

Théorie de l'apprentissage

Cours 10 (b)

Nicolas Baskiotis

`nicolas.baskiotis@lip6.fr`

Master 1 DAC

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

S2 (2015-2016)

Plan

1 Interlude théorique

Apprentissage supervisé et risque

Problématique de l'apprentissage supervisé

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

Minimisation du risque et risque bayésien

- Minimisation du risque

$$\begin{aligned} R_{\ell, P}(f) &= \int_{\mathcal{X} \times \mathcal{Y}} \ell(f(x), y) dP(x, y) = \int_{\mathcal{X} \times \mathcal{Y}} \ell(f(x), y) p(x, y) dx dy \\ &= \mathbb{E}[\ell(Y, f(X))] \end{aligned}$$

- Risque empirique sur les données $|D| = n : \frac{1}{n} \hat{R}_{n, \ell, P} \sum_{i=1}^n \ell(f(x^i), y^i)$
- Risque bayésien : $R_{\ell, P}^*(f) = \inf_{f: \mathcal{X} \rightarrow \mathcal{Y}} \int_{\mathcal{X} \times \mathcal{Y}} \ell(f(x), y) dP(x, y)$
- Objectif : à partir de données D trouver une fonction dont le risque est proche de l'optimal.

Consistance d'un algorithme

Définition

- Le risque : variable aléatoire

$$R_{\ell,P}(f) = \mathbb{E}[\ell(Y, f_D(X)|D)]$$

- 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 $|D| \rightarrow \infty$: $\lim_{|D| \rightarrow \infty} R_{\ell,P}(f) \rightarrow R_{\ell,P}^*$

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;

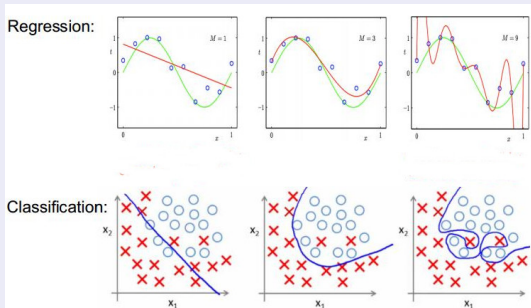
Exemple dans le cas d'attributs discrets ?

Risque empirique

Comment évaluer le risque ?

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

Mais sur-apprentissage ...

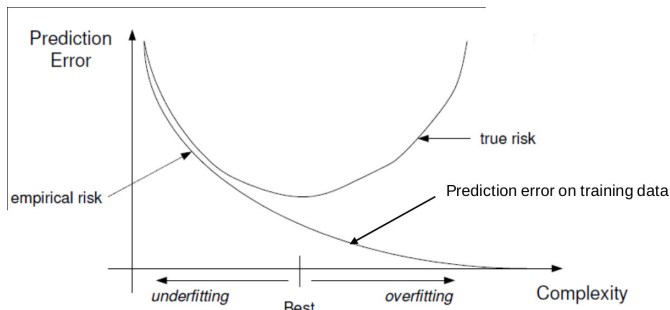


Solution : **restreindre** la famille de fonctions considérée : $f = \inf_{f \in \mathcal{F}} \hat{R}_n(f)$

Minimisation structurelle du risque

Principe

- On ne considère que les fonctions dans une famille \mathcal{F} :
$$f = \operatorname{argmin}_{f \in \mathcal{F}} \hat{R}_n(f) = \operatorname{argmin}_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \ell(y^i, f(x^i))$$
- vrai risque sur \mathcal{F} : $R_{\mathcal{F}}^*$
- risque empirique sur \mathcal{F} : $\hat{R}_{n, \mathcal{F}}^*$
- Condition nécessaire : $R_{\mathcal{F}}^* - R^* \geq 0$ 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 ($\ell(y, f(x)) = 1_{f(x) \neq y}$)
- $R^* = \inf_f P(y \neq f(x))$
- $f^* = \operatorname{arginf}_f P(y \neq f(x))$
- Propriétés :
 - ▶ $P(y \neq f^*(x)) \leq P(y \neq f(x))$ pour tout f
 - ▶ $f^* = \begin{cases} 1 & \text{si } \eta(x) > 1/2 \\ 0 & \text{si } \eta(x) \leq 1/2 \end{cases}$ avec $\eta(x) = \mathbb{E}[Y = 1|x]$

Notations

$$\begin{array}{l} R(f) = P(y \neq f(x)) \\ \hat{R}(f) = \frac{1}{n} \sum_i 1_{f(x_i) \neq y_i} \end{array} \left| \begin{array}{l} R^* = R(f^*) = \inf_f R(f) \\ R_{\mathcal{F}}^* = R(f_{\mathcal{F}}^*) = \inf_{f \in \mathcal{F}} R(f) \\ \hat{R}_{n, \mathcal{F}}^* = \inf_{f \in \mathcal{F}} \hat{R}_n(f) \end{array} \right| \begin{array}{l} f^* = \operatorname{arginf}_f R(f) \\ f_{\mathcal{F}}^* = \operatorname{arginf}_{f \in \mathcal{F}} R(f) \\ f_{n, \mathcal{F}}^* = \operatorname{argmin}_f \hat{R}_n(f) \end{array}$$

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

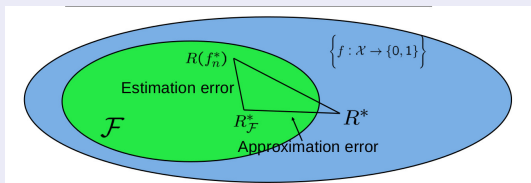
Erreurs d'estimation/d'approximation

Décomposition de l'erreur

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

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

Quelques théorèmes

- A combien est-on de l'optimal dans \mathcal{F} :

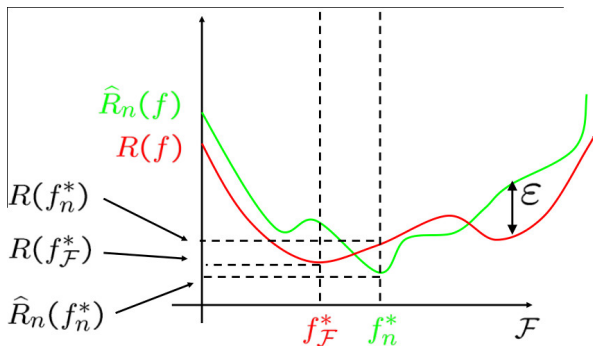
$$|R(f_{n,\mathcal{F}}^*) - R_{\mathcal{F}}^*| \leq 2 \sup_{f \in \mathcal{F}} |\hat{R}_n(f) - R(f)|$$

- A combien le risque empirique est éloigné du vrai risque :

$$|\hat{R}_n(f_{n,\mathcal{F}}^*) - R(f_n^*)| \leq \sup_{f \in \mathcal{F}} |\hat{R}_n(f) - R(f)|$$

⇒ Risque empirique par rapport au meilleur classifieur dans \mathcal{F} :

$$|\hat{R}_n(f_{n,\mathcal{F}}^*) - R_{\mathcal{F}}^*| \leq 3 \sup_{f \in \mathcal{F}} |\hat{R}_n(f) - R(f)|$$



Un cas concret : espace de recherche fini

Contexte

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

Variable de Bernouilli

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

Un cas concret : espace de recherche fini

Borne sur l'erreur

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

- $|\mathcal{F}_c|$: nombre de fonctions sans erreurs sur D
- Quelle est la probabilité de choisir une mauvaise, c'est-à-dire $e(f) \geq \epsilon$?

$$P(e(f_1) \geq \epsilon \text{ ou } e(f_2) \geq \epsilon \text{ ou } \dots \text{ ou } e(f_{|\mathcal{F}_c|}) \geq \epsilon) \leq \sum_k P(e(f_k) \geq \epsilon) \\ \geq \sum_k (1 - \epsilon)^m \geq |\mathcal{F}| (1 - \epsilon)^m \geq |\mathcal{F}| e^{-n\epsilon}$$

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

Cadre PAC : Probably Approximately Correct

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

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

- $P(e(f) - e_D(f) \geq \epsilon) \leq |\mathcal{F}|e^{-2m\epsilon^2}$
- $e(f) \leq e_D(f) + \sqrt{\frac{\ln|\mathcal{F}| + \ln\frac{1}{\delta}}{2m}}$
- Décomposition biais/variance ...
- Pour δ, ϵ 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 :
$$e(f) \leq e_D(f) + \sqrt{\frac{VC(\mathcal{F})(\ln \frac{2m}{VC(\mathcal{F})} + 1 + \ln \frac{4}{\delta})}{m}}$$