ENSIAS

Apprentissage Automatique
Partie 2

Pr. El AFIA

1. Bias-Variance

Capacité de Généralisation

La capacité de généralisation : $\boldsymbol{E_{gen}(g) \approx 0}$

On apprend f par minimisation de $E_{emp}(g)$, mais que peut-on dire sur $E_{gen}(g)$?

Il existe deux facteurs déterminant la capacité de généralisation:

- La complexité du modèle (la complexité de H).

- La complexité de l'échantillon (la complexité de N).

Réf.

Complexité du Modèle et Complexité de l’échantillon

Définition: Complexité du Modèle

La complexité du modèle indique le nombre des paramètres nécessaires pour atteindre une certaine performance de généralisation.

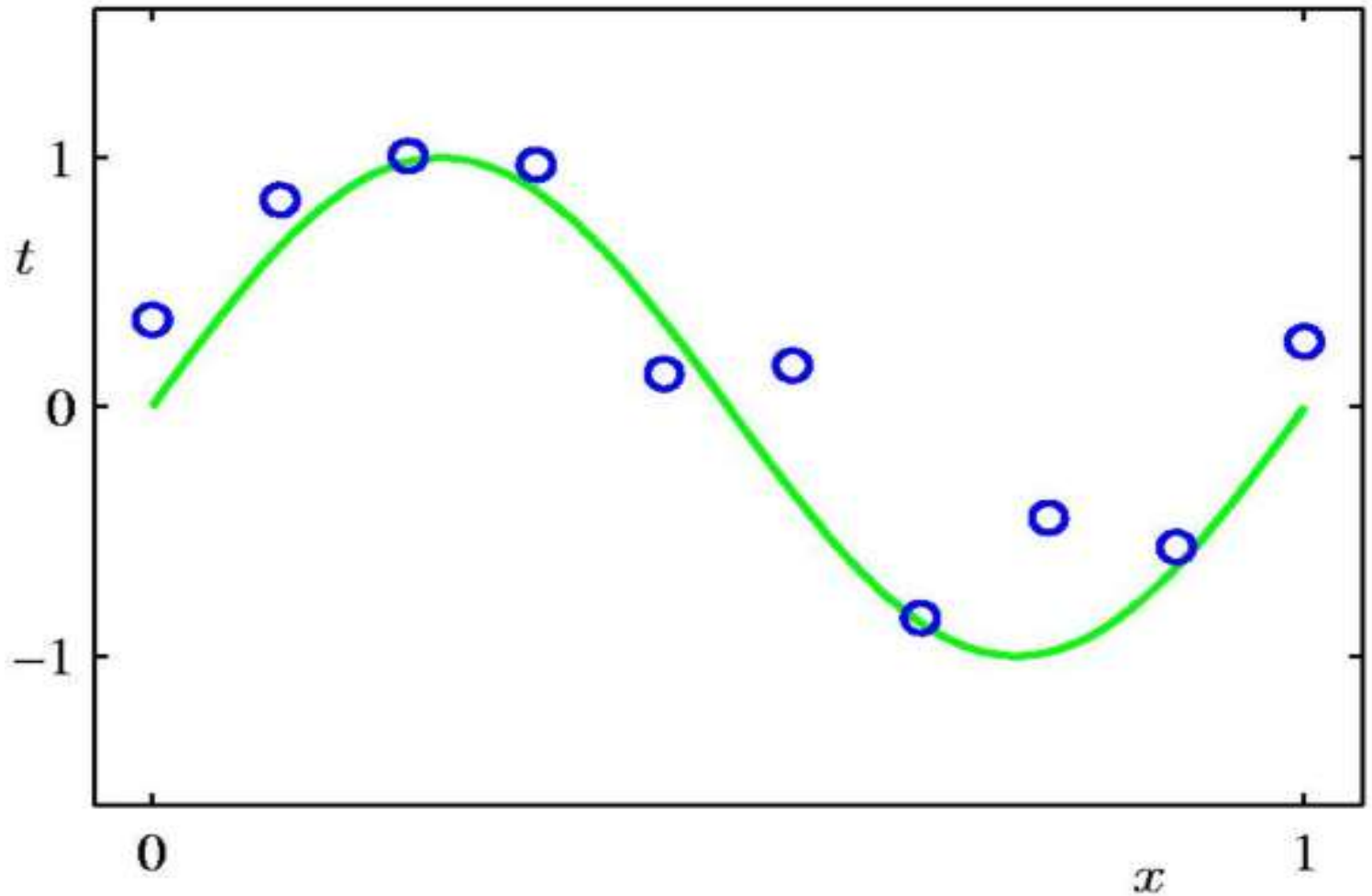
Définition: Complexité de l’échantillon

La complexité de l’échantillon indique le nombre des points de données d’entrainement nécessaire pour atteindre une certaine performance de généralisation.

Réf.

Complexité du Modèle

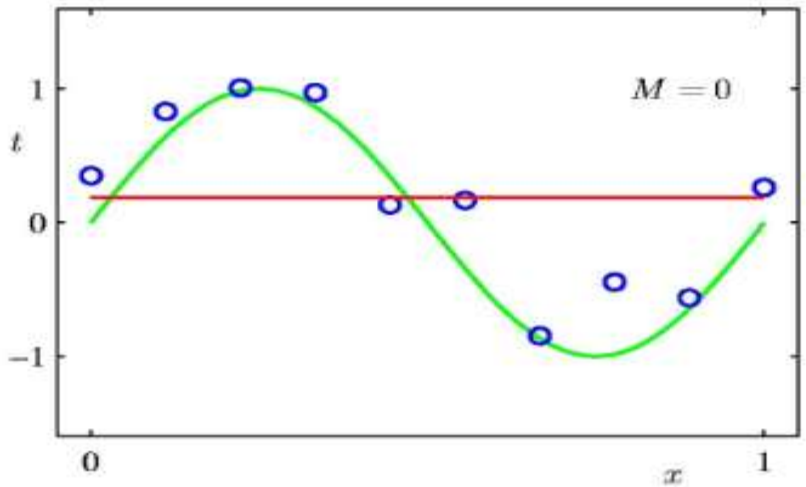
- Quelle est la meilleure hypothèse polynomiale pour ces données bruitées?

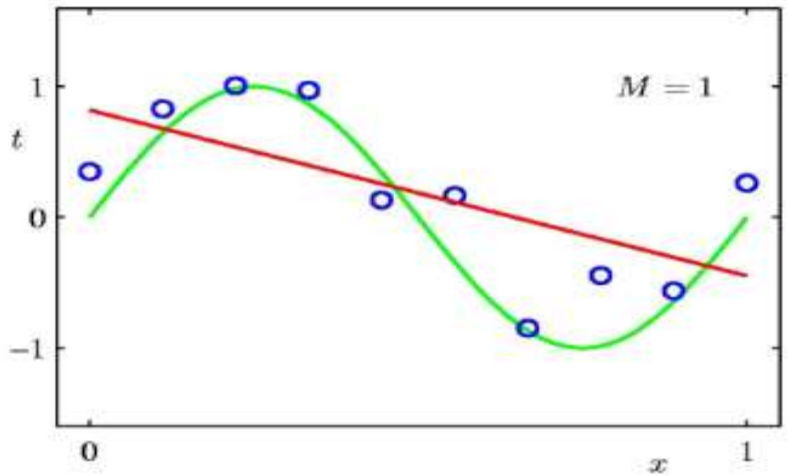


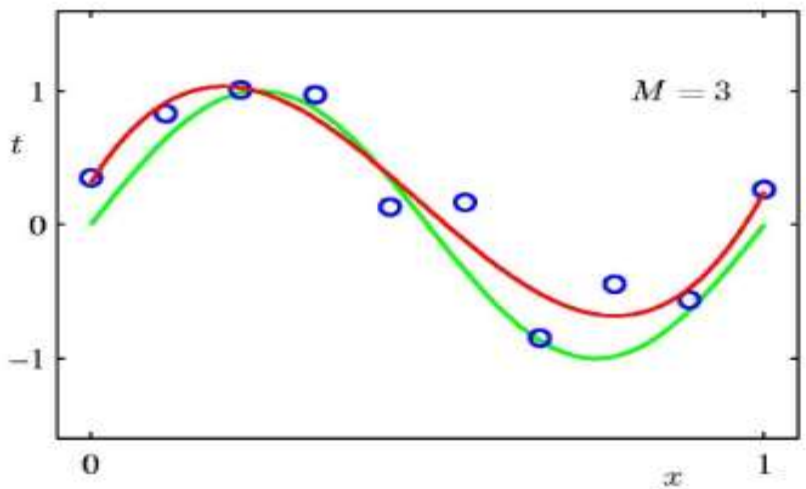
Réf.

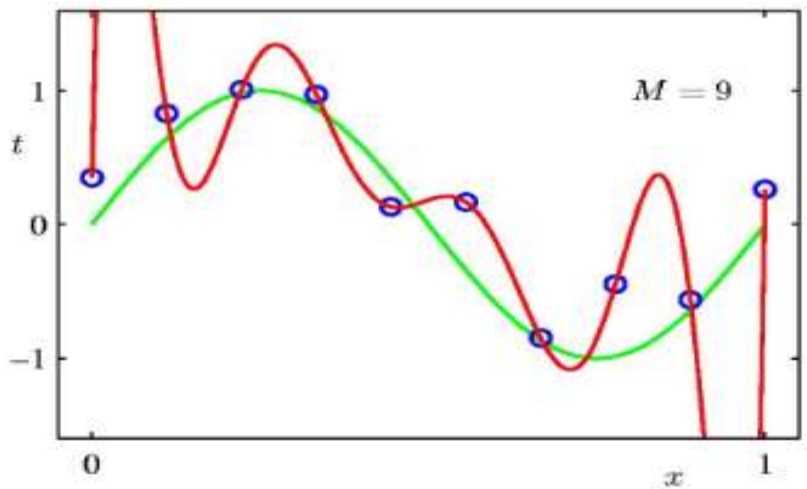
Complexité du Modèle

- Quelle est la meilleure hypothèse polynomiale pour ces données bruitées?





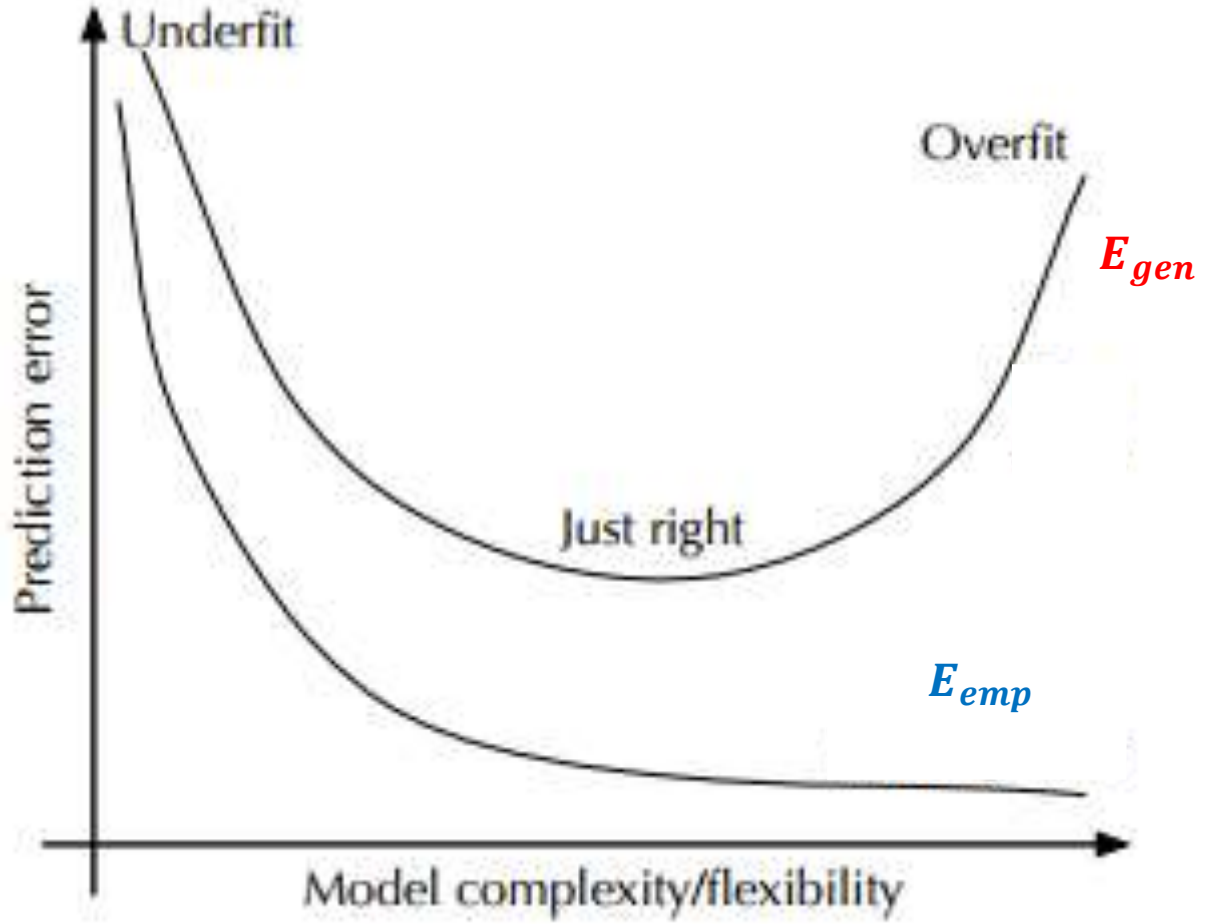




Réf.

Complexité du Modèle – Courbe d'apprentissage

- On suppose que la taille des données (la complexité de l'échantillon) est fixe.

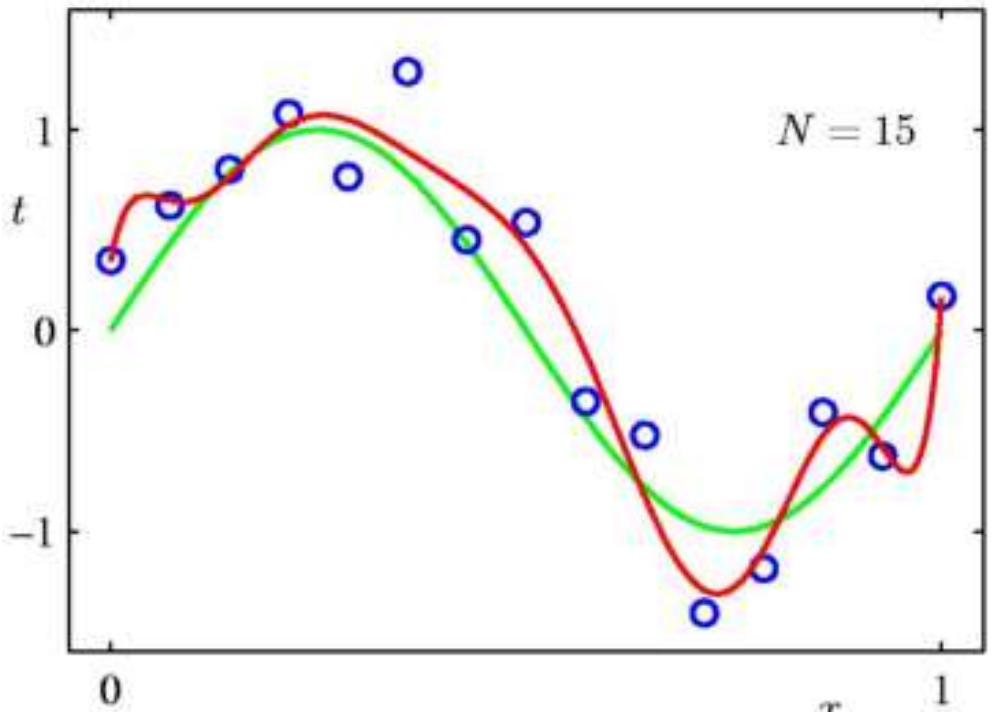


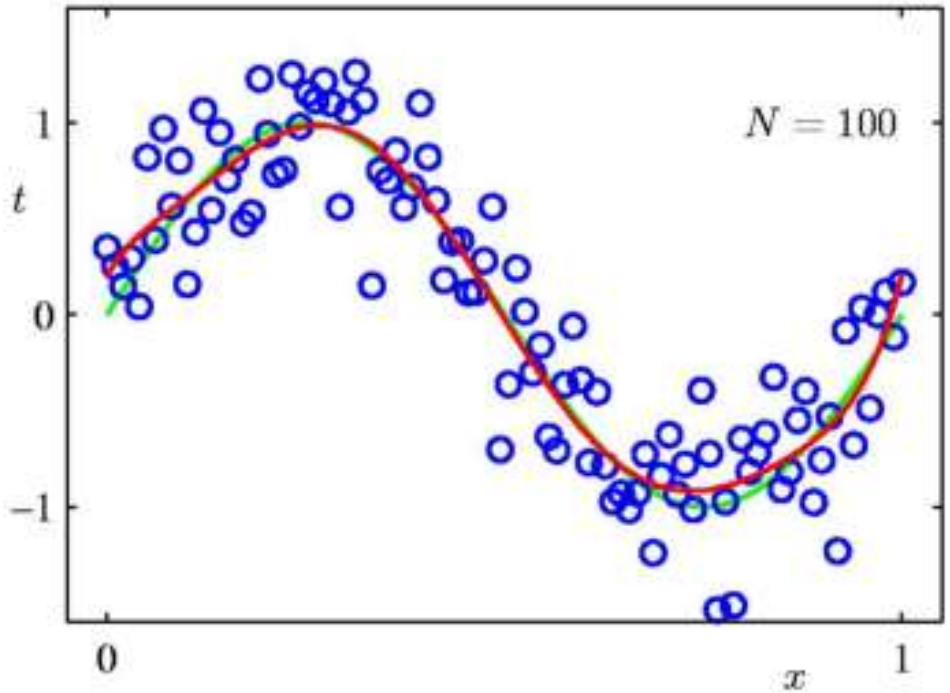
Réf.

Complexité de l'échantillon

- Choix de l'hypothèse est mauvais?

- Taille des données insuffisante?

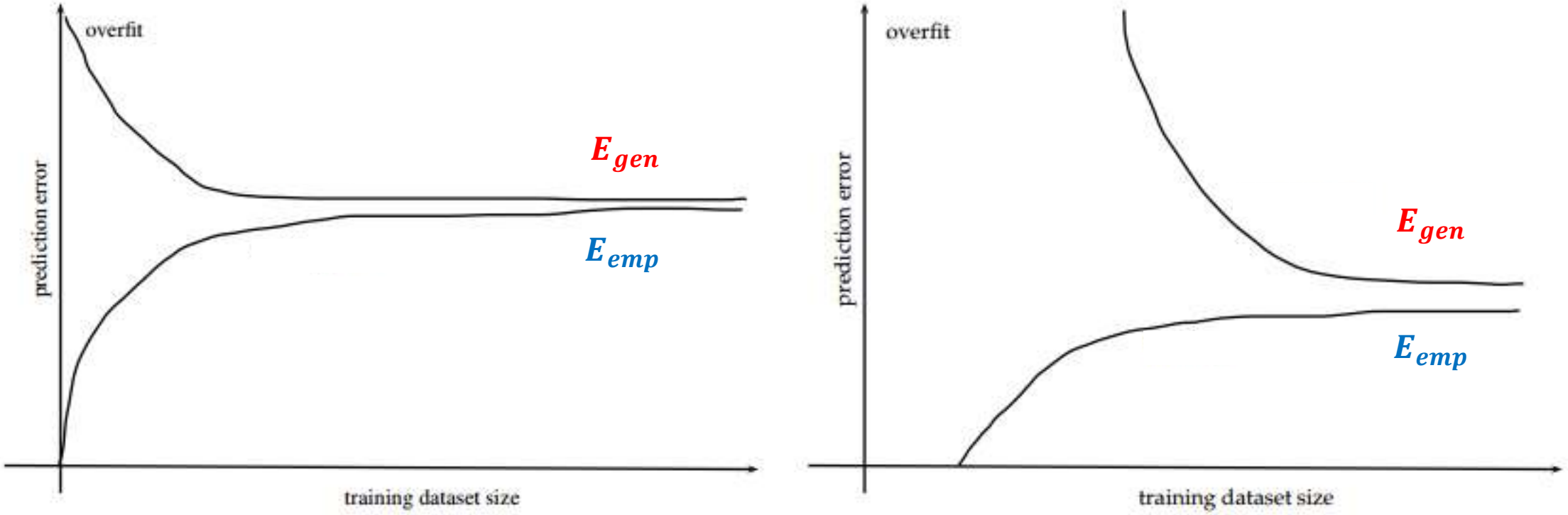




Réf.

Complexité de l'échantillon – Courbe d'apprentissage

- On suppose que la complexité du modèle est fixe.



- L'un de ces modèles est complexe et l'autre est simple. Lesquels?

Réf.

Autres Types de fonctions cibles

La mesure d'erreur utilisée dans ce cas est l'erreur quadratique :

$$e(h(x), y) = (h(x) - y)^2$$

Donc:

$$E_{emp}(h) = \frac{1}{n}\sum_{i=1}^{n}(h(x_i) - y_i)^2$$

$$E_{gen}(h) = E[(h(x) - y)^2]$$

Pour $h(x) = g^{(D)}(x)$ on a :

$$E_{gen}\left(g^{(D)}(x)\right) = E_x[(g^{(D)}(x) - y)^2]$$

Réf.

Autres Types de fonctions cibles

Or :

$$y = f(x) + \varepsilon(x)$$

$$E_{gen}\left(g^{(D)}(x)\right) = E_x\left[\left(g^{(D)}(x) - f(x) - \varepsilon(x)\right)^2\right]$$

$$E_D[E_{gen}\left(g^{(D)}\right)] = E_D[E_x\left[\left(g^{(D)}(x) - f(x) + \varepsilon(x)\right)^2\right]]$$

$$= E_x[E_D\left[\left(g^{(D)}(x) - \bar{g}(x) + \bar{g}(x) - f(x) + \varepsilon(x)\right)^2\right]]$$

$$= E_x[E_D\left[(g^{(D)}(x) - \bar{g}(x))^2 + \left(\bar{g}(x) - f(x)\right)^2 + \left(\varepsilon(x)\right)^2 + termes\right]]$$

Tel que:

$$\bar{g}(x) = \frac{1}{K}\sum_{k=1}^{K} g_k(x)$$

Réf.

Autres Types de fonctions cibles

$$= E_x[E_D[\left(g^{(D)}(x) - \bar{g}(x)\right)^2]] + E_x\left[\left(\bar{g}(x) - f(x)\right)^2\right] + E_x[\left(\varepsilon(x)\right)^2]$$

Donc :

$$\boldsymbol{E_D[E_{gen}\left(g^{(D)}\right)] = E_x[var(x)] + E_x[bias(x)] + \sigma^2}$$

$$\boldsymbol{E_D[E_{gen}\left(g^{(D)}\right)] = Variance + Bias + Bruit}$$

Réf.

Autres Types de fonctions cibles

$$\boldsymbol{E_D[E_{gen}(g^{(D)})] = Variance + Bias + Bruit}$$

- Le $\boldsymbol{bias}$ c'est le bruit déterministe.
- Le $\boldsymbol{\sigma^2}$ c'est le bruit stochastique.
- $\boldsymbol{Variance}$ c'est l'impact indirect du bruit.

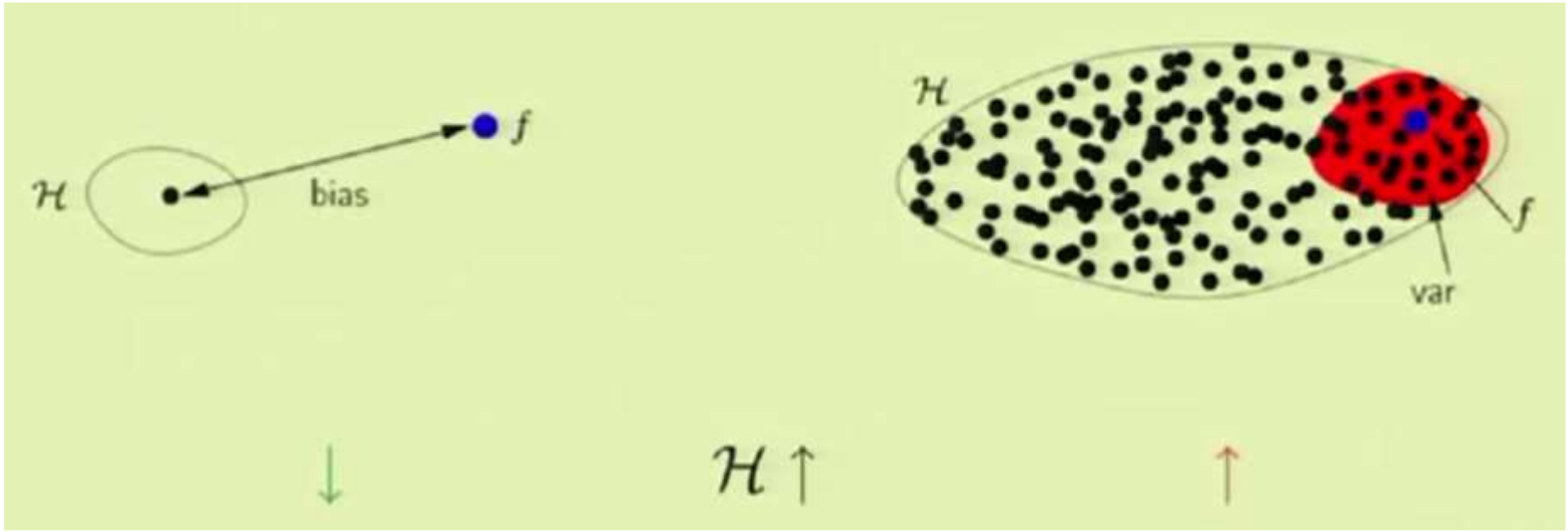
La décomposition bias/variance est un outil très utile pour comprendre l'impact du bruit sur la performance du modèle.

Réf.

Décomposition bias-variance

$$\boldsymbol{bias} = (\bar{\boldsymbol{g}}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}))^2$$

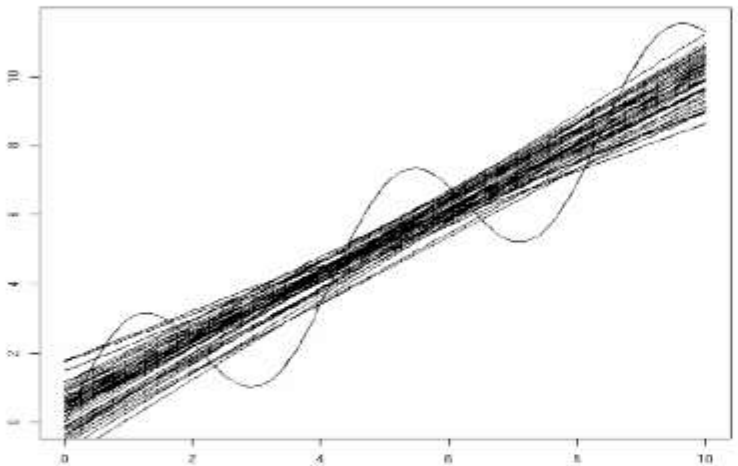
$$\boldsymbol{var} = \boldsymbol{E_D}\left[(\boldsymbol{g}^{(\boldsymbol{D})}(\boldsymbol{x}) - \bar{\boldsymbol{g}}(\boldsymbol{x}))^2\right]$$



$\bar{g}(x)$ c'est le centroide de la région rouge.

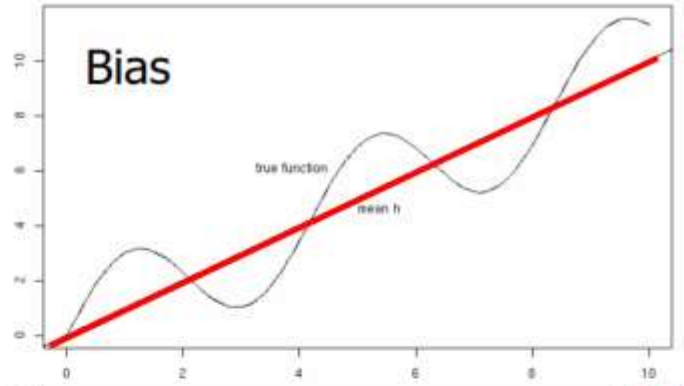
Réf.

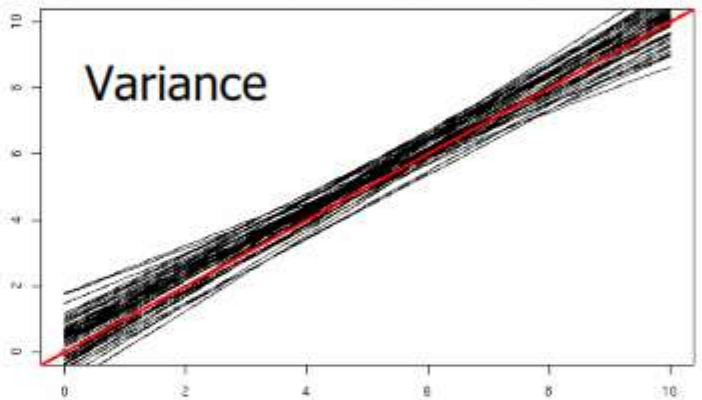
Bias – Variance – Bruit

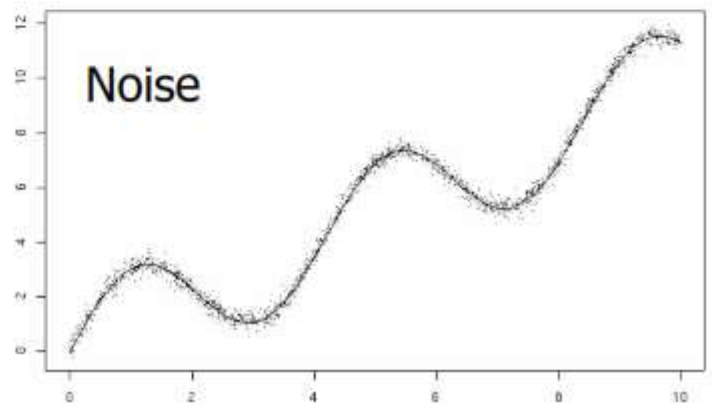


50 fits (20 examples each)

=







Réf.

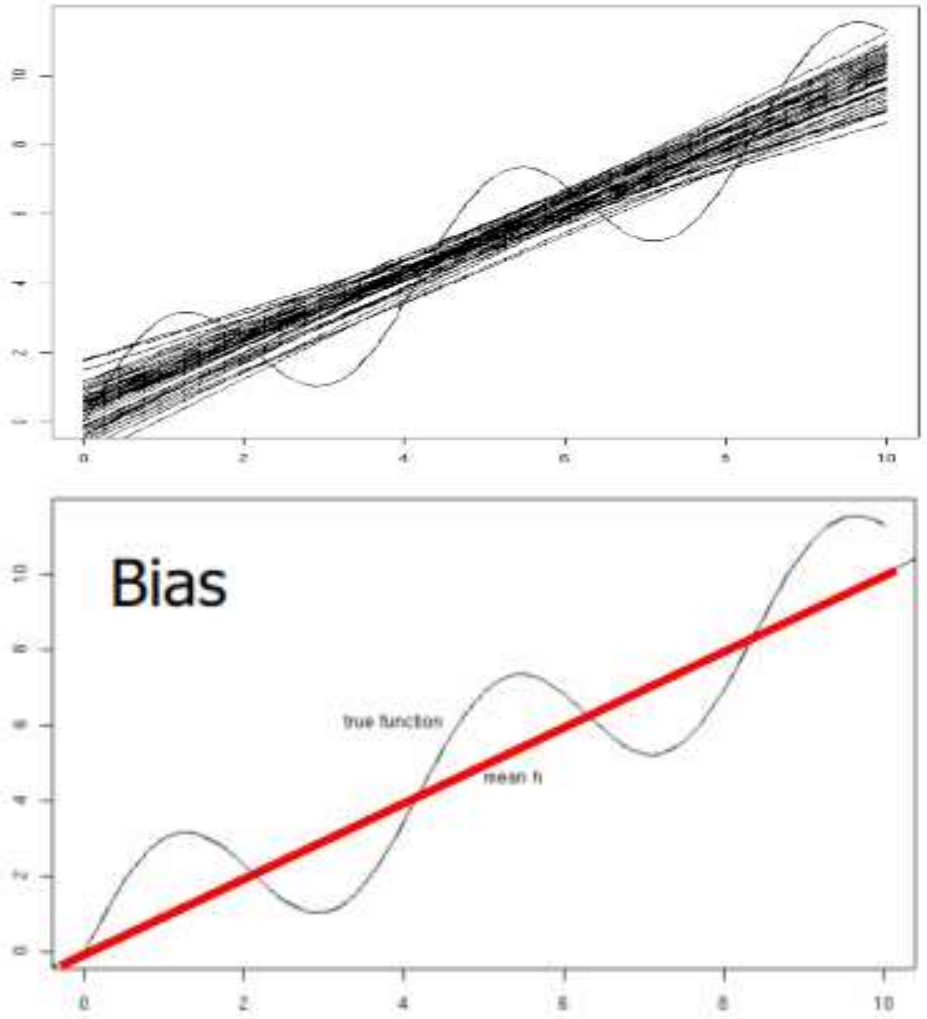
Bias

Le Bias c'est la différence entre la modèle cible $f(x)$ et le modèle qu'on espère apprendre $\bar{g}(x)$.

$$Bias^2 = E_x[(\bar{g}(x) - f(x))^2]$$

$$Bias^2 = \int_x (\bar{g}(x) - f(x))^2 p(x) dx$$

$$Bias^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\bar{g}(x_i) - f(x_i)\right)^2$$



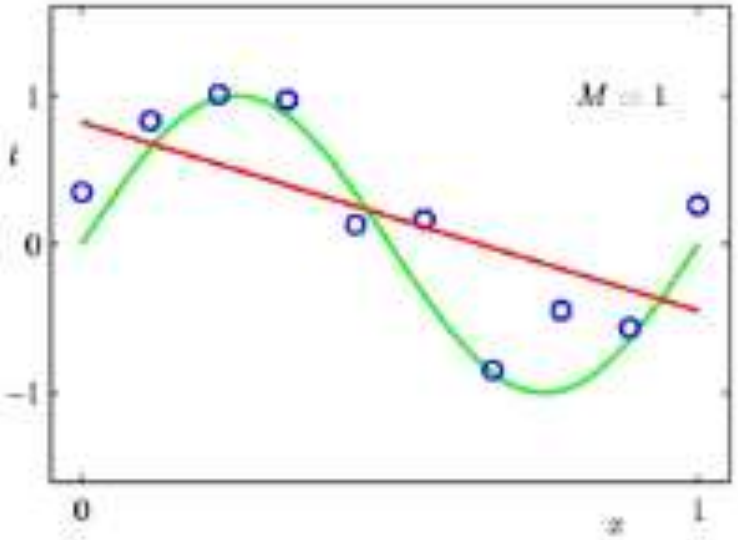
Bias

Pour un nombre fixe de données:

- Faibles approximateurs:

➢ Degré de polynôme faible.

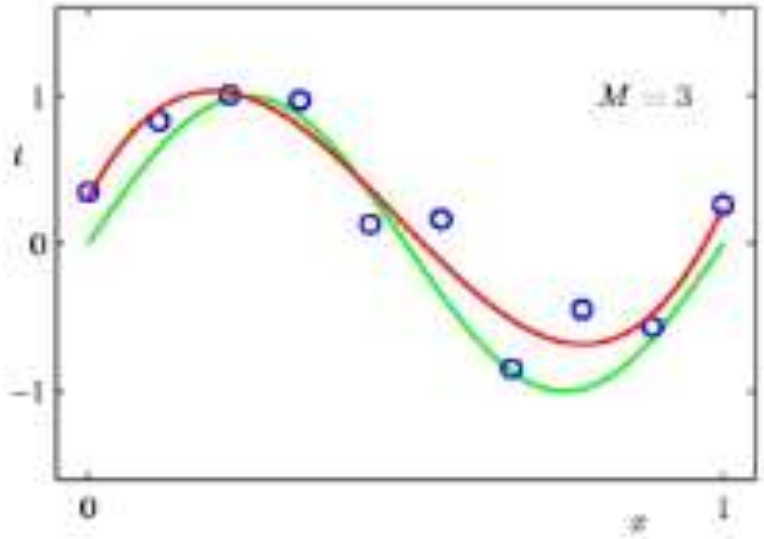
➢ Fort bias.



- Forts approximateurs:

➢ Degré de polynôme fort.

➢ Faible bias.



Le Bias disparait quand on choisit le modèle parfait.

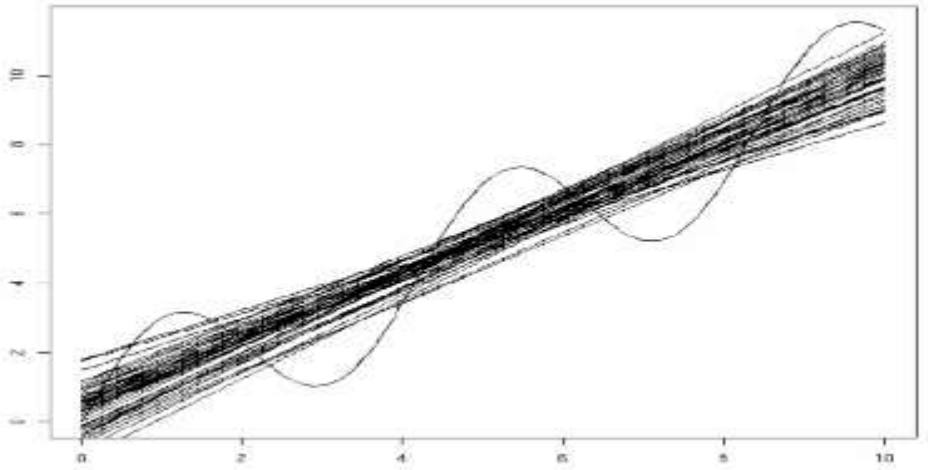
Variance

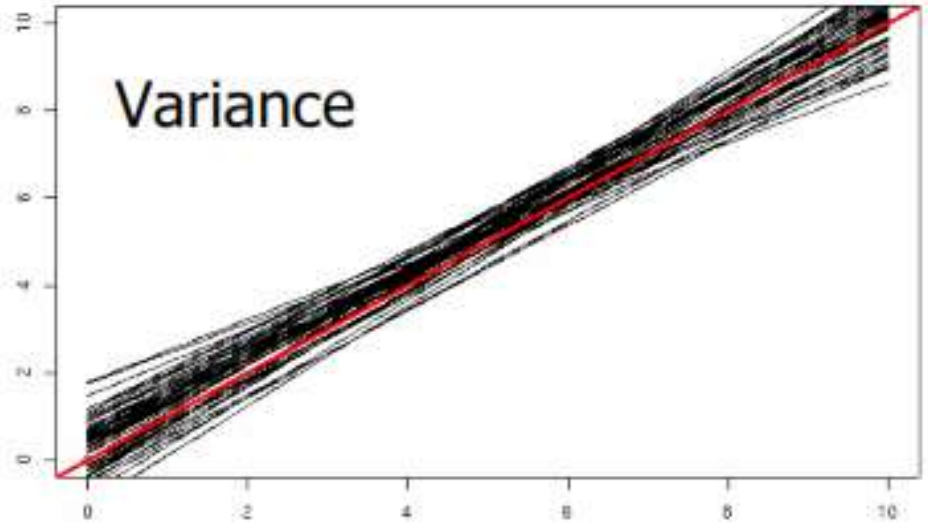
La Variance c'est la différence entre le modèle estimé dans un ensemble de données D (ce qu'on a appris dans D) $g^{(D)}(x)$ et le modèle qu'on espère apprendre $\bar{g}(x)$.

$$Variance = E_x[E_D[\left(g^{(D)}(x) - \bar{g}(x)\right)^2]]$$

$$Variance = \int_x E_D[\left(g^{(D)}(x) - \bar{g}(x)\right)^2]p(x)dx$$

$$= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{L}\sum_{j=1}^{L}\left(g^{(j)}(x_i) - \bar{g}(x_i)\right)^2$$





Variance

On remarque que la variance n'a pas une dépendance directe sur le modèle réel.
Pour un nombre fixe de données:

- Faibles approximateurs:
  - ➢ Degré de polynôme faible.
  - ➢ Faible variance.
  - ➢ Différents ensembles de données vont générer les MEME modèles estimés.
- Fort approximateurs:
  - ➢ Degré de polynôme fort.
  - ➢ Forte variance.
  - ➢ Différents ensembles de données vont générer DIFFERENTS modèles estimés.

La variance disparait quand $|D| \rightarrow \infty$.

Bias – Variance

- Les modèles **flexibles** possèdent un faible bias et une forte variane.
- Les modèles **rigides** possèdent un fort bias et une faible variance.
- Les modèles **optimaux** sont caractérisés par un équilibre bias/variance.

- Si le modèle est **très simple**, on dit que la solution est biaisée et ne s'adapte pas aux données.
- Si le modèle est **très complexe**, donc il est très sensible aux petits changements des données.

Réf.

Bruit

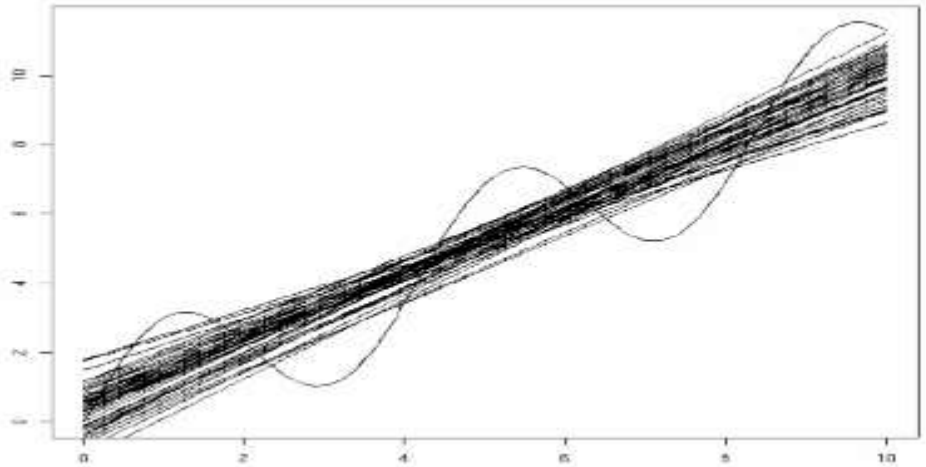
Le Bruit c'est la différence entre le modèle cible $f(x)$ et les valeurs observées y.
On sait que :

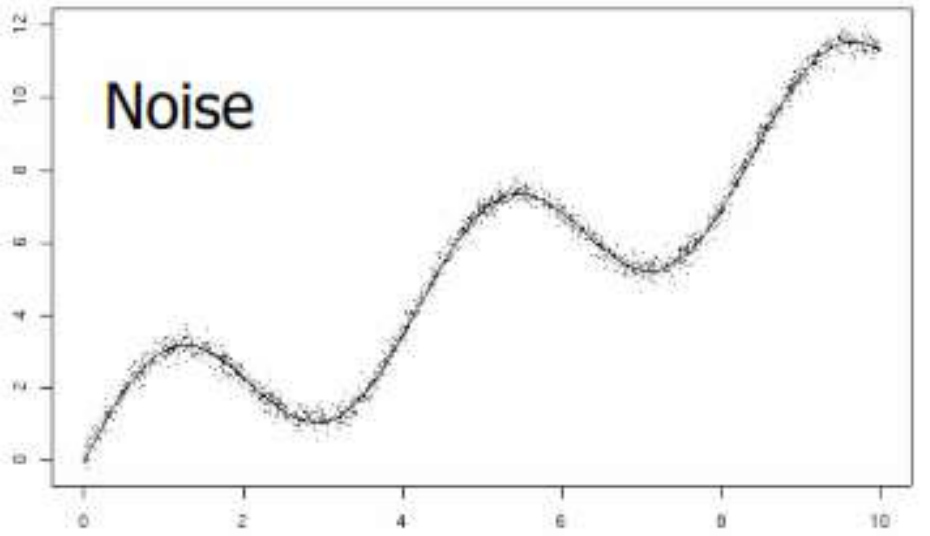
$$y = f(x) + \varepsilon$$

$$Bruit = E_x[(f(x) - y)^2] = E_x[\varepsilon^2]$$

$$Bruit = \int_x (f(x) - y)^2 p(x) dx$$

$$Bruit = \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(\varepsilon_i)^2$$





Bruit stochastique

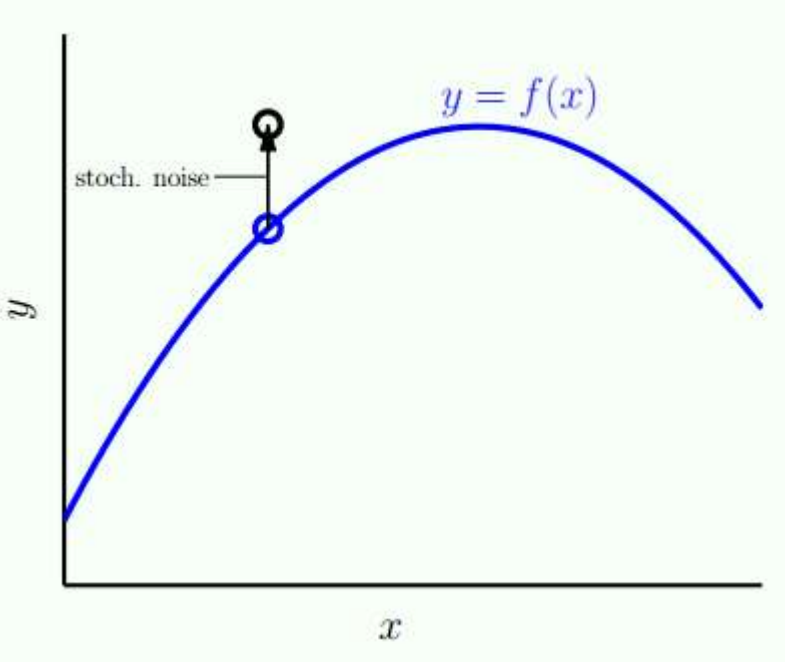
Le bruit stochastique c'est la partie de f qu'on ne peut pas modéliser.

On veut apprendre:

$$\boldsymbol{y_n = f(x_n)}$$

Mais malheureusement on observe:

$$\boldsymbol{y_n = f(x_n) + bruit\ stochastique}$$

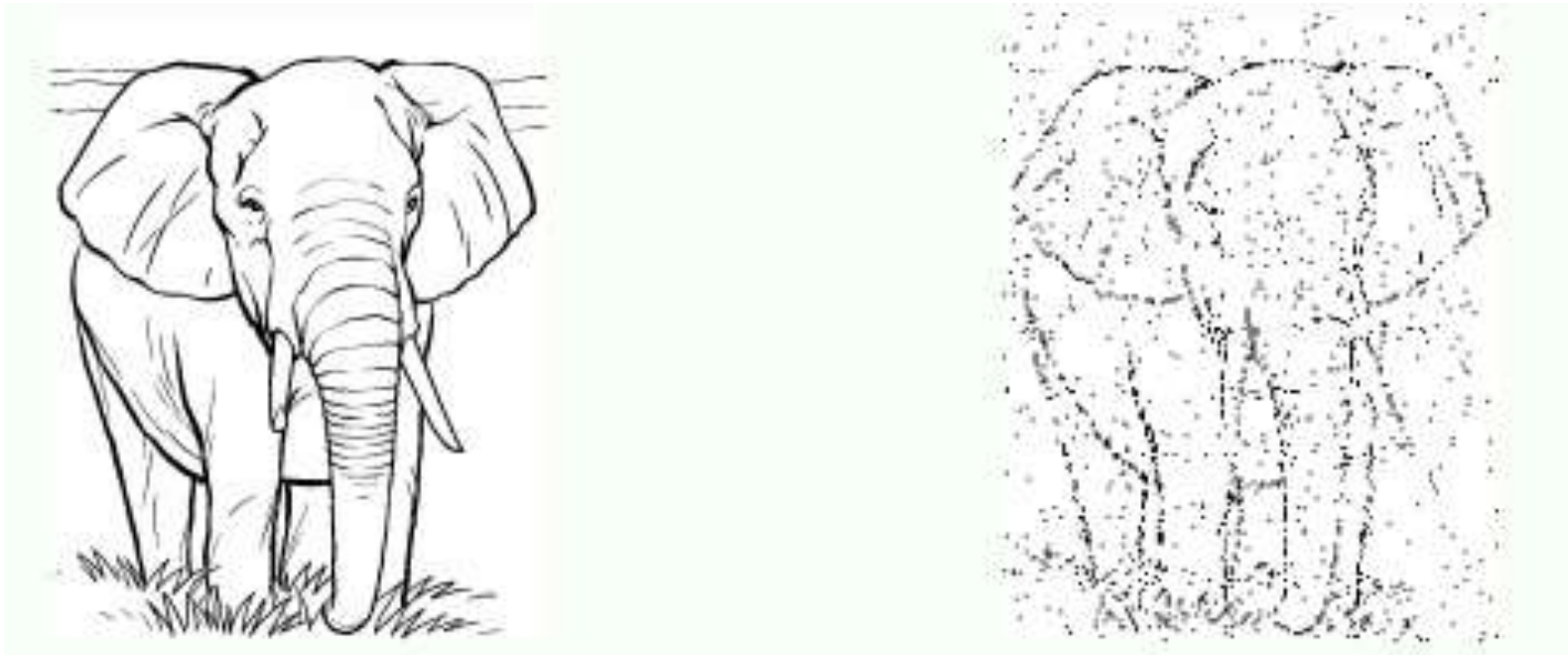


Personne ne peut modéliser le bruit stochastique.

Réf.

Bruit stochastique

Le bruit stochastique c'est l'erreur due aux mesures et aux fluctuations qu'on ne peut pas modéliser.



Réf.

Bruit déterministe

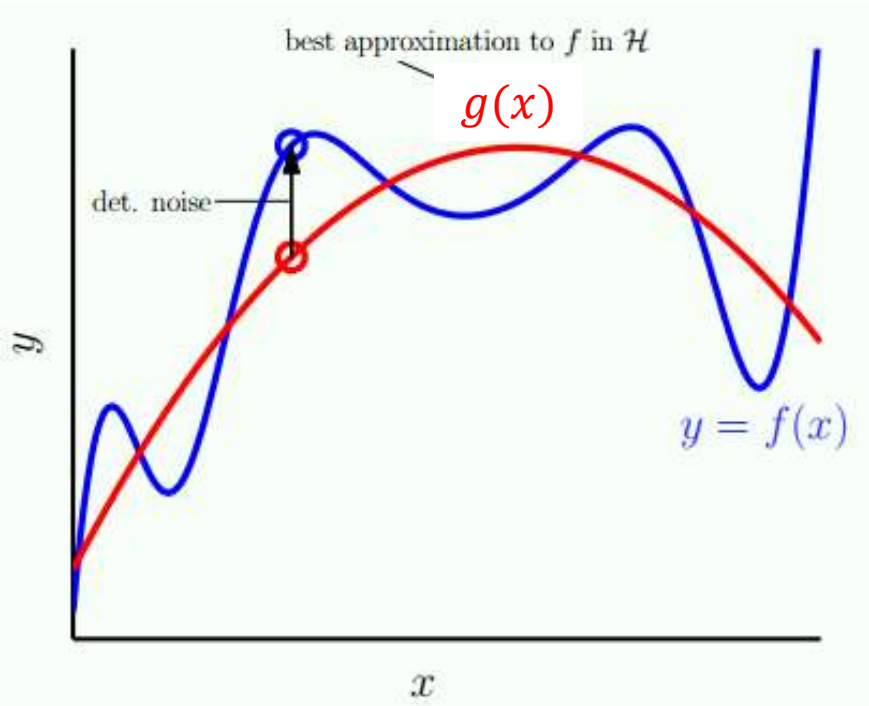
Le bruit déterministe c'est l'erreur (bias) due au choix du modèle.

On veut apprendre:

$$y_n = h(x_n)$$

Mais malheureusement on observe:

$$y_n = f(x_n) = h(x_n) + \textbf{bruit déterministe}$$



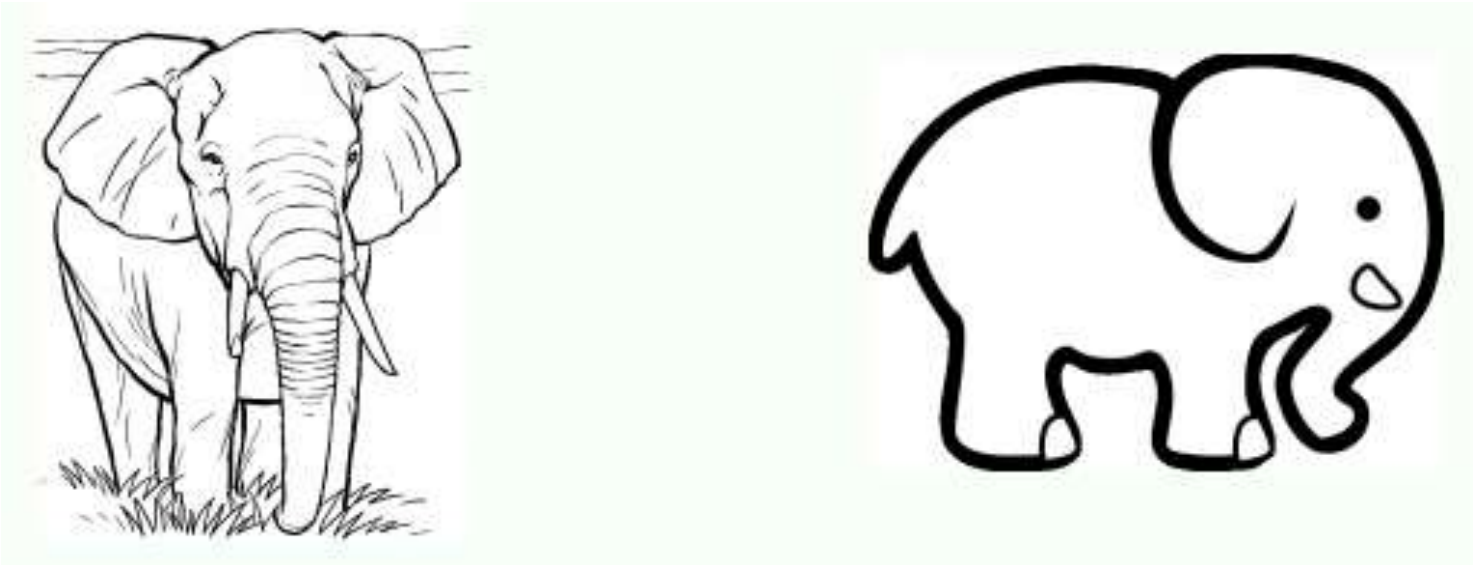
H ne peut pas modéliser le bruit déterministe.

Incapacité de h à modéliser correctement f.

Réf.

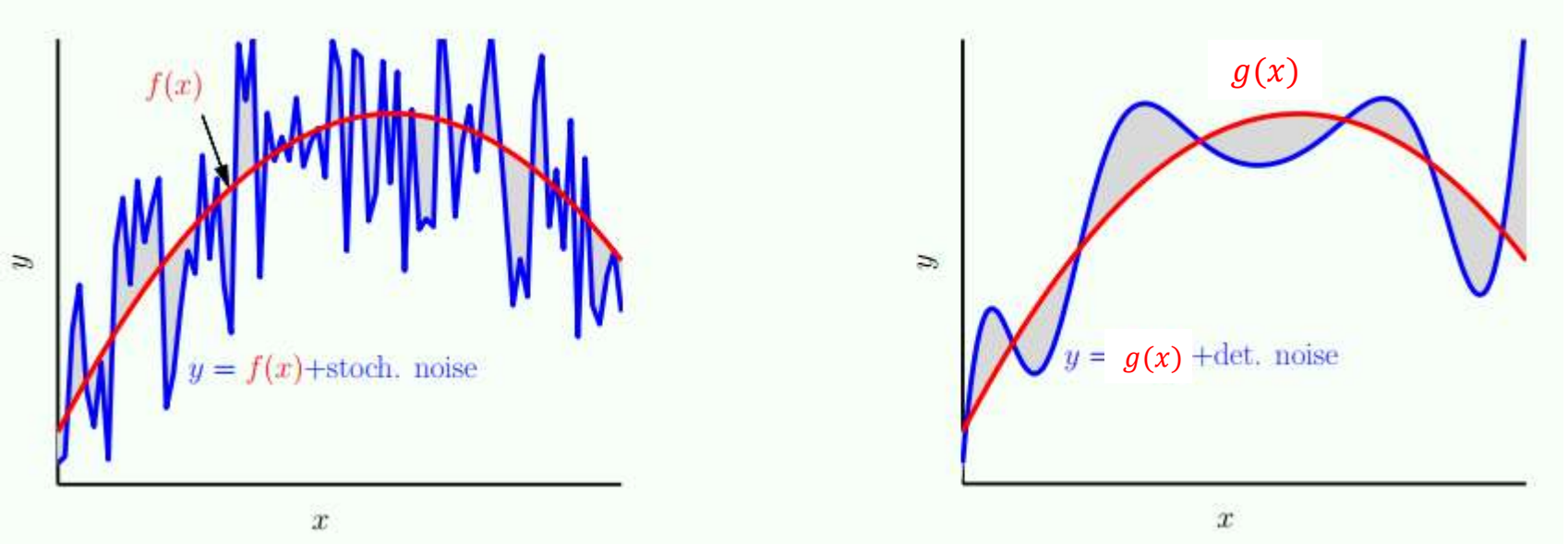
Bruit déterministe

Le bruit déterministe c'est la partie de f qu'on ne peut pas modéliser.



Réf.

Bruit stochastique – Bruit déterministe



- Source: mesures aléatoires.

Mesurer y_n une autre fois:

➢Le bruit stochastique change.

Changer H:

➢Le bruit stochastique est le même.

- Source: H ne peut pas modéliser f.

Mesurer y_n une autre fois:

➢Le bruit déterministe est le même.

Changer H:

➢Le bruit déterministe change.

Exemple

$f(x) = sin(\pi x)$

$x \in [-1,1]$

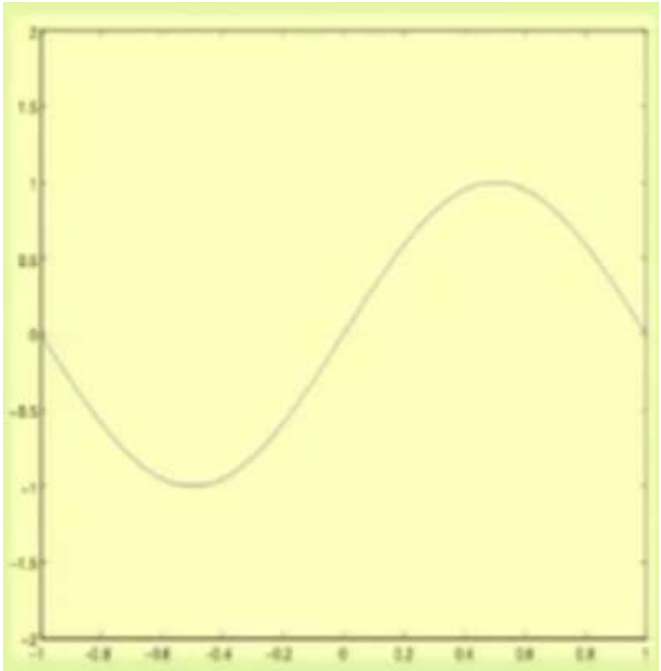
n=2 : (x_1, y_1) et (x_2, y_2)

On va utiliser deux modèles :

$$H_0: \quad h(x) = b$$

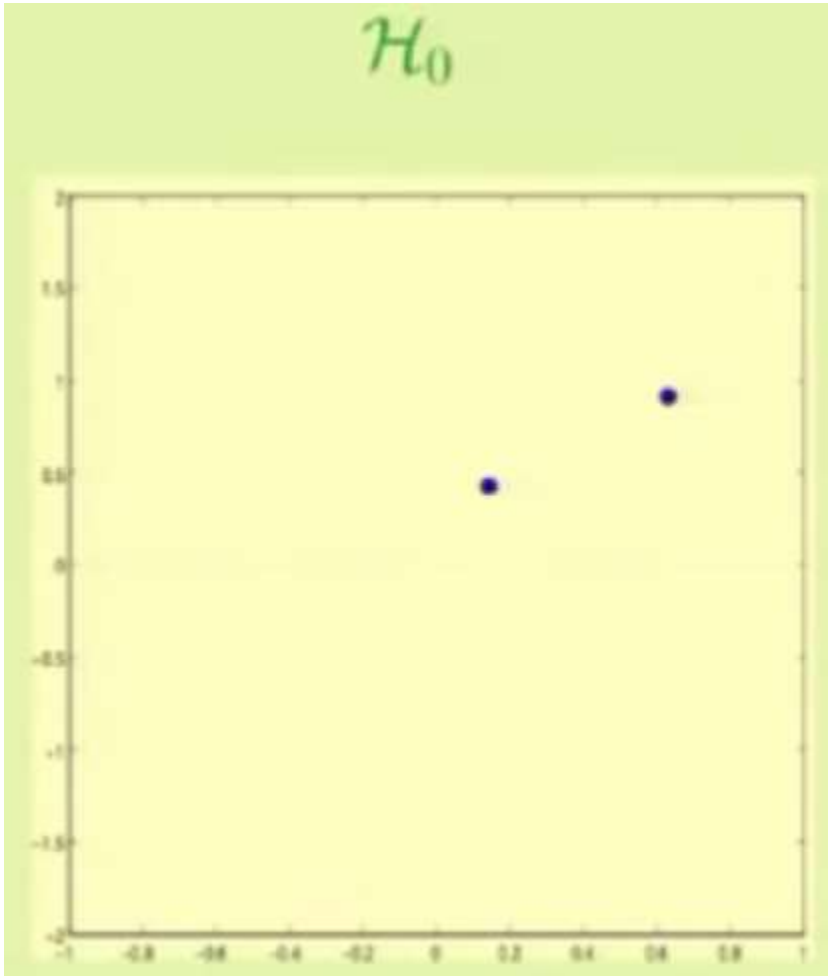
$$H_1: \quad h(x) = ax + b$$

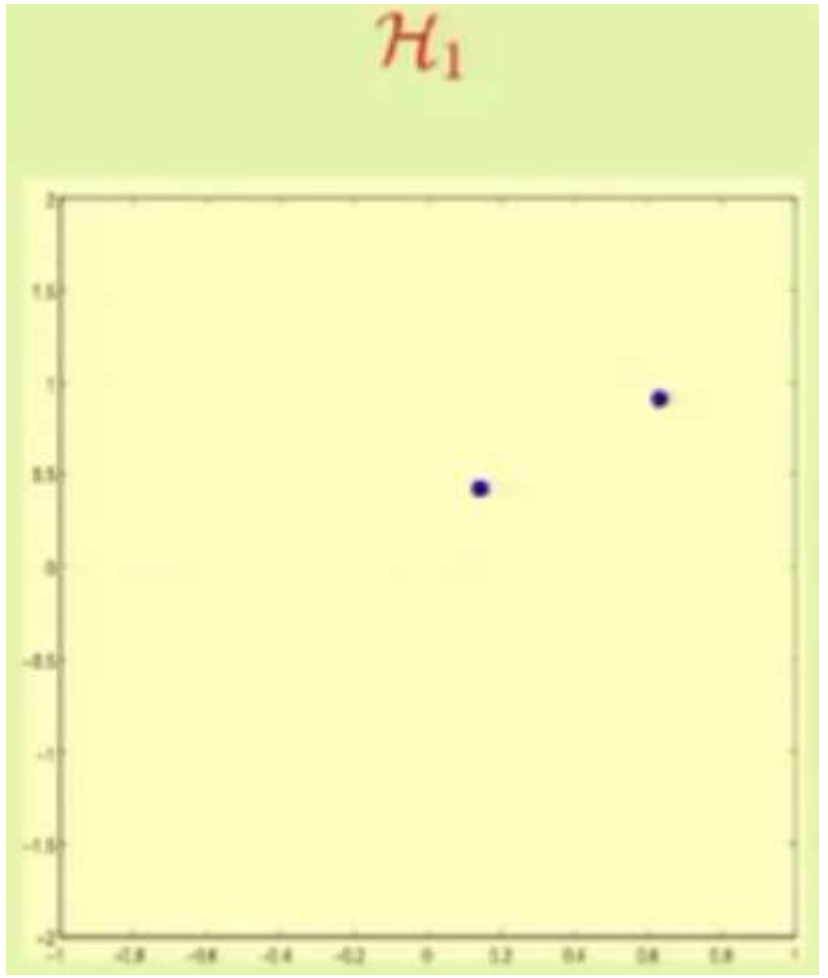
Quelle est la meilleure hypothèse H_0 ou H_1?



Réf.

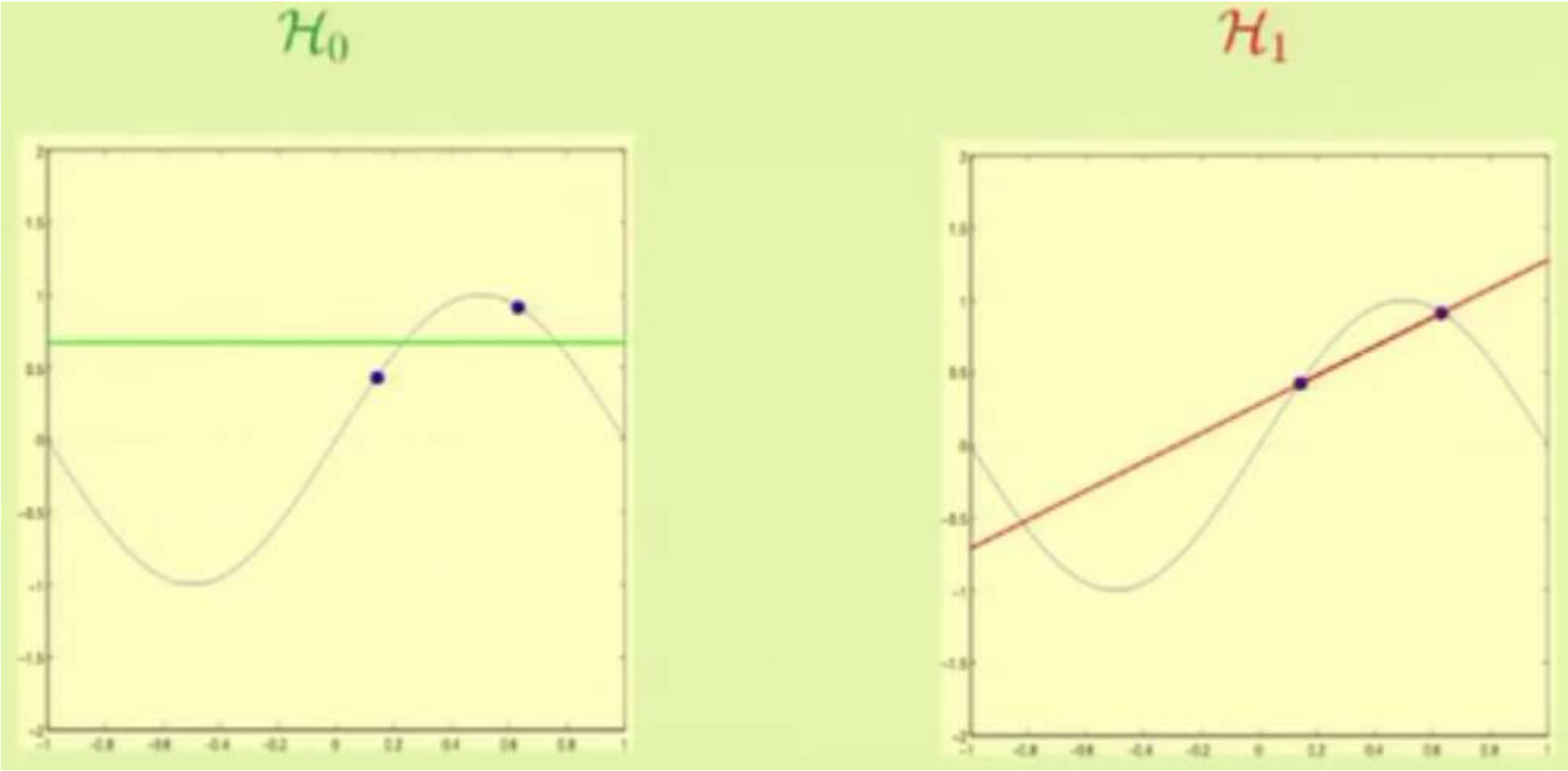
Exemple : Apprentissage





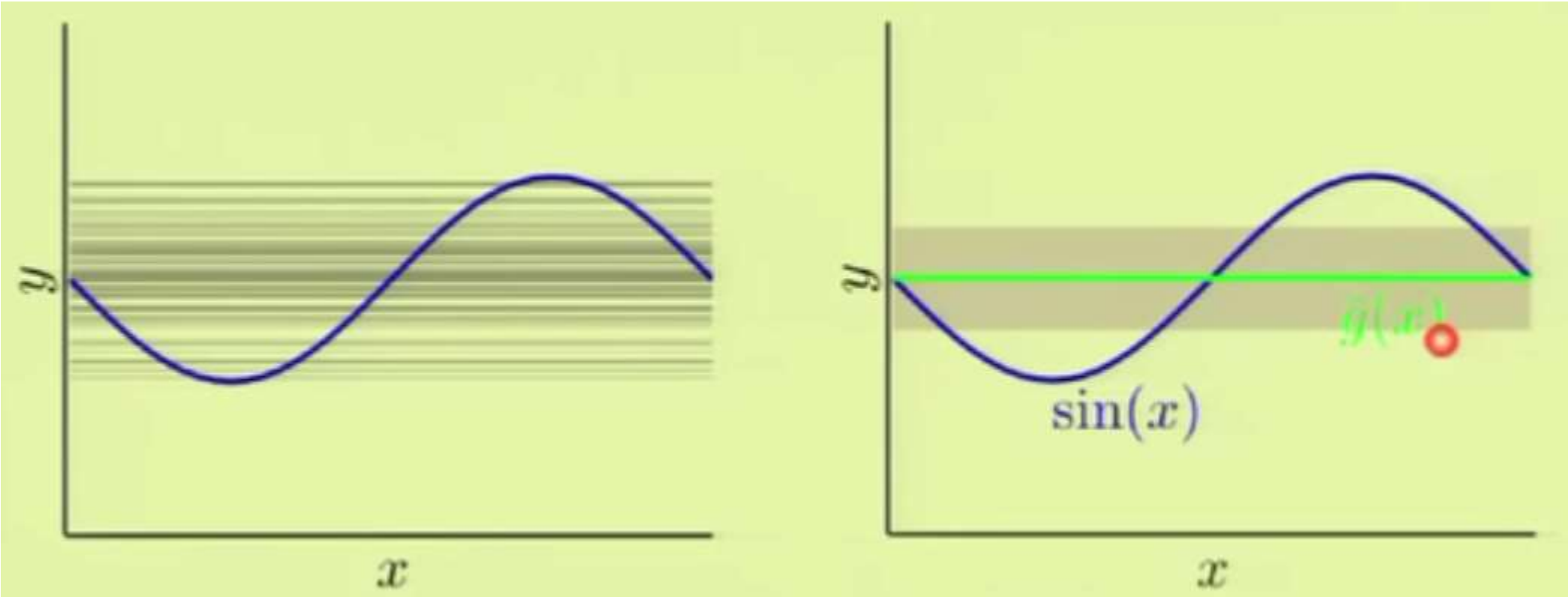
Réf.

Exemple : Apprentissage



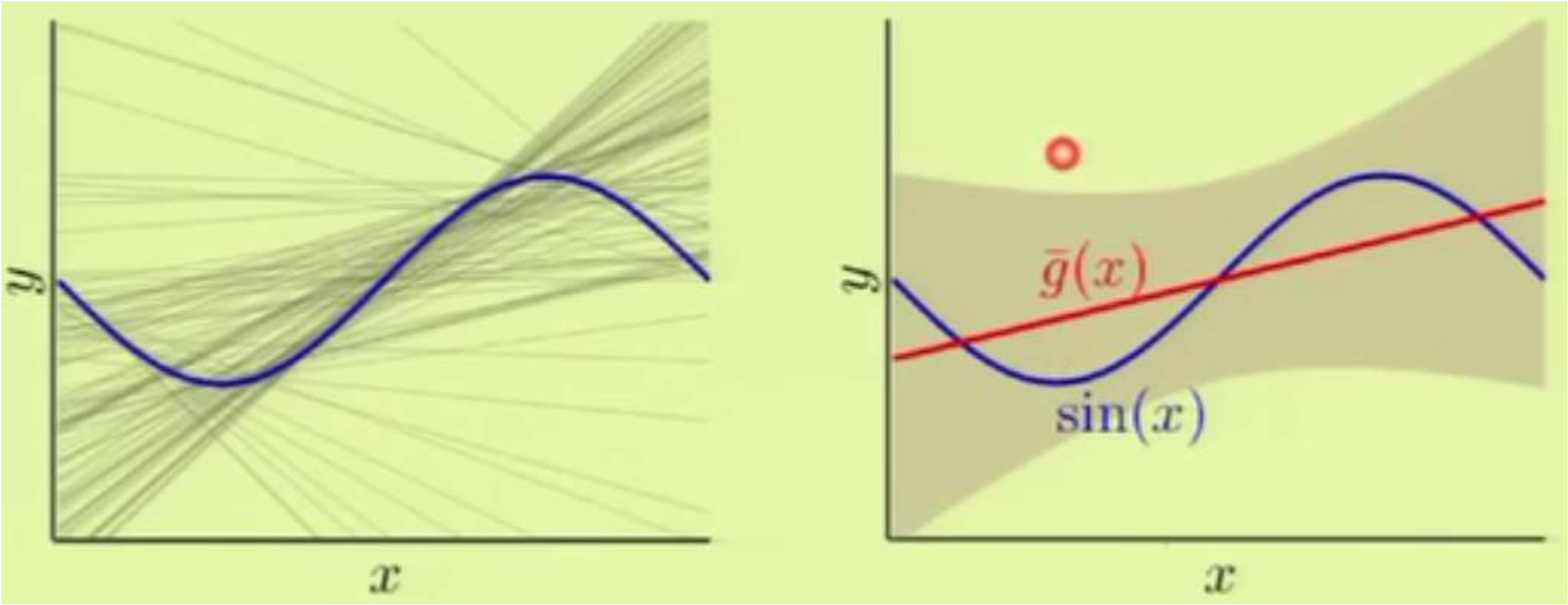
Réf.

Exemple : Bias et variance de H_0



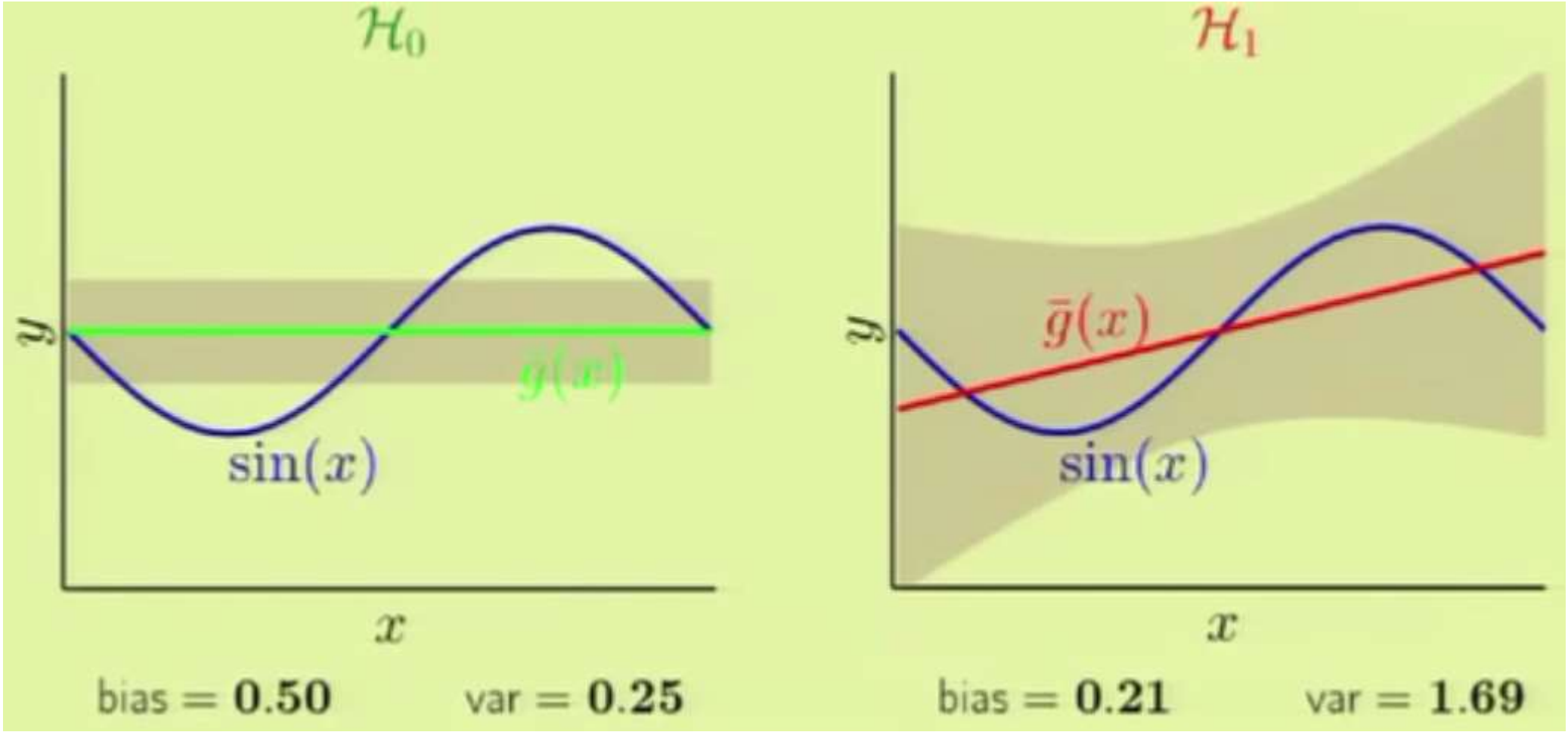
Réf.

Exemple : Bias et variance de H_1



Réf.

Exemple : Quelle est la meilleure H_0 ou H_1 ?

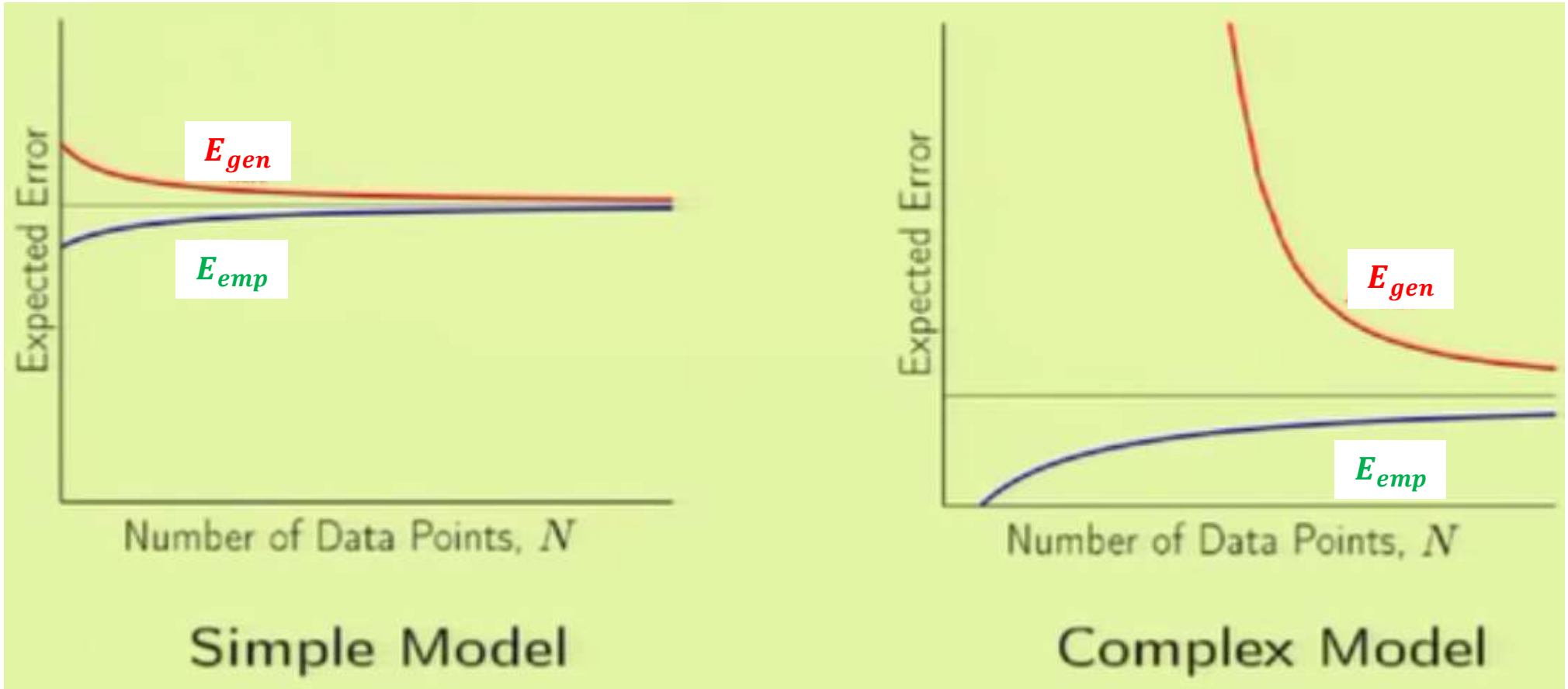


$$E_D[E_{gen}(g^{(D)})] = 0,75$$

$$E_D[E_{gen}(g^{(D)})] = 1,90$$

Réf.

Les courbes d'apprentissage



Réf.

2. Le ‘’Overfitting’’

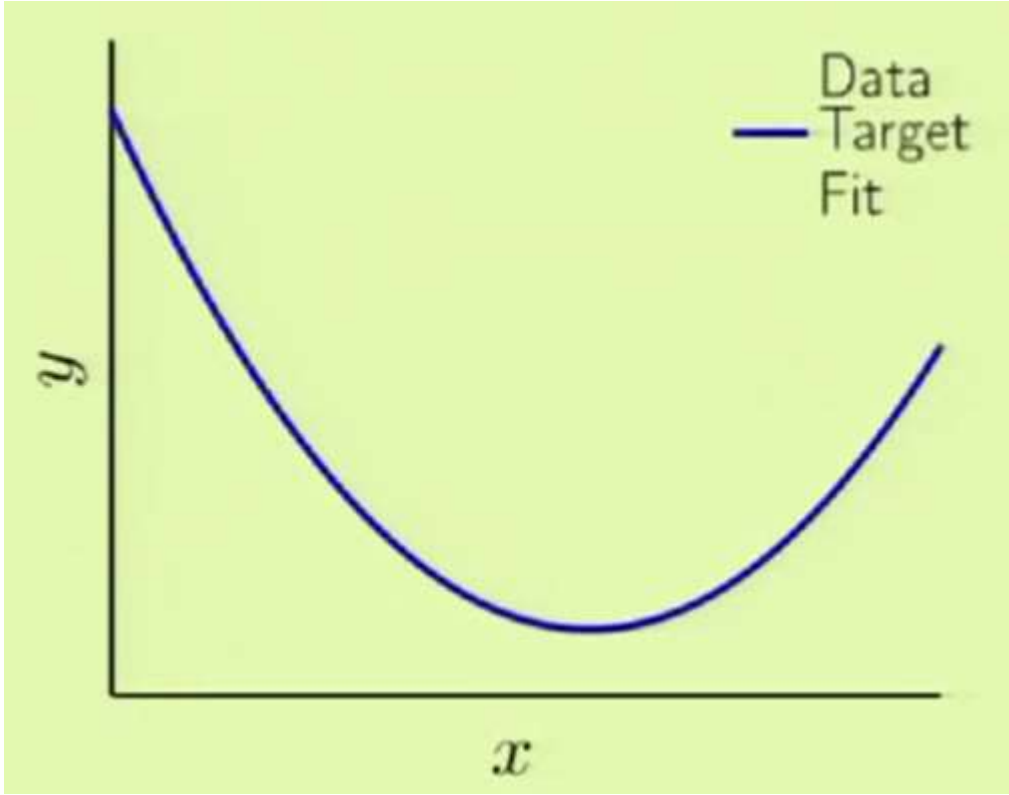
Définition de l'Overfitting

- C'est ce qui fait la différence entre un amateur et un professionnel du machine learning.
- C'est un terme comparatif qui veut dire le sur-apprentissage.
- C'est capturer les données plus que demandée.
- Il se manifeste quand on choisit une hypothèse dont E_{emp} est petite, mais E_{gen} est grande.
- Donc E_{emp} n'est plus le bon guide pour l'apprentissage.

Réf.

Causes de l'Overfitting

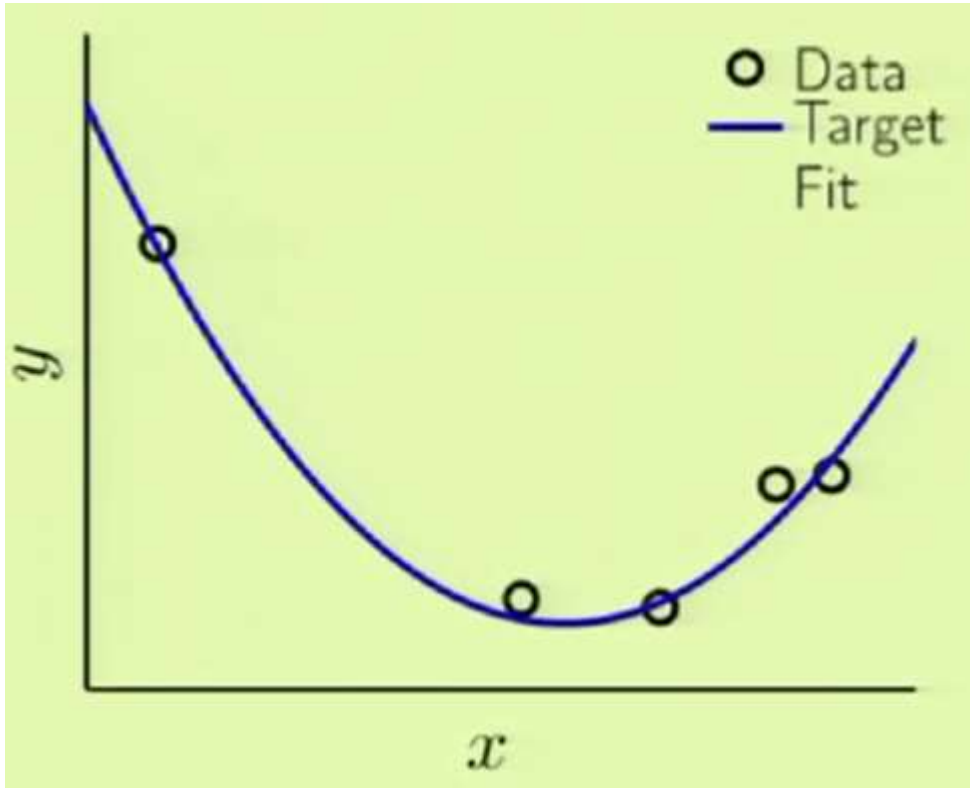
Considérons un problème de régression simple à une seule dimension.

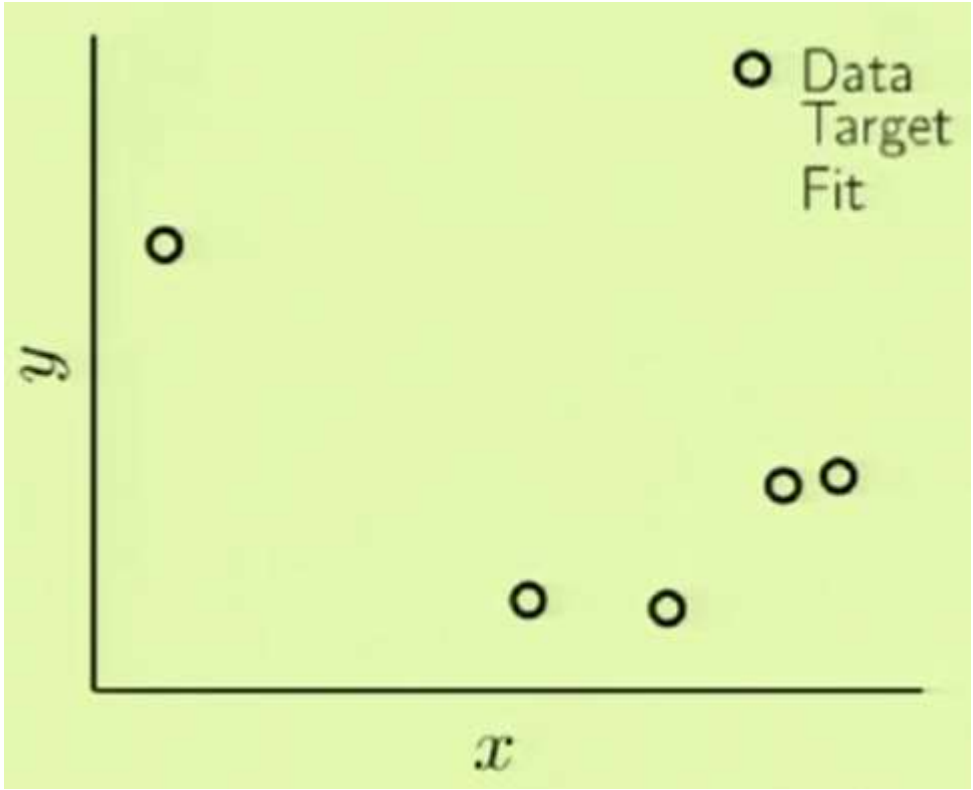


Réf.

Causes de l'Overfitting

- La fonction cible est inconnue.
- Elle est connue à travers 5 points de données bruités.

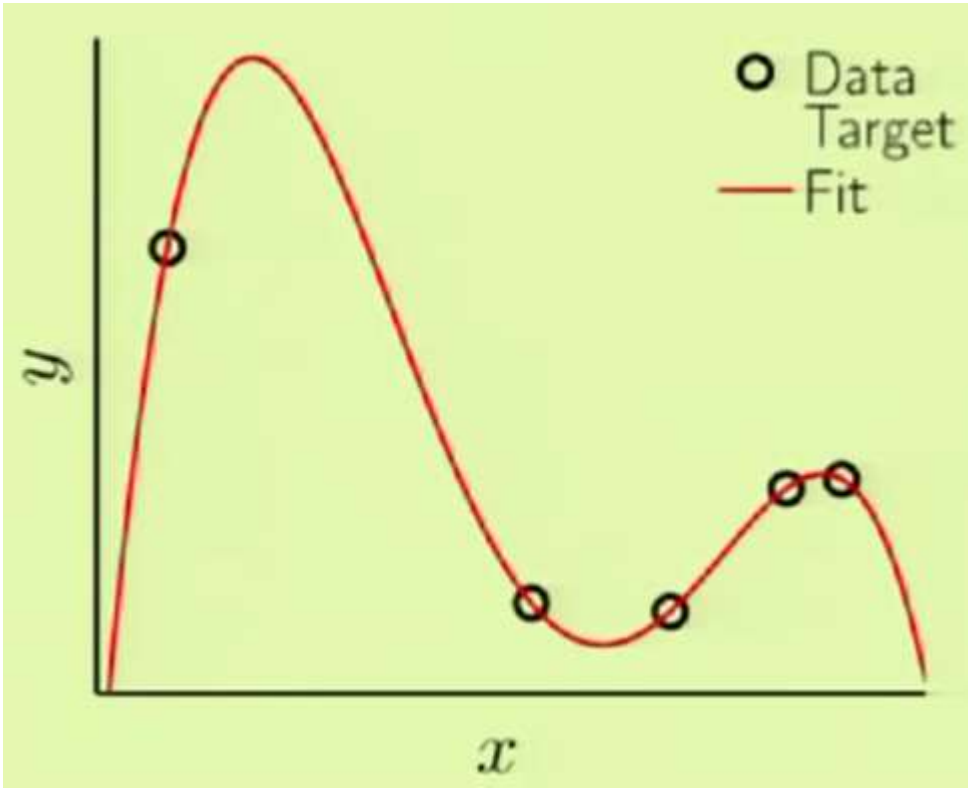




Réf.

Causes de l'Overfitting

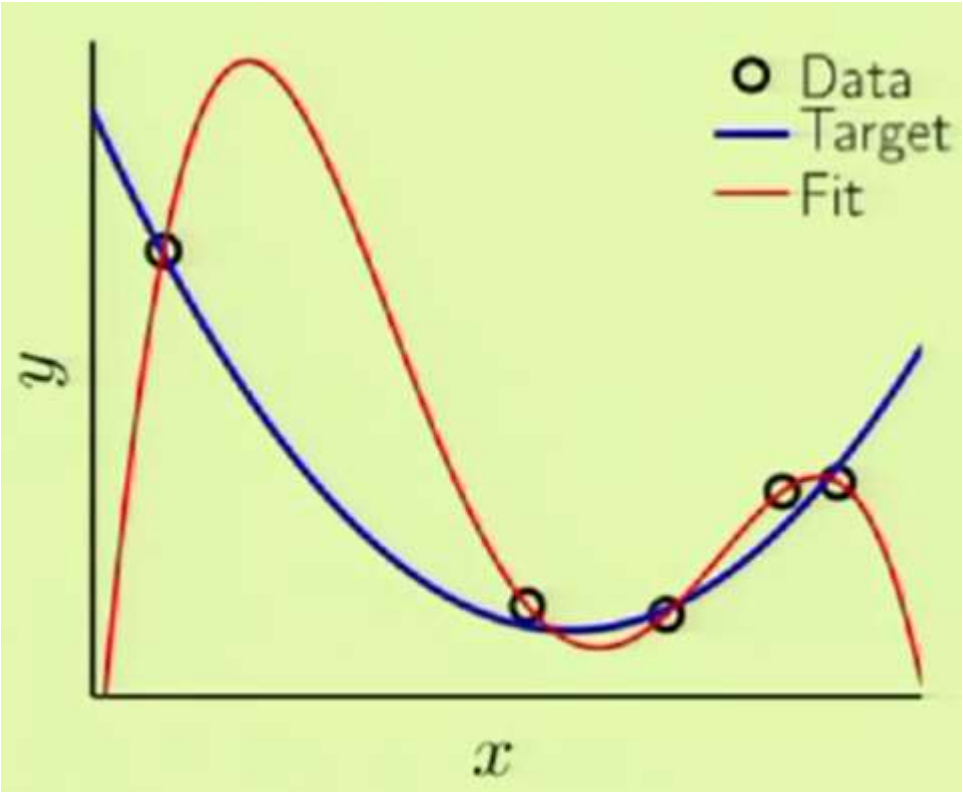
Capturons les 5 points : polynôme d'ordre 4.



Réf.

Causes de l'Overfitting

En réalité la fonction cible est un polynôme d'ordre 2 impactée par le bruit.



Réf.

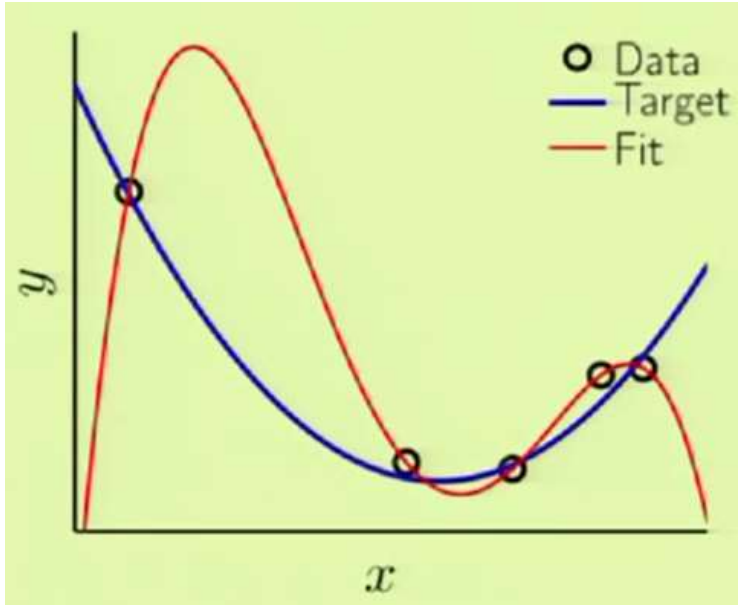
Causes de l'Overfitting

Il existe une grande différence entre la fonction cible et approximative.

- Scénario de sur-apprentissage des données.
- Le bruit a dévoyé l'apprentissage.
- Le modèle complexe utilise un nombre supplémentaire

de degrés de liberté pour capturer le bruit.

- $E_{emp} = 0$ et $E_{gen} = très\ grande$.
- Mauvaise généralisation

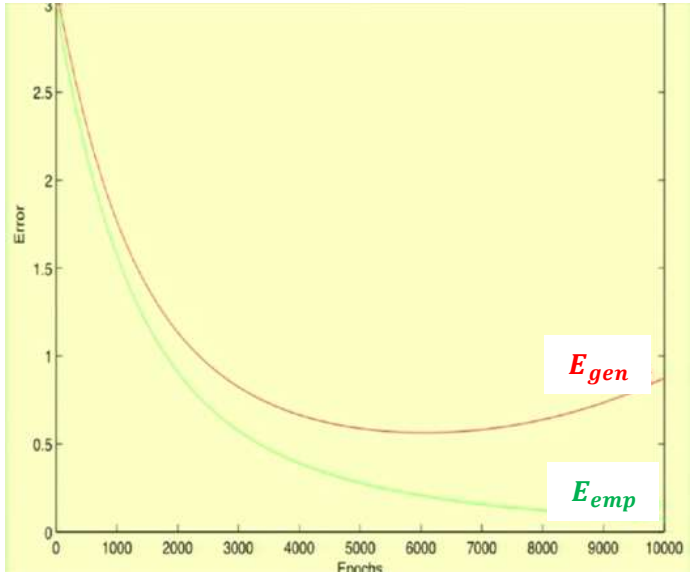


Réf.

Mauvaise généralisation Versus Overfitting

La mauvaise généralisation:

- Compare E_{emp} et E_{gen} du même modèle.
- Implique une grande différence entre E_{emp} et E_{gen}.



Le Overfitting:

- Compare E_{emp} et E_{gen} de deux modèles ou deux itérations différentes.
- Implique que lorsque $E_{emp}\ diminue \rightarrow E_{gen}\ augmente$.

Réf.

Le Overfitting chez la régression polynomiale

- Lorsque la relation entre x et y est non linéaire, on utilise la régression polynomiale pour capturer cette relation.
- Soit un problème de régression polynomiale à une seule dimension:

$$y = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_Q x^Q + \varepsilon$$

Q est le degré du polynôme.

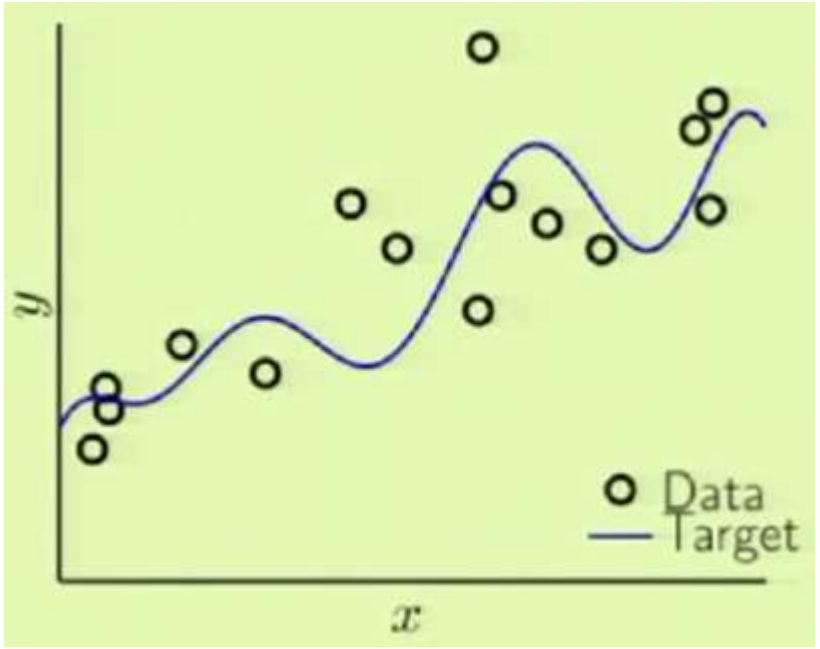
- La régression polynomiale est considérée comme un modèle linéaire en termes de $(\alpha_0, \alpha_1, \ldots, \alpha_Q)$ car les x sont des constantes.

Réf.

Quand survient le Overfitting?

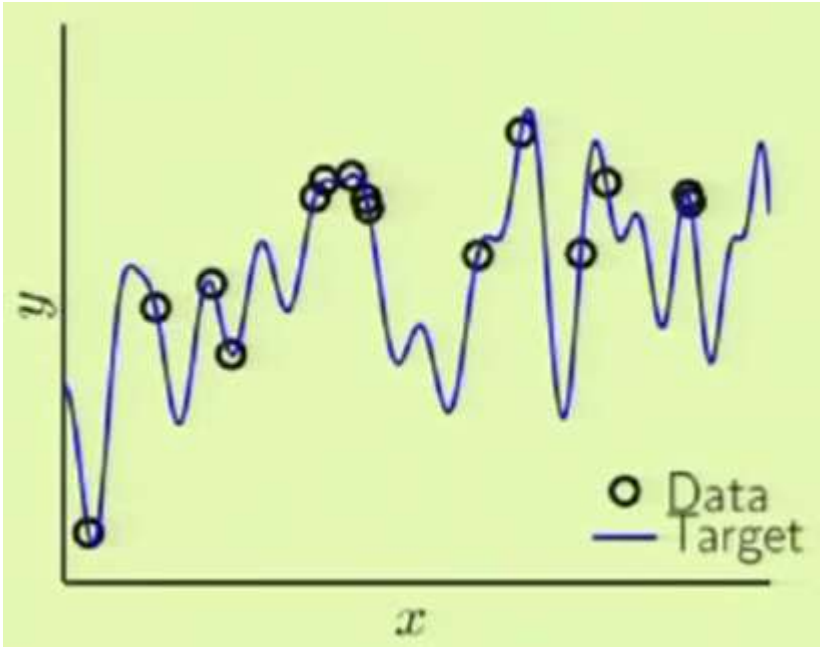
- La cible est une fonction polynomiale.
- L'ensemble de données D contient 15 points.

Cible polynomiale d'ordre 10 + Bruit



Fonction cible bruitée d'ordre inférieur

Cible polynomiale d'ordre 50 sans Bruit



Fonction cible non bruitée d'ordre supérieur

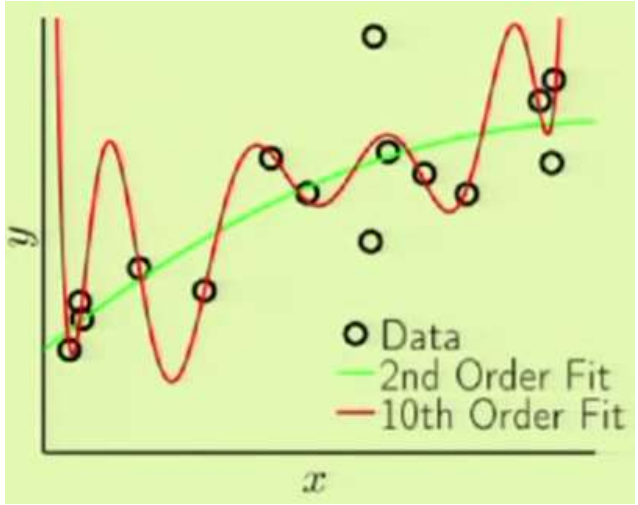
Réf.

Quand survient le Overfitting?

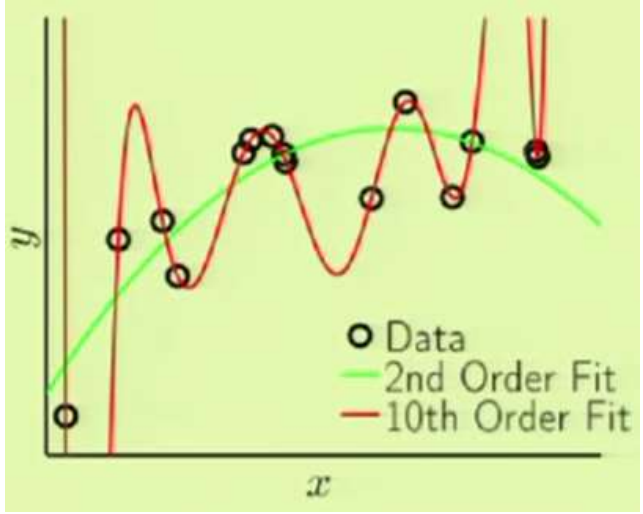
Pour capturer ces données, on a utilisé deux modèles :

- Fonction polynomiale d'ordre 2
- Fonction polynomiale d'ordre 10
- L'algorithme d'apprentissage ne voit pas la fonction cible mais seulement les données.

Fonction cible bruitée d'ordre inférieur



Fonction cible non bruitée d'ordre supérieur

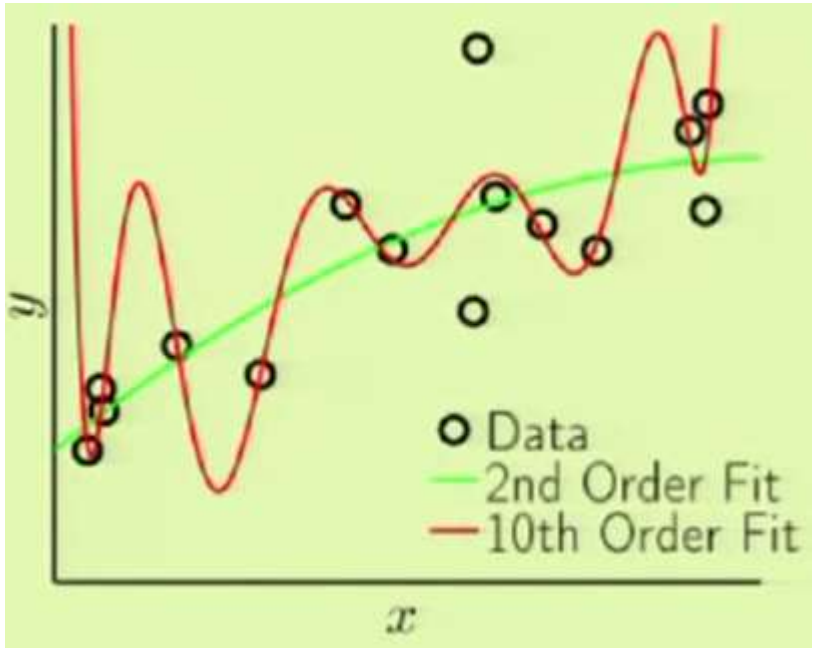


Réf.

Quand survient le Overfitting?

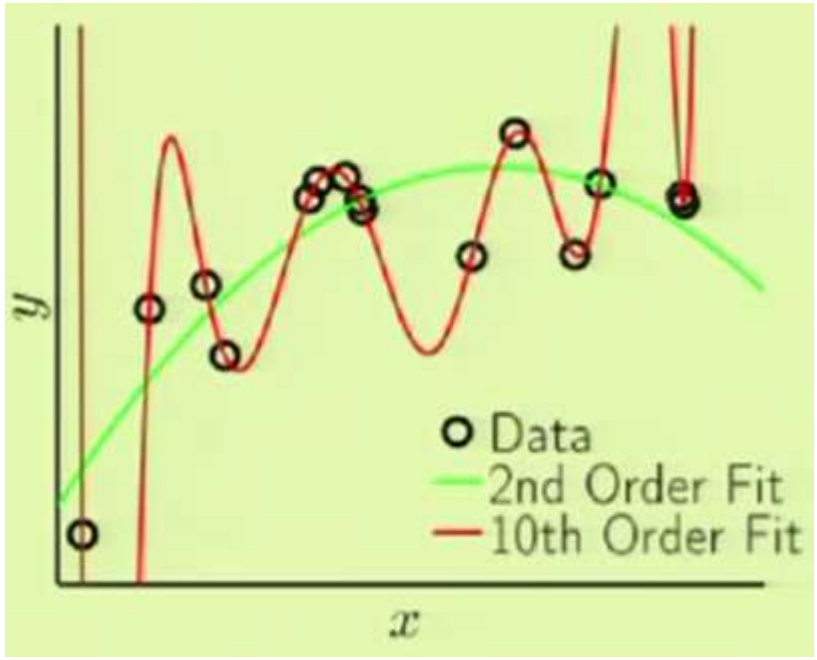
Le modèle simple est meilleur que le modèle complexe.

Fonction cible bruitée d'ordre inférieur



	2nd Order	10th Order
E_{emp}	0.050	0.034
E_{gen}	0.127	9.00

Fonction cible non bruitée d'ordre supérieur

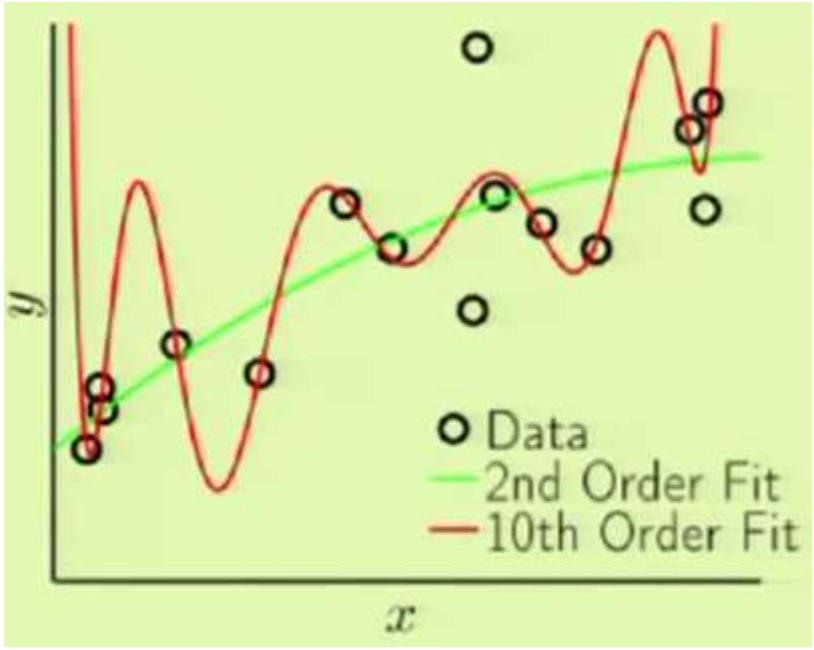


	2nd Order	10th Order
E_{emp}	0.029	10^{-5}
E_{gen}	0.120	7680

Réf.

Quand survient le Overfitting?

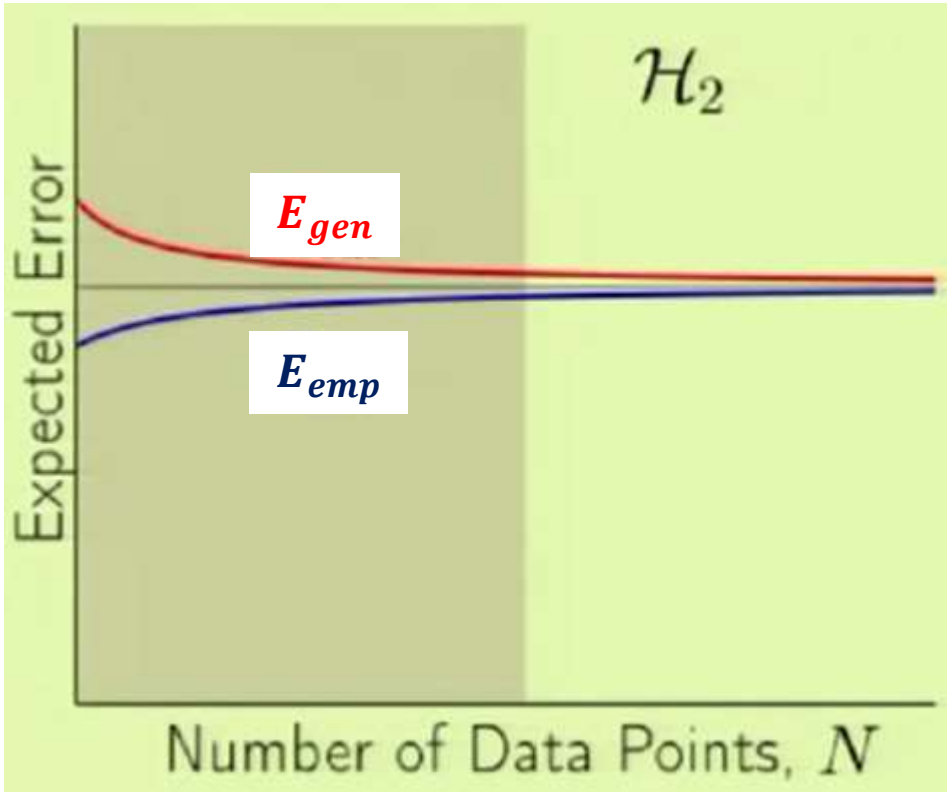
- Considérons le cas d'une cible polynomiale d'ordre 10.
- Soit deux modèles d'apprentissage : R et O.
  - R : un modèle restreint, il choisit H_2
  - O : un modèle de sur-apprentissage, il choisit H_{10}

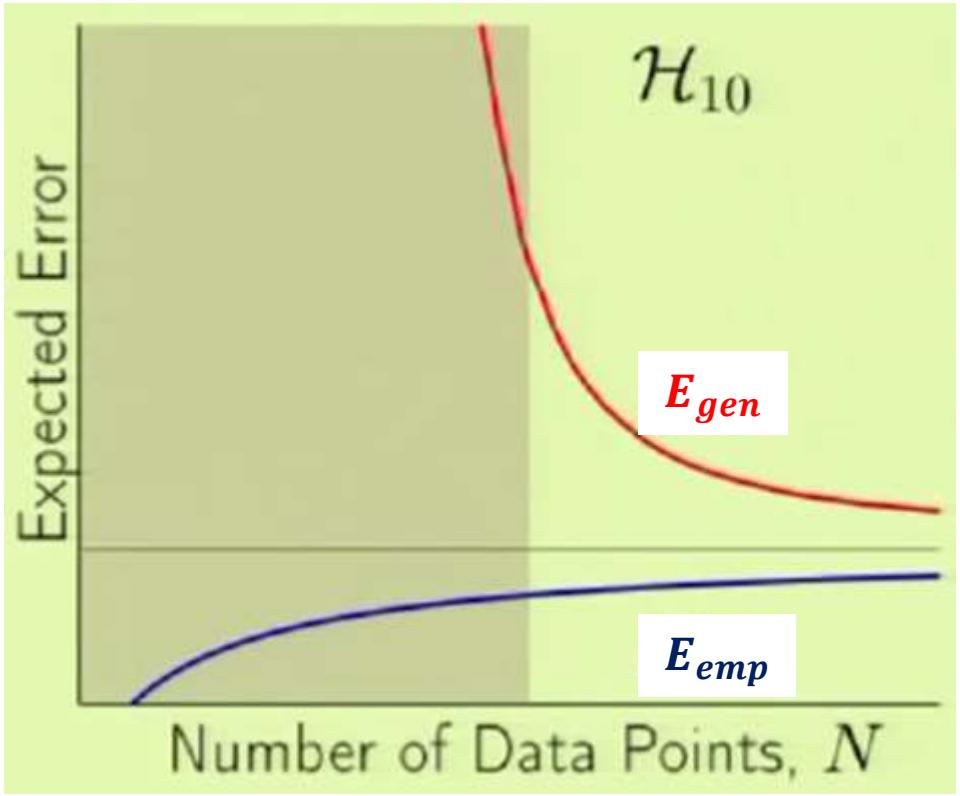


Réf.

Quand survient le Overfitting?

- Il existe un intervalle dans N où le modèle O possède une grande E_{gen} et une faible E_{emp}.
- Existence de l'Overfitting dans H_{10}





Réf.

Les catalyseurs de l'Overfitting

- Ils présentent les conditions qui impactent le phénomène de l'Overfitting.
- Considérons le cas d'une fonction cible bruitée:

$$y = f(x) + \varepsilon(x)$$

Tels que $\varepsilon(x)$ est un bruit blanc et:

$$\mathrm{y} = \sum_{q=0}^{Q_f} \alpha_q x^q + \varepsilon(x)$$

Q_f est le degré du polynôme qui décrit la fonction cible complexe f.

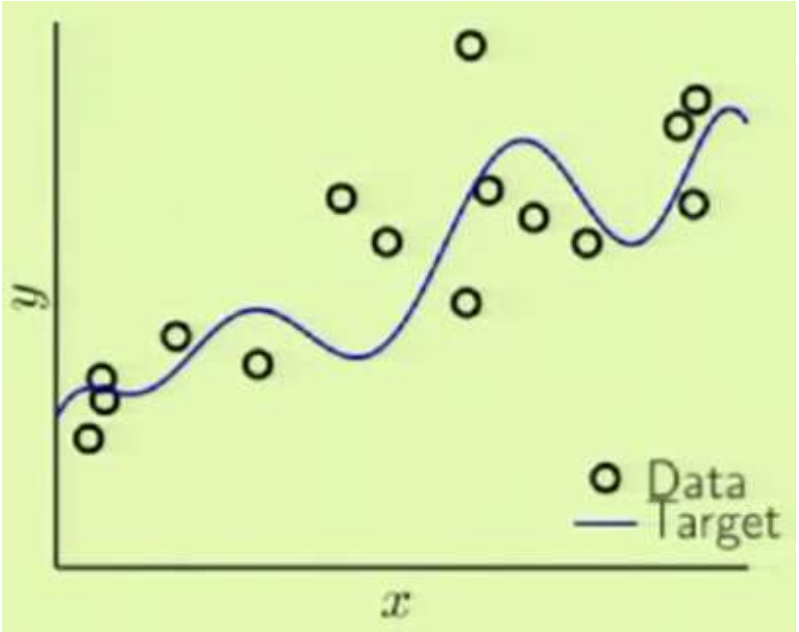
- Il faut noter que $\sum_{q=0}^{Q_f} \alpha_q x^q$ doit être normalisée afin de générer des α_q orthogonaux entre eux.
- On sait déjà que : $E_x[\varepsilon(x)] = \sigma^2$

Réf.

Les catalyseurs de l'Overfitting

❑ Les éléments qui impactent le overfitting sont:

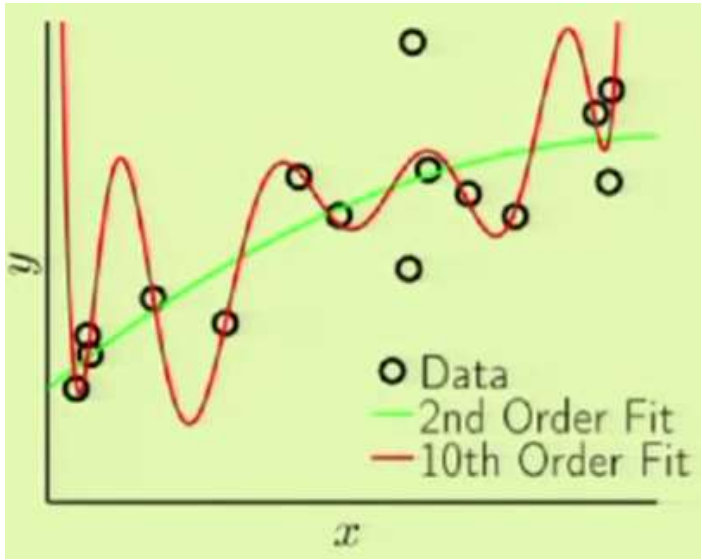
- σ^2 : Le bruit stochastique.
- Q_f : Le bruit déterministe.
- N : La taille de l'ensemble de données qui présente f.



Réf.

La mesure de l'Overfitting

- Supposons qu'on veut capturer les données: $(x_1, y_1), \ldots, (x_N, y_N)$ d'une fonction cible polynomiale d'ordre Q_f.
- On utilisera les modèles H_2 et H_{10}:
  - Le modèle H_{10} a généré $g_{10} \in H_{10}$.
  - Le modèle H_2 a généré $g_2 \in H_2$.



Donc:

$E_{emp}(g_{10}) \leq E_{emp}(g_2)$ car H_{10} contient beaucoup de degrés de liberté que H_2.

$E_{gen}(g_{10}) > E_{gen}(g_2)$ car H_{10} sur-apprend les données.

Réf.

La mesure de l'Overfitting

$$Mesure\ d'Overfitting = E_{gen}(modèle\ complexe) - E_{gen}(modèle\ simple)$$

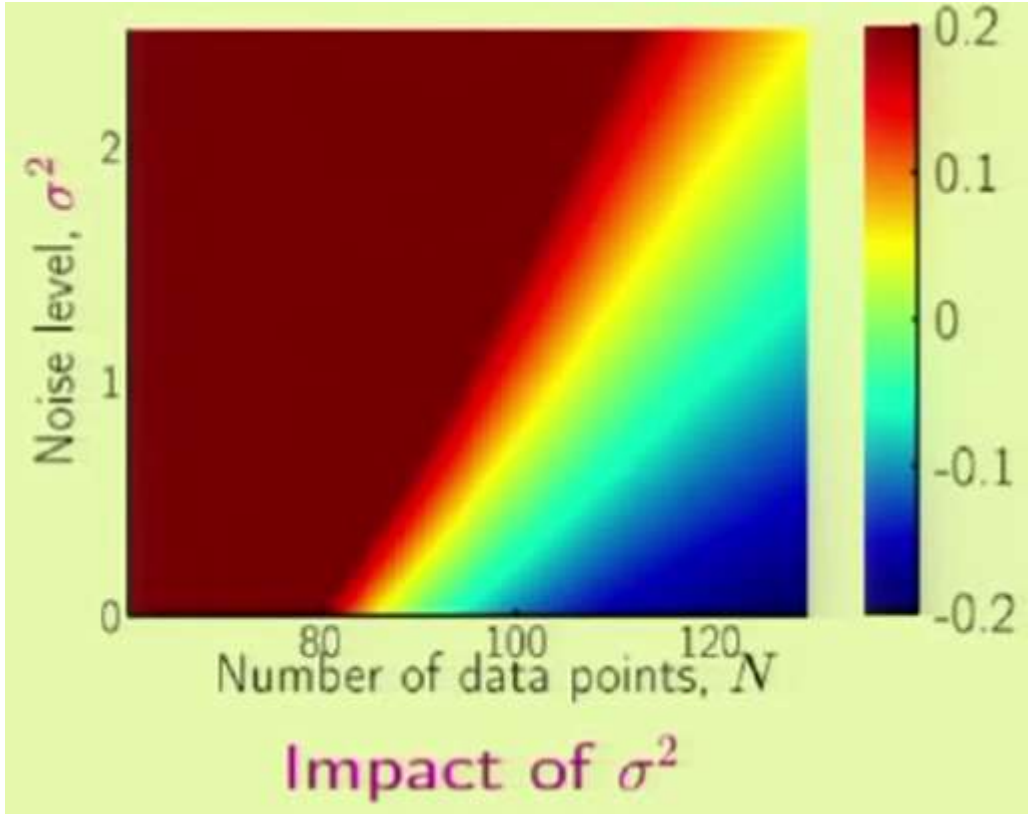
$$Mesure\ d'Overfitting = E_{gen}(g_{10}) - E_{gen}(g_2)$$

- Si $\boldsymbol{Mesure\ d'Overfitting > 0}$ → **Le modèle complexe est mauvais**
  → **Overfitting**.
- Si $\boldsymbol{Mesure\ d'Overfitting < 0}$ → **Le modèle complexe est bon**
  → **pas de Overfitting**.
- Si $\boldsymbol{Mesure\ d'Overfitting \approx 0}$ → **Les modèles sont pareils.**

Réf.

La mesure de l'Overfitting

Cette figure illustre l'impact de (σ^2, N) sur le Overfitting pour $Q_f = 20$.

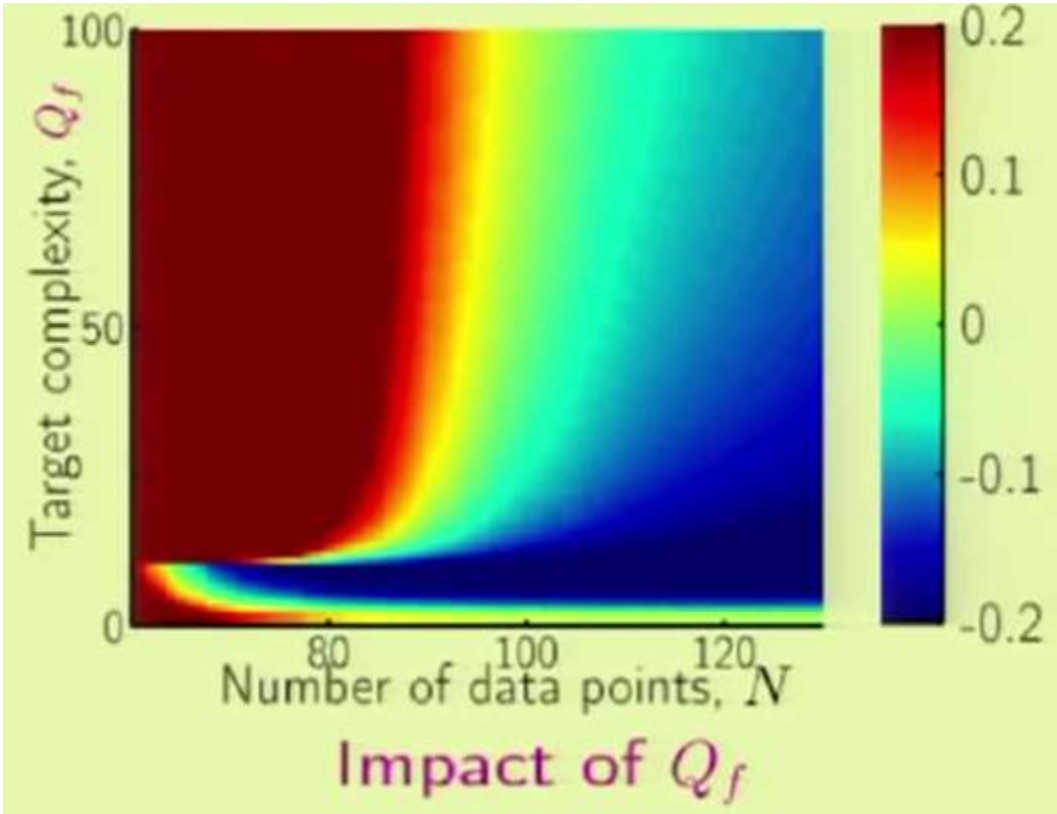


σ^2 *augmente* → *Bruit stochastique augmente* → *Overfitting augmente*

Réf.

La mesure de l'Overfitting

Cette figure illustre l'impact de (Q_f, N) sur le Overfitting pour $\sigma^2 = 0{,}1$.



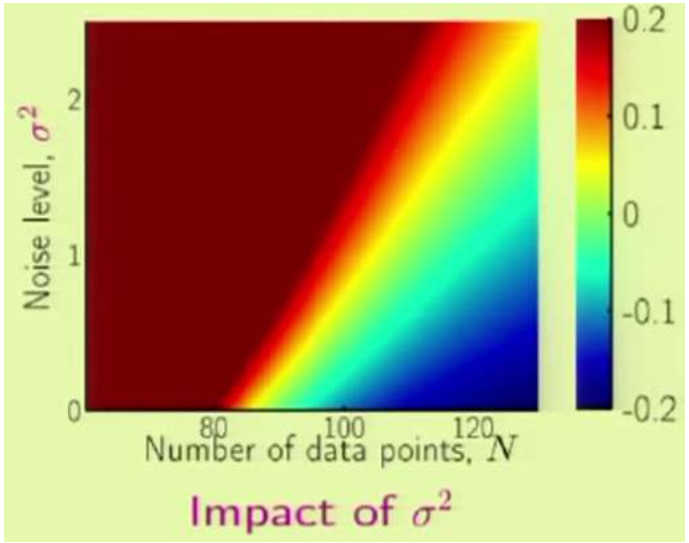
Q_f *augmente* → *bruit déterministe augmente* → *Overfitting augmente*

Réf.

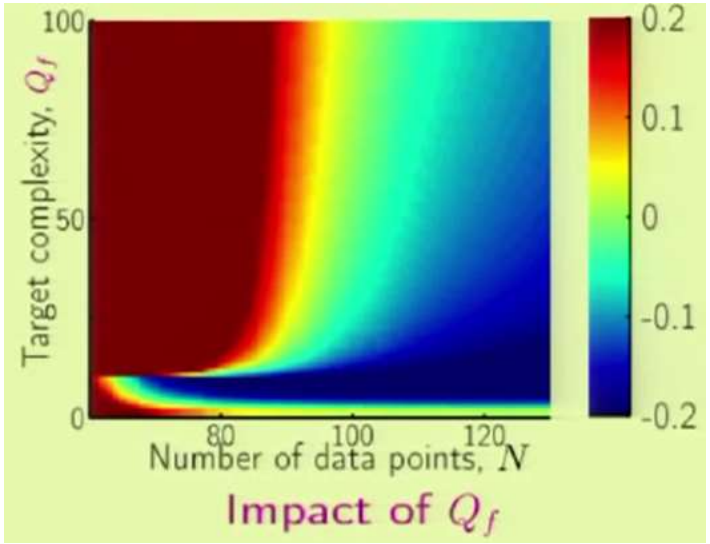
La mesure de l'Overfitting

Dans les deux cas : $N\ augmente \rightarrow Overfitting\ diminue$.

Bruit Stochastique



Bruit Déterministe



Le bruit stochastique (σ^2) impacte linéairement l'Overfitting.

Le bruit déterministe (Q_f) impacte non-linéairement l'Overfitting.

Réf.

Le bruit déterministe

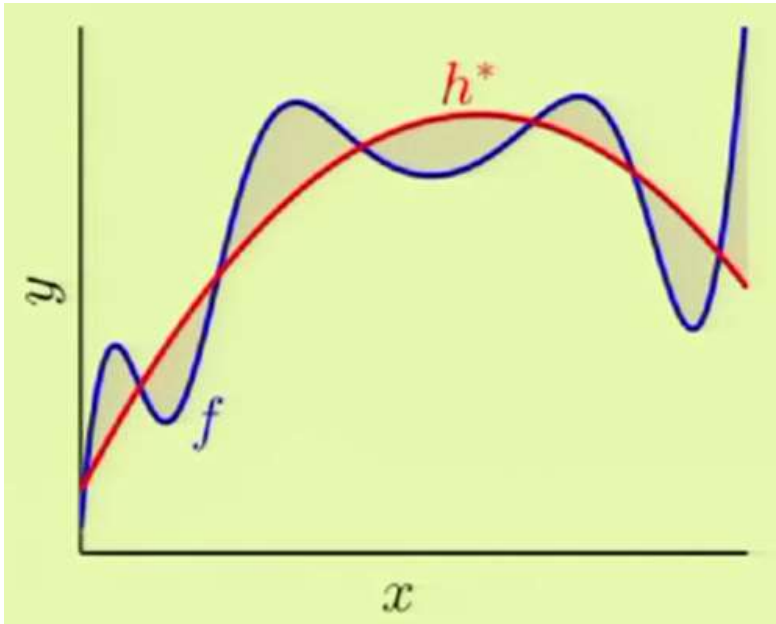
Pourquoi une cible à complexité supérieure impacte le Overfitting?

- Le bruit déterministe: C'est une partie de f que H ne peut pas capturer (modéliser) ou détecter.
- Car, on ne connait pas le signal de la fonction cible f.

$$\boldsymbol{Bruit\ déterministe = f(x) - h^*(x)}$$

- Pour un ensemble de points de données finis, l'algorithme d'apprentissage utilise ses degrés de liberté pour capturer le bruit.

➢ D'où le Overfitting.



Réf.

Bruit Stochastique Versus Bruit Déterministe

- Ces deux types de bruit ont le même impact sur le Overfitting.
- Mais ils diffèrent dans deux cas:

- **Le Bruit Stochastique dépend de la valeur mesurée y :**

$$y_n = f(x_n) + Bruit\ stochastique$$

- **Le Bruit Déterministe dépend de l'hypothèse h:**

$$y_n = f(x_n) = h(x_n) + Bruit\ déterministe$$

Réf.

Bruit Stochastique Versus Bruit Déterministe

- Si *la complexité de* H *est fixe et la compexité de* $f \uparrow$
- Si *la complexité de* f *est fixe et la complexité de* $H \downarrow$

Dans ces deux cas, que peut on dire sur le bruit déterministe et sur le overfitting?

Réf.

Bruit Stochastique Versus Bruit Déterministe

- Si *la complexité de* H *est fixe et la compexité de* $f \uparrow$

Donc :

$$Bruit\ déterministe \uparrow \rightarrow overfitting \uparrow$$

- Si *la complexité de* f *est fixe et la complexité de* $H \downarrow$

Donc :

$$Bruit\ déterministe \uparrow \rightarrow overfitting \downarrow$$

Réf.

Techniques de résolution du problème de l'Overfitting

Il existe deux outils qui permettent de remédier au problème de l'overfitting:

- **La régularisation :** freiner l'apprentissage.

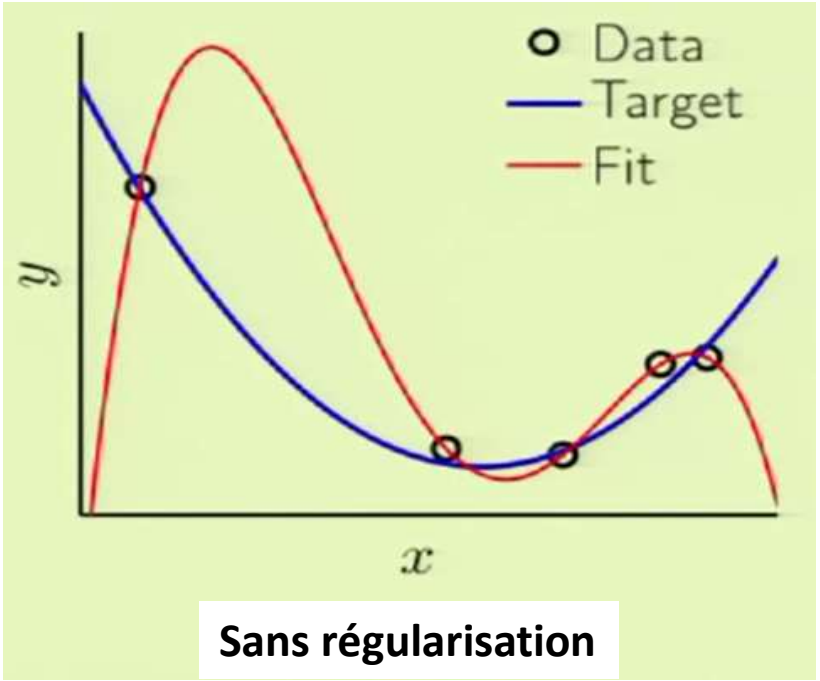
- **La validation :** Chercher le minimum de la courbe d'apprentissage.

Réf.

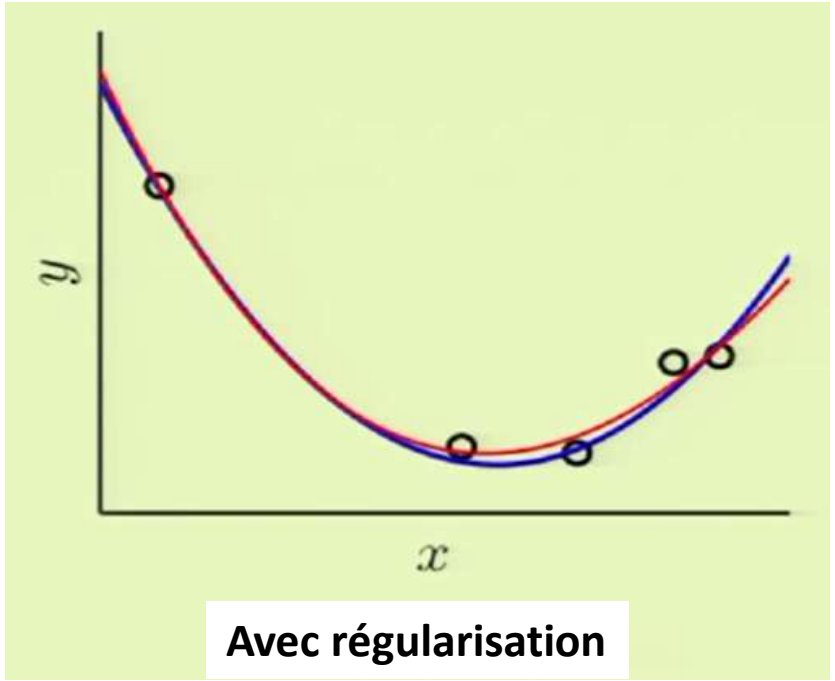
3. Régularisation

Définition de la régularisation

- La régularisation est l'un des moyens pour lutter contre le overfitting.
- Elle permet de forcer l'algorithme d'apprentissage à améliorer E_{gen} surtout quand le bruit existe.



Sans régularisation

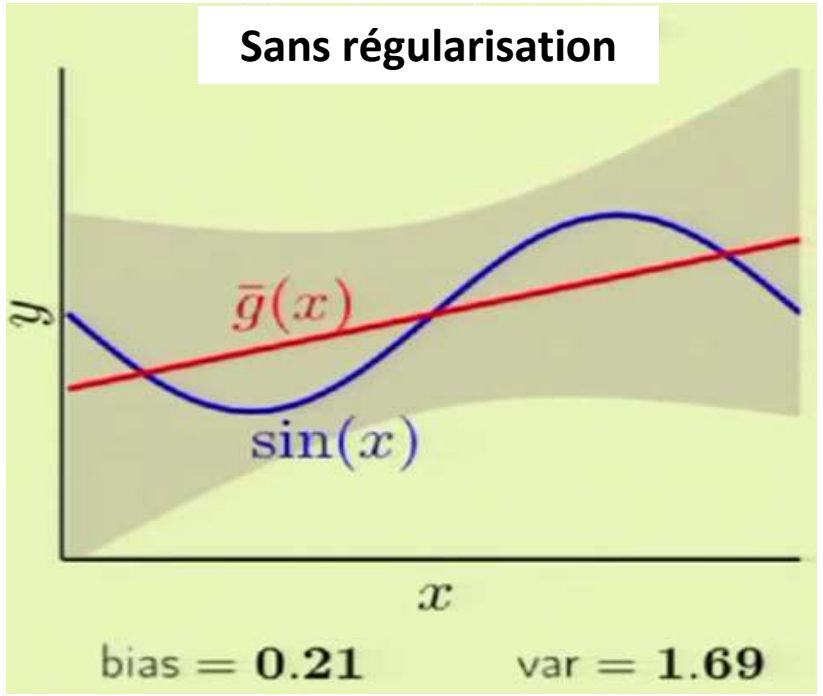


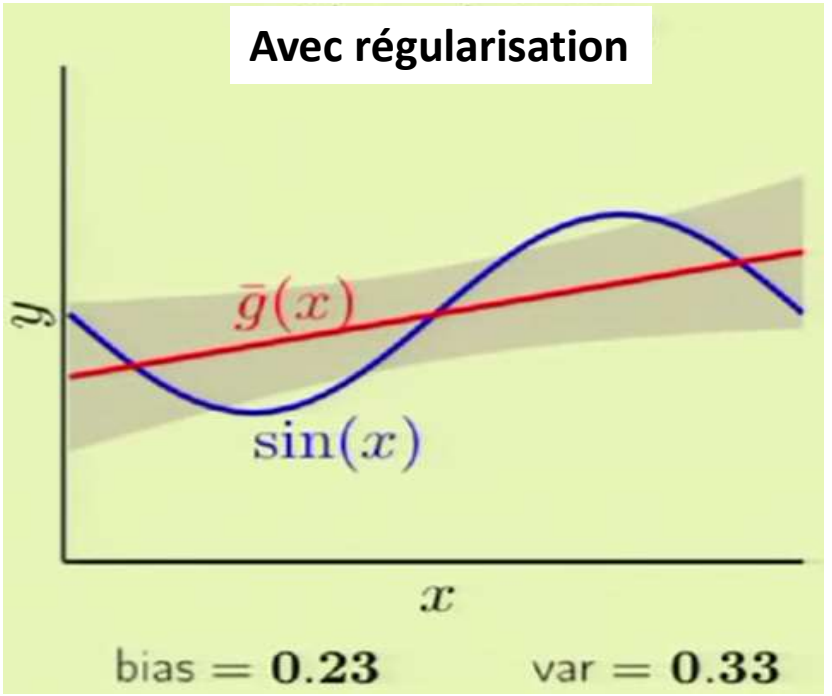
Avec régularisation

Réf.

Exemple

Existence de la régularisation → variance ↓ largement et bias ↑ faiblement





Le modèle régularisé est plus fort que le modèle non-régularisé.

Réf.

Solution sans contraintes

- Sans régularisation:

$$Min\ E_{emp}(w) = \left(\frac{1}{n}\sum_{i=1}^{n}(w^T X_i - y_i)^2\right)$$

$$= \frac{1}{n}(Xw - y)^T(Xw - y)$$

- Avec régularisation:

$$Min \quad E_{aug}(w) = E_{emp}(w) + \frac{\boldsymbol{\lambda}}{\boldsymbol{n}}\boldsymbol{w^T w}$$

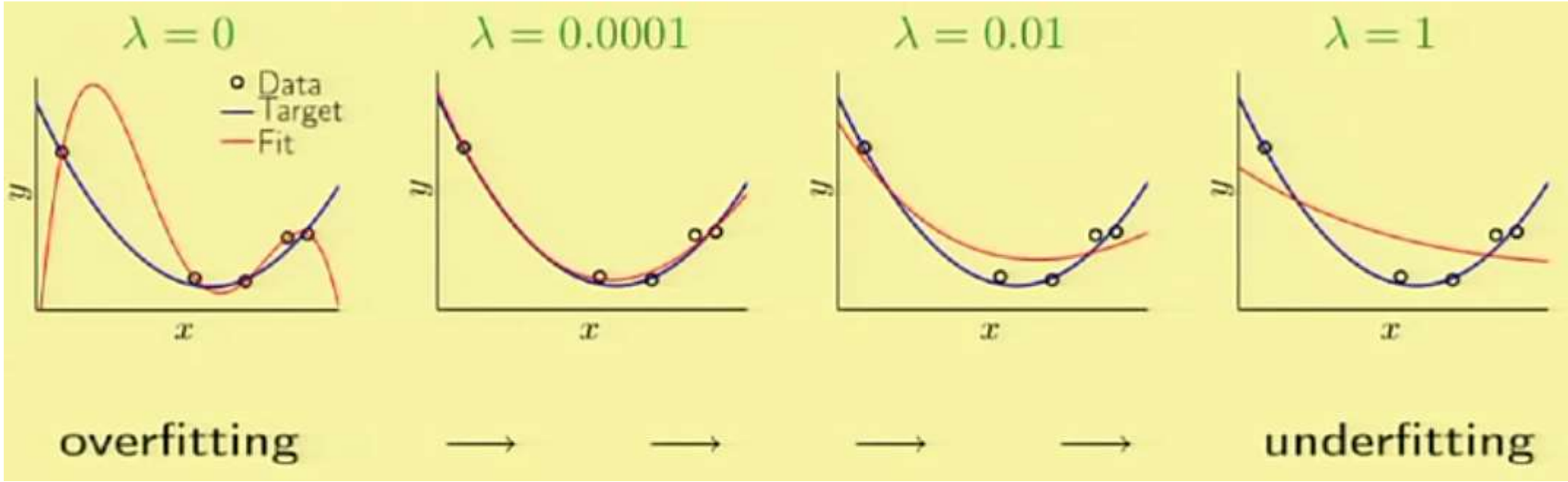
Un régulateur

$$= \frac{1}{n}(Xw - y)^T(Xw - y) + \frac{\boldsymbol{\lambda}}{\boldsymbol{n}}\boldsymbol{w^T w}$$

Réf.

La solution de l'optimisation de E_{aug}

- La valeur optimale de w dépend du choix de λ.
- Le choix de λ est très critique.
- Le choix du régulateur est largement heuristique.



Réf.

Le régulateur à dégradation de poids 'Weight Decay'

- Ce type de régulateur possède la forme suivante:

$$\boldsymbol{Régulateur = \frac{\lambda}{n} w^T w}$$

C'est l'un des régulateurs les plus célèbre du machine learning.

Il permet de minimiser la valeur des poids.

Réf.

Autres variances des régulateurs

- Au lieu de minimiser toutes les valeurs de poids, il existe des poids qui sont plus importants que d'autres.

$$\boldsymbol{Régulateur} = \sum_{q=0}^{Q} \gamma_q w_q^2$$

γ_q est une constante qui spécifie le type du régulateur qu'on travaille avec.

- Régulateur à ordre d'adaptation inférieur 'low order fit' $\Rightarrow \gamma_q = 2^q$
- Régulateur à ordre d'adaptation supérieur 'high order fit' $\Rightarrow \gamma_q = 2^{-q}$

- La forme générale de ces types de régulateur est:

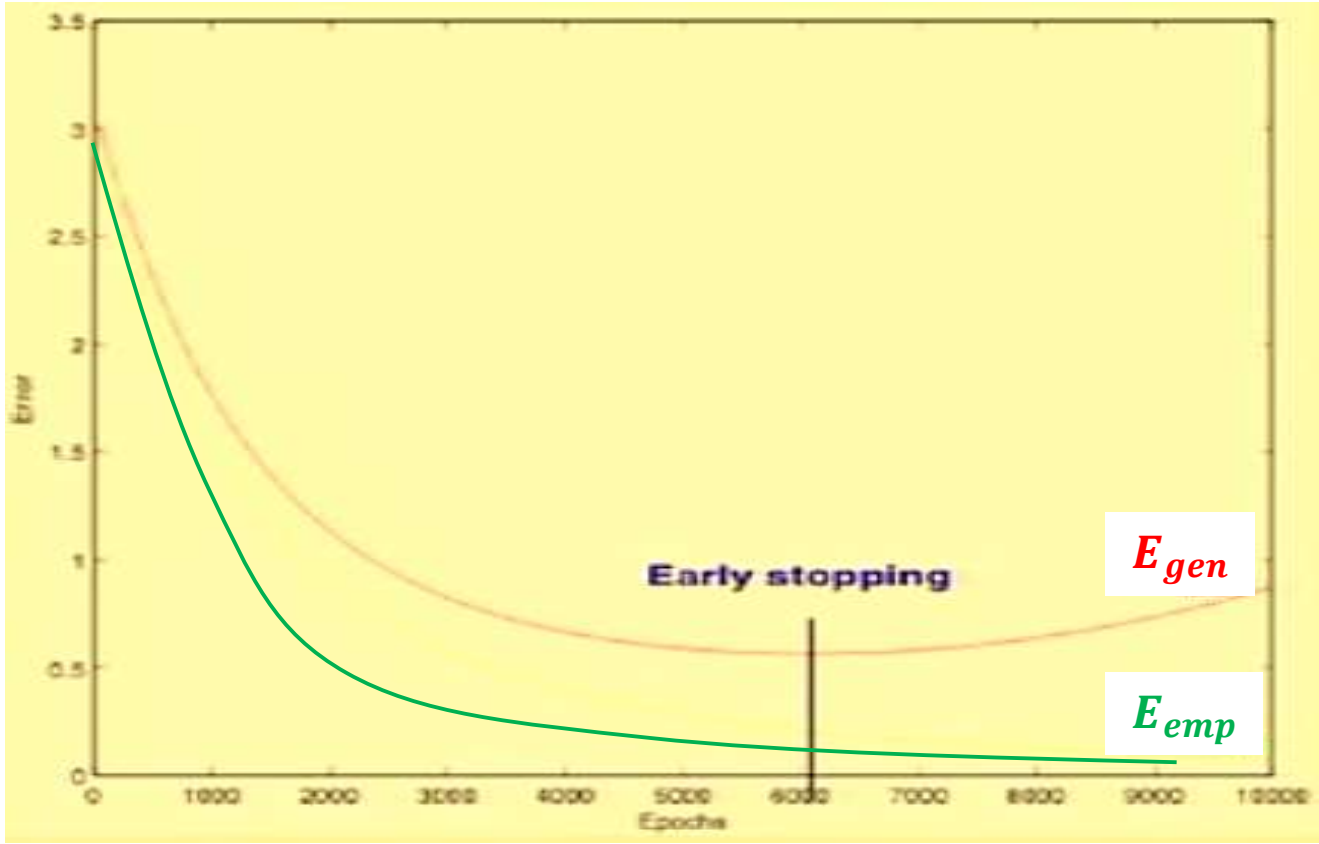
Le régulateur de Tikhonov :

$$\boldsymbol{w^T \Gamma^T \Gamma w}$$

Réf.

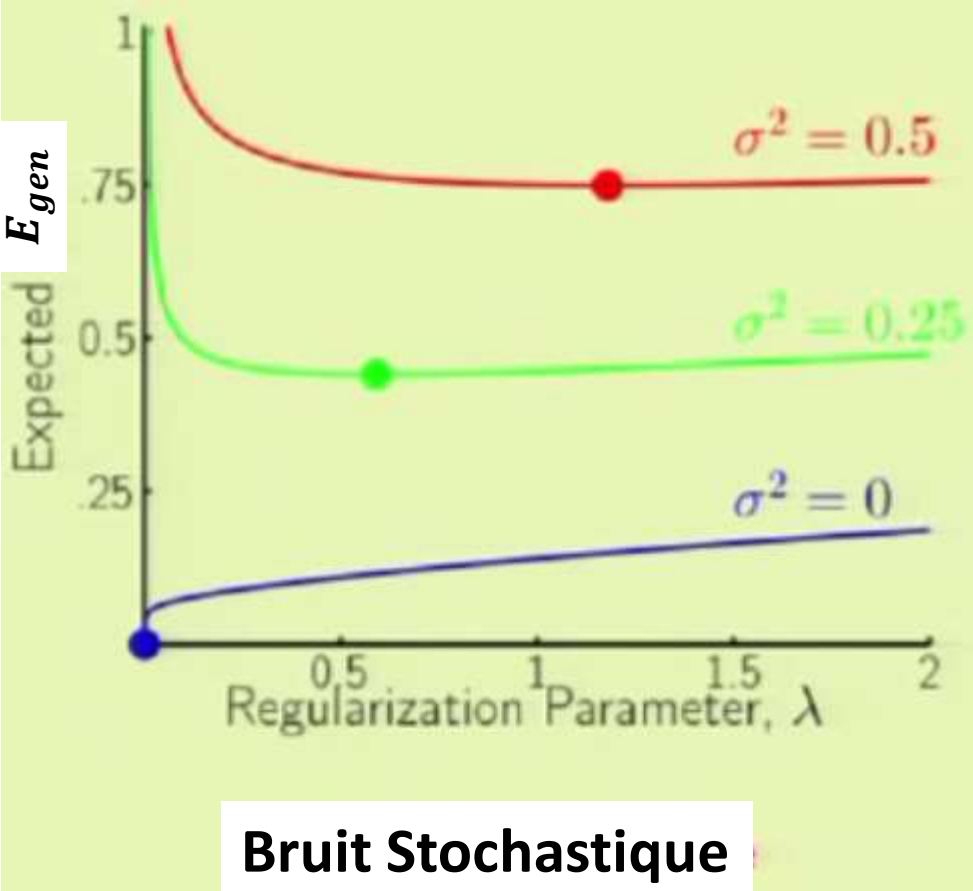
Régulateur à arrêt précoce

- Dans ce cas, on ne modifie pas la fonction objectif.
- Mais, on impose à l'algorithme un point d'arrêt qui sera fixer par la Validation.

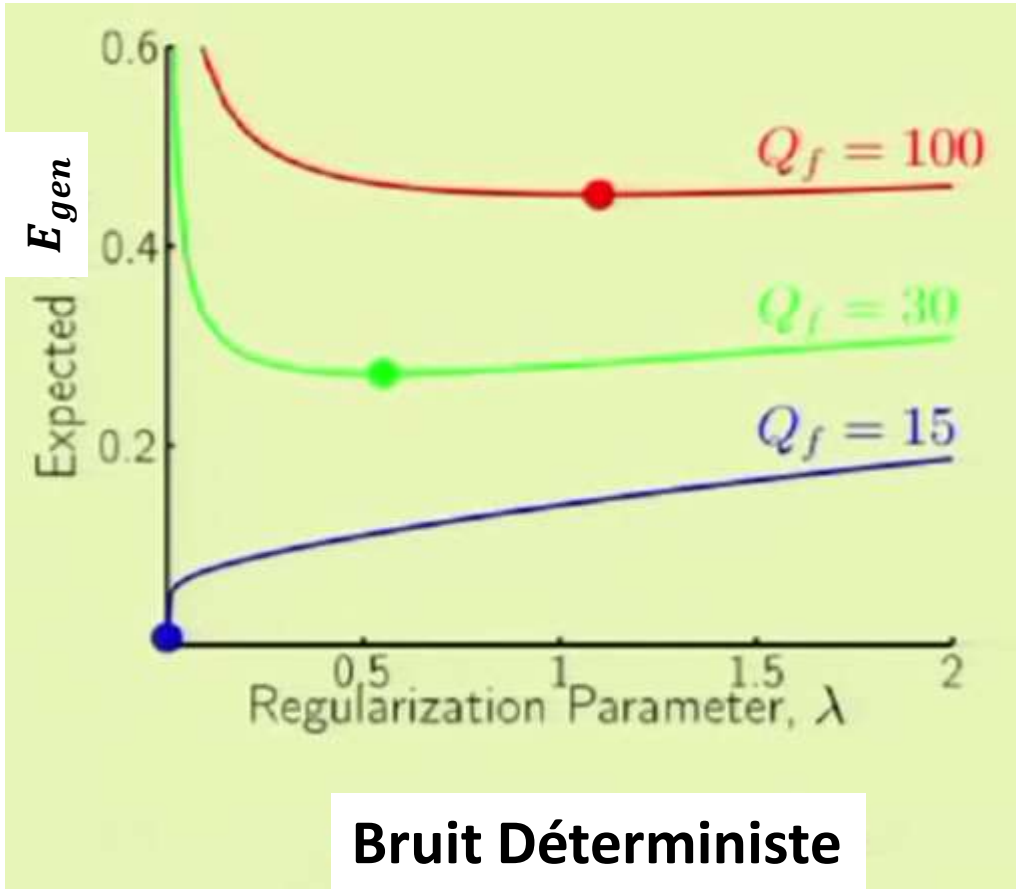


Réf.

Les valeurs optimales de λ



Bruit Stochastique



Bruit Déterministe

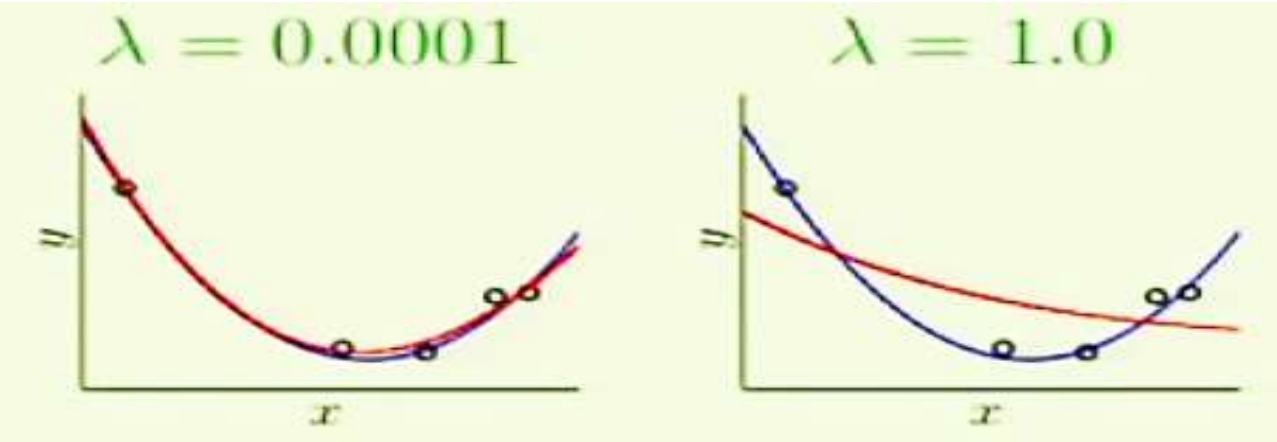
Si **Le bruit ↑** → **La valeur du paramètre de régularisation augmente ↑**

Réf.

4.Validation

Motivation

- Si la valeur de λ est correcte → la courbe adaptative est très proche de la cible.
- Si on utilise beaucoup de régularisation → la courbe adaptative n'est plus proche de la cible.



- Comment déterminer la valeur de λ?

La validation

Réf.

Utilisation de D_{val} plus qu'une fois

- M c'est le nombre des modèles à partir desquels on va choisir.
- Par exemple:
  - $Modèles = \{régression\ linéaire,\ ANNs, SVM\}$
  - $Modèles = \{2ème\ ordre, 5ème\ ordre, 10ème\ ordre\}\ d'un\ polynôme$
  - $Modèles = \{\lambda = 0.001, \lambda = 0.01, \lambda = 1\}\ d'un\ polynôme\ d'odre\ 5$
- Pour choisir entre ces modèles, il faut utiliser l'ensemble de validation.

Réf.

Utilisation de D_{val} plus qu'une fois

- Soit M modèles $H_1, \dots, H_M$

L'Algorithme de la sélection du modèle :

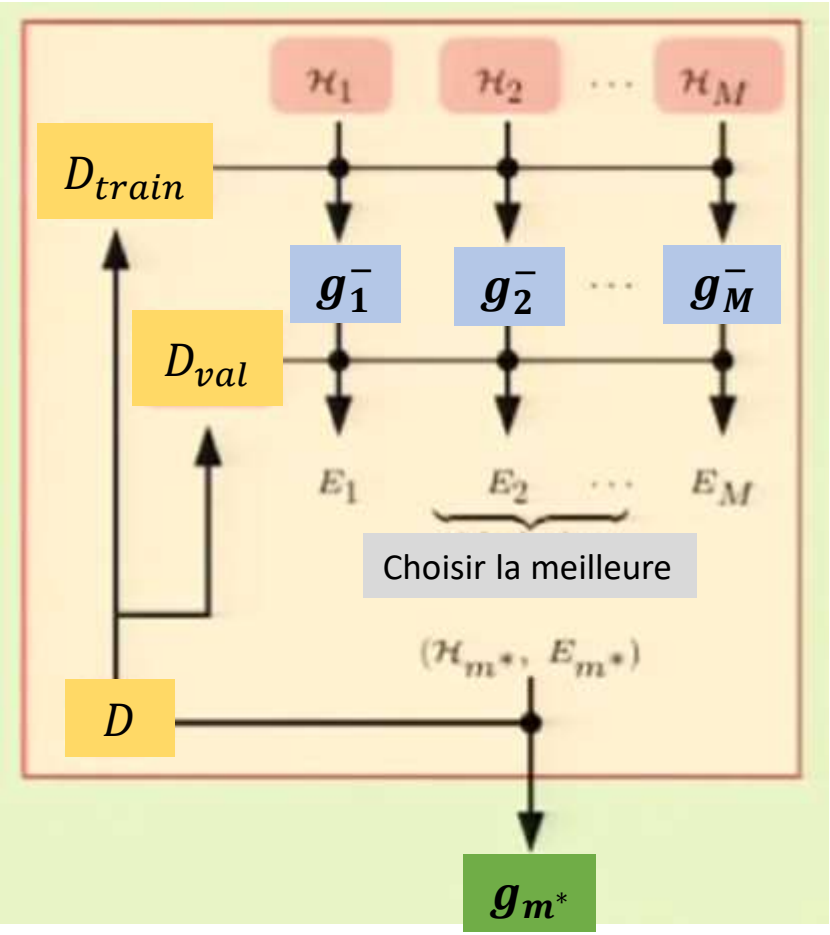
1: Utiliser D_{train} pour apprendre g_m^- pour chaque modèle.

2: Evaluer g_m^- en utilisant D_{val}:

$$E_m = E_{val}(g_m^-) \qquad m \in \{1, M\}$$

3: Choisir le modèle $m = m^*$ avec la plus petite E_m.

4: Retourner g_{m^*}.



Réf.

Contamination des données

- Estimations d'erreurs: $E_{train}, E_{test}, E_{val}$
- Contamination: un bias optimiste pour l'estimation de E_{gen}.

- **Ensemble d'entrainement** → Totalement contaminée
  → E_{emp} n'est pas une estimation de E_{gen}
- **Ensemble de test** → Non contaminée
  → E_{test}est une estimation non biaisée de E_{gen}
- **Ensemble de validation** → Peu contaminée
  → E_{val} est une estimation biaisée de E_{gen}

Réf.

Contamination des données

- Pour remédier au problème de la contamination de l'ensemble de validation, il faut utiliser un nombre d'ensembles de validation, de telle sorte si l'un est contaminé, on peut utiliser un autre.
- Donc l'estimation sera fiable.
- Cette technique est appelée :

Validation croisée

Réf.

5. La validation croisée

Leave-One-Out (LOO)

Il y a n façons pour partitionner les données en un ensemble d'entrainement de taille $n-1$ et un ensemble de validation de taille 1:

$$D_i = (x_1, y_1), \ldots, (x_{i-1}, y_{i-1}), \cancel{(x_i, y_i)}, (x_{i+1}, y_{i+1}), \ldots, (x_n, y_n)$$

- Soit $\boldsymbol{D_i}$ un ensemble de donnée après avoir retiré (x_i, y_i).

- Soit $\boldsymbol{g_i^-}$ l'hypothèse finale générée par D_i.

- Soit $\boldsymbol{e_i}$ l'erreur de validation générée par g_i^- en un point de donnée $\{(x_i, y_i)\}$:

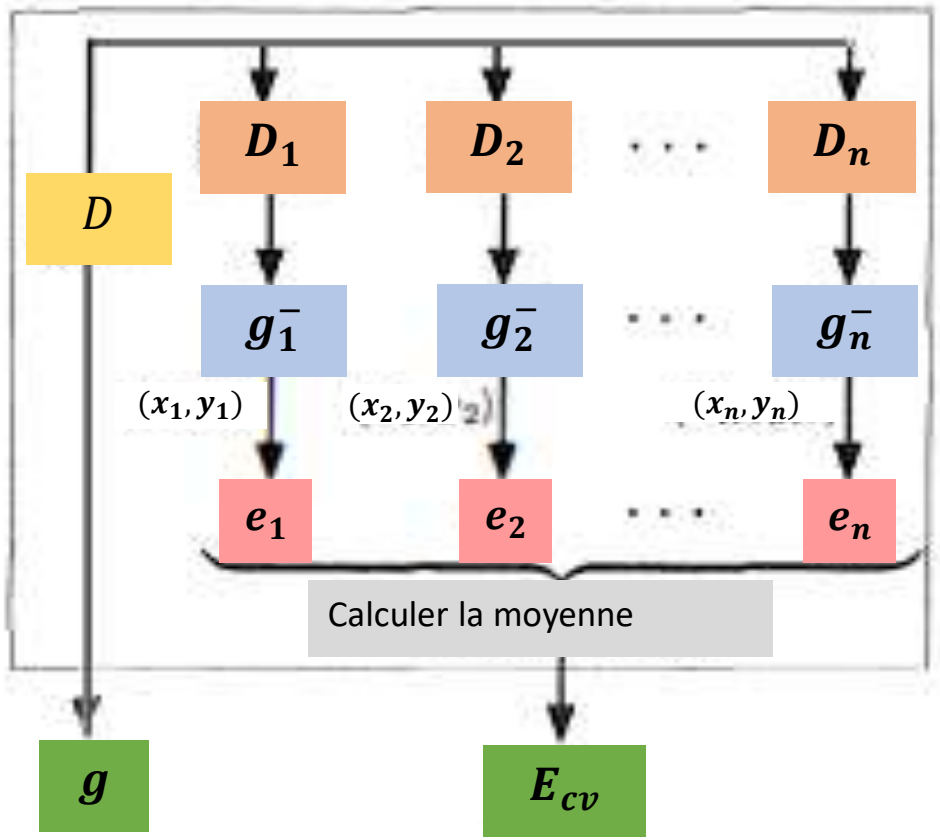
$$e_i = E_{val}(g_i^-) = e(g_i^-(x_i), y_i)$$

Réf.

Leave-One-Out (LOO)

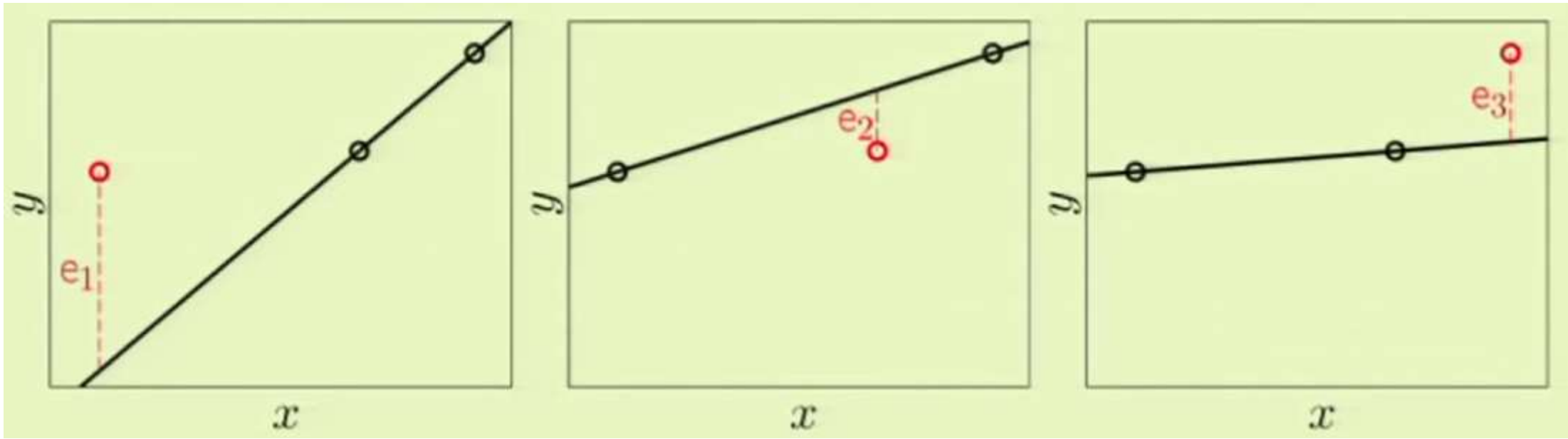
L'estimation de la validation croisée c'est la moyenne des valeurs de e_i :

$$E_{cv} = \frac{1}{n}\sum_{i=1}^{n} e_i$$



Réf.

Illustration de la validation croisée



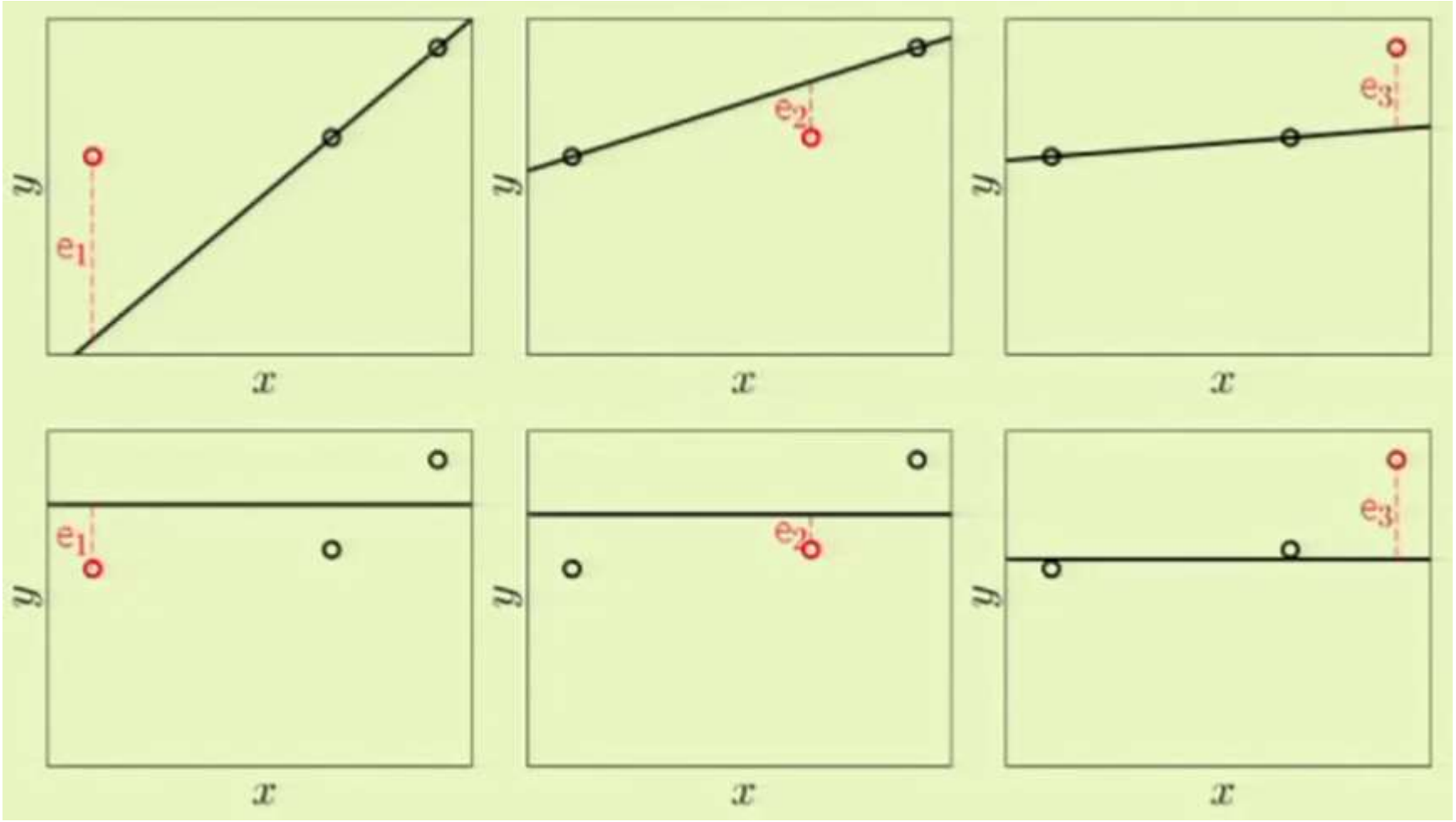
$$E_{cv} = \frac{1}{3}(e_1 + e_2 + e_3)$$

Réf.

Sélection du modèle à l'aide de la validation croisée

Modèle Linéaire

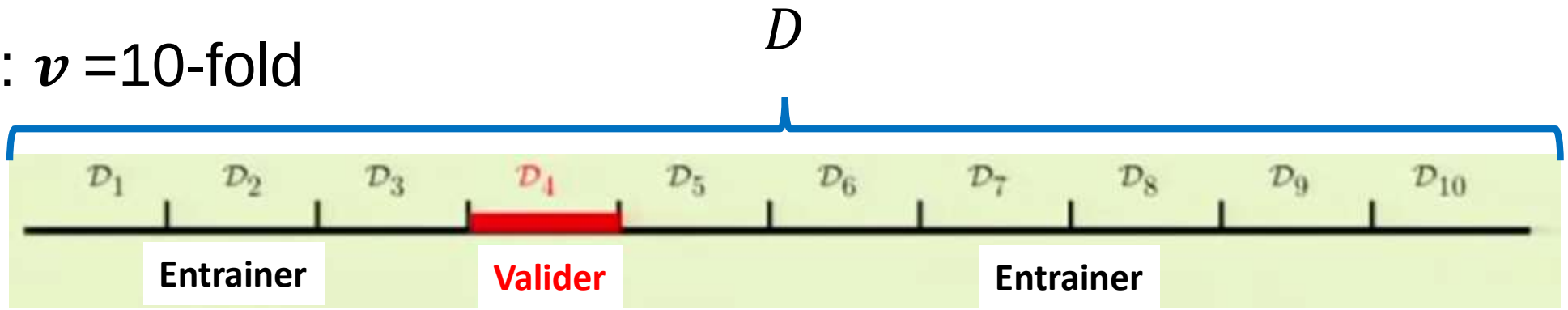
Modèle Constant



Réf.

La validation croisée "$v - fold$"

- LOO est utilisée pour les donnés de petite taille.
- LOO $\rightarrow n$ sessions d'entrainement avec $n - 1$ points.
- v = fold = sous-ensemble.
- Nombre des folds : $v = \frac{n}{k}$
- Nombre des données d'entrainement: $n - k$
- k c'est la taille d'un fold.
- Exemple: v =10-fold



Réf.

FIN