

A2DI: Théorie de l'apprentissage

John Klein

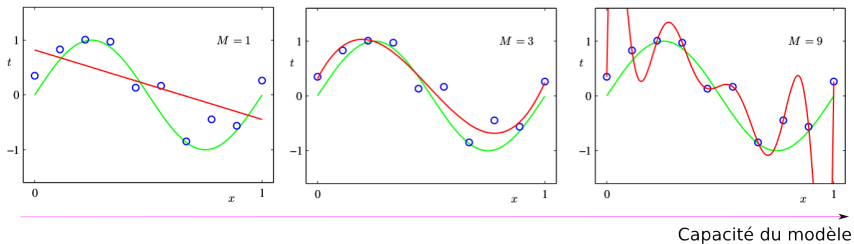
Lille1 Université - CRIStAL UMR CNRS 9189



La **théorie** de l'apprentissage cherche à répondre à 2 questions ?

- Peut-on apprendre ? (à quelles **conditions**)
- Si oui, le fait-on bien ? (**évaluation**)

Vers un juste milieu entre l'over et l'under-fitting.



[Bishop 2006]

Comment détecter l'over et l'under-fitting ?

- Il faut déjà s'accorder sur ce que se tromper veut dire.
- Soit f la fonction apprise par l'algorithme.
- On introduit une fonction de perte (loss) L :

Exemple en classification : **0-1 Loss**

$$L : \mathcal{C} \times \mathbb{X} \rightarrow \{0; 1\}, \quad (1)$$

$$(c^{(i)}, f(\mathbf{x}^{(i)})) \rightarrow \begin{cases} 0 & \text{si } c^{(i)} = f(\mathbf{x}^{(i)}) \\ 1 & \text{sinon} \end{cases}, \quad (2)$$

avec $f(\mathbf{x}^{(i)})$ la classe prédite par f pour l'exemple $\mathbf{x}^{(i)}$ dont la vraie classe est c_i .

Comment détecter l'over et l'under-fitting ?

- Il faut déjà s'accorder sur ce que se tromper veut dire.
- Soit f la fonction apprise par l'algorithme.
- On introduit une fonction de perte (loss) L :

Exemple en régression : perte en norme 2, **Quadratic Loss**

$$L : \mathbb{Y} \times \mathbb{X} \rightarrow [0; +\infty], \quad (3)$$

$$\left(y^{(i)}, f \left(\mathbf{x}^{(i)} \right) \right) \rightarrow \left(y^{(i)} - f \left(\mathbf{x}^{(i)} \right) \right)^2, \quad (4)$$

avec $f \left(\mathbf{x}^{(i)} \right)$ la valeur prédite par f pour l'exemple $\mathbf{x}^{(i)}$ dont la vraie valeur associée est $y^{(i)}$.

Comment détecter l'over et l'under-fitting ?

- Il faut déjà s'accorder sur ce que se tromper veut dire.
- Soit f la fonction apprise par l'algorithme.
- On introduit une fonction de perte (loss) L :

Exemple en régression : perte en norme 1, **Absolute Loss**

$$L : \mathbb{Y} \times \mathbb{X} \rightarrow [0; +\infty], \quad (5)$$

$$\left(y^{(i)}, f \left(\mathbf{x}^{(i)} \right) \right) \rightarrow \left| y^{(i)} - f \left(\mathbf{x}^{(i)} \right) \right|, \quad (6)$$

avec $f \left(\mathbf{x}^{(i)} \right)$ la valeur prédite par notre algorithme pour l'exemple $\mathbf{x}^{(i)}$ dont la vraie valeur associée est $y^{(i)}$.

Comment détecter l'over et l'under-fitting ?

- On ne peut pas supposer que nos données $\mathcal{D} = \left\{ \left(\mathbf{x}^{(i)}, y^{(i)} \right) \right\}_{i=1}^n$ représentent tous les cas possibles.
- En revanche, on peut supposer que \mathcal{D} est iid selon la loi génératrice des données $p_{X,Y}$.
- Notre objectif sera de minimiser la perte attendue sous $p_{X,Y}$ ou erreur de généralisation :

$$Err_{\text{gen}}(f) = \mathbb{E}_{X,Y} [L] = \int_{\mathbb{X}} \int_{\mathbb{Y}} L(y, f(\mathbf{x})) dp_{X,Y}. \quad (7)$$

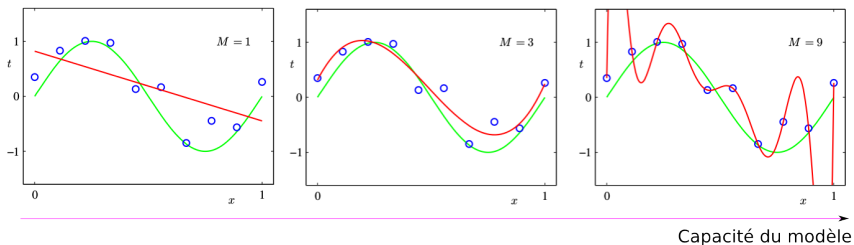
Comment détecter l'over et l'under-fitting? → en estimant Err_{gen}

- La perte attendue ne peut être calculée explicitement car la distribution *de la nature* est bien sûr inconnue.
- On peut en revanche, dans le cas supervisé, calculer la perte empirique, ou erreur de train

$$Err_{\text{train}}(f, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(\mathbf{x}_i)). \quad (8)$$

Comment détecter l'**over** et l'**under-fitting**? → en estimant Err_{gen}

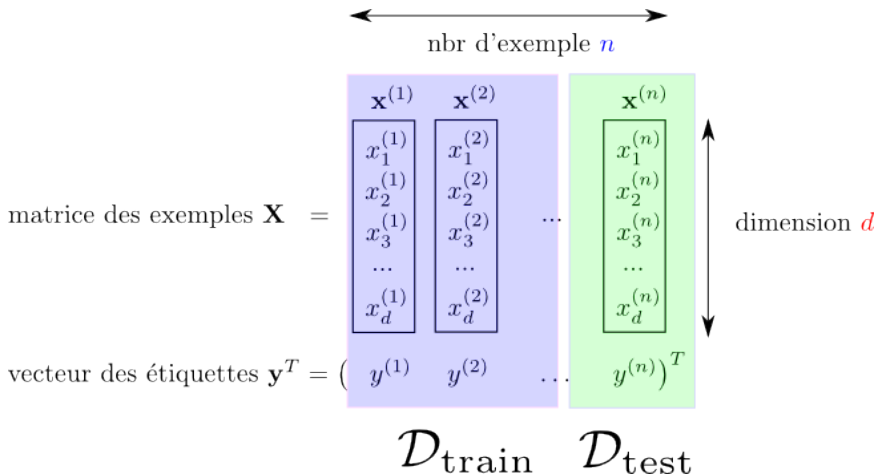
- L'**erreur de train** décroît avec la capacité du modèle.
- Elle permet donc de détecter de l'**under-fitting** mais pas l'**over-fitting** !
- Le problème est que les données \mathcal{D} proviennent de $p_{X,Y}$ mais que la fonction prédictrice f est obtenue à partir de \mathcal{D} .
- Err_{train} n'est donc pas un très bon estimateur de Err_{gen} .



[Bishop 2006]

Comment détecter l'over et l'under-fitting? → en estimant Err_{gen}

- Une autre approche consiste à scinder \mathcal{D} en 2!



Comment détecter l'over et l'under-fitting? → en estimant Err_{gen}

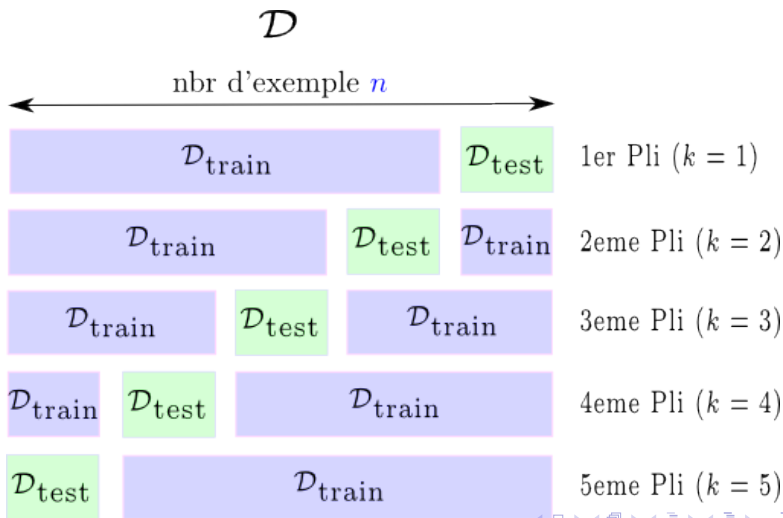
- J'utilise alors l'ensemble d'apprentissage $\mathcal{D}_{\text{train}}$ pour estimer f dans un 1er temps.
- J'utilise ensuite l'ensemble de test $\mathcal{D}_{\text{test}}$ pour estimer Err_{gen} dans un second temps :

$$Err_{\text{test}}(f, \mathcal{D}) = \frac{1}{\#\mathcal{D}_{\text{test}}} \sum_{(y_i, f(\mathbf{x}_i)) \in \mathcal{D}_{\text{test}}} L(y_i, f(\mathbf{x}_i)). \quad (9)$$

- Cette fois-ci, Err_{test} sera faible si je n'ai ni over ni under-fitting!
- Par contre, Err_{test} est un estimateur un peu pessimiste de la qualité de l'apprentissage car beaucoup moins de données sont disponibles pour trouver f .
- Ce protocole est appelé hold out.

Comment détecter l'over et l'under-fitting? → en estimant Err_{gen}

- Pour une meilleure estimation, il faut faire plusieurs scissions.
- On parle alors de validation croisée à K plis (K fold crossvalidation).



Comment détecter l'over et l'under-fitting? → en estimant Err_{gen}

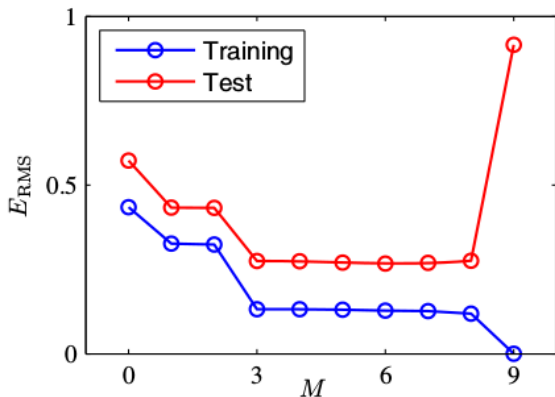
- Soit $\mathcal{D}_{\text{test}}^{(k)}$ l'ensemble de test correspondant au $k^{\text{ème}}$ pli.
- L'erreur de test devient alors :

$$Err_{\text{test}}(f, \mathcal{D}) = \frac{1}{n} \sum_{k=1}^K \sum_{(y_i, f(\mathbf{x}_i)) \in \mathcal{D}_{\text{test}}^{(k)}} L(y_i, f(\mathbf{x}_i)). \quad (10)$$

- Une K -CV requiert K apprentissage → coût non négligeable.
- On coupe \mathcal{D} arbitrairement mais de sorte à conserver à peu près la même proportion d'exemples par classe.
- Cas particulier : $K = n$, on parle de Leave-one-out cross validation (**LOOCV**).

Comment détecter l'over et l'under-fitting? → en estimant Err_{gen}

- L'erreur de test fournit typiquement un graphe en "U".



[Bishop 2006]

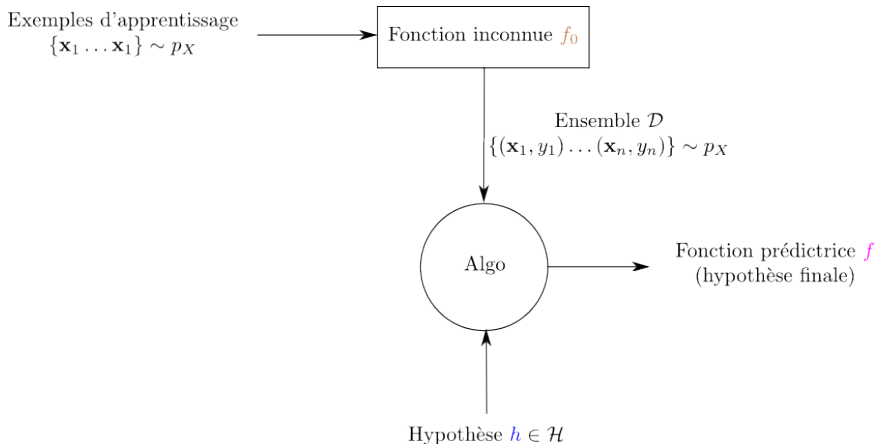
Comment détecter l'over et l'under-fitting? → en maîtrisant Err_{gen}
Approche fréquentiste : trouver une borne de $|Err_{\text{gen}} - Err_{\text{train}}|$.

- Soit \mathcal{H} l'ensemble des hypothèses de notre modèle = ens. des fonctions apprenables.
- Le volume de \mathcal{H} dépend de certains hyperparamètres (degré pour la régression polynomiale, k pour k -ppv, etc.).
- Apprendre consiste à piocher f dans \mathcal{H} de sorte à avoir

$$f = \arg \min_{h \in \mathcal{H}} Err_{\text{gen}}(h).$$

Comment détecter l'over et l'under-fitting? → en maîtrisant Err_{gen}
Approche fréquentiste : trouver une borne de $|Err_{\text{gen}} - Err_{\text{train}}|$.

- On se place dans schéma de travail suivant :



Comment détecter l'over et l'under-fitting? \rightarrow en maîtrisant Err_{gen}
Approche fréquentiste : trouver une borne de $|Err_{\text{gen}} - Err_{\text{train}}|$.

- L'objectif est une inégalité du style :

$$\mathbb{P}(\text{bad thing}) \leq \epsilon. \quad (11)$$

Comment détecter l'over et l'under-fitting? → en maîtrisant Err_{gen}
Approche fréquentiste : trouver une borne de $|Err_{\text{gen}} - Err_{\text{train}}|$.

- L'objectif est une inégalité du style :

$$\mathbb{P}(|Err_{\text{gen}}(h) - Err_{\text{train}}(h, \mathcal{D})| > \tau) \leq \epsilon. \quad (12)$$

Comment détecter l'over et l'under-fitting? \rightarrow en maîtrisant Err_{gen}
Approche fréquentiste : trouver une borne de $|Err_{\text{gen}} - Err_{\text{train}}|$.

- L'objectif est une inégalité du style :

$$\mathbb{P}(\text{worst thing}) \leq \epsilon. \quad (13)$$

Comment détecter l'over et l'under-fitting? → en maîtrisant Err_{gen}
Approche fréquentiste : trouver une borne de $|Err_{\text{gen}} - Err_{\text{train}}|$.

- L'objectif est une inégalité du style :

$$\mathbb{P} \left(\max_{h \in \mathcal{H}} |Err_{\text{gen}}(h) - Err_{\text{train}}(h, \mathcal{D})| > \tau \right) \leq \epsilon. \quad (14)$$

- Cette garantie n'a d'intérêt que si je peux exprimer ϵ en fonction de τ et de n !

Comment détecter l'over et l'under-fitting? → en maîtrisant Err_{gen}
Approche fréquentiste : trouver une borne de $|Err_{\text{gen}} - Err_{\text{train}}|$.

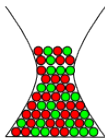
- L'objectif est une inégalité du style :

$$\mathbb{P} \left(\max_{h \in \mathcal{H}} |Err_{\text{gen}}(h) - Err_{\text{train}}(h, \mathcal{D})| > \tau \right) \leq \epsilon. \quad (14)$$

- Cette garantie n'a d'intérêt que si je peux exprimer ϵ en fonction de τ et de n !



Pause Proba !



urne

Proba de mauvaise détection
"out sample" μ



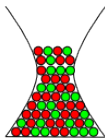
échantillons

Fréquence de mauvaise détection
"in sample" ν

- Posons $Z_i = 1 - \mathbb{I}_{y_i}(\textcolor{red}{f}(X_i))$.
- Z_i est une v.a. **binaire** avec $Z_i \sim \text{Ber}(\mu)$.
- Z_i représente une **mauvaise** détection induite par X_i .
- Posons $S_n = \frac{1}{n} \sum_{i=1}^n Z_i$.
- On a $\mathbb{E}[S_n] = \mu$.



Pause Proba !



urne

Proba de mauvaise détection
"out sample" μ



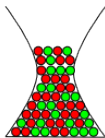
échantillons

Fréquence de mauvaise détection
"in sample" ν

- Posons $Z_i = 1 - \mathbb{I}_{y_i}(\textcolor{violet}{f}(X_i))$.
- Z_i est une v.a. **binaire** avec $Z_i \sim \text{Ber}(\mu)$.
- Z_i représente une **mauvaise** détection induite par X_i .
- Posons $S_n = \frac{1}{n} \sum_{i=1}^n Z_i$.
- On a $\mathbb{E}[S_n] = \mu$.



Pause Proba !



urne

Proba de mauvaise détection
"out sample" μ



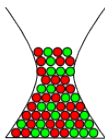
échantillons

Fréquence de mauvaise détection
"in sample" ν

- Posons $Z_i = 1 - \mathbb{I}_{y_i}(\textcolor{violet}{f}(X_i))$.
- Z_i est une v.a. **binaire** avec $Z_i \sim \text{Ber}(\mu)$.
- Z_i représente une **mauvaise** détection induite par X_i .
- Posons $S_n = \frac{1}{n} \sum_{i=1}^n Z_i$.
- On a $\mathbb{E}[S_n] = \mu$.



Pause Proba !



urne

Proba de mauvaise détection
"out sample" μ



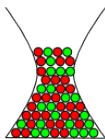
échantillons

Fréquence de mauvaise détection
"in sample" ν

- Posons $Z_i = 1 - \mathbb{I}_{y_i}(\textcolor{violet}{f}(X_i))$.
- Z_i est une v.a. **binaire** avec $Z_i \sim \text{Ber}(\mu)$.
- Z_i représente une **mauvaise** détection induite par X_i .
- Posons $S_n = \frac{1}{n} \sum_{i=1}^n Z_i$.
- On a $\mathbb{E}[S_n] = \mu$.



Pause Proba !



urne

Proba de mauvaise détection
"out sample" μ



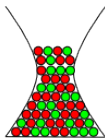
échantillons

Fréquence de mauvaise détection
"in sample" ν

- Posons $Z_i = 1 - \mathbb{I}_{y_i}(\textcolor{violet}{f}(X_i))$.
- Z_i est une v.a. **binaire** avec $Z_i \sim \text{Ber}(\mu)$.
- Z_i représente une **mauvaise** détection induite par X_i .
- Posons $S_n = \frac{1}{n} \sum_{i=1}^n Z_i$.
- On a $\mathbb{E}[S_n] = \mu$.



Pause Proba !



urne

Proba de mauvaise détection
"out sample" μ



échantillons

Fréquence de mauvaise détection
"in sample" ν

- Posons $Z_i = 1 - \mathbb{I}_{y_i}(\textcolor{violet}{f}(X_i))$.
- Z_i est une v.a. **binaire** avec $Z_i \sim \text{Ber}(\mu)$.
- Z_i représente une **mauvaise** détection induite par X_i .
- Posons $S_n = \frac{1}{n} \sum_{i=1}^n Z_i$.
- On a $\mathbb{E}[S_n] = \mu$.



Pause Proba !

- Le résultat suivant s'applique à notre cas :

Inégalité de Hoeffding

Soit $(Z_i)_{i=1}^n$ une suite de v.a. indépendantes et bornées dans $[0; 1]$. Soit $S_n = \frac{1}{n} \sum_{i=1}^n Z_i$ et $\mu = \mathbb{E}[S_n]$. On a alors :

$$\mathbb{P}(|\mu - S_n| > \tau) \leq 2e^{-2n\tau^2} \quad (15)$$

- Ceci est un exemple d'**inégalité de concentration** mesurant la **dévi**ation d'une v.a. par rapport à une valeur.
- On dit que $\mu = S_n$ est "**probably approximately correct**" (PAC).
- Contrairement au TCL, les Z_i ne sont **pas identiquement distribuées**.

Comment détecter l'over et l'under-fitting? → en maîtrisant Err_{gen}
Approche fréquentiste : trouver une borne de $|Err_{\text{gen}} - Err_{\text{train}}|$.
(retour à notre problème).

- Hoeffding nous permet d'écrire :

$$\mathbb{P}(|Err_{\text{gen}}(h) - Err_{\text{train}}(h, \mathcal{D})| > \tau) \leq 2e^{-2n\tau^2}. \quad (16)$$

- C'est gagné! ?
- Une minute.. il manque le max !

Comment détecter l'over et l'under-fitting? → en maîtrisant Err_{gen}
Approche fréquentiste : trouver une borne de $|Err_{\text{gen}} - Err_{\text{train}}|$.
(retour à notre problème).

- Hoeffding nous permet d'écrire :

$$\mathbb{P}(|Err_{\text{gen}}(h) - Err_{\text{train}}(h, \mathcal{D})| > \tau) \leq 2e^{-2n\tau^2}. \quad (16)$$

- C'est gagné ?
- Une minute.. il manque le max !

Comment détecter l'over et l'under-fitting? → en maîtrisant Err_{gen}
Approche fréquentiste : trouver une borne de $|Err_{\text{gen}} - Err_{\text{train}}|$.
(retour à notre problème).

- Hoeffding nous permet d'écrire :

$$\mathbb{P}(|Err_{\text{gen}}(h) - Err_{\text{train}}(h, \mathcal{D})| > \tau) \leq 2e^{-2n\tau^2}. \quad (16)$$

- C'est gagné ?
- Une minute.. il manque le max !

ATTENTION

$$\mathbb{P}\left(\max_{h \in \mathcal{H}} Z_h > \tau\right) \neq \max_{h \in \mathcal{H}} \mathbb{P}(Z_h > \tau)! \quad (17)$$

Contre-Exemple

La v.a. $Z_h \sim \text{Ber}(0.5)$ représente un pile-ou-face. (valeur 1 = “pile”).
Supposons que $\#\mathcal{H} = 10$. On a

$$\begin{aligned} \mathbb{P}\left(\max_{h \in \mathcal{H}} Z_h > 0\right) &= \mathbb{P}(\{Z_1 = 1\} \text{ OU } \{Z_2 = 1\} \dots \text{ OU } \{Z_{10} = 1\}), \\ &= 1 - \mathbb{P}(\{Z_1 = 0\} \text{ ET } \{Z_2 = 0\} \dots \text{ ET } \{Z_{10} = 0\}), \\ &= 1 - \prod_{j=1}^{10} \mathbb{P}(\{Z_j = 0\}), \\ &= 1 - 0.5^{10}, \\ &\approx 0.99902, \\ &\neq 0.5 = \max_{h \in \mathcal{H}} \mathbb{P}(Z_h > 0) \end{aligned}$$

Comment détecter l'over et l'under-fitting? → en maîtrisant Err_{gen}
Approche fréquentiste : trouver une borne de $|Err_{\text{gen}} - Err_{\text{train}}|$.

- La seule possibilité pour atteindre notre objectif est d'utiliser la borne d'union :

$$\mathbb{P}(A \text{ OU } B) \leq \mathbb{P}(A) + \mathbb{P}(B). \quad (18)$$

- Au final, quand \mathcal{H} est fini, on a :

$$\mathbb{P}\left(\max_{h \in \mathcal{H}} |Err_{\text{gen}}(h) - Err_{\text{train}}(h, \mathcal{D})| > \tau\right) \leq 2\#\mathcal{H}e^{-2n\tau^2}. \quad (19)$$

- Quand \mathcal{H} est infini, on peut (sous certaines conditions) utiliser une autre quantité que $\#\mathcal{H}$ appelée dimension de Vapnik-Chervonenski (VC).

Message principal

Message principal

Est-ce possible d'apprendre en **théorie** ?

Message principal

Est-ce possible d'apprendre en *théorie*? Oui, en faisant en sorte que $Err_{gen} \approx 0$.

Message principal

Est-ce possible d'apprendre en **théorie**? Oui, en faisant en sorte que $Err_{gen} \approx 0$.

Est-ce possible d'apprendre en **pratique**?

Message principal

Est-ce possible d'apprendre en **théorie**? Oui, en faisant en sorte que $Err_{gen} \approx 0$.

Est-ce possible d'apprendre en **pratique**? Oui, en faisant en sorte que $Err_{train} \approx 0$ et que $Err_{train} \approx Err_{gen}$.

Approche Bayésienne :

- D'un point de vue Bayésien, seul un **conditionnement** par rapport aux **données** réellement observées \mathcal{D} est valable.
- Soit h^* la meilleure fonction apprenable :

$$h^* = \arg \min_{h \in \mathcal{H}} Err_{\text{gen}}(h). \quad (20)$$

- On inférerait alors h^* en estimant les probabilités

$$P(h^* = h | \mathcal{D}). \quad (21)$$

- Il conviendrait de poser un modèle probabiliste sur les h pour attaquer le problème d'inférence.