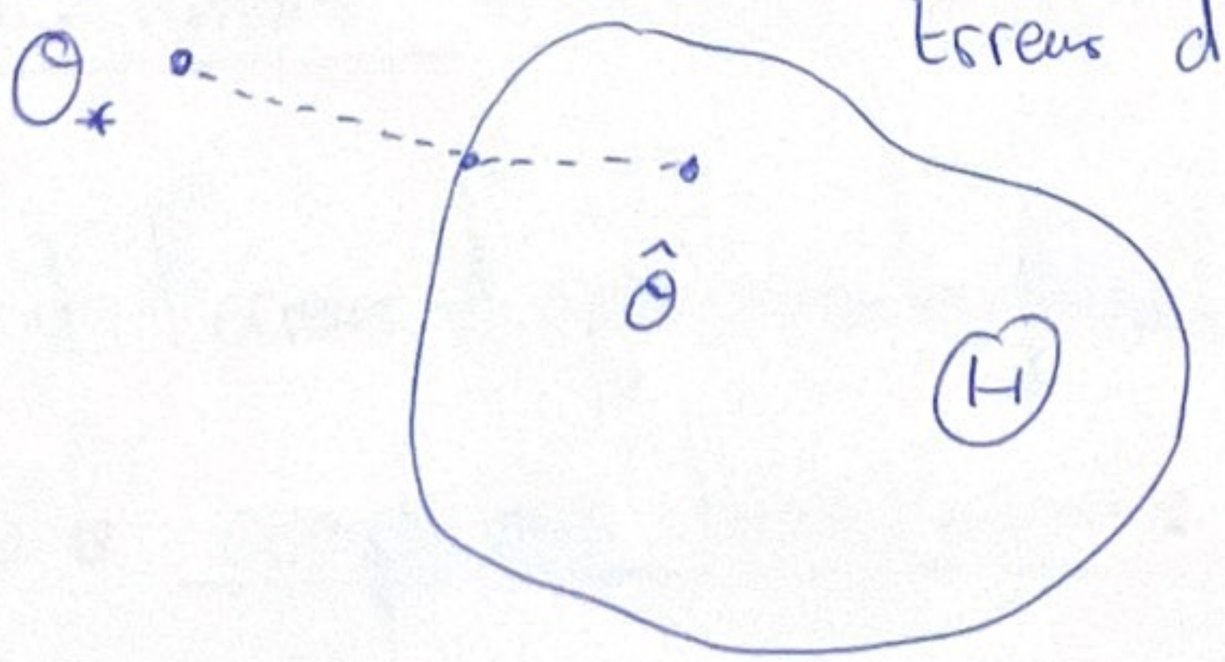
Sélection de Modèles et Méthodes Parcimonieuses

I. Décomposition du risque

Soit $\mathcal{H}$ une famille de paramètres, et soit $\hat{\theta} \in \mathcal{H}$ un estimateur.

$$\underbrace{R(f_{\hat{\theta}}) - R^*}_{\text{Risque de l'estimateur}} = \underbrace{\left\{ R(f_{\hat{\theta}}) - \inf_{\theta' \in \mathcal{H}} R(f_{\theta'}) \right\}}_{\text{Erreur d'estimation}} + \underbrace{\left\{ \inf_{\theta' \in \mathcal{H}} R(f_{\theta'}) - R^* \right\}}_{\text{Erreur d'approximation}}$$

Risque de Bayes



1) L'erreur d'estimation

• Elle caractérise à quel point le risque de notre estimateur est proche du meilleur estimateur de la classe.

• Elle est reliée à la variance de l'estimateur.

Plus (H) est compliquée, plus il faudra de points pour que cette erreur soit petite.

2) L'erreur d'approximation

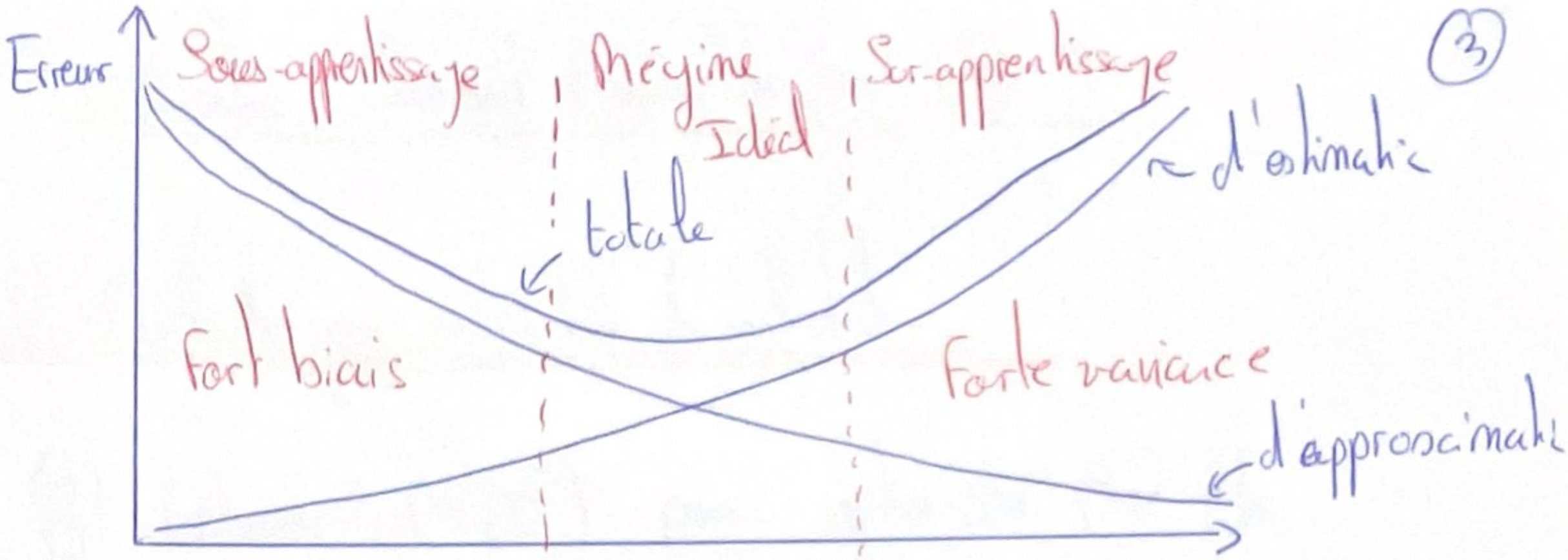
• Modélise à quel point (H) modélise la réalité fidèlement.

• Plus (H) est grande, plus l'erreur d'approximation sera faible.

• Il s'agit d'un terme de biais

3) Compromis Biais / Variance

L'erreur d'estimation et l'erreur d'approximation forment un compromis appelé le compromis biais/variance



(3)

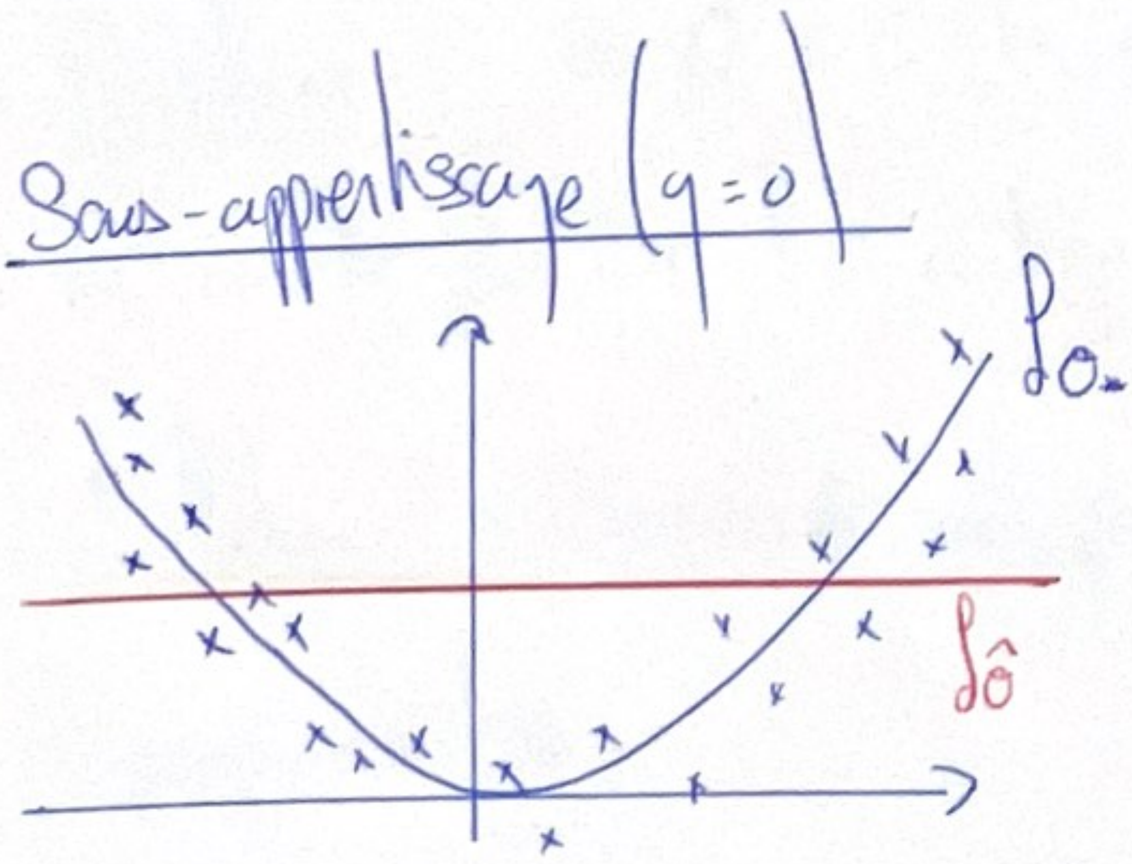
4) Régression Polynômiale

Vérité : $y = x^2 + \text{bruit}$.

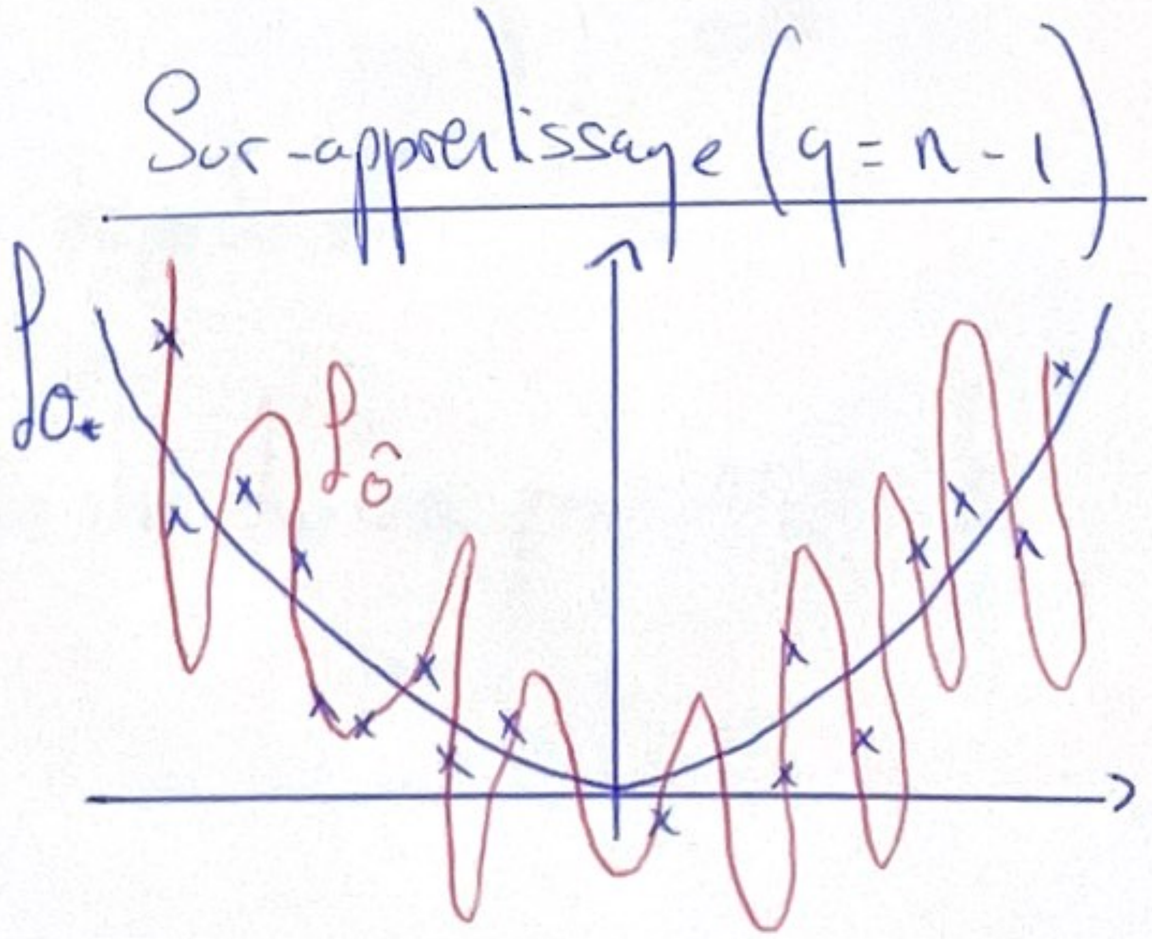
$$\hat{\theta} \in \operatorname{argmin} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \theta^T \varphi_q(x_i) \right)^2$$

où $\varphi_q(x) = \begin{pmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^q \end{pmatrix}$

Sous-apprentissage ($q = 0$)



Sur-apprentissage ($q = n - 1$)



Question : Comment trouver un bon (H) ?

II Méthodes par ensemble de validation

Estimer $R(\hat{\theta})$ puis optimiser (H) (les "hyperparamètres".

1) Découpage Train / Val

$(x,y), (x_1,y_1), (x_2,y_2), \dots, (x_n,y_n)$ i.i.d $\mathbb{P}$.

On peut estimer

$R(f_{\hat{\theta}}) = E_{(x,y)}(\ell(y, f_{\hat{\theta}}(x)))$ par $\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f_{\hat{\theta}}(x_i))$.

Problème : si $\hat{\theta} = \hat{\theta}((x_1,y_1), \dots, (x_n,y_n))$, cette idée ne marche plus car il y a trop de dépendances.

Solution : Réserver une partie des données pour régler les hyperparamètres.

$\underbrace{(x_1, y_1), \dots, (x_{|\text{Entrainement}|}, y_{|\text{Entrainement}|})}_{\text{Jeu de données d'entraînement}} +$

$(x_{|\text{Entrainement}|+1}, y_{|\text{Entrainement}|+1}), \dots, (x_{|\text{Entrainement}|+|\text{Validation}|}, y_{|\text{Entrainement}|+|\text{Validation}|})$

Jeu de données de validation.

Maintenant si $\hat{\theta} = \hat{\theta}$(Jeu de données d'entraînement)

Alors $\hat{\theta}$ $\perp$ Jeu de données de validation.

On a alors $R(f_{\hat{\theta}}) \approx \frac{1}{|\text{Validation}|} \sum_{(x,y) \in \text{Validation}} P(y, f_{\hat{\theta}}(x))$ (*)

Pour régler les hyperparamètres, on peut donc

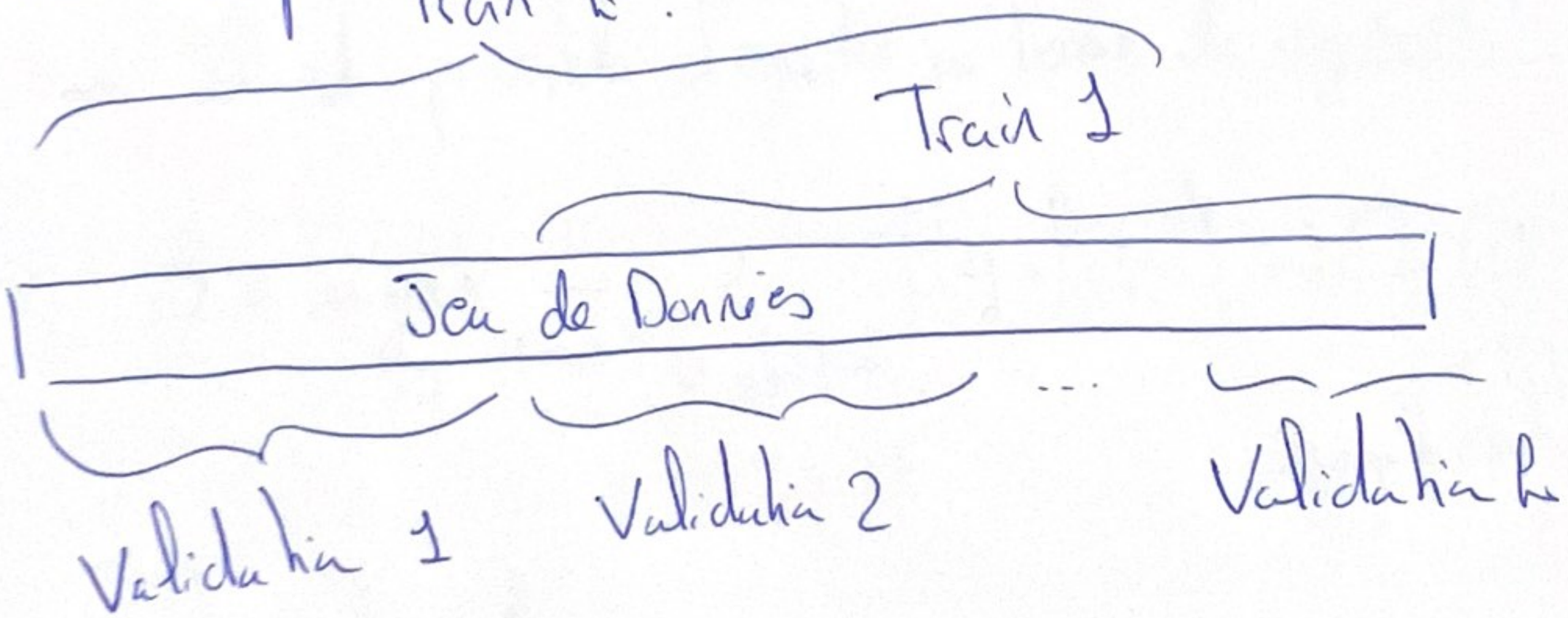
(i) Estimer $\hat{\theta}$ pour plusieurs valeurs d'hyperparamètres

(ii) Sélectionner les hyperparamètres qui minimisent (*).

Remarque: Il est aussi bon de réserver une partie des données, appelée ensemble de test, pour tester ~~les données~~ à la toute fin.

2) Validation Croisée

L'idée de la validation croisée est de stabiliser l'étape de spliting du jeu de données en moyennant sur plusieurs découpes. Train k.



III. Méthodes par pénalisation (Régularisation)

L'erreur d'entraînement se comporte à peu près comme l'erreur d'approximation, elle tend vers 0 quand la "complexité" de $\mathcal{H}$ tend vers $+\infty$.

$\Rightarrow$ L'erreur d'entraînement capture mal l'explosion de la variance liée à la complexification

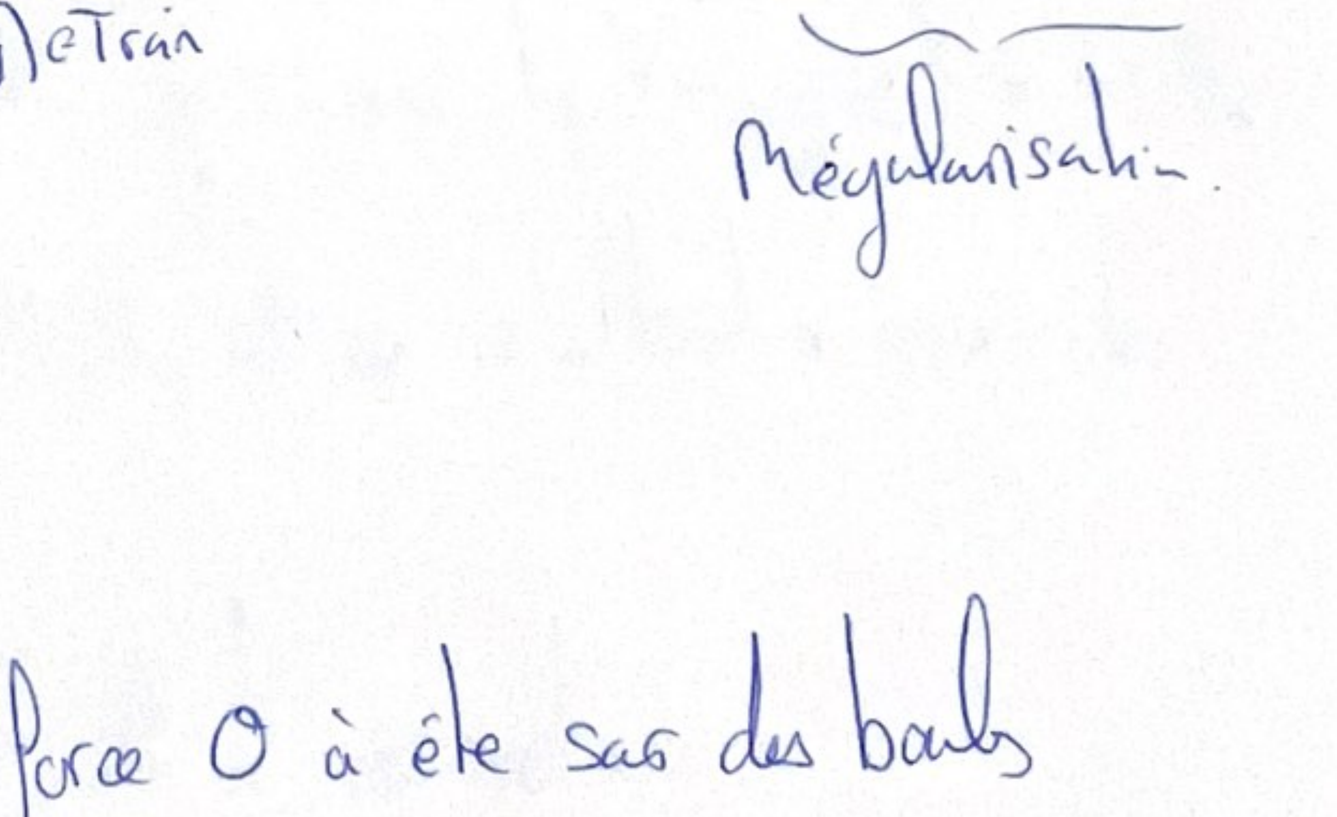
$\Rightarrow$ Il est possible d'ajouter un terme de régularisation

$$\hat{\theta} \in \underset{\theta}{\operatorname{argmin}} \frac{1}{|Train|} \sum_{(x,y) \in Train} P(y, f_\theta(x)) + \lambda \Omega(\theta)$$

Régularisation.

1) Régularisation l_2

$\Omega(\theta) = \frac{1}{2}\|\theta\|^2$ $\Rightarrow$ force θ à être sur des boules de petite norme



Exemple: Régression Ridge

Weight Decay dans les réseaux de neurones.

2) Régularisation l_0 :

$\Omega(\theta) = \|\theta\|_0$ = nb de composantes de θ non nulles.

=> Force θ à avoir peu de coefficients non nuls.

=> Réduit le nombre de variables explicatives.

⚠ Problèmes difficiles à résoudre numériquement

3) Régularisation l_1 :

💡 $\|.\|_1$ = meilleure approximation convexe de $\|.\|_0$

= plus grande fonction g convexe t.q $\forall x, g(x) \leq \|x\|_0$.

=> On remplace $\|.\|_0$ par $\|.\|_1$, et on s'attend à avoir des effets similaires.

Le problème est maintenant "facile à résoudre".