Introduction à l'apprentissage supervisé
Un peu de théorie — exemple de la classification binaire

Geneviève Robin

Plan

Objectifs du cours

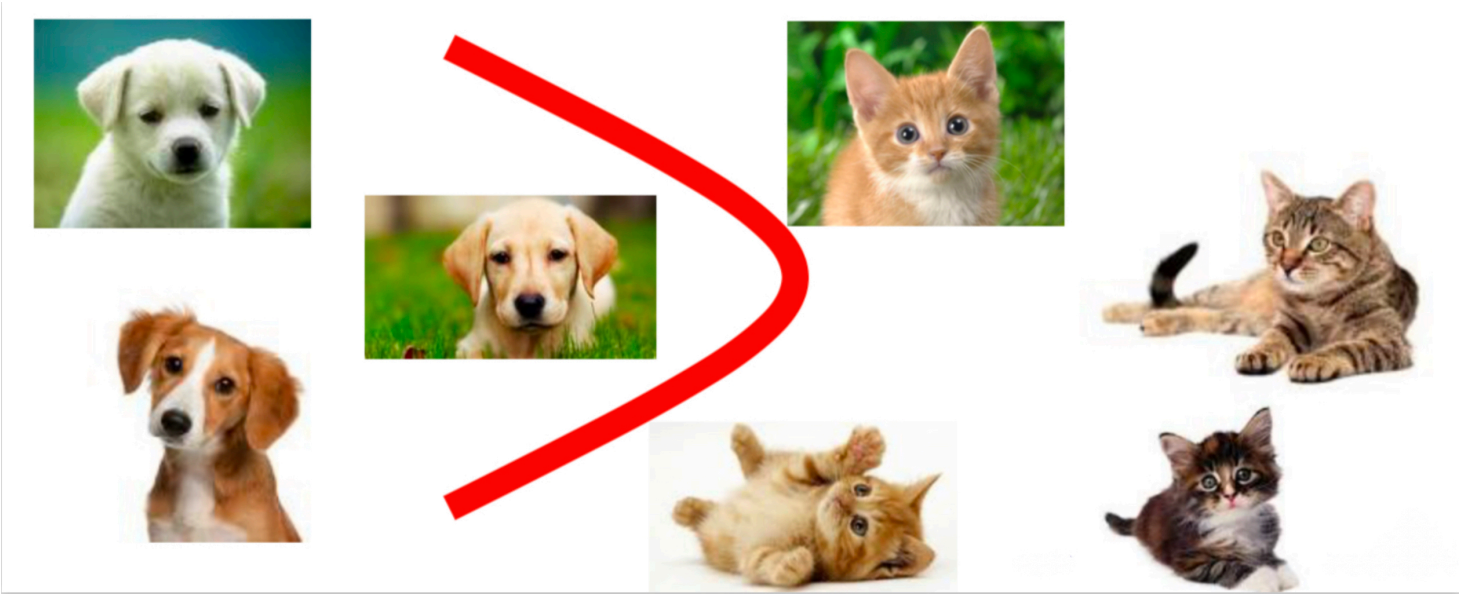
- Introduire quelques notions théoriques aux fondements de l'apprentissage statistique.
- Approfondir la formulation mathématique des problèmes d'apprentissage pour mieux comprendre chaque problème.
- Comprendre les notions clés régissant les méthodes d'apprentissage classiques et leurs performances théoriques.

Exemple de la classification binaire (supervisée)

Principe général

On cherche une fonction!

- Exemple : Distinguer les chats des chiens dans des images



- Dans quel **espace** cherche-t-on cette fonction ?

Modèle de la classification binaire (supervisée)

- Jeu de données : $\{(X_i, Y_i)\}_{1 \leq i \leq n}$ où
  - X_i encode une image (pixels, features, etc.)
  - Y_i est une étiquette binaire (chat ou chien)

- Vision probabiliste des couples (X_i, Y_i)

1. (X, Y) est une variable aléatoire de distribution P sur $\mathbb{R}^d \times \{0, 1\}$.
2. Données $\{(X_i, Y_i)\}_{1 \leq i \leq n}$ i.i.d. de distribution P.

Règles de décision

- Une règle de décision en classification binaire supervisée est une fonction

$$g : \mathbb{R}^d \to \{0, 1\}$$

aussi appelée **classifieur**.

- Exemples de classifieurs :
  - Classifieurs linéaires (e.g. régression logistique)

$$g_{\beta_0,\beta}(x) = \mathbb{I}(\beta^\top x + \beta_0 \geq 0),$$

avec $\theta \in \mathbb{R}^d$ et $\theta_0 \in \mathbb{R}$.

  - En règle générale

$$g_f(x) = \mathbb{I}(f(x) \geq 0),$$

où $f : \mathbb{R}^d \to \mathbb{R}$ peut être modélisée par régression logistique, abres de décisions, SVM, etc.

Plan

Introduction intuitive

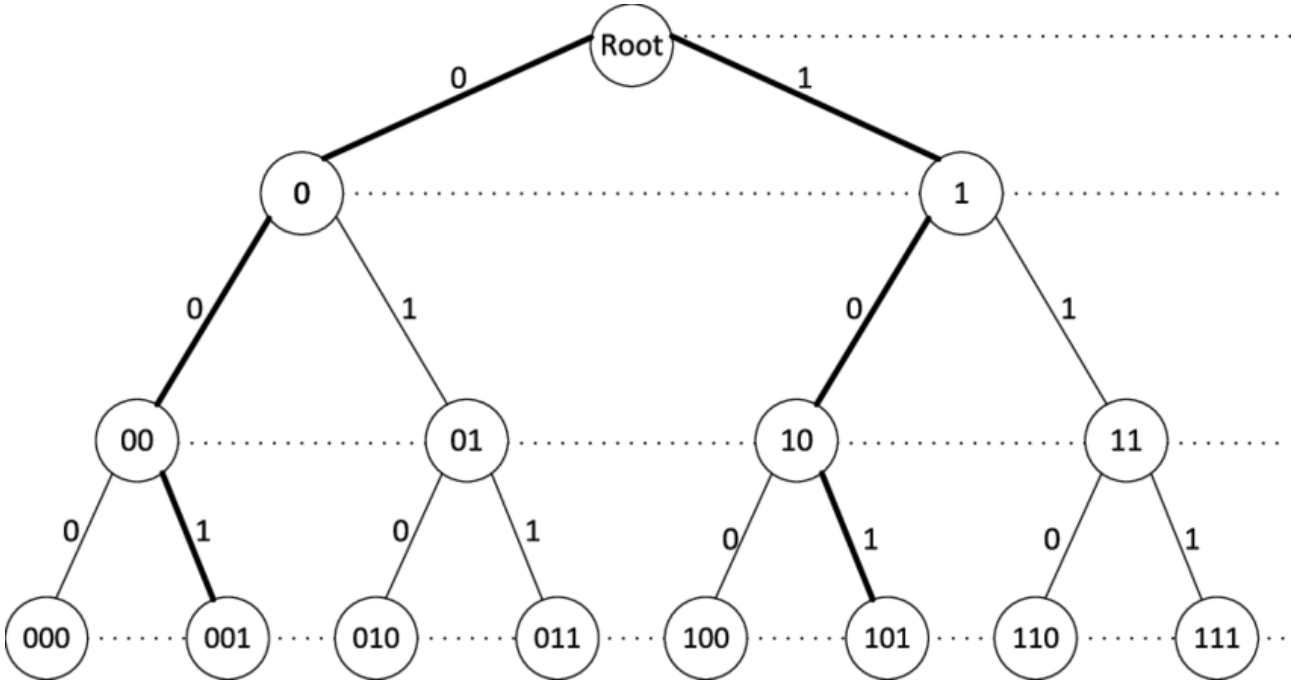
- On suppose que la “Nature” a choisi une fonction (un classifieur) parmi un ensemble fini de K fonctions.

- Supposons que nous ayons à disposition un oracle qui répond “oui” ou “non” lorsqu'on pose une question à propos de cette fonction.

  Quel est le nombre optimal n de questions à poser pour découvrir cette fonction inconnue ?

Réponse grâce à l'Information de Shannon

- L'entropy est le nombre de bits nécessaire pour coder un ensemble de K symboles (ici K classifieurs)



- Le nombre de questions optimal est donné par $n = \frac{\log(K)}{\log(2)}$.

Application du principe à un jeu de données

- Soit un espace de features $\mathcal{X}$ et un espace de label $\mathcal{Y} = \{0, 1\}$
- **Question :** De combien de données $(X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$ a-t-on besoin pour trouver une fonction donnée parmi un ensemble de K fonctions $f : \mathcal{X} \to \{0, 1\}$?
- Le nombre d'exemples nécessaire est $n = \frac{\log K}{\log 2}$
- **Procédure :** On cherche X_i tel que la moitié des K fonctions prend la valeur 1 et l'autre moitié 0, puis on demande à l'oracle si la fonction recherchée prend la valeur 0 ou 1 en X_i. Puis on applique cela n fois.

Apprentissage Probablement Approximativement Correct (PAC)

- En pratique il n'est pas facile de trouver un tel X_i qui permette de séparer l'ensemble de fonctions en deux.

- **Nouvelle question :** De combien de données (X_i, Y_i) a-t-on besoin pour trouver une fonction donnée f_0 parmi un ensemble de K fonctions $f_k : \mathcal{X} \to \{0, 1\}$, telle qu'avec probabilité au moins $1 - \delta$ (probablement) la fonction $\hat{f}$ trouvée est ε-proche de f_0 (approximativement) ?

- Il faut un nombre d'exemples de l'ordre de

$$n = \frac{\log K - \log \delta}{\varepsilon}.$$

Plan

Exemple de la classification binaire

- (X, Y) un coupe de variables aléatoires de distribution P sur $\mathbb{R}^d \times \{-1, +1\}$

1. **Point de vue génératif** — Distribution jointe P est une mixture
   - Densités conditionnellement à la classe f_+ et f_-
   - Paramètre de mixture $p = \mathbb{P}\{Y = +1\}$

2. **Point de vue discriminatif** — Distribution jointe P décrite par (P_X, η)
   - Distribution marginale $X \sim P_X = df_X/d\lambda_d$
   - Fonction de probabilité postérieure

$$\eta(x) = \mathbb{P}\{Y = 1 | X = x\}, \forall x \in \mathbb{R}^d.$$

Classifieur, mesure d'erreur, optimalité

- Classifieur $g : \mathbb{R}^d \to \{-1, +1\}$
- Erreur de classification $L(g) = \mathbb{P}\{g(X) \neq Y\}$

$$L(g) = \mathbb{E}\left(\eta(X)\mathbb{I}\{g(X) = -1\} + (1 - \eta(X))\mathbb{I}\{g(X) = 1\}\right)$$

- Règle de Bayes $g^*(x) = 2\mathbb{I}\{\eta(x) > 1/2\} - 1$, $\forall x \in \mathbb{R}^d$
- Erreur du classifieur de Bayes : $L^* = L(g^*) = \mathbb{E}\left\{\min(\eta(X), 1 - \eta(X))\right\}$
- Risque excédentaire

$$L(g) - L^* = 2\mathbb{E}\left\{\left|\eta(X) - \frac{1}{2}\mathbb{I}\left\{g(X) \neq g^*(X)\right\}\right|\right\}$$

Lien avec l'estimation paramétrique

- Soit $\hat{\eta}$ un estimateur de η basé sur un échantillon D_n
- On considère $\hat{g}$ l'estimateur plug-in construit à partir de $\hat{\eta}$

$$\hat{g}(x) = 2\mathbb{I}\{\hat{\eta}(x) > 1/2\} - 1$$

- On a, conditionnellement à l'échantillon D_n

$$L(\hat{g}) - L^* \leq 2\mathbb{E}_X\left(|\hat{\eta}(X) - \eta(X)|\right)$$

- Problème, l'estimation de η est difficile lorsque les données sont de grande dimension.

Convexification du risque

- Données de classification binaire avec $Y \in \{-1, +1\}$
- Règle de décision à valeur réelle $f : \mathbb{R}^d \to \mathbb{R}$
- Fonction de coût $\varphi : \mathbb{R} \to \mathbb{R}_+$ convexe, croissante, $\varphi(0) = 1$
- φ-risque moyen

$$A(f) = \mathbb{E}(\varphi(-Yf(X)))$$

- Principaux exemples :

$$\varphi(x) = \mathrm{e}^x, \log_2(1 + \mathrm{e}^x), (1 + x)_+$$

- Remarque : $L(\mathrm{sgn}(f)) \leq A(f)$

Plan

Erreur empirique / erreur de généralisation

- On suppose que les couples (X_i, Y_i) sont des copies i.i.d. de (X, Y) de loi $\mathcal{L}$ inconnue
- on note $\mathcal{D}_n = \Big\{(X_1, Y_1), \ldots, (X_n, Y_n)\Big\}$

On se donne une fonction classifiante déterministe $c : \mathbb{R}^d \in \mathcal{C}$, on définit

Erreur empirique ou erreur visible

$$R_n(c) = \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, c(X_i)).$$

Erreur de généralisation

$$R(c) = \mathbb{E}_{\mathcal{L}}\Big(\ell(Y_+, c(X_+))\Big)$$

où (X_+, Y_+) est un couple indépendant de $\mathcal{D}_n$

Remarques

- En classification, on prend souvent $\ell(y, y') = \mathbb{1}_{y \neq y'}$, dans ce cas $1 - R_n(c)$ est appelé "accuracy" de c.
- On a
$$R(c) = \mathbb{E}_{\mathcal{L}}\big(R_n(c)\big).$$
- On voudrait retrouver
$$c^\star = \operatorname*{argmin}_c R(c)$$

Classifieur bayésien

$c^\star = \operatorname{argmin}_c R(c)$ est, dans le cas de la classification et de la perte 0/1, le classifieur bayésien.

Mais on se restreint le plus souvent à un sous-ensemble $\mathcal{G}$ (par exemple les fonctions constantes sur une partition $\mathcal{A}$)

$$c_{\mathcal{G}}^{\text{oracle}} = \operatorname*{argmin}_{c \in \mathcal{G}} R(c)$$

puis, comme la loi $\mathcal{L}$ est inconnue, on remplace R par R_n

$$\hat{c}_{\mathcal{G}} = \operatorname*{argmin}_{c \in \mathcal{G}} R_n(c).$$

On a bien sûr

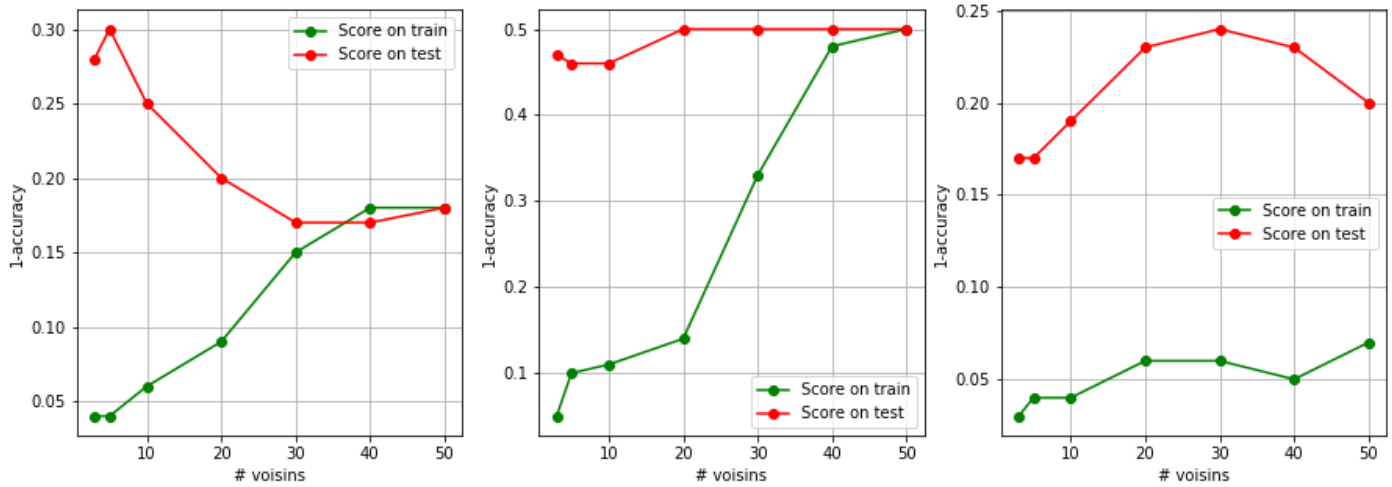
$$R(\hat{c}_{\mathcal{G}}) \geq R(c_{\mathcal{G}}^{\text{oracle}}) \geq R(c^\star).$$

Plan

Erreur visible / erreur de généralisation

Ce que l'on veut comparer

On veut comparer $R_n(\hat{c}_{\mathcal{G}})$ et $R(c^\star)$ pour mesurer "l'optimisme" quand on calcule $R_n(\hat{c}_{\mathcal{G}})$.



Première borne de risque

On montre que, avec probabilité plus grande que $1 - \varepsilon$

$$R(\hat{c}_{\mathcal{G}}) \leq R(c^{\star}) + \text{erreur d'approximation} + \sqrt{\frac{2\log(2|\mathcal{G}|\varepsilon^{-1})}{n}}$$

Borne "risque visible/erreur de généralisation"

On montre que, avec probabilité plus grand que $1 - \varepsilon$

$$R(\hat{c}_{\mathcal{G}}) \leq R_n(\hat{c}_{\mathcal{G}}) + \sqrt{\frac{2\log(2|\mathcal{G}|\varepsilon^{-1})}{n}}$$

Borne sur l'erreur de généralisation

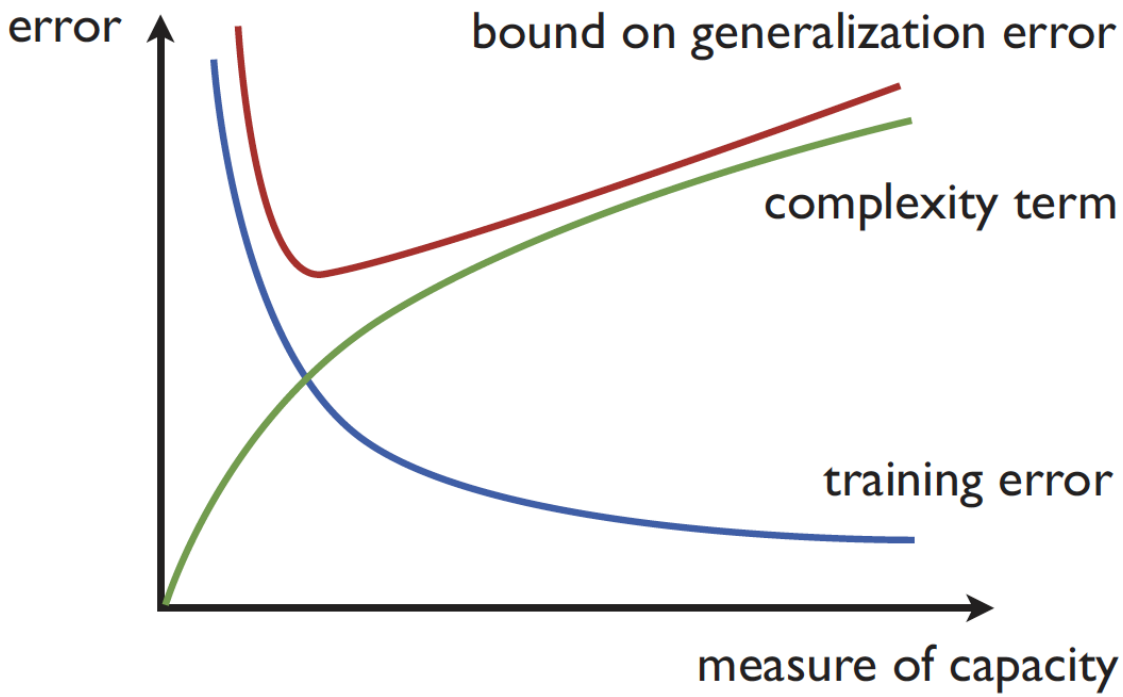


Figure 1: In **mohri2012foundations**

Objectifs principaux du cours

- **Consistence** d'une suite de règles de décision $\left(\hat{f}\right)_{n\geq 1}$:

$$L(\hat{f}_n) \to L^* \text{ en probabilité lorsque } n \to \infty.$$

- **Bornes supérieures** : Soit $\hat{f}_n \in \mathcal{F}$. Avec probabilité au moins $1-\delta$, il existe une constante c telle que

$$L(\hat{f}_n) - \inf_{\mathcal{F}} L \leq C(\mathcal{F}, n) + c\sqrt{\frac{\log(1//delta)}{n}},$$

où $C(\mathcal{F}, b) = \mathcal{O}(1/\sqrt{n})$

Plan

Principe de l'algorithme kNN

1. Calcul des distances
   - Calcul des distances entre paires $d(x, X_i)$ pour $i = 1, \ldots, n$.
2. Entraînement
   - Tri des données de la plus proche de x, $X_{(1)}$ à la plus éloignée $X_{(n)}$
3. Prédictioin $\hat{h}(x, k) =$ Vote majoritaire des k plus proches voisins $X_{(1)}, \ldots, X_{(k)}$
   - On considère les labels $Y_{(1)}, \ldots, Y_{(k)}$ et on prend le vote majoritaire :

$$\hat{h}(x, k) = \text{argmax}_c \{ \sum_{l=1}^{k} \mathbb{I}\{Y_{(l)} = c\} \}.$$

Hyperparamètres du modèle : nombre de voisin k, distance d.

Théorie pour les kNN

- Rappel : l'erreur de classification s'écrit $L(h) = \mathbb{P}(Y \neq h(X))$ et $L^* = \inf L$.

- Résultat de consistence :

$$\mathbb{E}L(\hat{h}(/cdot, k)) \to L^*$$

  sous les conditions $k_n \to \infty$ et $k_n/n \to 0$ lorsque $n \to \infty$

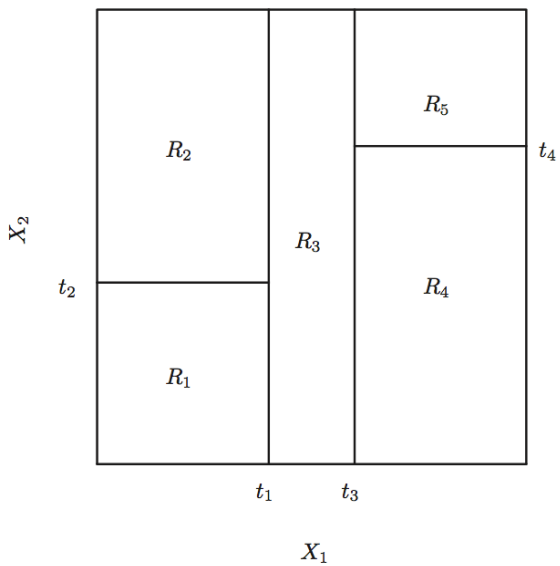
- Pas de forme close pour la valeur optimale de k_n : en pratique on fait de la validation croisée.

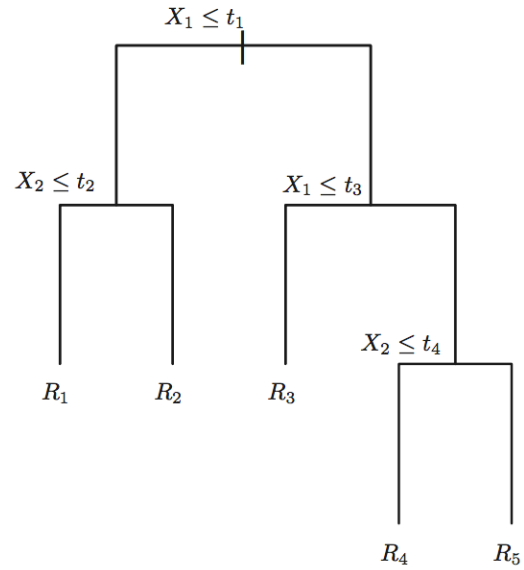
- Pas de résultat théorique en général concernant le choix de la distance d.

Arbres de décision

On dénote la partition par $c = \bigcup_j \gamma_j$ avec les cellules γ_j

- On cherche la cellule $\gamma(x)$ où x se trouve
- On considère les données d'entraînement se trouvant dans la cellule $\gamma(x)$
- On prédit $\hat{h}(x, c) =$ Vote majoritaire sur les données d'apprentissage de la cellule $\gamma(x)$





Consistence des arbres de décision

- Dans le cas d'une partition régulière composée de cellules hypercubiques de $\mathbb{R}^d$ d'arêtes de longueur δ_n :

$$\mathbb{E}L(\hat{h}(\cdot, \delta_n)) \to L^*$$

  Sous condition que $n\delta_n^d \to \infty$ et $\delta_n \to 0$ lorsque $n \to \infty$ (il faut suffisamment de point dans chaque cellule et la taille de chaque cellule doit tendre vers 0).

- Dans le cas d'une partition apprise à partir des données, il existe d'autres résultats théoriques qui sortent du cadre de ce cours (théorie VC et Rademacher).

Plan