrai risque :
  $|\hat{R}_n(f^{\star}_{n,\mathcal{F}}) - R(f^{\star}_n)| \leq sup_{f\in\mathcal{F}}|\hat{R}_n(f) - R(f)|$

$\Rightarrow$ Risque empirique par rapport au meilleur classifieur dans $\mathcal{F}$ :
$|\hat{R}_n(f^{\star}_{n,\mathcal{F}}) - R^{\star}_{\mathcal{F}}| \leq 3sup_{f\in\mathcal{F}}|\hat{R}_n(f) - R(f)|$



# Un cas concret : espace de recherche fini

## Contexte

- $\mathcal{F}$ est fini
- Soit $f$ telle que pas d'erreurs sur $E$
- Probabilité que la vrai erreur $e(f)$ de $f$ soit plus grande que $\epsilon$ donné ?

## Variable de Bernouilli

- $P(\text{une erreur}|e(f) = \epsilon) = \epsilon$ par définition
- $P(\text{une erreur}|e(f) \geq \epsilon) \geq \epsilon$
- $P(\text{pas erreur}|e(f) \geq \epsilon) = 1 - P(\text{erreur}|e(f) \geq \epsilon) \leq 1 - \epsilon$
- $P(\text{pas d'erreurs sur n points}|e(f) \geq \epsilon) \leq (1 - \epsilon)^n \leq e^{-\epsilon n}$

# Un cas concret : espace de recherche fini

## Borne sur l'erreur

$P(\text{pas d'erreurs sur n points} | e(f) \geq \epsilon) \leq (1-\epsilon)^n \leq e^{-\epsilon n}$

- $|\mathcal{F}_c| = \{f_1, \ldots, f_K\}$ : nombre de fonctions sans erreurs sur $D$
- Quelle est la probabilité de choisir une mauvaise, c'est-à-dire $e(f) \geq \epsilon$ ?
  $P(e(f_1) \geq \epsilon \text{ ou } e(f_2) \geq \epsilon \text{ ou } \cdots e(f_K) \geq \epsilon) \leq \sum_k P\left(e(f_k) \geq \epsilon\right)$
  $\leq \sum_k (1-\epsilon)^m \leq K(1-\epsilon)^m \leq Ke^{-n\epsilon}$

$\Rightarrow$ $P(e(f) \geq \epsilon) \leq |\mathcal{F}_c| e^{-n\epsilon}$ (Haussler, 1988)

## Cadre PAC : Probably Approximately Correct

- $\text{PAC}_{\epsilon,\delta}$ : $P(e(f) \geq \epsilon) \leq \delta$
- Soit on choisit $\epsilon$ et $\delta$, puis on calcule $n$ : $n \geq \frac{ln|\mathcal{F}| + ln\frac{1}{\delta}}{\epsilon}$
- Soit on choisit $n$ et $\delta$, puis on calcule $\epsilon$ : $\epsilon \geq \frac{ln|H| + ln\frac{1}{\delta}}{m}$

# Limites de la borne de Haussler

## Questions

- Si pas de fonction d'erreur nulle ?
- Si $|\mathcal{F}|$ est vraiment grand ?
- Si $|\mathcal{F}|$ est infini ?

## Généralisation et remarques

- $P(e(f) - e_E(f) \geq \epsilon)) \leq |\mathcal{F}|e^{-2m\epsilon^2}$
- $e(f) \leq e_E(f) + \sqrt{\frac{ln|\mathcal{F}| + ln\frac{1}{\delta}}{2m}}$
- Décomposition biais/variance ...
- Pour $\delta, \epsilon$ fixés, $m \geq \frac{1}{2\epsilon^2}(ln|\mathcal{F}| + ln|\frac{1}{\delta}|)$

# Cas de famille infinie de fonctions

## Question

- Combien de points en 1D un classifieur linéaire peut-il séparer ?
- En 2D ?
- Et un arbre de profondeur 2 ?

## Définitions

- Un ensemble de points est *shattered* (pulvérisé) par un espace de fonction si pour tout partitionnement des points en deux ensembles il existe une fonction qui sépare les deux partitions.
- La VC-dimension (Vapnik-Chervonenkis) de $\mathcal{F}$ sur un espace de données $\mathcal{X}$ est la taille du plus grande ensemble fini de points de $X$ pulvérisé par $\mathcal{F}$.
- Fonctions linéaires : en dimension $d$, VC-dimension de $d+1$
- Borne PAC : $e(f) \leq e_D(f) + \sqrt{\frac{VC(\mathcal{F})(ln\frac{2m}{VC(\mathcal{F})}+1+ln\frac{4}{\delta})}{m}}$

# En pratique . . .

## Régularisation !

- En ne regardant que l'erreur empirique $\Rightarrow$ sur-apprentissage
- Possible d'avoir une estimation du biais et de la variance dans des situations simples (histogramme, régression, . . . ) mais nécessite beaucoup d'évaluations et peu pratique

$\Rightarrow$ Modification pour l'estimation du risque empirique : introduction de la pénalisation

## Deux types de régularisation

- Ivanov : $\hat{f} = argmin_f\ L(f)$tq $\Omega(f) \leq \delta$
  Contrainte sur l'espace de recherche de la fonction $f$
- Thikonov : $\hat{f} = argmin_f\ L(f) + \Omega(f)$
  Compromis entre expressivité et erreur empirique.
- Si $L$ et $\Omega$ en norme de degré 2 $\Rightarrow$ ridge regression, ridge classification, . . .
- Si $L$ en norme 2 et $\Omega$ en norme 1 ou inversement $\Rightarrow$ LASSO
- Si $L$ en norme 1 et $\Omega$ en norme 2 $\Rightarrow$ SVM linéaire