Apprentissage Automatique

Rym Guibadj

LISIC, ULCO, EILCO

Apprentissage automatique

Quelques domaines de l'IA

- Représentation des connaissances
- Résolution de problèmes
- Reconnaissance de la parole
- Reconnaissance de l'écriture
- Reconnaissance des visages
- Robotique
- **Apprentissage automatique** (artificiel)

Apprentissage automatique

Définition

L'apprentissage automatique est un **sous-domaine de l'intelligence artificielle** qui se concentre sur le développement de techniques permettant d'**apprendre à partir de données**. L'objectif principal de l'apprentissage automatique est de permettre aux machines d'**acquérir des connaissances ou des compétences sans être explicitement programmées**.

Apprentissage automatique

- On dira qu'une machine **apprend** dès lors qu'elle change en fonction de données en entrée ou de réponses à son environnement de sorte à ce que **ses performance futures deviennent meilleures**

- L'objectif de l'apprentissage automatique est de **concevoir des programmes pouvant s'améliorer automatiquement avec l'expérience**

Pourquoi l'apprentissage automatique ?

- Certaines tâches ne sont bien définies que via un ensemble d'exemples
- Pour découvrir des relations importantes dans des données (fouille de données)
- Les machines peuvent ne pas fonctionner sur tous les environnements
- La quantité de connaissances disponibles à propos de certaines situations sont telles que le cerveau humain ne puisse les expliciter
- L'environnement change constamment

Des applications diverses :

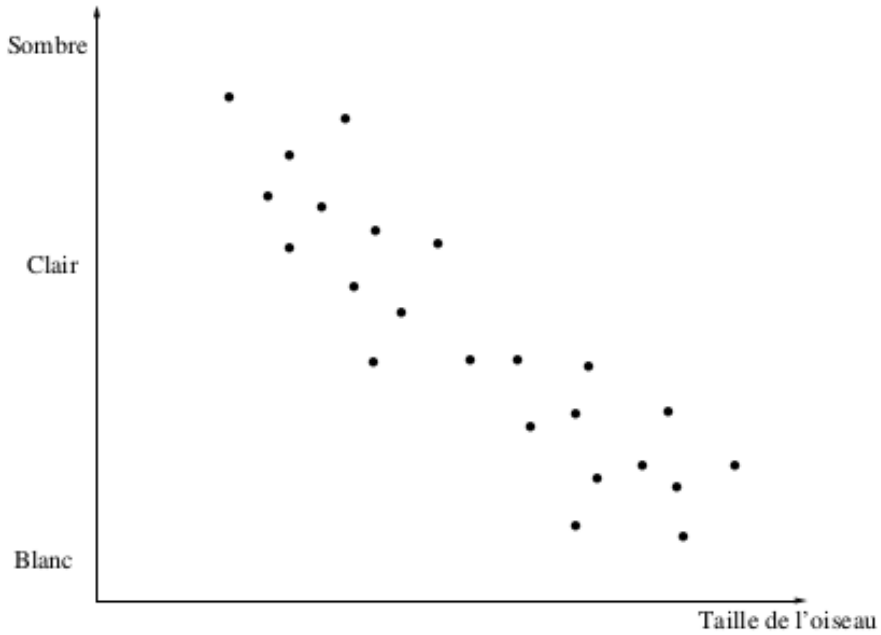
- Reconnaissance de la parole.
- Diagnostique médical.
- Moteurs de recherche.
- Les jeux.
- Conduite autonome.
- ...

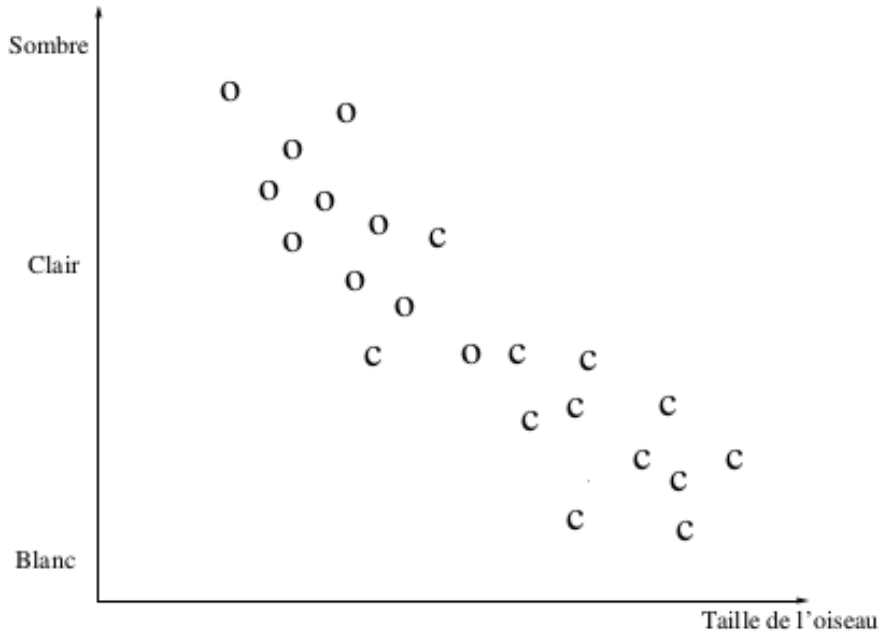
Exemple d'apprentissage :

- Imaginons un étang sur lequel nagent des oies et des cygnes (nous admettons qu'il n'y a pas d'autres oiseaux dans cette région).
- Deux personnes arrivent dont l'un est expert et l'autre débutant.
- Le débutant souhaite apprendre à distinguer une oie d'un cygne. Il doit se contenter de mesurer ce qui lui paraît caractéristique : le niveau de gris du plumage et la taille de la bête.

Exemple d'apprentissage :

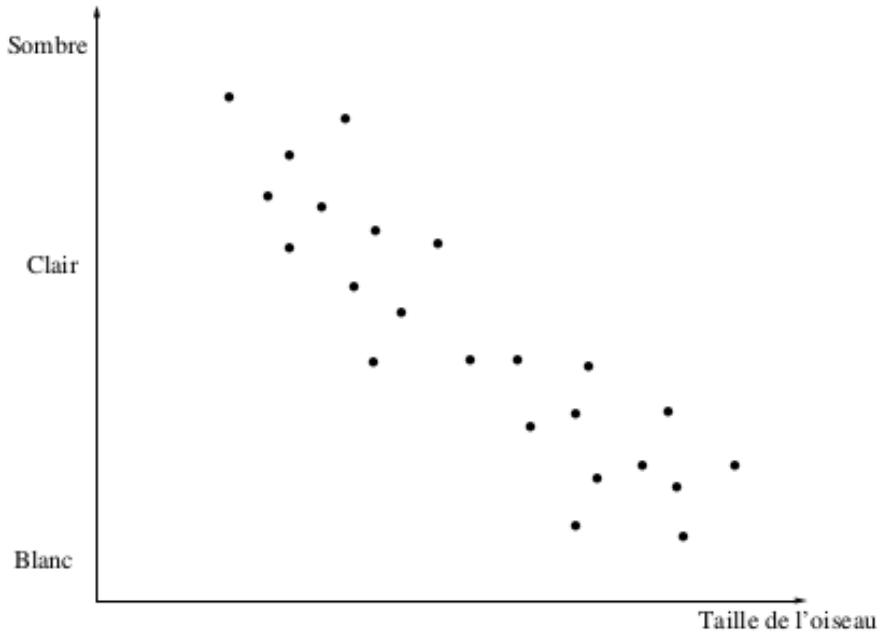
- Le premier graphique représente les mesures de chaque oiseau prise par l'amateur
- Le second graphique représente les mêmes oiseaux étiquetés par l'expert

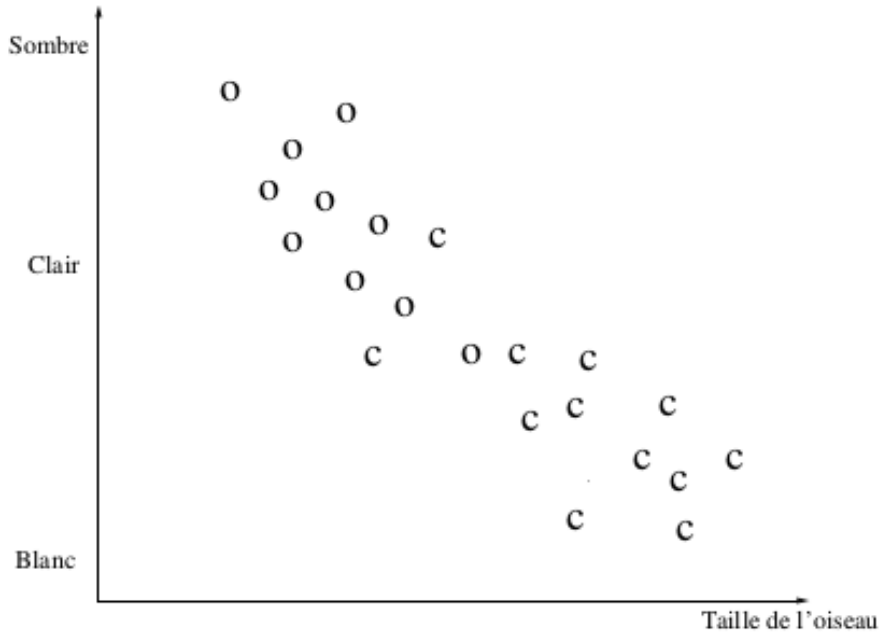




Exemple d'apprentissage :

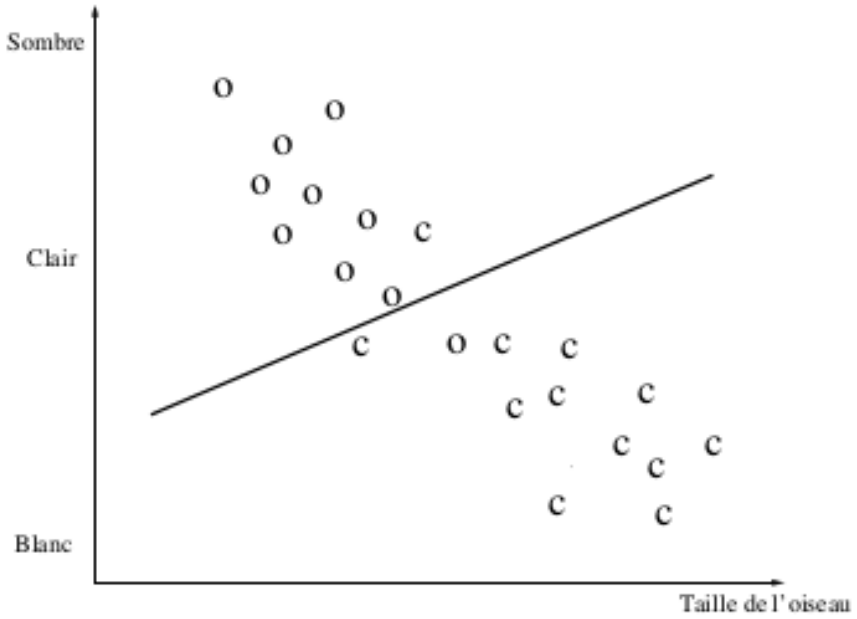
- Il faut maintenant que l'amateur trouve une règle permettant de séparer les exemples en minimisant l'erreur.

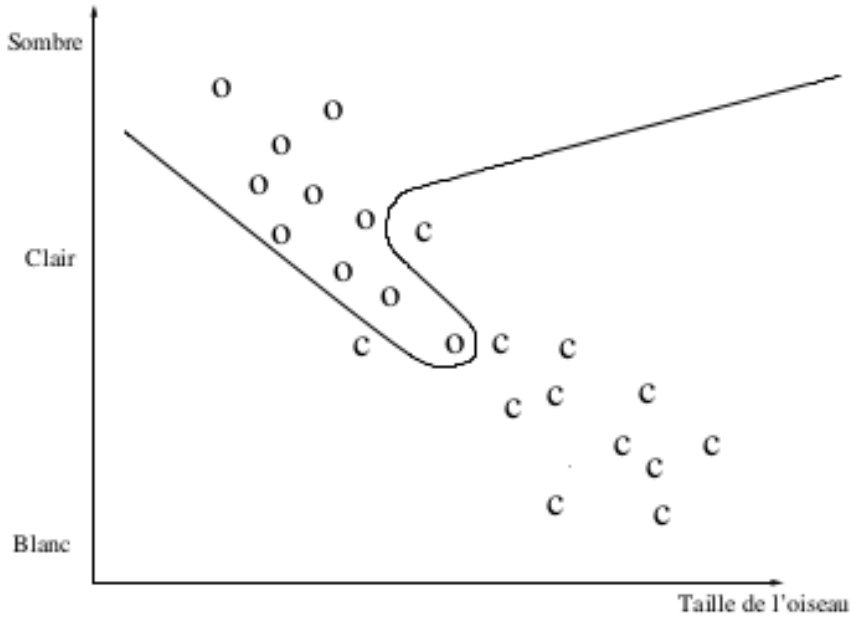




Exemple d'apprentissage :

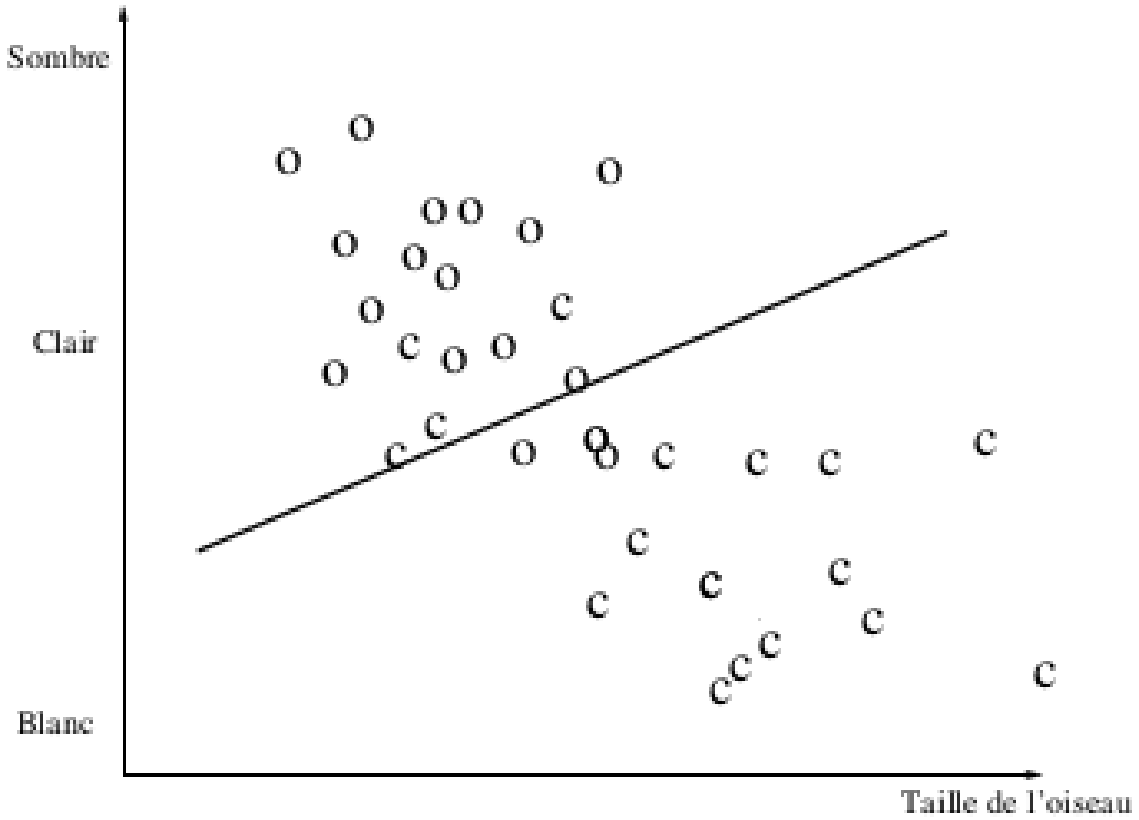
- Une règle de décision simple et une règle de décision complexe pour séparer les oies des cygnes.





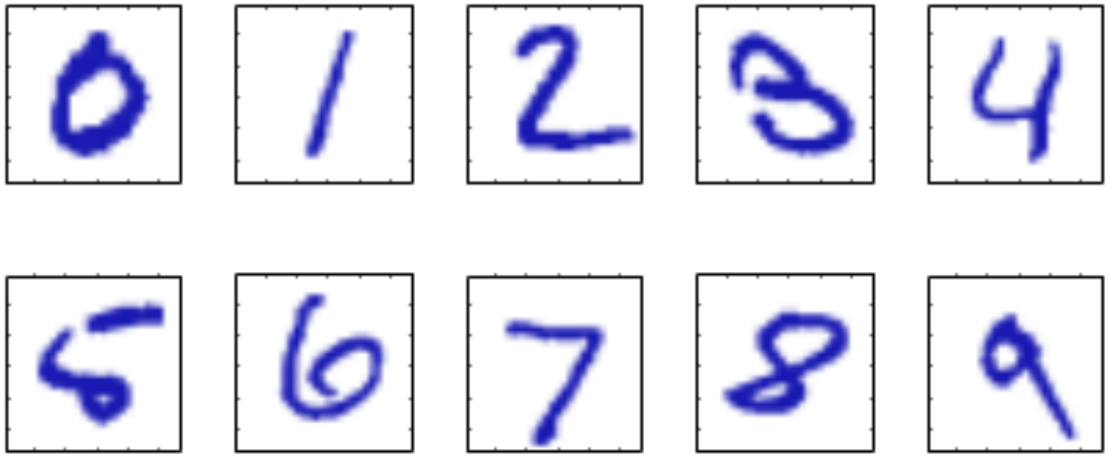
Exemple d'apprentissage :

- Le test de la règle simple sur d'autres oiseaux.



Exemple d'apprentissage :

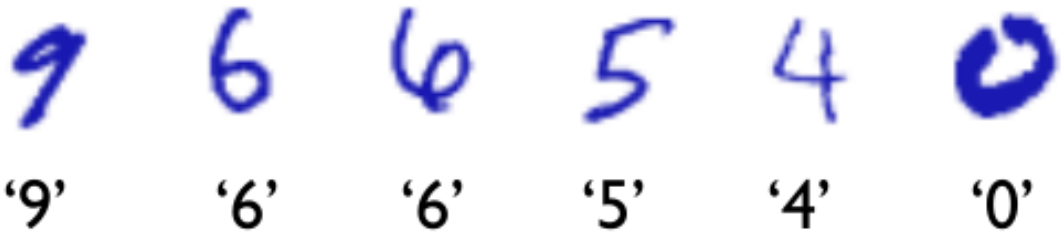
Comment reconnaître des caractères manuscrits



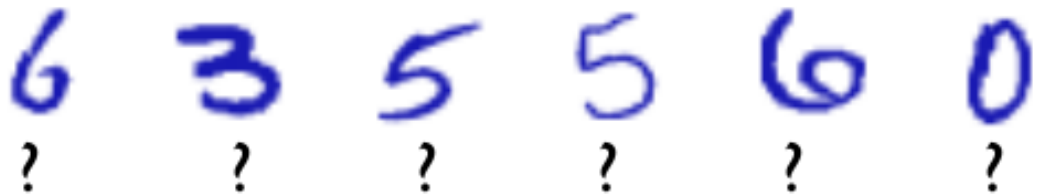
- **Par énumération de règles :** trop fastidieux, difficile de couvrir tous les cas.
- **En donnant à l'ordinateur la capacité d'apprendre :** laisser l'ordinateur faire des essais et apprendre de ces erreurs.

Données d'entraînement et généralisation

On fournit à l'algorithme d'apprentissage des **données d'entraînements**

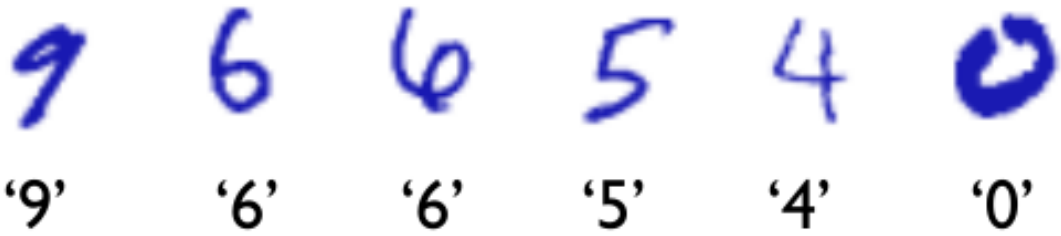


L'algorithme retourne un programme capable de **généraliser** à de nouvelles données

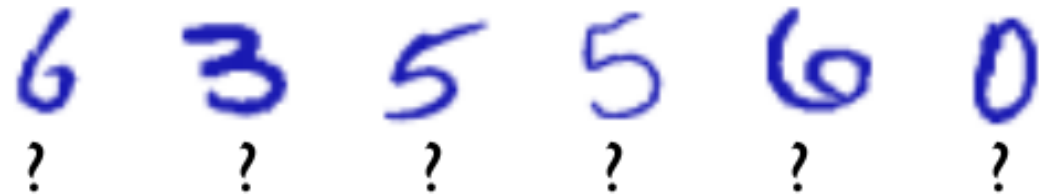


Données d'entraînement, généralisation et modèle

On note l'**ensemble d'entraînement :** $\mathcal{D} = \{(x_1, y_1), ..., (x_N, y_N)\}$. On appelle x_i une **entrée** et y_i la **cible**.

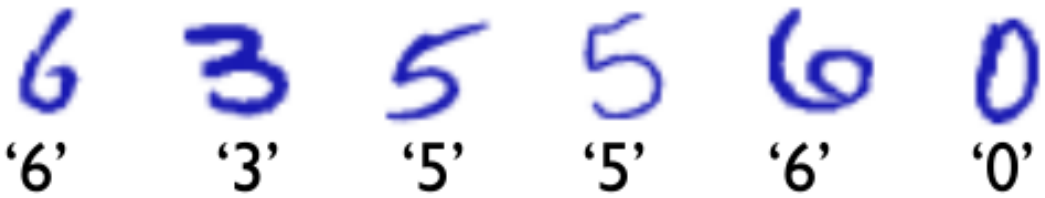


Le ≪programme≫ généré par l'algorithme d'apprentissage $f(x)$ est appelé **un modèle** capable de généraliser à de nouvelles données.



Ensemble de tests

On utilise un ensemble de test D_{test} pour mesurer la performance de généralisation de notre modèle $f(x)$



Types d'apprentissage :

- **Apprentissage supervisé :** il ya une cible à prédire

$$\mathcal{D} = \{(x_1, y_1), ..., (x_N, y_N)\}$$

  - Chaque exemple est associé à une étiquette
  - Objectif : prédire l'étiquette de chaque donnée
  - Le système apprend à classer les données

- **Apprentissage non supervisé :** cible n'est pas fournie

$$\mathcal{D} = \{x_1, ..., x_N\}$$

  - Les exemples ne sont pas étiquetés
  - Objectif : trouver une structure aux données
  - Le système apprend une classification des données

Types d'apprentissage :

- **Apprentissage par renforcement**
  - Les exemples sont (parfois) associés à une récompense ou une punition
  - Objectif : trouver les actions qui maximisent les récompenses
  - Le système apprend une politique de décision

Types d'apprentissage :

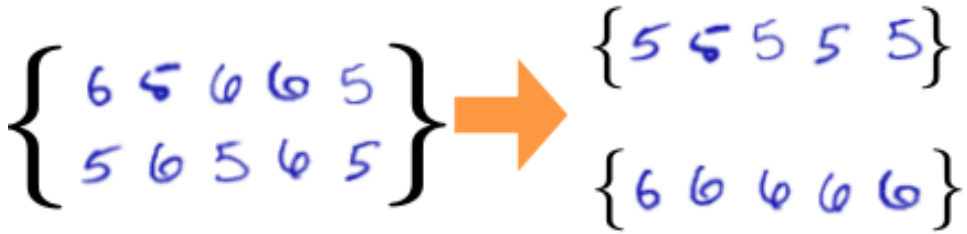
Apprentissage supervisé : il ya une cible à prédire

- **classification** : la cible est une classe $y \in \{1, ..., K\}$. Exemple : reconnaissance de caractères
  - x : vecteur des intensités de tous les pixels de l'image
  - y : identité du caractère
- **régression** : la cible est un nombre réel $y \in R$. Exemple : prédiction de la valeur d'une action à la bourse
  - x : vecteur contenant l'information sur l'activité économique de la journée
  - y : valeur d'une action à la bourse le lendemain

Types d'apprentissage :

Apprentissage non supervisé : cible n'est pas fournie.

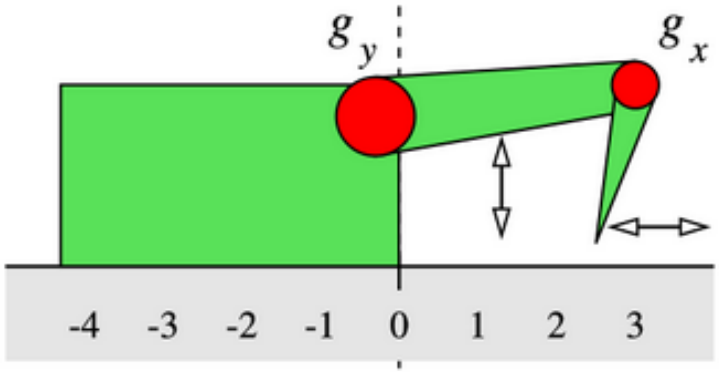
- Partitionnement / *clustering*
- Exemple : partitionnement des caractères manuscrits



Types d'apprentissage :

Apprentissage par renforcement

- Exemple : robot motorisé qui avancent sur différents types de sols
  - Observe les effets de ses actions
  - Déduit de ses observations la qualité de ses actions
  - Améliore ses actions futures



	1	2	3	4	5
1	0 0 0 0	0 0 0 0	0 0 0 -1	0 0 0 0	0 0 0 0
2	1 0 0 0	0 0 0 11	0 0 0 4	0 0 0 0	0 0 0 0
3	-31 0 -39 0	-29 32 -65 27	0 43 0 0	0 16 0 0	0 0 0 0
4	-50 0 -66 0	-47 -5 -52 54	-47 0 -43 0	-26 0 -26 0	-4 0 0 0
5	-74 0 -47 0	-79 -16 -71 0	-100 0 -62 0	-37 0 0 0	0 0 0 0

ailab.hs-weingarten.de

Types d'apprentissage :

Apprentissage par renforcement

- Exemple : configuration automatique de machines virtuelles d'un cluster de serveur linux dans un contexte de charge dynamique
  - 3 paramètres pertinents ont été choisis pour chacune des 3 VMs utilisées.
  - 3 actions sont possibles sur chaque paramètre : incrémentation, décrémentation ou invariance de celui-ci.
  - Les récompenses sont définies par le SLO (service level objectives) auquel on soustrait le temps de réponse.

Apprentissage supervisé

- Aussi appelé **analyse discriminante**
- Les données d'apprentissage sont etiquetées
  - Un **expert** ou **oracle** doit préalablement étiqueter des exemples.
- Le processus se passe en deux phases :
  - La **phase d'apprentissage** (*hors ligne*) : déterminer un modèle de données étiquetées
  - La **phase de test** (*en ligne*) : prédire l'étiquette d'une nouvelle donnée, connaissant le modèle préalablement appris.

Définition formelle

- Données d'apprentissage
  - N couples entrée sortie $(x_n, y_n)_{1<=n<=N}$ avec $x_n \in X$ et $y_n \in Y$
  - On suppose que ces données sont tirées selon une loi (de probablilité) inconnue
- Objectif de l'apprentissage
  - déterminer une fonction de prédiction $f : X \rightarrow Y$ qui soit en accord avec le données d'apprentissage

Les différentes représentations

- Si f est une fonction continue on parle alors de **régression**.
- Si f est une fonction discrète on parle alors de **classification**.
- Si f est une fonction binaire on parle alors d'**apprentissage de concept**.

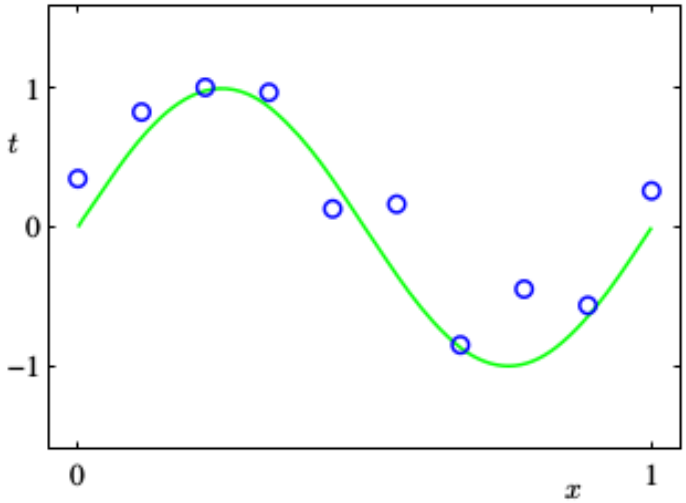
Méthodes d'apprentissage supervisé

Liste non exhaustive

- Régression polynomiale
- Les k plus proches voisins
- Arbre de décision
- Les réseaux de neurones
- Les machines à vecteur support
- etc.

Régression

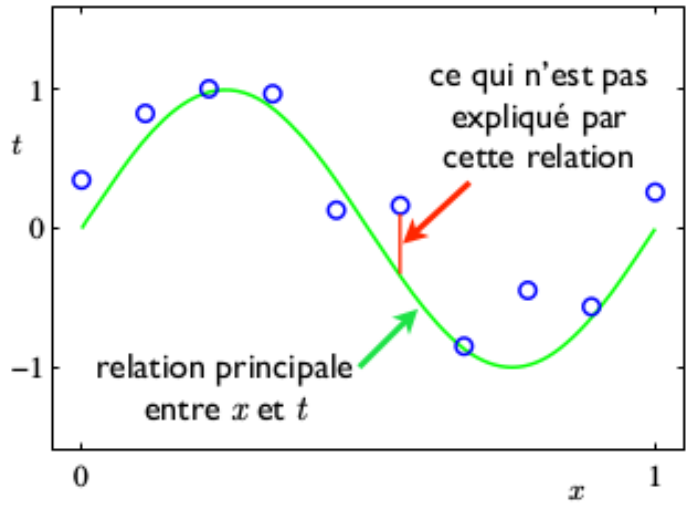
- Exemple simple : régression en une dimension ($1D$)
  - entrée : scalaire x
  - cible : scalaire y
- Données d'entraînements $\mathcal{D}$ contiennent :
  - $(x_1, ..., x_N)^T$
  - $(y_1, ..., y_N)^T$



- **Objectif** : faire une prédiction $\hat{y}$ pour une nouvelle entrée $\hat{x}$

Régression

- Exemple simple : régression en une dimension ($1D$)
  - entrée : scalaire x
  - cible : scalaire y
- Données d'entraînements $\mathcal{D}$ contiennent :
  - $(x_1, ..., x_N)^T$
  - $(y_1, ..., y_N)^T$



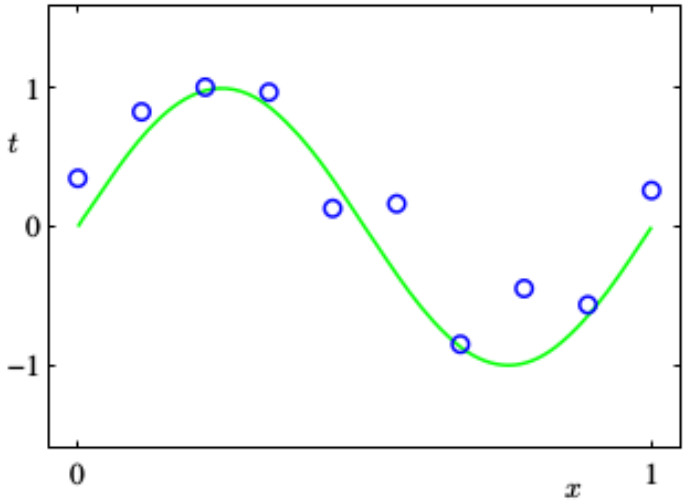
- **Objectif** : faire une prédiction $\hat{y}$ pour une nouvelle entrée $\hat{x}$

Régression plymoniale, modèle

- On va supposer qu'une bonne prédiction aurait une forme polynomiale

$$f(x, w) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M$$

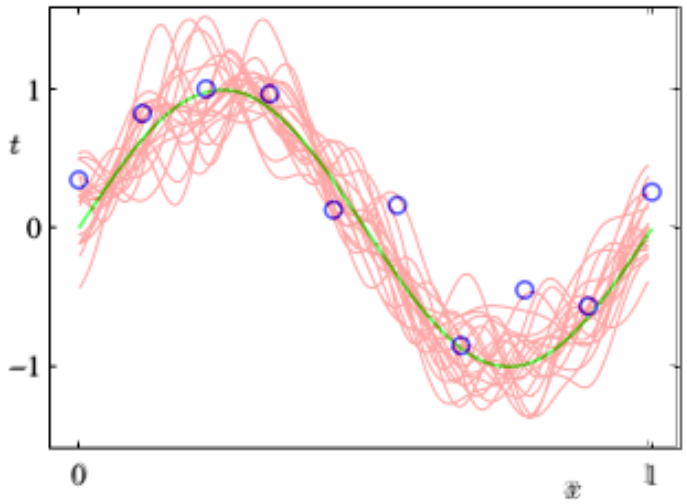
$$f(x, w) = \sum_{j=0}^{M} w_j x^j$$



- $f(x, w)$ est notre modèle. Il représente nos hypothèses sur le problème à résoudre
  - La valeur de M représente le degré du polynôme.
  - Le vecteur $w = (w_1, w_2, \ldots, w_M)$ représente les paramètres du modèle qu'on doit trouver

Régression : minimisation de perte (coût,erreur)

- Comment trouver w ? (problème d'apprentissage)
- On cherche le $w*$ qui minimise la somme de notre perte / erreur / coût sur l'ensemble d'entraînement

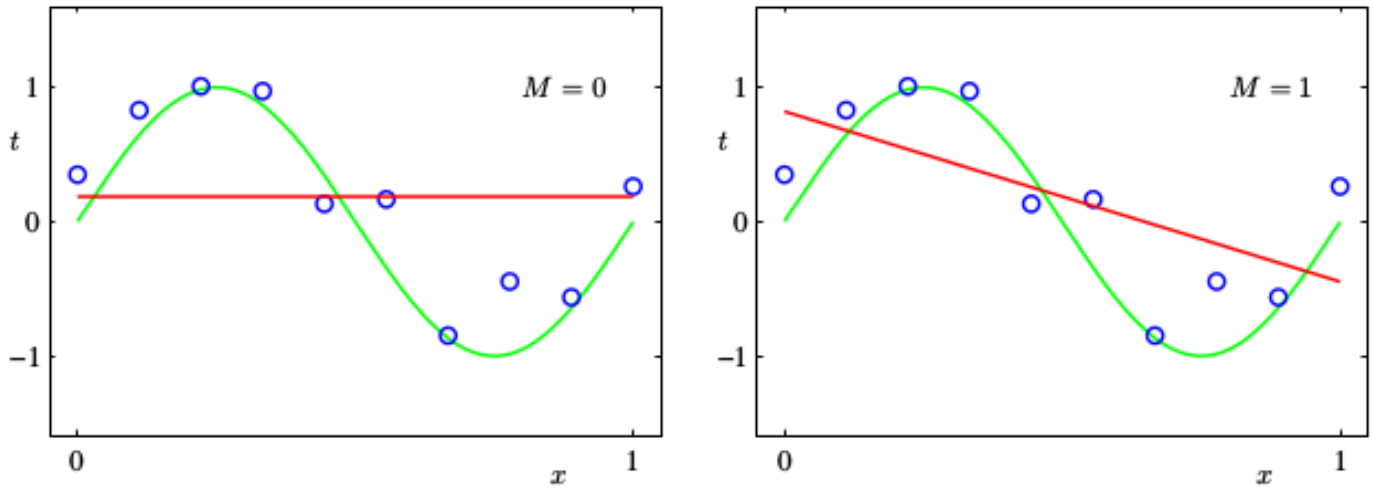


$$E(w) = \frac{1}{2}\sum_{n=1}^{N}\{f(x_n, w) - y_n\}^2$$

- Un algorithme d'apprentissage résoudrait ce problème : trouver $w*$ à partir des données

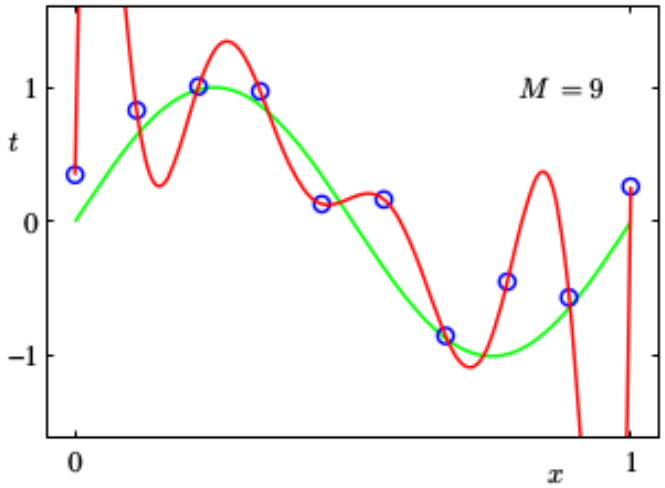
Régression : sous-apprentissage

- Comment choisir M (le degré du polynôme) ?
  - De trop petites valeurs auront une grande perte sur l'ensemble d'entraînement : situation de **sous-apprentissage**



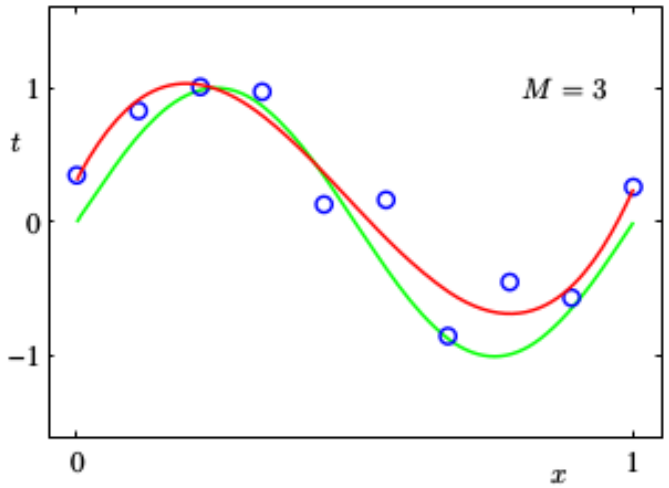
Régression : sur-apprentissage

- Comment choisir M (le degré du polynôme) ?
  - De trop grandes valeurs vont seulement apprendre l'ensemble d'entraînement "par coeur" : situation de **sur-apprentissage**
  - Apprendre à prédire ce que n'est pas prévisible à partir de x seulement (ex. du bruit)



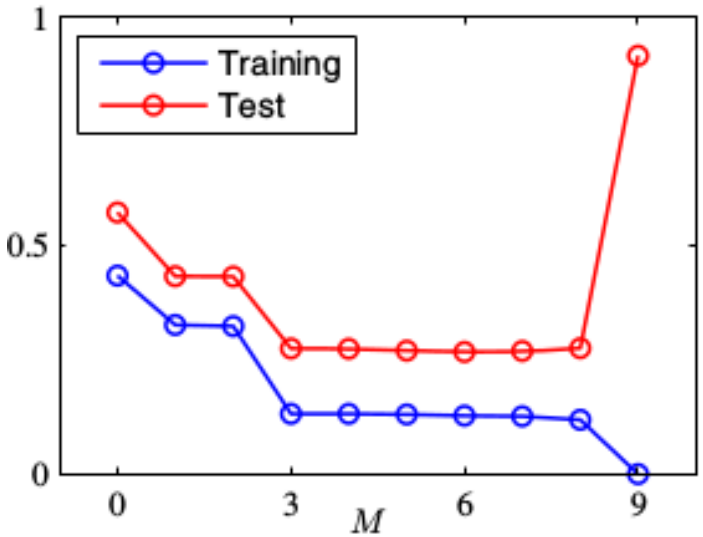
Régression : sélection de modèle

- Comment choisir M (le degré du polynôme) ?
  - On voudrait une valeur intérmédiaire qui permet de retrouver la tendance générale de la relation entre x et y, sans le bruit
  - c'est ce qui va permettre de bien généraliser à de nouvelles entrées !
  - trouver cette meilleure valeur de M s'apelle de la **sélection de modèle**.
  - Il existe différentes techniques pour trouver le meilleur M.



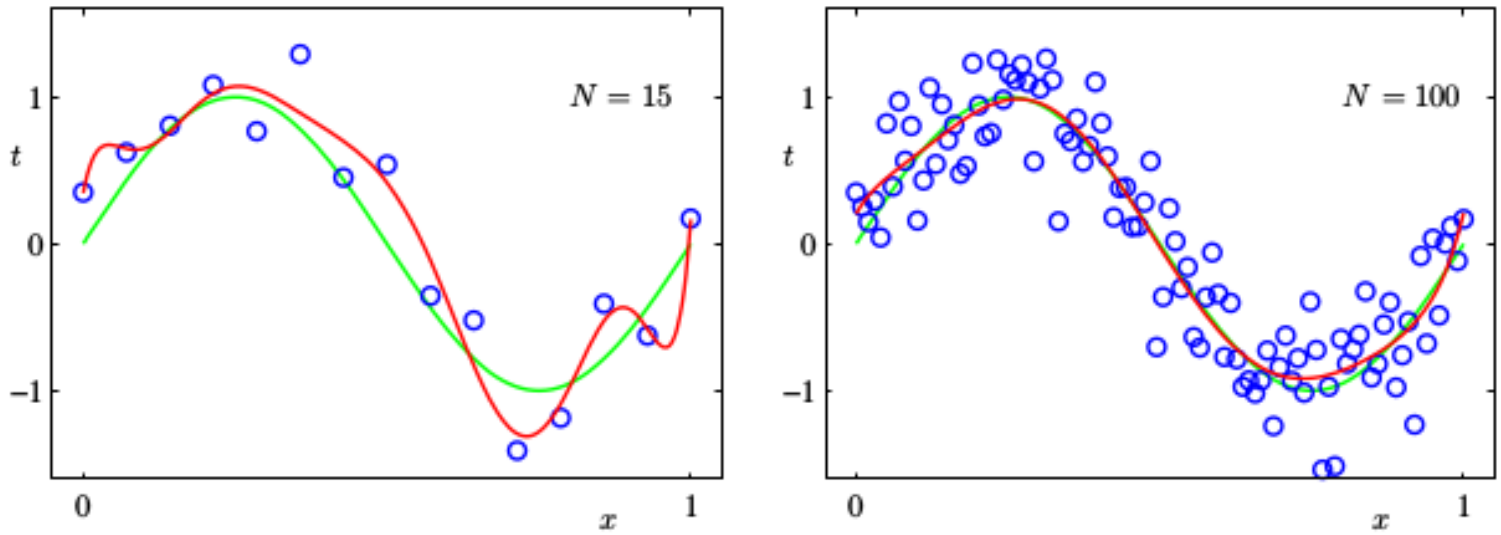
Régression : capacité d'un modèle, performance

- La **capacité** d'un modèle est son aptitude à apprendre "par coeur". Exemple : plus M est grand, plus le modèle a de capacité.
- Plus la capacité est grande, plus la différence entre l'erreur d'entraînement er l'erreur de test augmente.
- En régression, l'erreur sur tout un ensemble est souvent mesurée par la racine de la moyenne des erreurs au carré $E_{RMS} = \sqrt{2E(w*)/N}$



Régression : généralisation vs. quantité de données

- Plus la quantité de données d'entraînement augmente, plus le modèle entraîné va bien généraliser



Régression : régularisation

- Comment utiliser un grand M avec peu de données
- **Régularisation** : on réduit la capacité autrement
  - On pénalise la somme des carrés des paramètres

$$\widetilde{E}(x) = \frac{1}{2}\sum_{n=1}^{N}\{f(x_n, w) - y_n\}^2 + \frac{\lambda}{2}||w||^2$$

  - où $||w||^2 = w^T w = w_0^2 + w_1^2 + ... + w_M^2$

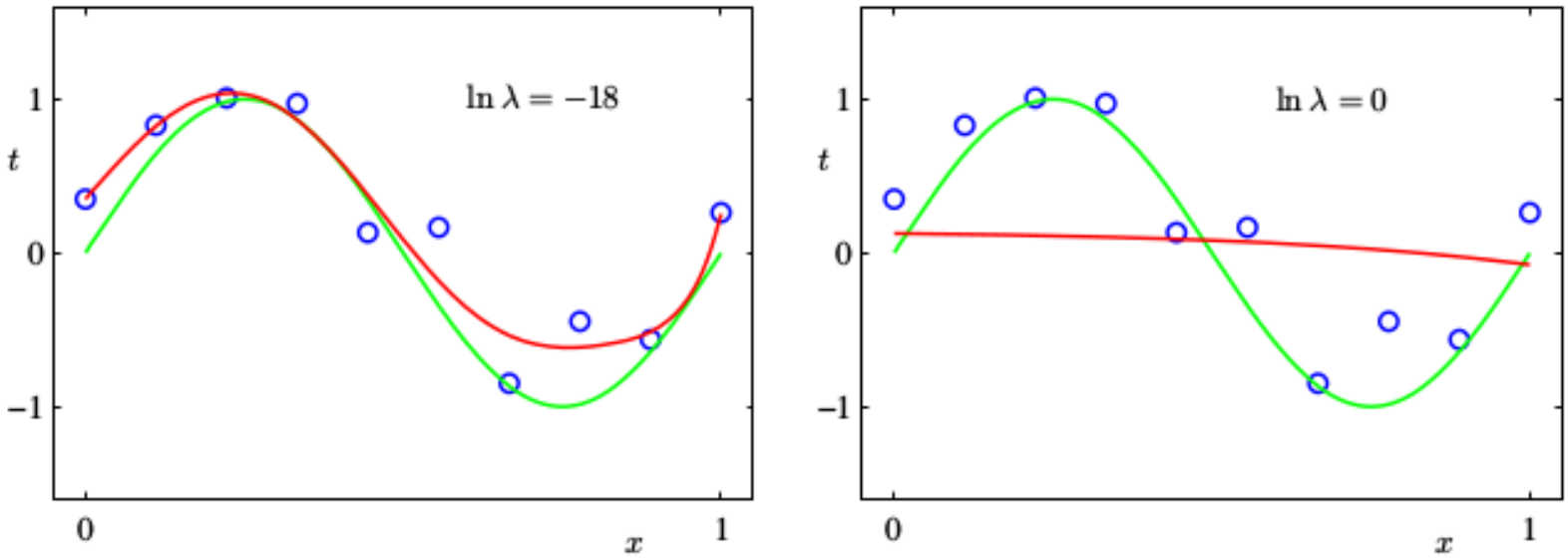
Régression : régularisation

- Valeurs des paramètres $w*$ pour différents M, sans régularisation

	$M=0$	$M=1$	$M=3$	$M=9$
w_0^*	0.19	0.82	0.31	0.35
w_1^*		−1.27	7.99	232.37
w_2^*			−25.43	−5321.83
w_3^*			17.37	48568.31
w_4^*				−231639.30
w_5^*				−231639.30
w_6^*				640042.26
w_7^*				−1061800.52
w_8^*				1042400.18
w_9^*				−557682.99
w_9^*				125201.43

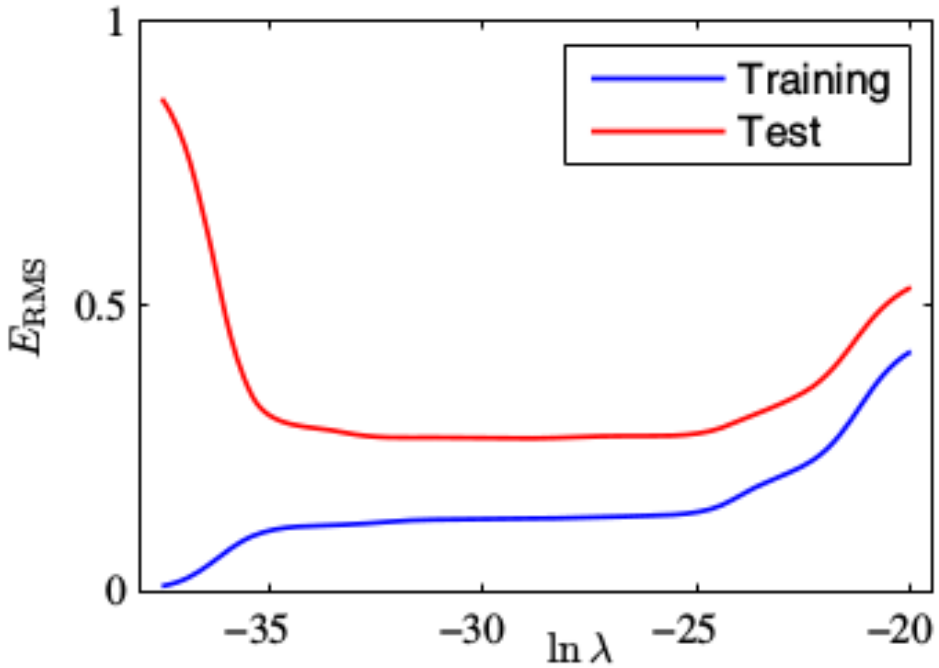
Régression : régularisation

- Plus la régularisation est forte, moins le modèle sera flexible (donc il aura moins de capacité)



Régression : régularisation

- Comme M, la force de la régularisation a une influence sur l'erreur d'entrainement et de test



Régression : sélection de modèle

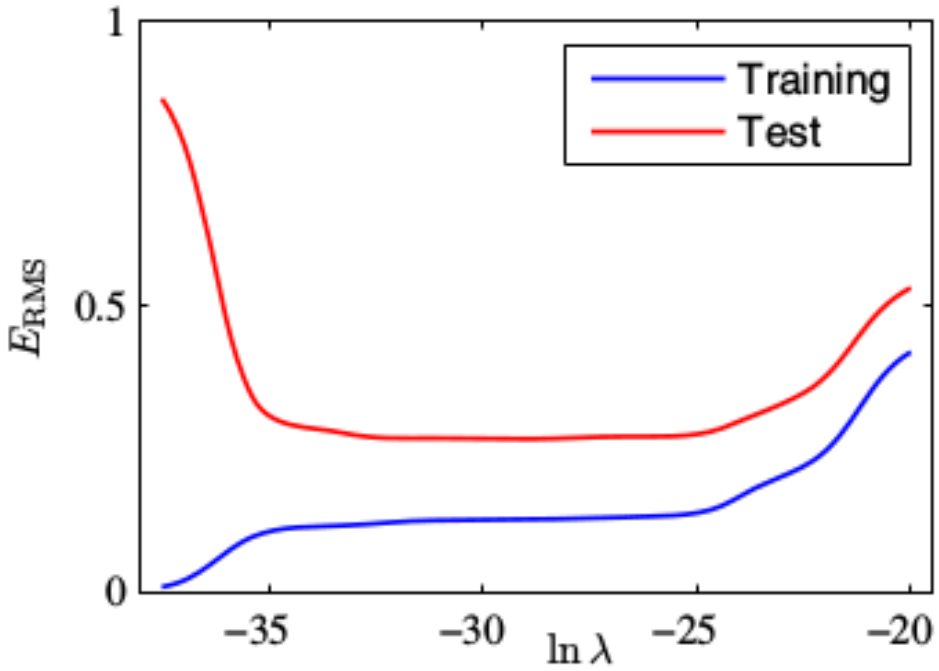
- Soit l'algorithme d'apprentissage qui optimise

$$\widetilde{E}(x) = \frac{1}{2}\sum_{n=1}^{N}\{f(w_n, w) - y_n\}^2 + \frac{\lambda}{2}||w||^2$$

- On appelle M et λ des **hyper-paramètres** : ils doivent être déterminés avant l'entraînement.
- La **sélection de modèle** est le choix de la valeur de ces **hyper-paramètres**.

Régression : hyper-paramètres

- Le choix des hyper-paramètres va influencer la performance sur de nouvelles données (test)



Régression : ensemble de validation

- Solution I : on réserve des données d'entraînement pour comparer différentes valeurs
  - garde la majorité pour l'ensemble d'entraînement $\mathcal{D}_{train}$ (ex. 80%)
  - le reste, $\mathcal{D}_{valid}$ (ex. 20%), servira à comparer les hyper-paramètres
- On appelle $\mathcal{D}_{valid}$ l'**ensemble de validation**

Régression : algorithme de sélection de modèle

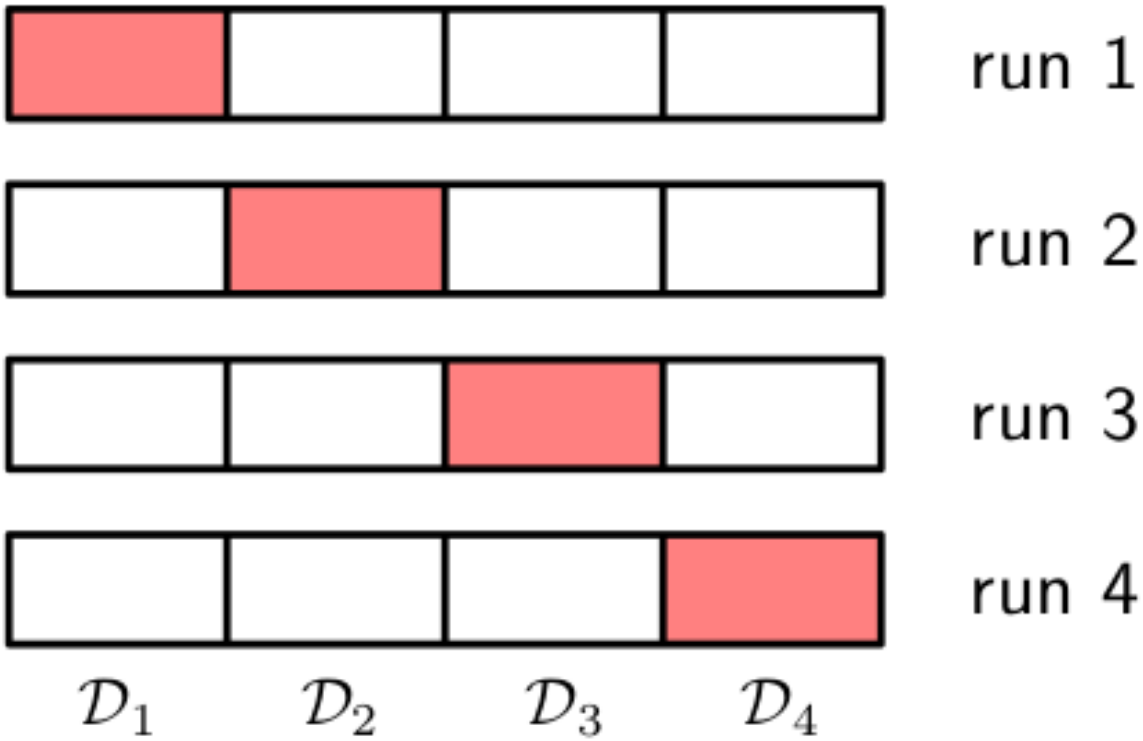
- Pour chaque valeur d'hyper-paramètres à comparer
  - obtenir un modèle entraîné à partir de $\mathcal{D}_{train}$
  - évaluer la performance du modèle sur $\mathcal{D}_{valid}$
- retourner le choix d'hyper-paramètres ayant donné le modèle avec la meilleure performance sur $\mathcal{D}_{valid}$

Régression : S-fold cross-validation

- Lorsqu'on a peu de données, 20% est trop peu pour estimer la performance de généralisation
- On pourrait répéter la procédure de séparation *train/valid* plus d'une fois
- **S-fold cross-validation** : divise les données en S portions différentes. Chaque portion est utilisée une fois en tant que $\mathcal{D}_{valid}$

Régression : S-fold cross-validation

- Exemple : $S = 4$



Régression : S-fold cross-validation

Sélection de modèle avec S-fold cross-validation

Pour $s = 1...S$
  Pour chaque valeur d'hyper-paramètres à comparer
    obtenir un modèle entraîné à partir de $\mathcal{D}_{train} = \mathcal{D} \backslash \mathcal{D}_s$
    évaluer la performance du modèle sur $\mathcal{D}_{valid} = \mathcal{D}_s$
  retourner la valeur des hyper-paramètres ayant la meilleure performance moyenne sur les ensembles $\mathcal{D}_{valid}$

- Si $S = N$, on parle alors de méthode **leave-one-out**

Régression : S-fold cross-validation

- Comment déterminer la liste des valeurs d'hyper-paramètres à comparer ?
- recherche sur une grille
  - détermine une liste de valeur pour chaque hyper-paramètre
  - construit la liste de toutes les combinaisons possibles