A2DI: Théorie de l'apprentissage

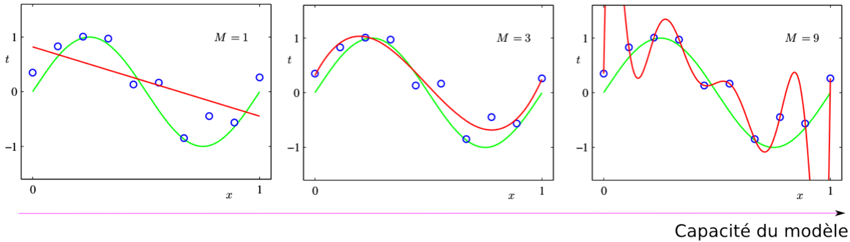
John Klein

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.
- On introduit une fonction de perte (loss) L :
  Exemple en classification : **0-1 Loss**

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

$$(c_i, \hat{c}(\mathbf{x_i})) \rightarrow \begin{cases} 0 & \text{si } c_i = \hat{c}(\mathbf{x_i}) \\ 1 & \text{sinon} \end{cases}, \tag{2}$$

avec $\hat{c}(\mathbf{x_i})$ la classe prédite par notre algorithme pour l'exemple $\mathbf{x_i}$ dont le 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.
- On introduit une fonction de perte (loss) L :
  Exemple en régression : perte en norme 2, **Quadratic Loss**

$$\begin{aligned} L : \mathbb{Y} \times \mathbb{X} &\rightarrow [0; +\infty], & (3) \\ (y_i, f(\mathbf{x_i})) &\rightarrow (y_1 - f(\mathbf{x_i}))^2, & (4) \end{aligned}$$

avec $f(\mathbf{x_i})$ 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 ?

- Il faut déjà s'accorder sur ce que se tromper veut dire.
- On introduit une fonction de perte (loss) L :
  Exemple en régression : perte en norme 1, **Absolute Loss**

$$\begin{aligned} L : \mathbb{Y} \times \mathbb{X} &\rightarrow [0; +\infty], & (5) \\ (y_i, f(\mathbf{x_i})) &\rightarrow |y_1 - f(\mathbf{x_i})|, & (6) \end{aligned}$$

avec $f(\mathbf{x_i})$ 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 $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ représentent tous les cas possibles.
- En revanche, on peut supposer que $\mathcal{D}$ est iid selon la loi génératrice des données $p_{X,Y}$ (inconnue).
- Pour apprendre, on supposera parfois que cette distribution appartient à une famille de distributions paramétrée par $\boldsymbol{\theta}$.
- Notre objectif sera de minimiser la perte attendue sous $p_{X,Y}$ ou erreur espérée :

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

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

- On ne peut pas supposer que nos données $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ représentent tous les cas possibles.

- En revanche, on peut supposer que $\mathcal{D}$ est iid selon la loi génératrice des données $p_{X,Y}$ (inconnue).

- Pour apprendre, on supposera parfois que cette distribution appartient à une famille de distributions paramétrée par $\boldsymbol{\theta}$.

- Notre objectif sera de minimiser la perte attendue sous $p_{X,Y}$ ou erreur espérée :

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

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

- On ne peut pas supposer que nos données $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ représentent tous les cas possibles.
- En revanche, on peut supposer que $\mathcal{D}$ est iid selon la loi génératrice des données $p_{X,Y}$ (inconnue).
- Pour apprendre, on supposera parfois que cette distribution appartient à une famille de distributions paramétrée par $\boldsymbol{\theta}$.
- Notre objectif sera de minimiser la perte attendue sous $p_{X,Y}$ ou erreur espérée :

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

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

- On ne peut pas supposer que nos données $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ représentent tous les cas possibles.
- En revanche, on peut supposer que $\mathcal{D}$ est iid selon la loi génératrice des données $p_{X,Y}$ (inconnue).
- Pour apprendre, on supposera parfois que cette distribution appartient à une famille de distributions paramétrée par $\boldsymbol{\theta}$.
- Notre objectif sera de minimiser la perte attendue sous $p_{X,Y}$ ou erreur espérée :

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

Comment détecter l'over et l'under-fitting ? $\rightarrow$ 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

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

- L'erreur de généralisation se définit alors comme l'écart entre la perte attendue et l'erreur de train :

$$Err_{\text{gen}}\left(f\right)=Err_{\text{esp}}-Err_{\text{train}}. \quad (9)$$

Comment détecter l'over et l'under-fitting ? $\rightarrow$ 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

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

- L'erreur de généralisation se définit alors comme l'écart entre la perte attendue et l'erreur de train :

$$Err_{\text{gen}}\left(f\right)=Err_{\text{esp}}-Err_{\text{train}}. \tag{9}$$

Comment détecter l'over et l'under-fitting ? $\rightarrow$ 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

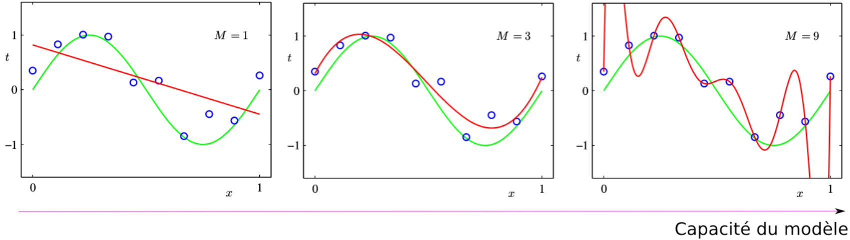
$$Err_{\text{train}}\left(f,\mathcal{D}\right)=\frac{1}{N}\sum_{i=1}^{N}L\left(y_i,f\left(\mathbf{x}_i\right)\right). \tag{8}$$

- L'erreur de généralisation se définit alors comme l'écart entre la perte attendue et l'erreur de train :

$$Err_{\text{gen}}\left(f\right)=Err_{\text{esp}}-Err_{\text{train}}. \tag{9}$$

Comment détecter l'over et l'under-fitting ? $\rightarrow$ en estimant Err_{esp}

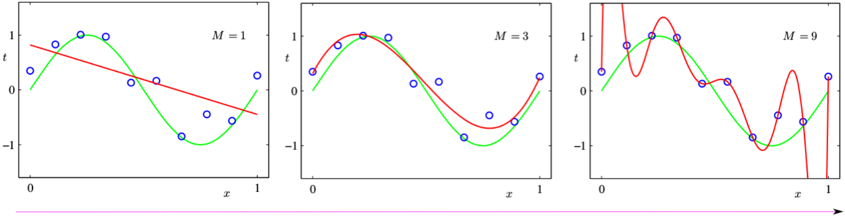
- L'erreur de train décroît avec la capacité du modèle.
- Elle permet donc de détecter de l'under-fitting mais pas l'over-fitting !
- Le problème est que les données $\mathcal{D}$ proviennent de $p_{X,Y}$ mais que la fonction prédictrice f est obtenue à partir de $\mathcal{D}$.
- Err_{train} n'est donc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



[Bishop 2006]

Comment détecter l'over et l'under-fitting ? $\rightarrow$ en estimant Err_{esp}

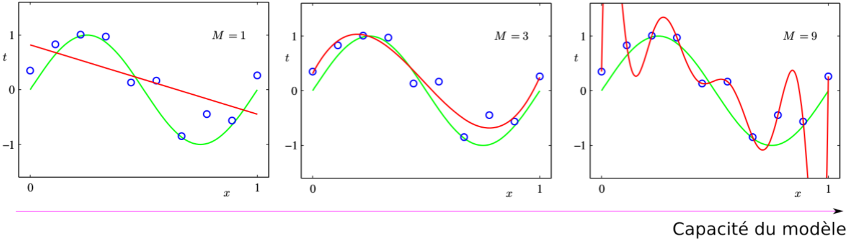
- L'erreur de train décroît avec la capacité du modèle.
- Elle permet donc de détecter de l'under-fitting mais pas l'over-fitting !
- Le problème est que les données $\mathcal{D}$ proviennent de $p_{X,Y}$ mais que la fonction prédictrice f est obtenue à partir de $\mathcal{D}$.
- Err_{train} n'est donc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



[Bishop 2006]

Comment détecter l'over et l'under-fitting ? $\rightarrow$ en estimant Err_{esp}

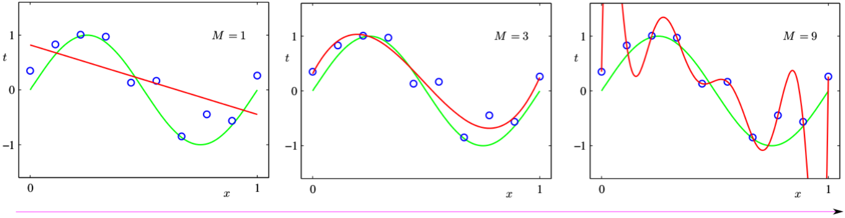
- L'erreur de train décroît avec la capacité du modèle.
- Elle permet donc de détecter de l'under-fitting mais pas l'over-fitting !
- Le problème est que les données $\mathcal{D}$ proviennent de $p_{X,Y}$ mais que la fonction prédictrice f est obtenue à partir de $\mathcal{D}$.
- Err_{train} n'est donc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



[Bishop 2006]

Comment détecter l'over et l'under-fitting ? $\rightarrow$ en estimant Err_{esp}

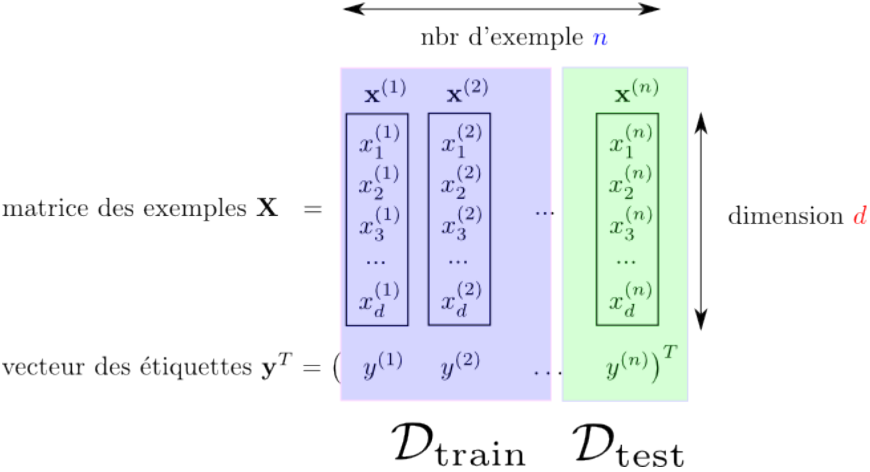
- L'erreur de train décroît avec la capacité du modèle.
- Elle permet donc de détecter de l'under-fitting mais pas l'over-fitting !
- Le problème est que les données $\mathcal{D}$ proviennent de $p_{X,Y}$ mais que la fonction prédictrice f est obtenue à partir de $\mathcal{D}$.
- Err_{train} n'est donc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



[Bishop 2006]

Comment détecter l'over et l'under-fitting ? $\rightarrow$ en estimant Err_{esp}

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



Comment détecter l'over et l'under-fitting ? $\rightarrow$ en estimant Err_{esp}

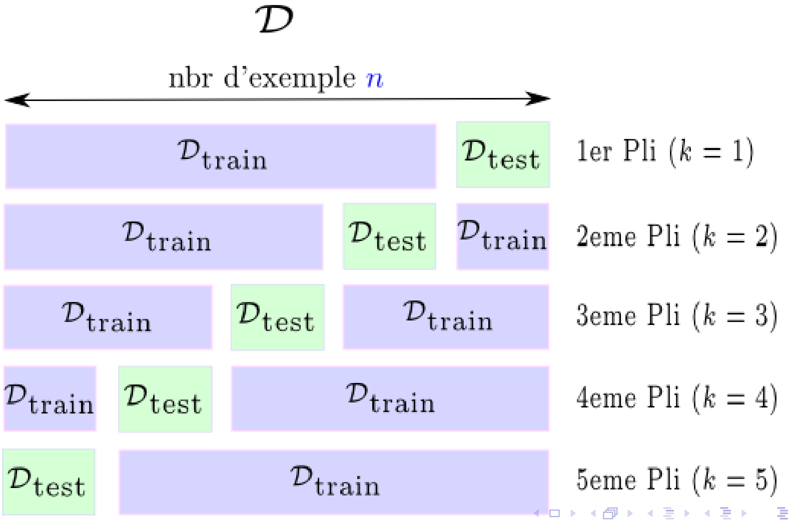
- J'utilise alors l'ensemble d'apprentissage $\mathcal{D}_{\text{train}}$ pour estimer f dans un 1er temps.
- J'utilise ensuite l'ensemble de test $\mathcal{D}_{\text{test}}$ pour estimer Err_{esp} dans un second temps :

$$Err_{\text{test}}\left(f,\mathcal{D}\right) = \frac{1}{\sharp\mathcal{D}_{\text{test}}} \sum_{(y_i, f(\mathbf{x}_i)) \in \mathcal{D}_{\text{test}}} L\left(y_i, f\left(\mathbf{x}_i\right)\right). \tag{10}$$

- Cette fois-ci, Err_{test} sera faible si je n'ai ni over ni under-fitting !

Comment détecter l'over et l'under-fitting ? $\rightarrow$ en estimant Err_{esp}

- 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 ? $\rightarrow$ en estimant Err_{esp}

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

$$Err_{\text{test}}(f, \mathcal{D}) = \frac{1}{n} \sum_{k=1}^{K} \sum_{(y_i, f(\mathbf{x}_i)) \in \mathcal{D}_{\text{test}}^{(k)}} L(y_i, f(\mathbf{x}_i)) . \tag{11}$$

- Une K-CV requiert K apprentissages $\rightarrow$ 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 ? $\rightarrow$ en estimant Err_{esp}

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

- L'erreur de test devient alors :

$$Err_{\text{test}}(f, \mathcal{D}) = \frac{1}{n} \sum_{k=1}^{K} \sum_{(y_i, f(\mathbf{x}_i)) \in \mathcal{D}^{(k)}_{\text{test}}} L(y_i, f(\mathbf{x}_i)) . \tag{11}$$

- Une K-CV requiert K apprentissages $\rightarrow$ 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 ? $\rightarrow$ en estimant Err_{esp}

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

$$Err_{\text{test}}(f, \mathcal{D}) = \frac{1}{n} \sum_{k=1}^{K} \sum_{(y_i, f(\mathbf{x}_i)) \in \mathcal{D}_{\text{test}}^{(k)}} L(y_i, f(\mathbf{x}_i)) . \tag{11}$$

- Une K-CV requiert K apprentissages $\rightarrow$ 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 ? $\rightarrow$ en estimant Err_{esp}

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

$$Err_{\text{test}}(f, \mathcal{D}) = \frac{1}{n} \sum_{k=1}^{K} \sum_{(y_i, f(\mathbf{x}_i)) \in \mathcal{D}_{\text{test}}^{(k)}} L(y_i, f(\mathbf{x}_i)). \quad (11)$$

- Une K-CV requiert K apprentissages $\rightarrow$ 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 ? $\rightarrow$ en estimant Err_{esp}

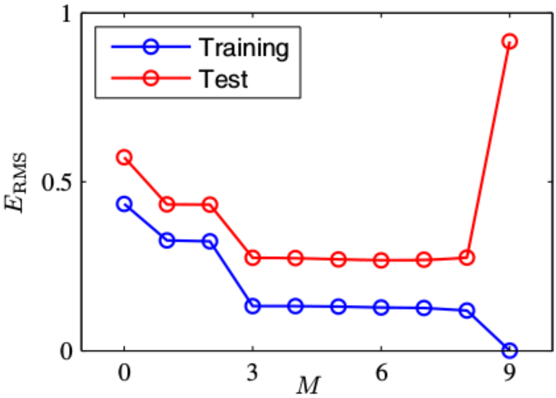
- Soit $\mathcal{D}_{\text{test}}^{(k)}$ l'ensemble de test correspondant au $k^{\text{ème}}$ pli.
- L'erreur de test devient alors :

$$Err_{\text{test}}(f, \mathcal{D}) = \frac{1}{n} \sum_{k=1}^{K} \sum_{(y_i, f(\mathbf{x}_i)) \in \mathcal{D}_{\text{test}}^{(k)}} L(y_i, f(\mathbf{x}_i)). \tag{11}$$

- Une K-CV requiert K apprentissages $\rightarrow$ 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 ? $\rightarrow$ en estimant Err_{esp}

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



[Bishop 2006]

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

- Soit $\mathcal{H}$ l'ensemble des hypothèses = notre modèle (degré pour le régression polynomiale, k pour k-ppv, etc.).

- Pour un $h \in \mathcal{H}$, remplaçons nos notations :

$$Err_{\text{train}}(f, \mathcal{D}) \leftarrow Err_{\text{train}}(h, \mathcal{D}), \tag{12}$$
$$Err_{\text{gen}}(f) \leftarrow Err_{\text{gen}}(h). \tag{13}$$

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

- Soit $\mathcal{H}$ l'ensemble des hypothèses = notre modèle (degré pour le régression polynomiale, k pour k-ppv, etc.).
- Pour un $h \in \mathcal{H}$, remplaçons nos notations :

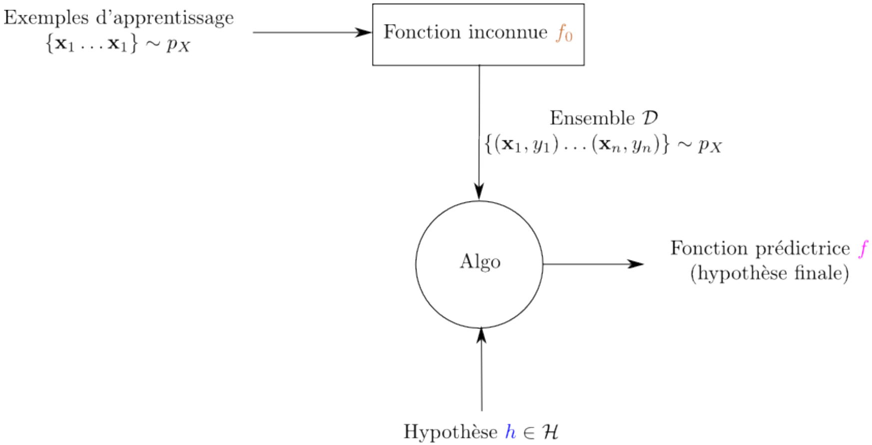
$$Err_{\mathrm{train}}(f, \mathcal{D}) \leftarrow Err_{\mathrm{train}}(h, \mathcal{D}), \tag{12}$$

$$Err_{\mathrm{gen}}(f) \leftarrow Err_{\mathrm{gen}}(h). \tag{13}$$

Comment prévenir l'over et l'under-fitting ? $\rightarrow$ en maîtrisant Err_{gen}

Approche fréquentiste : trouver une borne de $|Err_{\text{gen}}|$.

- On se place dans schéma de travail suivant :



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

- L'objectif est une inégalité du style :

$$\mathbb{P}\left(\text{bad thing}\right) \leq \epsilon. \tag{14}$$

- Attention, la majorité des h ont de mauvaises performances (Err_{esp} grande) !
- Ce qui nous intéresse est que Err_{train} reflète bien les performances de h !
- Cette garantie n'a d'intérêt que si je peux exprimer ϵ en fonction de τ et de n !

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

- L'objectif est une inégalité du style :

$$\mathbb{P}\left(\left|Err_{\text{esp}}\left(h\right) - Err_{\text{train}}\left(h, \mathcal{D}\right)\right| > \tau\right) \leq \epsilon. \tag{14}$$

- Attention, la majorité des h ont de mauvaises performances (Err_{esp} grande) !
- Ce qui nous intéresse est que Err_{train} reflète bien les performances de h !
- Cette garantie n'a d'intérêt que si je peux exprimer ϵ en fonction de τ et de n !

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

- L'objectif est une inégalité du style :

$$\mathbb{P}\left(\text{worst thing}\right) \leq \epsilon. \tag{14}$$

- Attention, la majorité des h ont de mauvaises performances (Err_{esp} grande) !
- Ce qui nous intéresse est que Err_{train} reflète bien les performances de h !
- Cette garantie n'a d'intérêt que si je peux exprimer ϵ en fonction de τ et de n !

Comment prévenir l'over et l'under-fitting ? $\rightarrow$ en maîtrisant Err_{gen}

Approche fréquentiste : trouver une borne de $|Err_{\text{gen}}|$.

- L'objectif est une inégalité du style :

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

- Attention, la majorité des h ont de mauvaises performances (Err_{esp} grande) !
- Ce qui nous intéresse est que Err_{train} reflète bien les performances de h !
- Cette garantie n'a d'intérêt que si je peux exprimer ϵ en fonction de τ et de n !

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

- L'objectif est une inégalité du style :

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

- Attention, la majorité des h ont de mauvaises performances (Err_{esp} grande) !
- Ce qui nous intéresse est que Err_{train} reflète bien les performances de h !
- Cette garantie n'a d'intérêt que si je peux exprimer ϵ en fonction de τ et de n !

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

- L'objectif est une inégalité du style :

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

- Attention, la majorité des h ont de mauvaises performances (Err_{esp} grande) !
- Ce qui nous intéresse est que Err_{train} reflète bien les performances de h !
- Cette garantie n'a d'intérêt que si je peux exprimer ϵ en fonction de τ et de n !

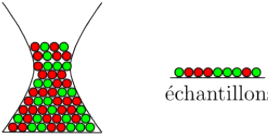
Comment prévenir l'over et l'under-fitting ? $\rightarrow$ en maîtrisant Err_{gen}
Approche fréquentiste : trouver une borne de $|Err_{\mathrm{gen}}|$.

- L'objectif est une inégalité du style :

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

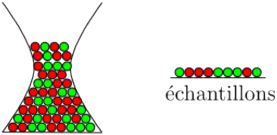
- Attention, la majorité des h ont de mauvaises performances (Err_{esp} grande) !
- Ce qui nous intéresse est que Err_{train} reflète bien les performances de h !
- Cette garantie n'a d'intérêt que si je peux exprimer ϵ en fonction de τ et de n !

Pause Proba !



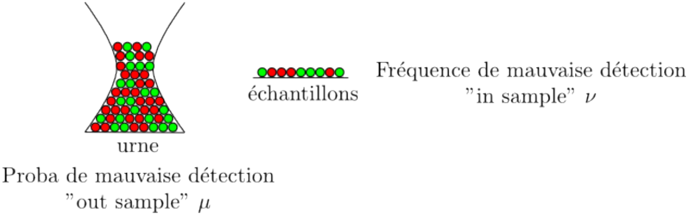
- Posons $Z_i = 1 - \mathbb{I}_{y_i}(f(X_i))$.
- Z_i est une v.a. binaire avec $Z_i \sim \mathrm{Ber}(\mu)$.
- Z_i représente une mauvaise détection induite par X_i.
- Posons $S_n = \frac{1}{n} \sum_{i=1}^{n} Z_i$.
- On a $\mathbb{E}[S_n] = \mu$.

Pause Proba !



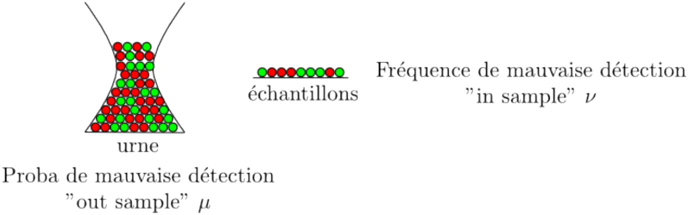
- Posons $Z_i = 1 - \mathbb{I}_{y_i}\left(f\left(X_i\right)\right)$.
- Z_i est une v.a. binaire avec $Z_i \sim \mathrm{Ber}\left(\mu\right)$.
- Z_i représente une mauvaise détection induite par X_i.
- Posons $S_n = \frac{1}{n} \sum_{i=1}^{n} Z_i$.
- On a $\mathbb{E}\left[S_n\right] = \mu$.

Pause Proba !



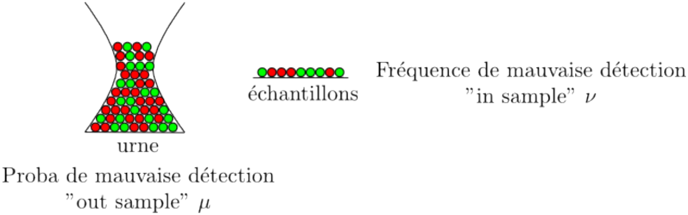
- Posons $Z_i = 1 - \mathbb{I}_{y_i}\left(f\left(X_i\right)\right)$.
- Z_i est une v.a. binaire avec $Z_i \sim \mathrm{Ber}\left(\mu\right)$.
- Z_i représente une mauvaise détection induite par X_i.
- Posons $S_n = \frac{1}{n}\sum_{i=1}^{n} Z_i$.
- On a $\mathbb{E}\left[S_n\right] = \mu$.

Pause Proba !



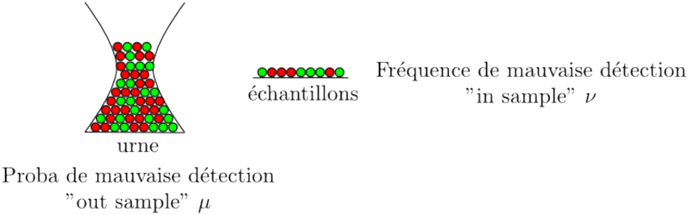
- Posons $Z_i = 1 - \mathbb{I}_{y_i}\left(f\left(X_i\right)\right)$.
- Z_i est une v.a. binaire avec $Z_i \sim \mathrm{Ber}\left(\mu\right)$.
- Z_i représente une mauvaise détection induite par X_i.
- Posons $S_n = \frac{1}{n}\sum_{i=1}^{n} Z_i$.
- On a $\mathbb{E}\left[S_n\right] = \mu$.

Pause Proba !



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

Pause Proba !



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

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 déviation d'une v.a. par rapport à une valeur.
- On dit que $\mu = S_n$ est "probably approximately correct" (PAC).

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

- Hoeffding nous permet d'écrire :

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

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

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

- Hoeffding nous permet d'écrire :

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

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

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

- Hoeffding nous permet d'écrire :

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

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

ATTENTION

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

Contre-Exemple

La v.a. $Z_h \sim \mathrm{Ber}\left(0.5\right)$ représente un pile-ou-face. (valeur $1 =$ "pile"). Suppons que $\sharp\mathcal{H} = 10$. On a

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

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

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

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

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

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

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

Comment prévenir l'over et l'under-fitting ? $\rightarrow$ en maîtrisant Err_{gen}
Approche Bayésienne : on intègre !

- D'un point de vue Bayésien, seul un conditionnement par rapport aux données réellement observées $\mathcal{D}$ est valable.

- On s'intéresse donc à la quantité suivante, appelée perte attendue a posteriori :

$$\rho(f|\mathcal{D}) = \mathbb{E}_{f|\mathcal{D}}[Err_{\text{esp}}] = \int Err_{\text{esp}}(f)\, dp_{f|\mathcal{D}}. \tag{20}$$

- Il n'est pas toujours évident de calculer cette intégrale mais dans l'absolu, cela reste faisable.

- Minimiser cette quantité par rapport à f est un problème complexe $\rightarrow$ modèle paramétrique $f \leftrightarrow \theta$.

Comment prévenir l'over et l'under-fitting ? $\rightarrow$ en maîtrisant Err_{gen}

Approche Bayésienne : on intègre !

- D'un point de vue Bayésien, seul un conditionnement par rapport aux données réellement observées $\mathcal{D}$ est valable.

- On s'intéresse donc à la quantité suivante, appelée perte attendue a posteriori :

$$\rho(f|\mathcal{D}) = \mathbb{E}_{f|\mathcal{D}}\left[Err_{\text{esp}}\right] = \int Err_{\text{esp}}(f)\, dp_{f|\mathcal{D}}. \tag{20}$$

- Il n'est pas toujours évident de calculer cette intégrale mais dans l'absolu, cela reste faisable.

- Minimiser cette quantité par rapport à f est un problème complexe $\rightarrow$ modèle paramétrique $f \leftrightarrow \theta$.

Comment prévenir l'over et l'under-fitting ? $\rightarrow$ en maîtrisant Err_{gen}

Approche Bayésienne : on intègre !

- D'un point de vue Bayésien, seul un conditionnement par rapport aux données réellement observées $\mathcal{D}$ est valable.
- On s'intéresse donc à la quantité suivante, appelée perte attendue a posteriori :

$$\rho\left(f|\mathcal{D}\right) = \mathbb{E}_{f|\mathcal{D}}\left[Err_{\mathrm{esp}}\right] = \int Err_{\mathrm{esp}}\left(f\right) dp_{f|\mathcal{D}}. \tag{20}$$

- Il n'est pas toujours évident de calculer cette intégrale mais dans l'absolu, cela reste faisable.
- Minimiser cette quantité par rapport à f est un problème complexe $\rightarrow$ modèle paramétrique $f \leftrightarrow \theta$.

Comment prévenir l'over et l'under-fitting ? $\rightarrow$ en maîtrisant Err_{gen}

Approche Bayésienne : on intègre !

- D'un point de vue Bayésien, seul un conditionnement par rapport aux données réellement observées $\mathcal{D}$ est valable.
- On s'intéresse donc à la quantité suivante, appelée perte attendue a posteriori :

$$\rho(f|\mathcal{D}) = \mathbb{E}_{f|\mathcal{D}}[Err_{\text{esp}}] = \int Err_{\text{esp}}(f)\, dp_{f|\mathcal{D}}. \tag{20}$$

- Il n'est pas toujours évident de calculer cette intégrale mais dans l'absolu, cela reste faisable.
- Minimiser cette quantité par rapport à f est un problème complexe $\rightarrow$ modèle paramétrique $f \leftrightarrow \theta$.

$\boxed{\text{Message principal}}$

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 $Err_{train} \approx Err_{esp}$.