

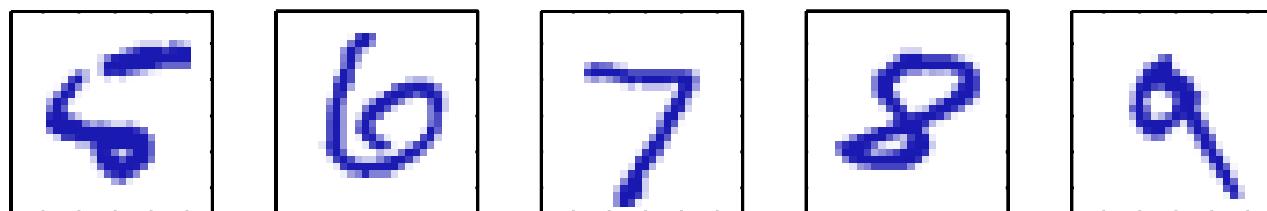
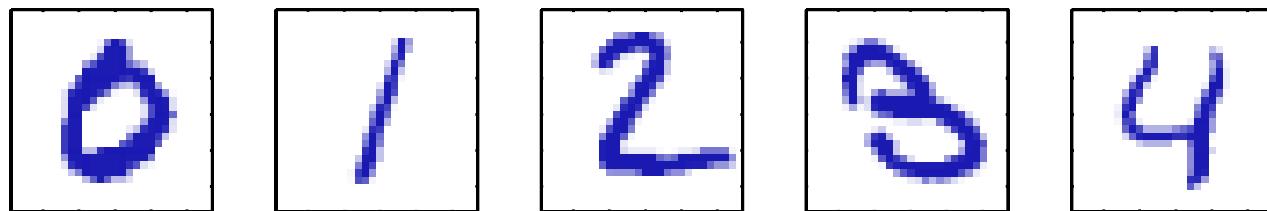
Apprentissage automatique

Concepts fondamentaux - motivation

APPRENTISSAGE AUTOMATIQUE

Sujets: motivation

- Comment développer une intelligence artificielle ?
 - exemple : reconnaître des caractères manuscrits

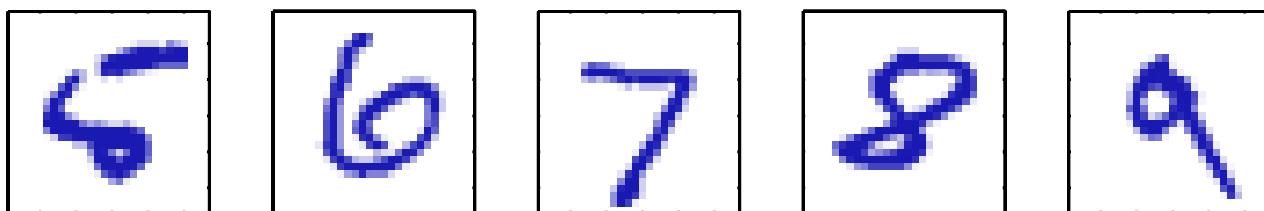
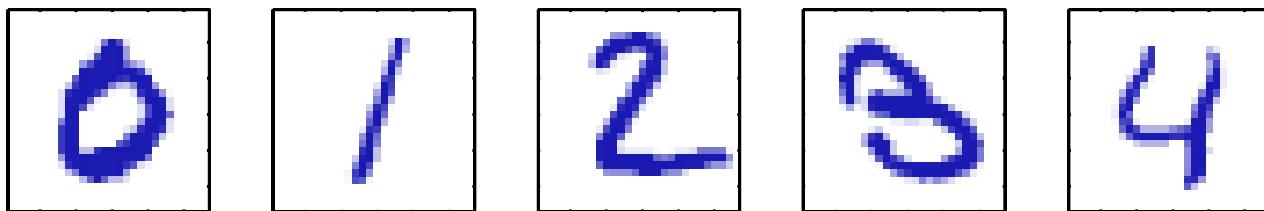


- Par énumération de règles ?
 - si intensité du pixel à la position (15,24) est plus grand que 50, et pixel à la position ... alors c'est un «3»

APPRENTISSAGE AUTOMATIQUE

Sujets: motivation

- Comment développer une intelligence artificielle ?
 - exemple : reconnaître des caractères manuscrits

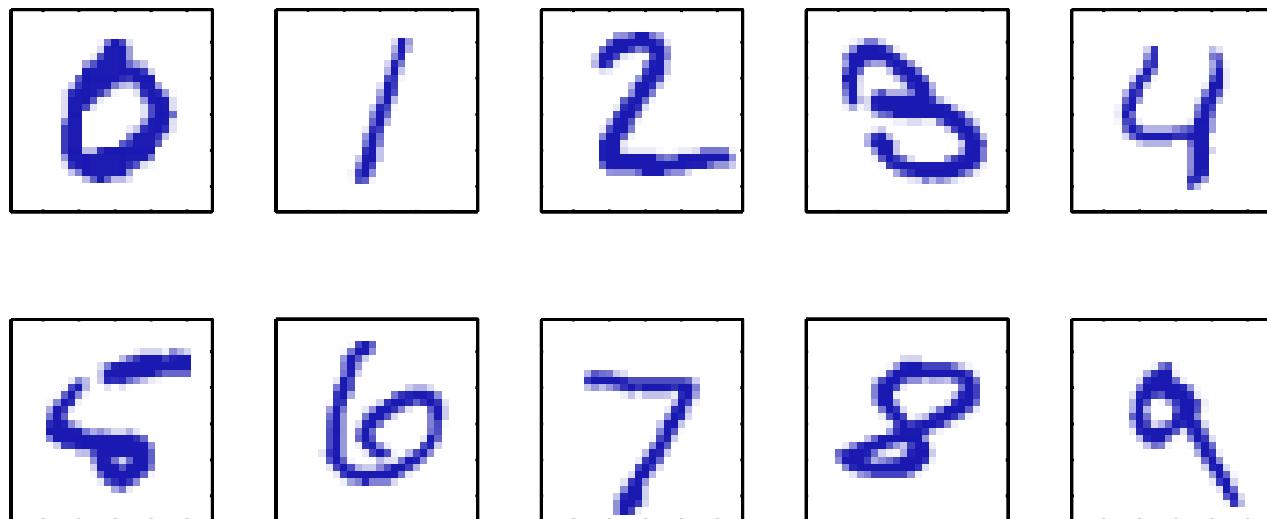


- ~~Par énumération de règles ?~~
 - trop fastidieux, difficile de couvrir tous les cas d'espèce

APPRENTISSAGE AUTOMATIQUE

Sujets: motivation

- Comment développer une intelligence artificielle ?
 - exemple : reconnaître des caractères manuscrits

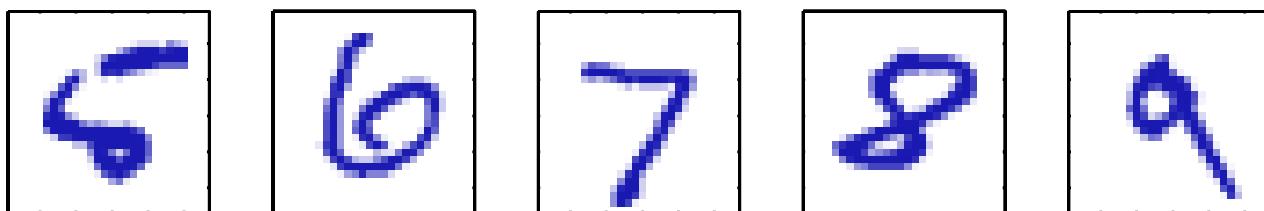
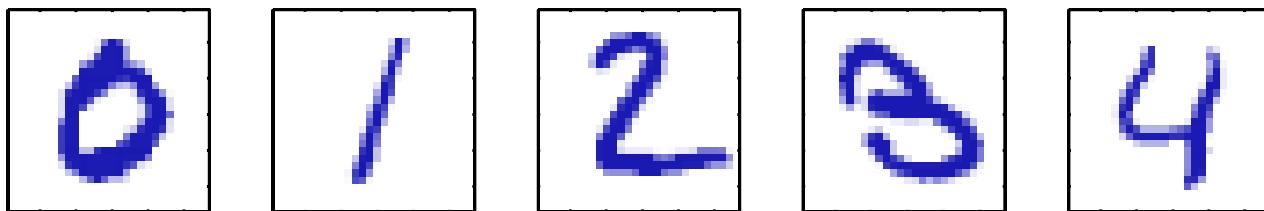


- En donnant à l'ordinateur la capacité d'apprendre à le faire!
 - laisser l'ordinateur faire des essais et apprendre de ses erreurs

APPRENTISSAGE AUTOMATIQUE

Sujets: motivation

- Comment développer une intelligence artificielle ?
 - exemple : reconnaître des caractères manuscrits



- Apprentissage automatique / *machine learning* :
 - le domaine s'intéressant à l'étude de tels algorithmes

APPRENTISSAGE AUTOMATIQUE

Sujets: données d'entraînement vs. généralisation

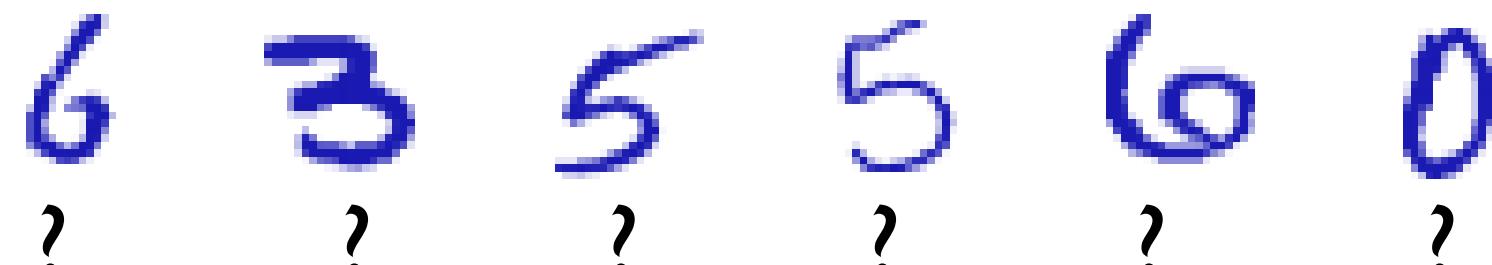
- Les algorithmes d'apprentissage procèdent comme suit :

- › on fournit à l'algorithme des **données d'entraînement** ...



'9' '6' '6' '5' '4' '0'

- › ... et l'algorithme retourne un «programme» capable de **généraliser** à de nouvelles données



Apprentissage automatique

Concepts fondamentaux - notation et nomenclature

APPRENTISSAGE AUTOMATIQUE

Sujets: données d'entraînement vs. généralisation

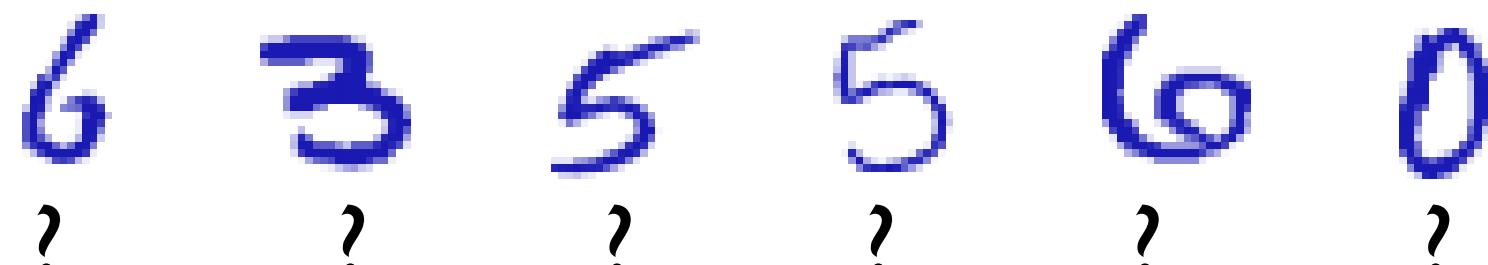
- Les algorithmes d'apprentissage procèdent comme suit :

- › on fournit à l'algorithme des **données d'entraînement** ...



'9' '6' '6' '5' '4' '0'

- › ... et l'algorithme retourne un «programme» capable de **généraliser** à de nouvelles données

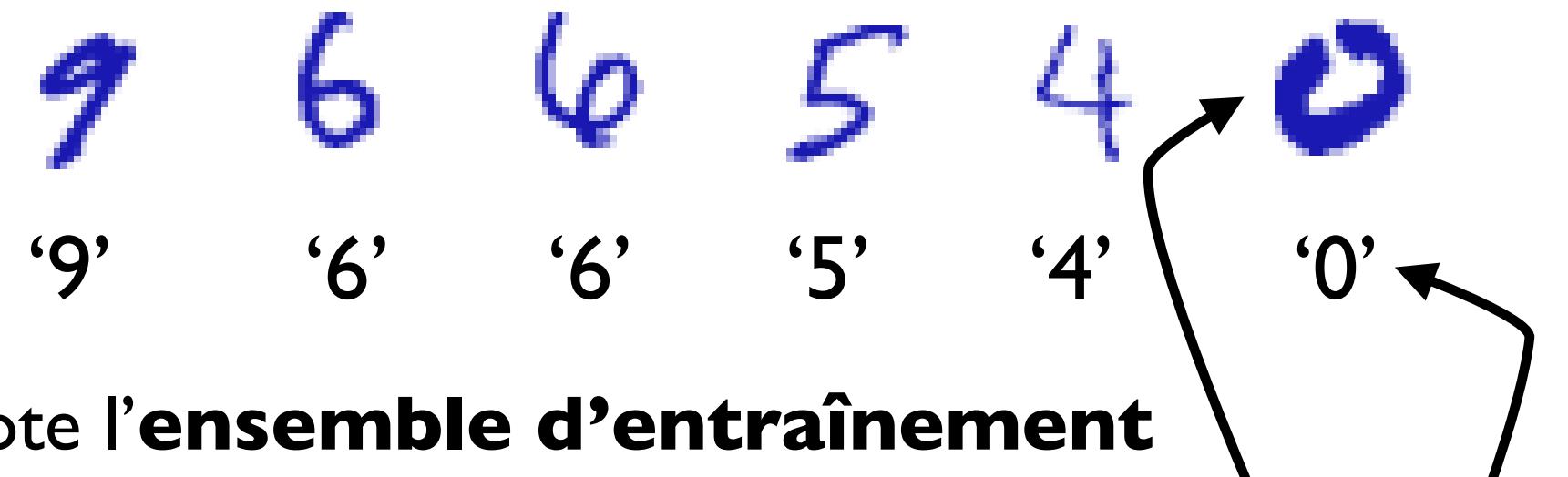


APPRENTISSAGE AUTOMATIQUE

Sujets: ensemble d'entraînement, entrée, cible

- Les algorithmes d'apprentissage procèdent comme suit :

- on fournit à l'algorithme des **données d'entraînement** ...



- on note l'**ensemble d'entraînement**

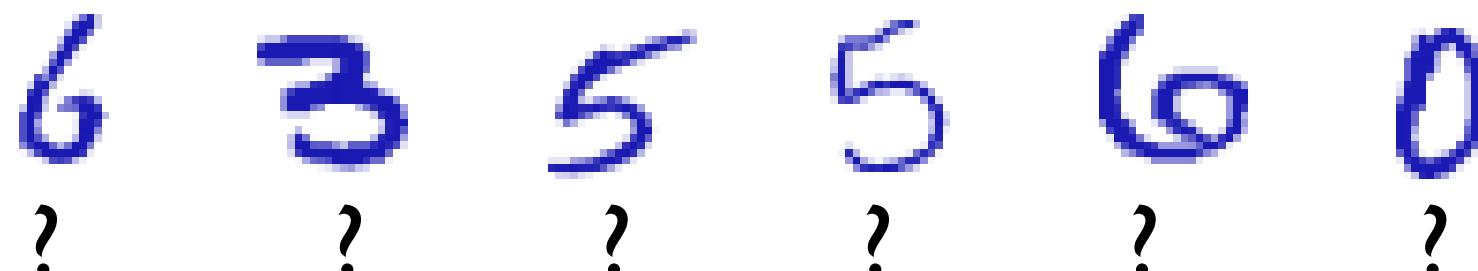
$$\mathcal{D} = \{(\mathbf{x}_1, t_1), \dots, (\mathbf{x}_N, t_N)\}$$

- on appelle \mathbf{x}_n une **entrée** et t_n la **cible**

APPRENTISSAGE AUTOMATIQUE

Sujets: modèle

- Les algorithmes d'apprentissage procèdent comme suit :
 - on note le «programme» généré par l'algorithme d'apprentissage $y(x)$
 - on va aussi appeler $y(x)$ un **modèle**
 - ... et l'algorithme retourne un «programme» capable de **généraliser** à de nouvelles données

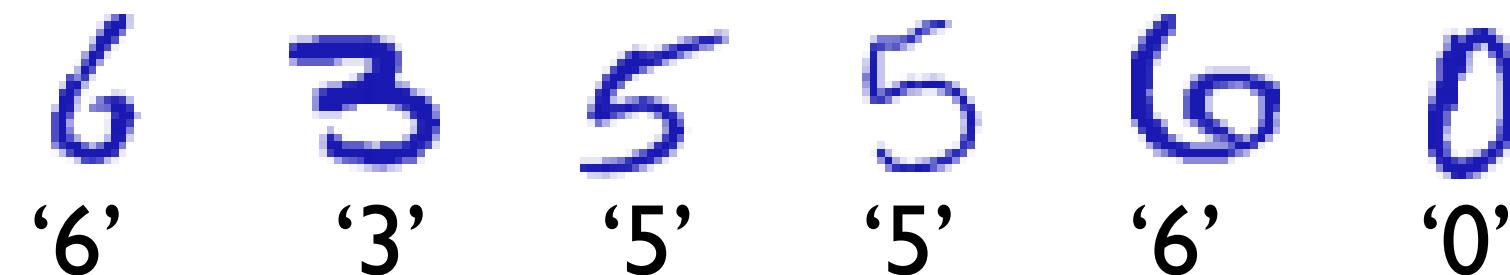


APPRENTISSAGE AUTOMATIQUE

Sujets: ensemble de test

- Les algorithmes d'apprentissage procèdent comme suit :

- on utilise un **ensemble de test** $\mathcal{D}_{\text{test}}$ pour mesurer la performance de généralisation de notre modèle $y(x)$
- ... et l'algorithme retourne un «programme» capable de **généraliser** à de nouvelles données



Apprentissage automatique

Concepts fondamentaux - types d'apprentissage

APPRENTISSAGE AUTOMATIQUE

Sujets: types d'apprentissage

- Il existe différents types d'apprentissage
 - apprentissage supervisé : il y a une cible à prédire

$$\mathcal{D} = \{(\mathbf{x}_1, t_1), \dots, (\mathbf{x}_N, t_N)\}$$

- apprentissage non-supervisé : cible n'est pas fournie

$$\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$$

- apprentissage par renforcement (non couvert dans ce cours)

TYPES D'APPRENTISSAGE

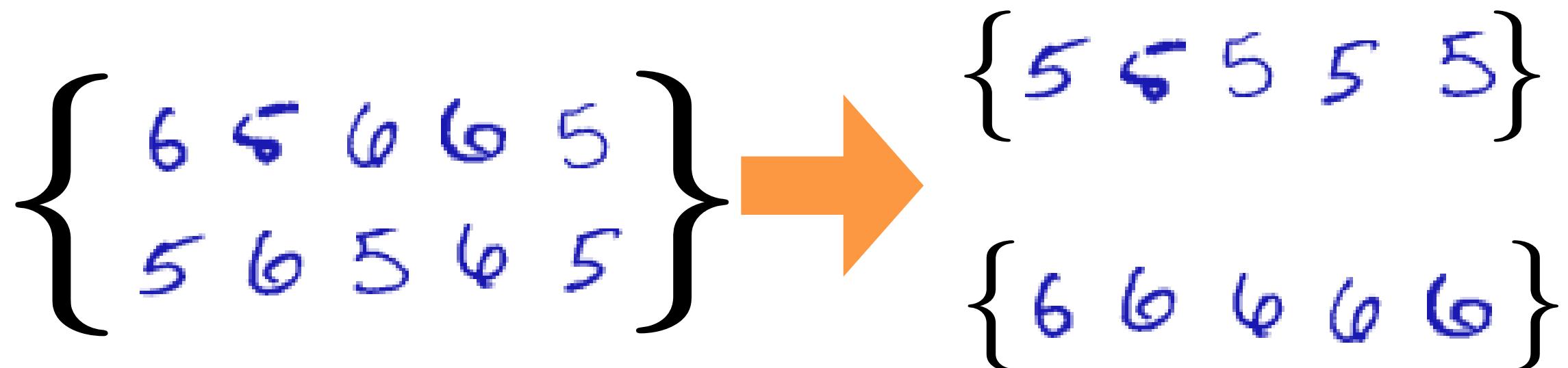
Sujets: apprentissage supervisé, classification, régression

- L'apprentissage supervisé est lorsqu'on a une cible à prédire
 - **classification :** la cible est un indice de classe $t \in \{1, \dots, K\}$
 - exemple : reconnaissance de caractères
 - ✓ x : vecteur des intensités de tous les pixels de l'image
 - ✓ t : identité du caractère
 - **régression :** la cible est un nombre réel $t \in \mathbb{R}$
 - exemple : prédiction de la valeur d'une action à la bourse
 - ✓ x : vecteur contenant l'information sur l'activité économique de la journée
 - ✓ t : valeur d'une action à la bourse le lendemain

TYPES D'APPRENTISSAGE

Sujets: apprentissage non-supervisé, partitionnement

- L'apprentissage non-supervisé est lorsqu'une cible n'est pas explicitement donnée
 - partitionnement de données / *clustering*

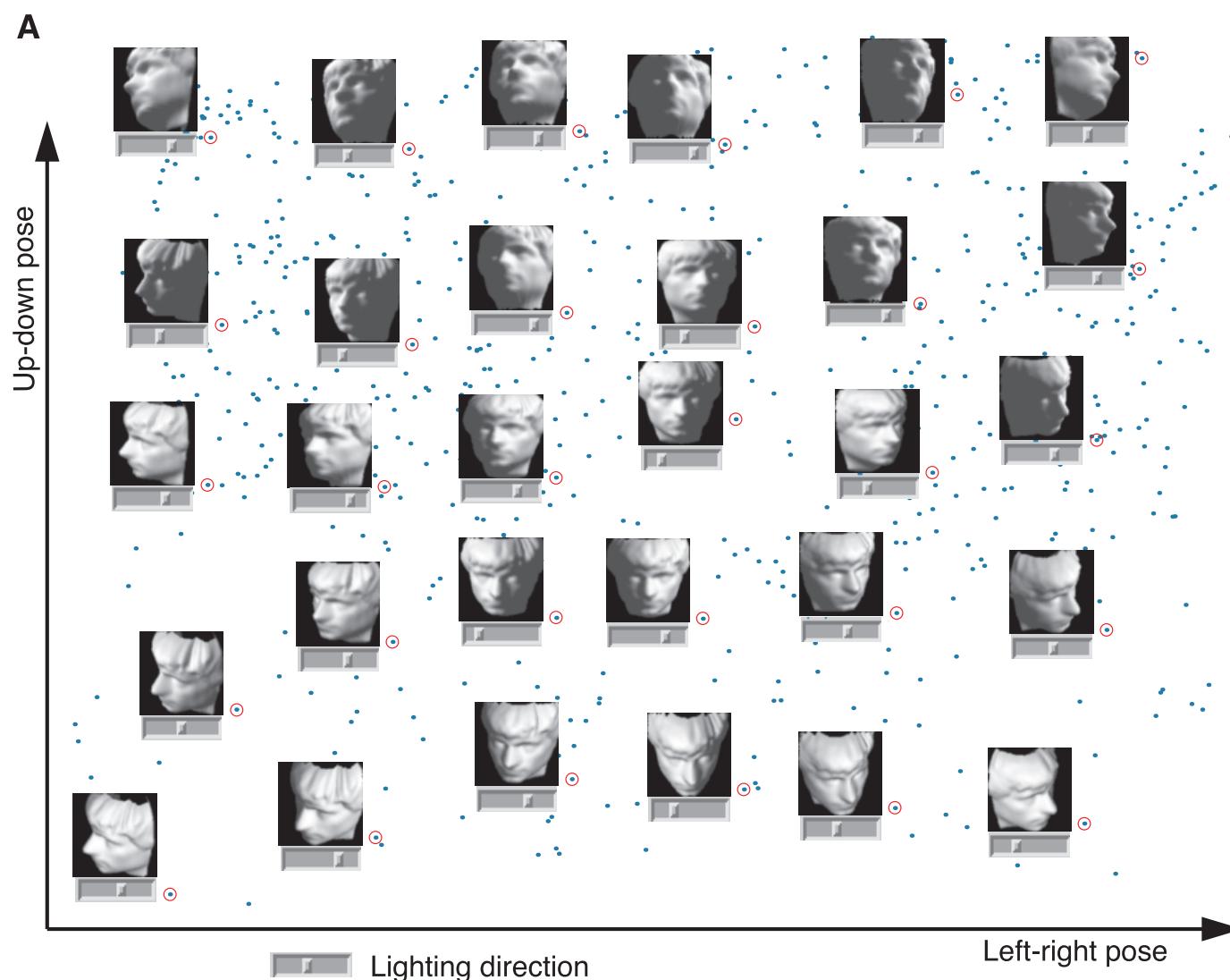


TYPES D'APPRENTISSAGE

Sujets: apprentissage non-supervisé, visualisation

- L'apprentissage non-supervisé est lorsqu'une cible n'est pas explicitement donnée
 - visualisation de données

Tenenbaum, de Silva,
Langford, (2000)

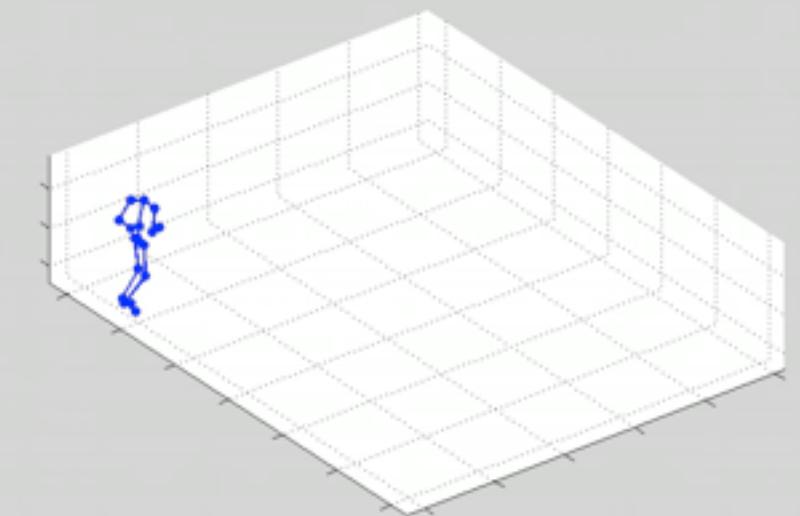


TYPES D'APPRENTISSAGE

Sujets: apprentissage non-supervisé, estimation de densité

- L'apprentissage non-supervisé est lorsqu'une cible n'est pas explicitement donnée
 - estimation de densité : apprendre la loi de probabilité $p(\mathbf{x})$ ayant généré les données
 - pour générer de nouvelles données réalistes
 - pour distinguer les «vrais» données des «fausses» données (*spam filtering*)
 - compression de données

Taylor, Hinton, Roweis
(2006)



001

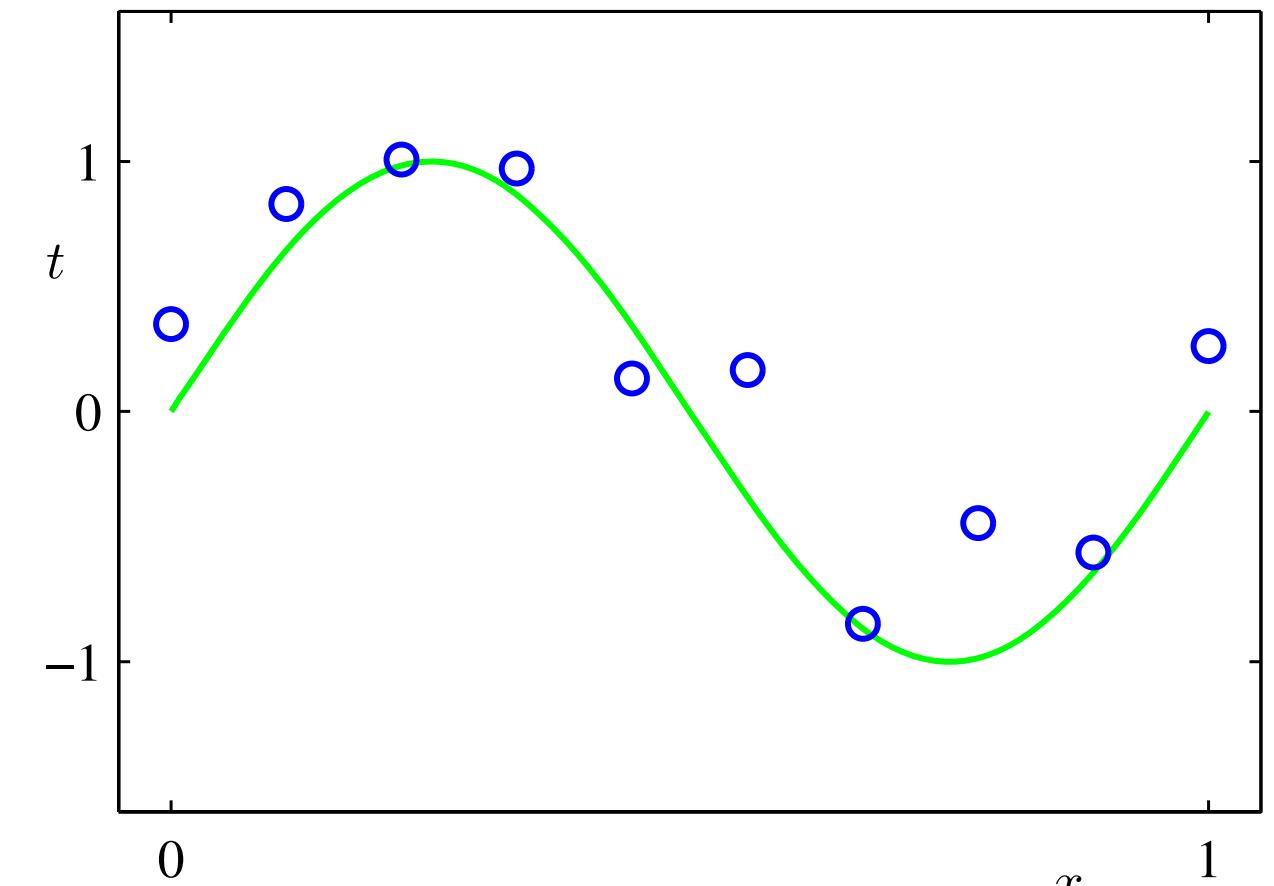
Apprentissage automatique

Concepts fondamentaux - exemple : régression polynomiale

EXEMPLE: RÉGRESSION

Sujets: régression 1D

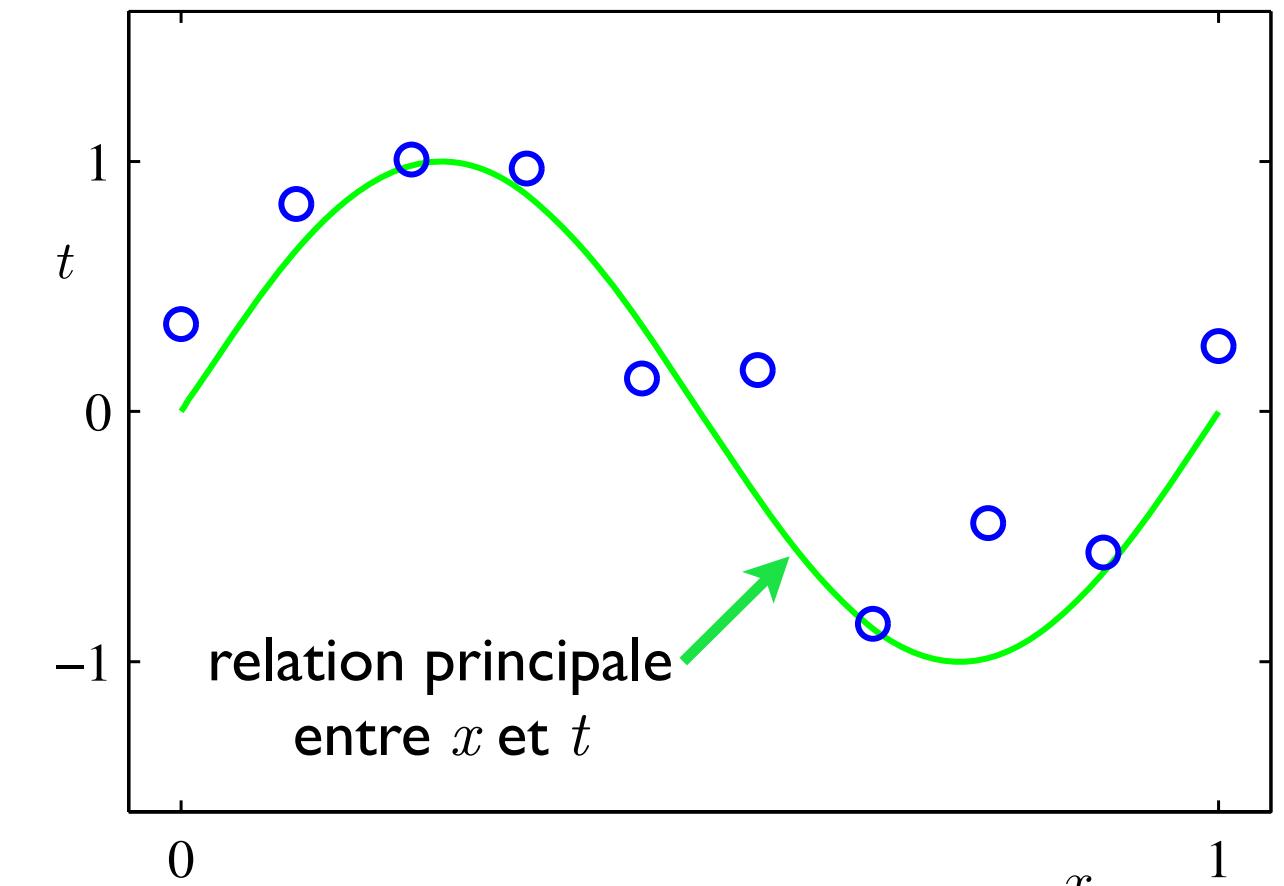
- Exemple simple: régression en une dimension
 - entrée : scalaire x
 - cible : scalaire t
- Données d'entraînement \mathcal{D} contiennent :
 - $\mathbf{x} \equiv (x_1, \dots, x_N)^T$
 - $\mathbf{t} \equiv (t_1, \dots, t_N)^T$
- Objectif :
 - faire une prédiction \hat{t} pour une nouvelle entrée \hat{x}



EXEMPLE: RÉGRESSION

Sujets: régression 1D

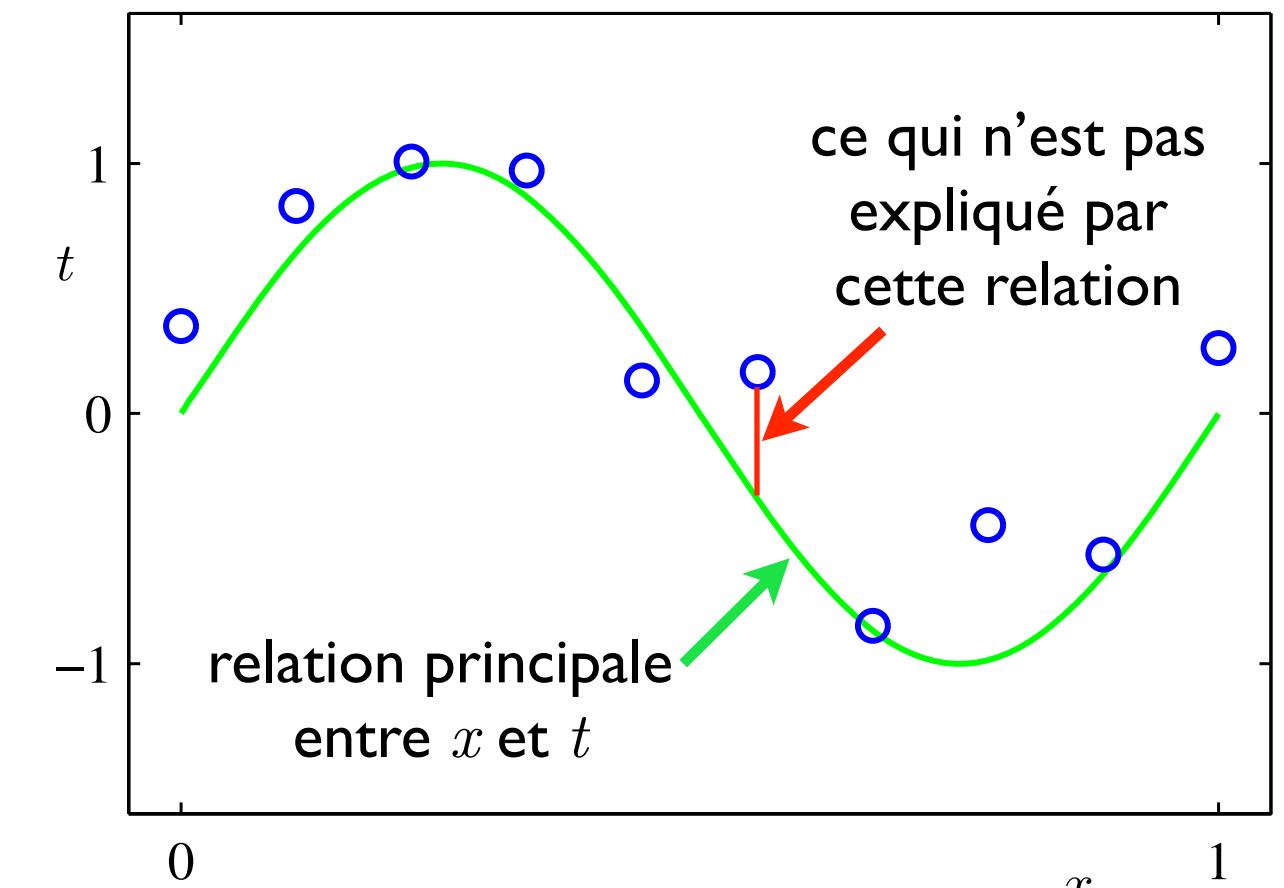
- Exemple simple: régression en une dimension
 - entrée : scalaire x
 - cible : scalaire t
- Données d'entraînement \mathcal{D} contiennent :
 - $\mathbf{x} \equiv (x_1, \dots, x_N)^T$
 - $\mathbf{t} \equiv (t_1, \dots, t_N)^T$
- Objectif :
 - faire une prédiction \hat{t} pour une nouvelle entrée \hat{x}



EXEMPLE: RÉGRESSION

Sujets: régression 1D

- Exemple simple: régression en une dimension
 - entrée : scalaire x
 - cible : scalaire t
- Données d'entraînement \mathcal{D} contiennent :
 - $\mathbf{x} \equiv (x_1, \dots, x_N)^T$
 - $\mathbf{t} \equiv (t_1, \dots, t_N)^T$
- Objectif :
 - faire une prédiction \hat{t} pour une nouvelle entrée \hat{x}



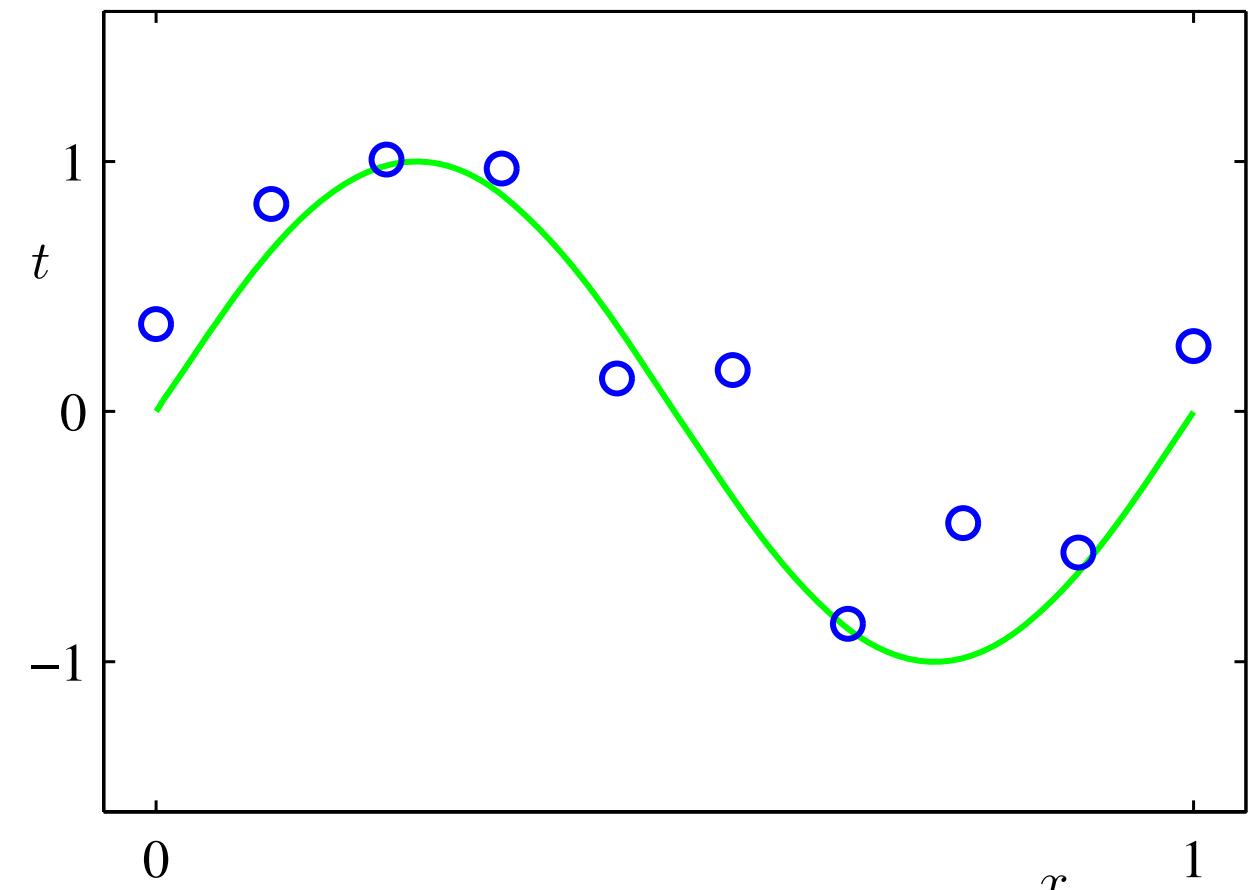
EXEMPLE: RÉGRESSION

Sujets: régression polynomiale, modèle

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

$$\begin{aligned}y(x, \mathbf{w}) &= w_0 + w_1 x + w_2 x^2 + \dots + w_M x^M \\&= \sum_{j=0}^M w_j x^j\end{aligned}$$

- $y(x, \mathbf{w})$ est notre **modèle**
 - représente nos hypothèses sur le problème à résoudre
 - a normalement des paramètres, qu'on doit trouver (\mathbf{w} ici)



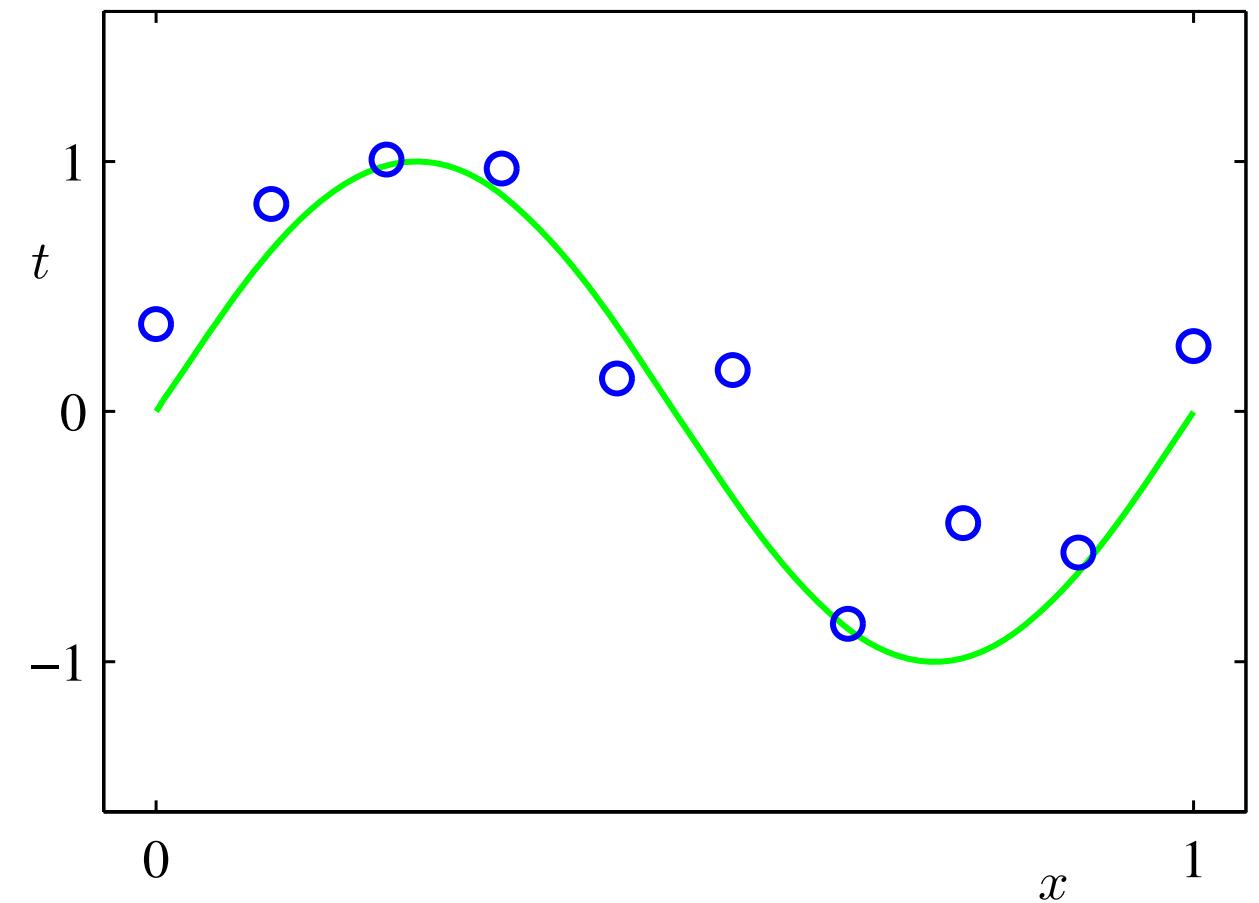
EXEMPLE: RÉGRESSION

Sujets: régression polynomiale, modèle

- On peut voir un modèle comme un «programme» représenté mathématiquement

```
def predict(x,w):  
    x_poly = x ** np.arange(len(w))  
    return np.dot(x_poly,w)
```

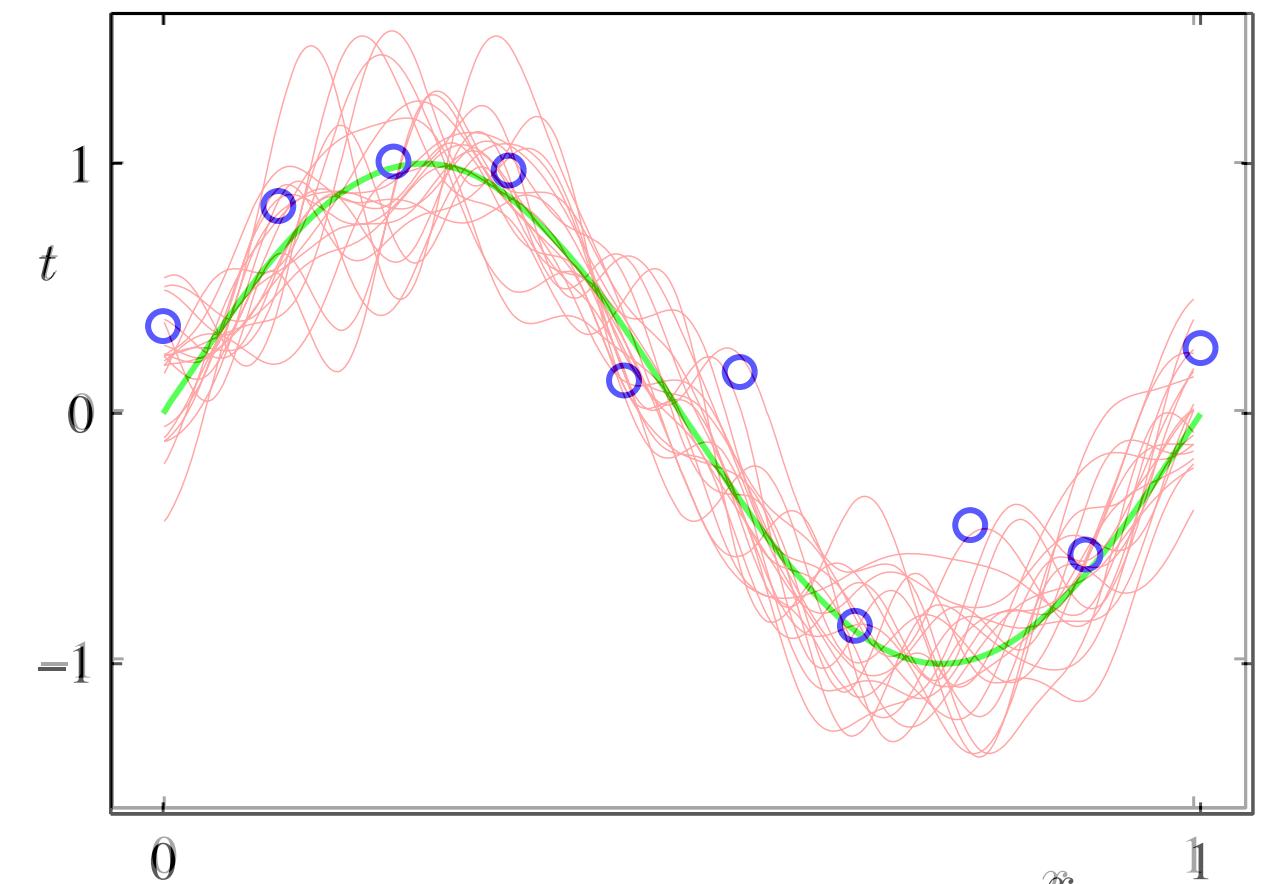
- $y(x, w)$ est notre **modèle**
 - représente nos hypothèses sur le problème à résoudre
 - a normalement des paramètres, qu'on doit trouver (w ici)



EXEMPLE: RÉGRESSION

Sujets: minimisation de perte (coût, erreur)

- Comme trouver w ? (**problème d'apprentissage**)
 - on cherche le w^* qui minimise la somme de notre perte / erreur / coût sur l'ensemble d'entraînement
- $$E(w) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, w) - t_n\}^2$$
- le « $\frac{1}{2}$ » n'est pas important (mais simplifiera certains calculs)
- Un algorithme d'apprentissage résoudrait ce problème
 - à partir des données, il va retourner w^*



Apprentissage automatique

Concepts fondamentaux - sur-apprentissage / sous-apprentissage

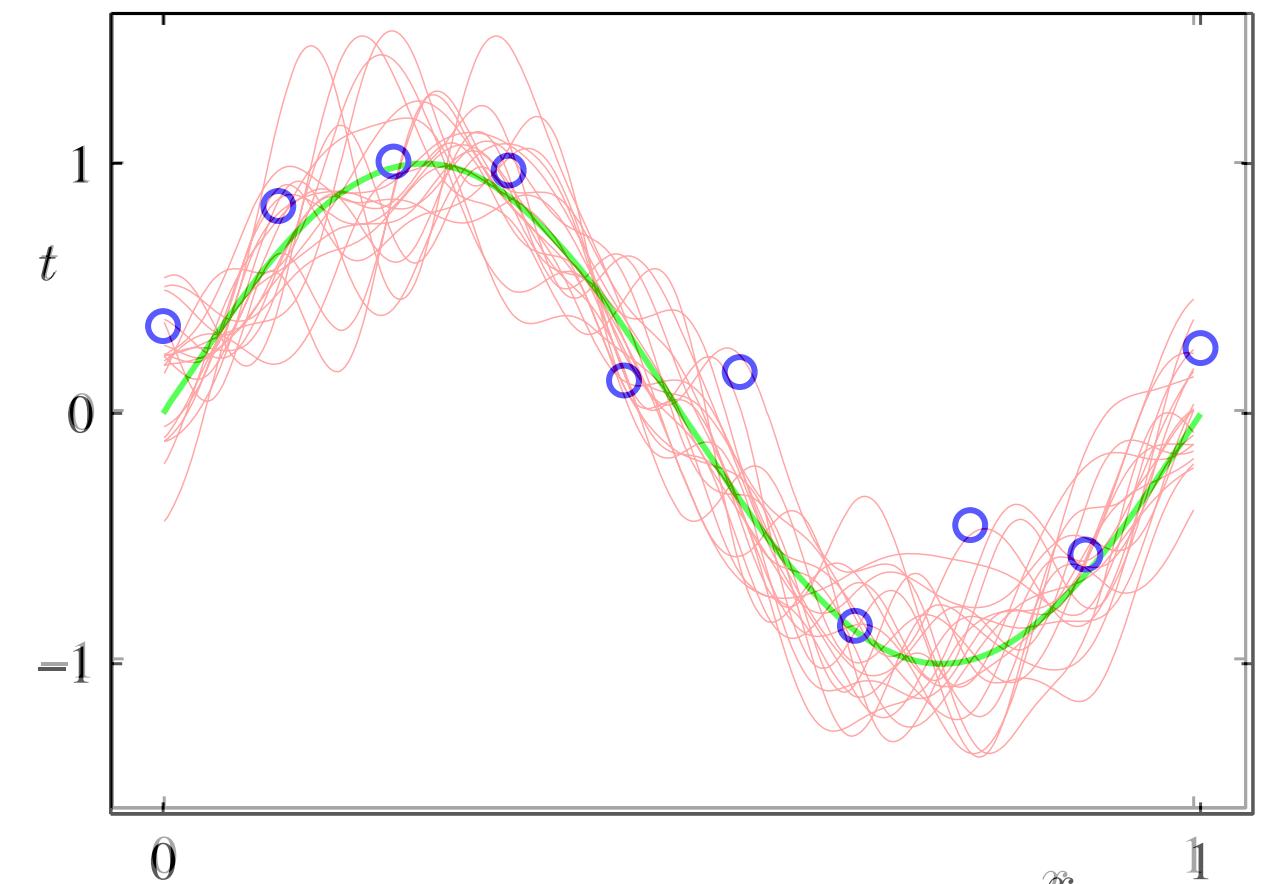
EXEMPLE: RÉGRESSION

Sujets: minimisation de perte (coût, erreur)

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

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

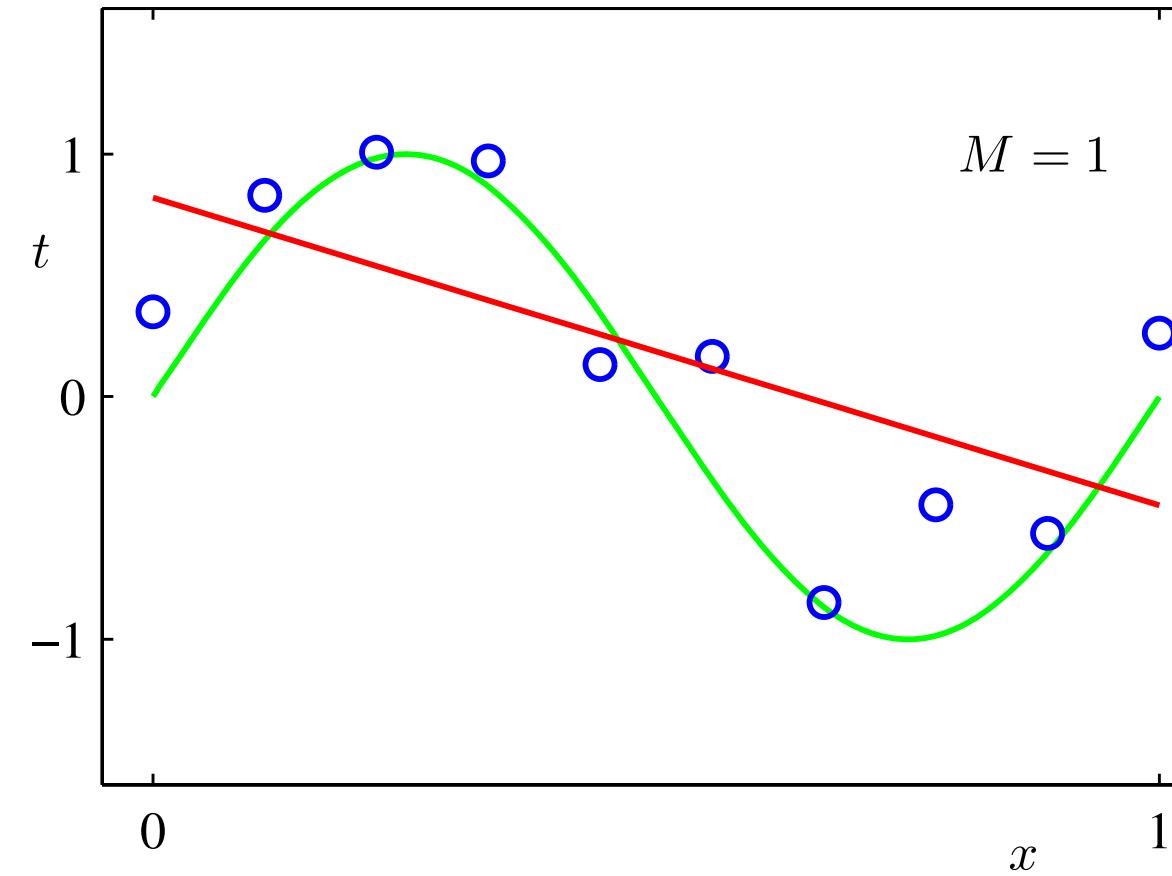
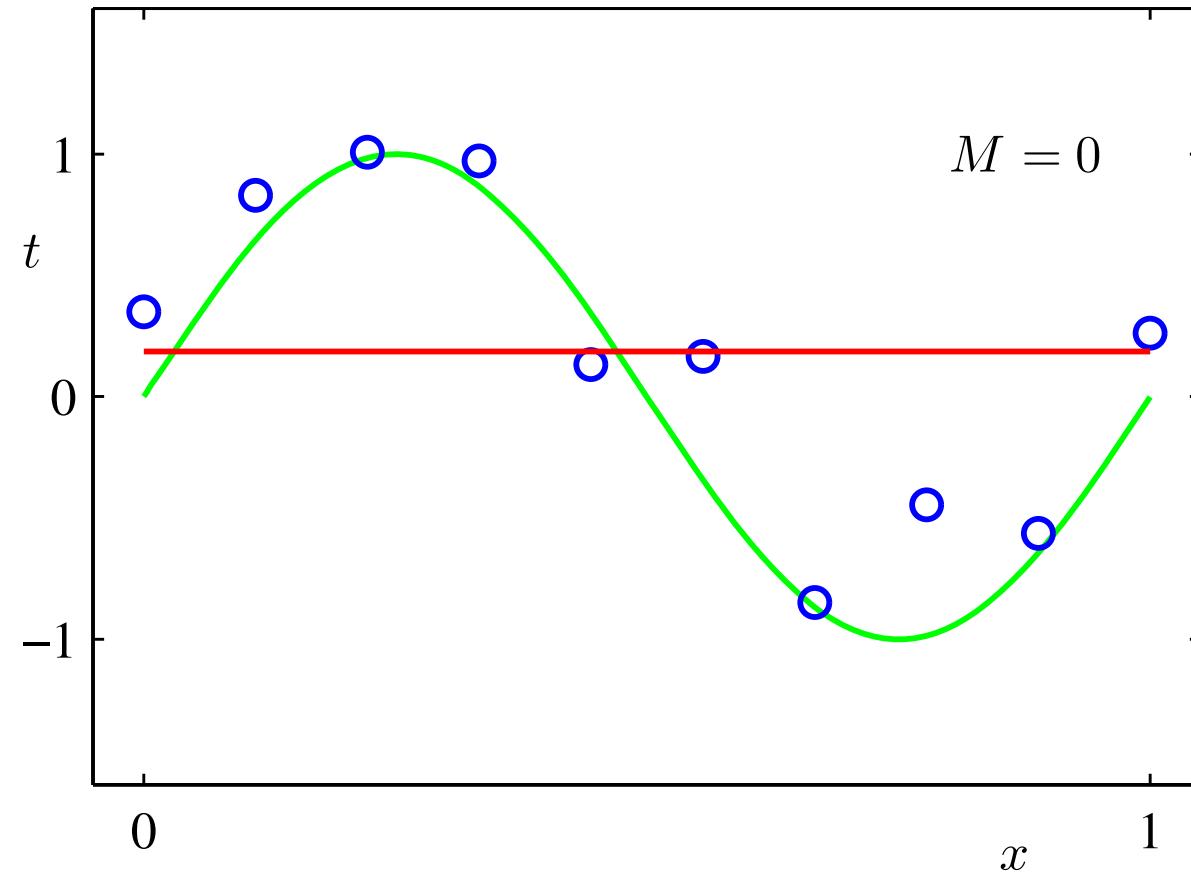
- le « $\frac{1}{2}$ » n'est pas important (mais simplifiera certains calculs)
- Un algorithme d'apprentissage résoudrait ce problème
 - à partir des données, il va retourner w^*



EXEMPLE: RÉGRESSION

Sujets: sous-apprentissage (*underfitting*)

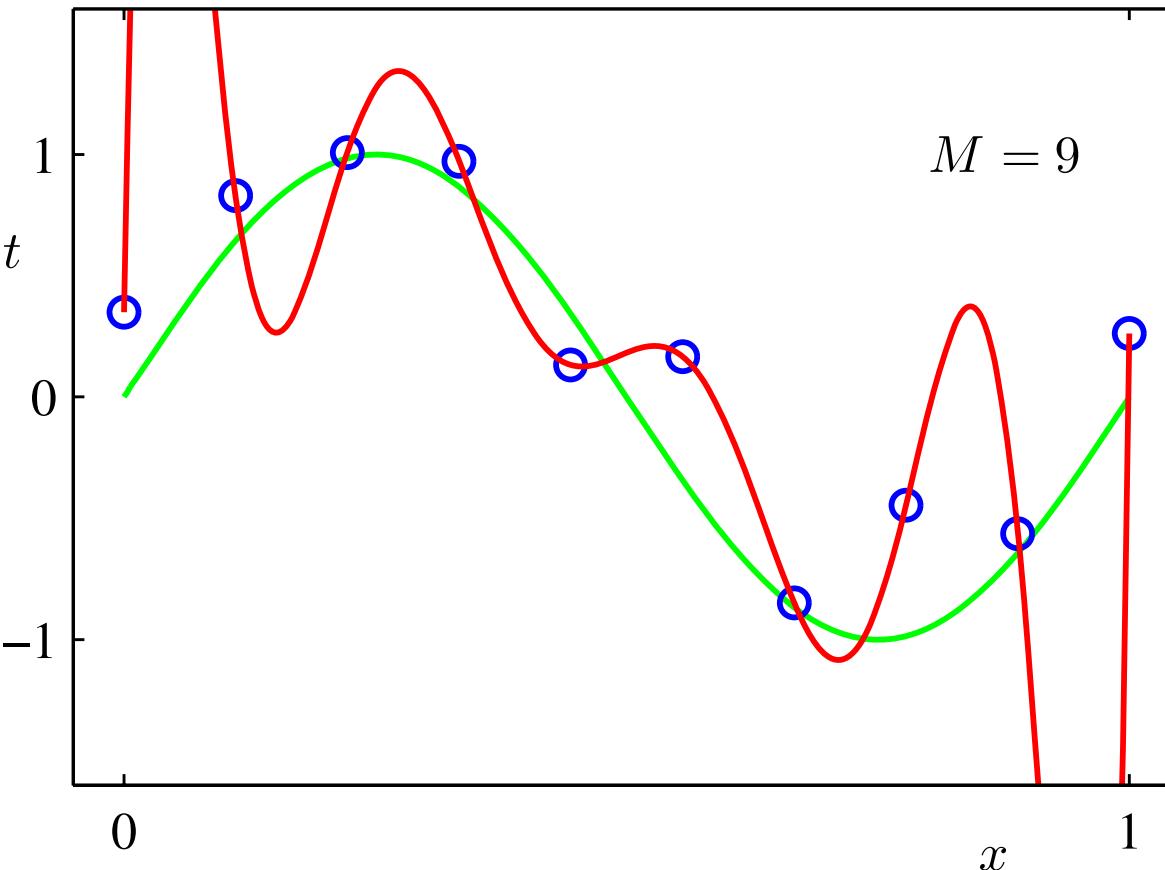
- Comme choisir M ?
 - de trop petites valeurs auront une grande perte sur l'ensemble d'entraînement : situation de **sous-apprentissage**



EXEMPLE: RÉGRESSION

Sujets: sur-apprentissage (*overfitting*)

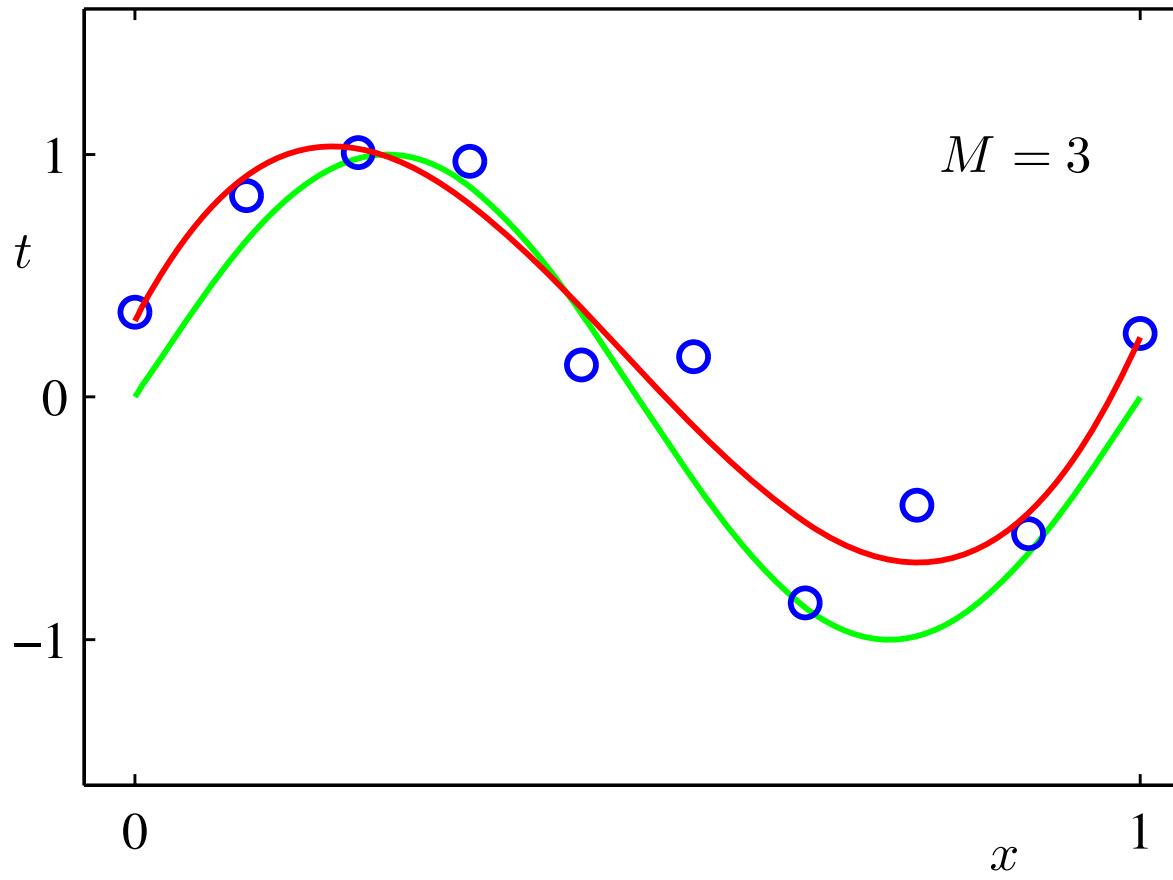
- Comme choisir M ?
 - de trop grandes valeurs vont seulement apprendre l'ensemble d'entraînement «par cœur» : situation de **sur-apprentissage**
 - va apprendre à prédire ce qui n'est pas prévisible à partir de x seulement (p. ex. du bruit)



EXEMPLE: RÉGRESSION

Sujets: sélection de modèle

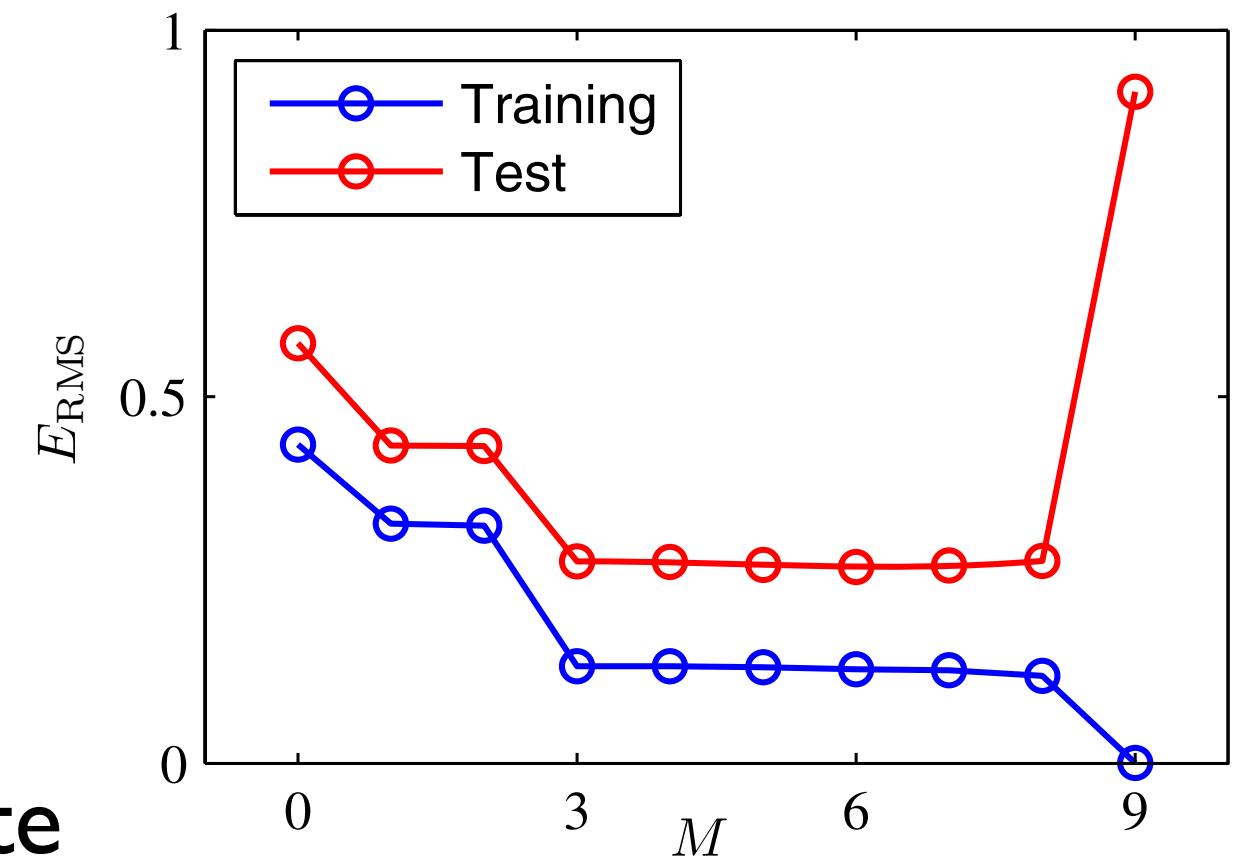
- Comme choisir M ?
 - on voudrait une valeur intermédiaire qui permet de retrouver la tendance générale de la relation entre x et t , sans le bruit
 - c'est ce qui va permettre de bien généraliser à de nouvelles entrées !
 - trouver cette meilleure valeur de M s'appelle de la **sélection de modèle**
 - on va voir plus tard différentes techniques



EXEMPLE: RÉGRESSION

Sujets: capacité d'un modèle, performance

- **Capacité** d'un modèle
 - aptitude d'un modèle à apprendre «par cœur»
 - exemple : plus M est grand, plus le modèle a de capacité
- Plus la capacité est grande, plus la différence entre l'erreur d'entraînement et 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 (*root-mean-square error*)

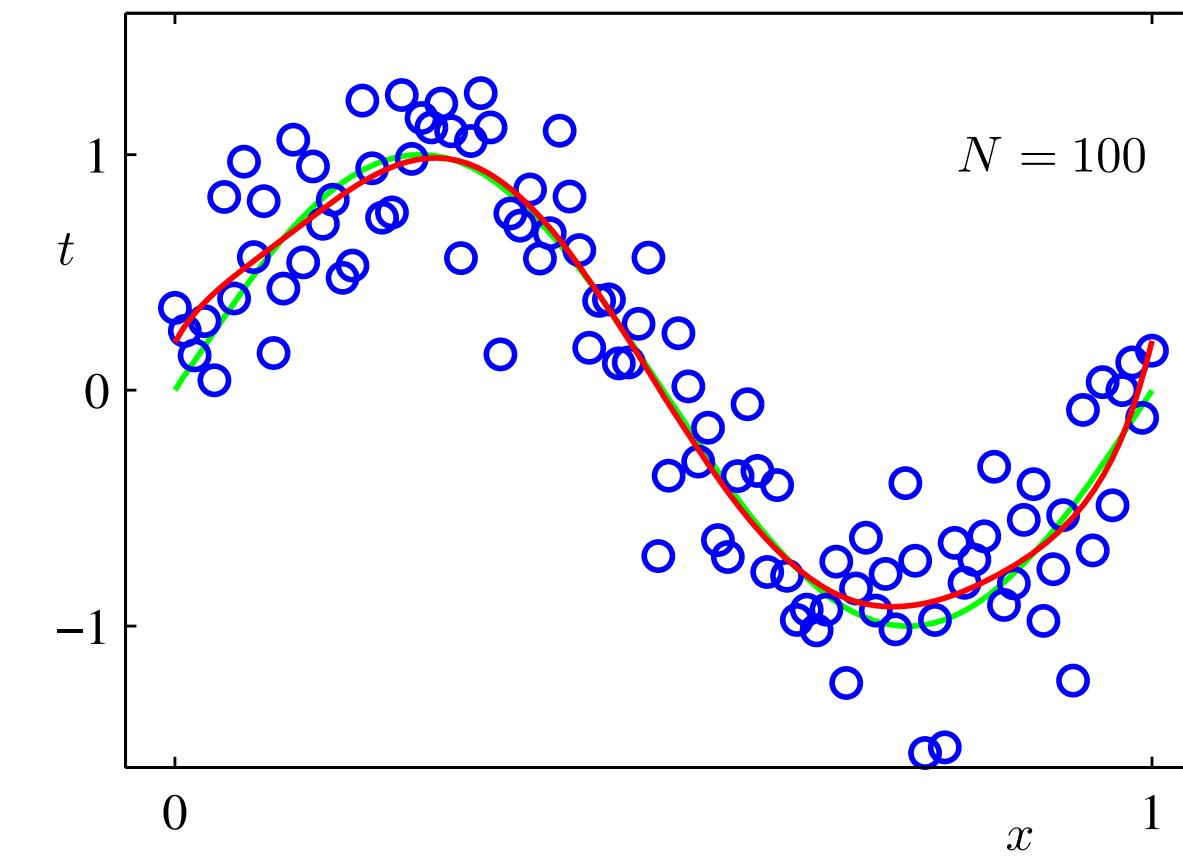
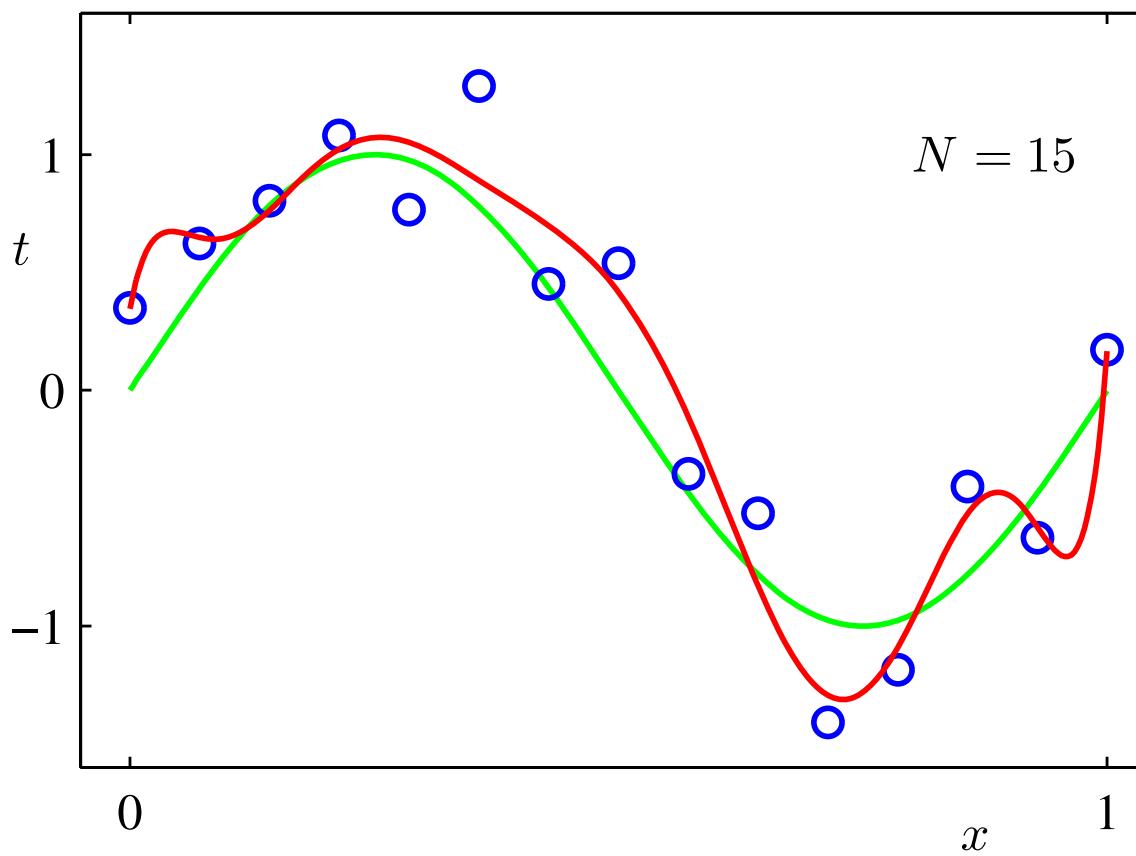


$$E_{\text{RMS}} = \sqrt{2E(\mathbf{w}^*)/N}$$

EXEMPLE: RÉGRESSION

Sujets: généralisation vs. quantité de données

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



Apprentissage automatique

Concepts fondamentaux - régularisation

EXEMPLE: RÉGRESSION

Sujets: régularisation

- Comment utiliser un grand M avec peu de données
 - par exemple, si on connaît le «vrai» M
- **Régularisation** : on réduit la capacité autrement
 - exemple : on pénalise la somme du carré des paramètres (i.e. la norme Euclidienne au carré)

$$\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

contrôle la capacité

- où $\|\mathbf{w}\|^2 \equiv \mathbf{w}^T \mathbf{w} = w_0^2 + w_1^2 + \dots + w_M^2$

EXEMPLE: RÉGRESSION

Sujets: régularisation

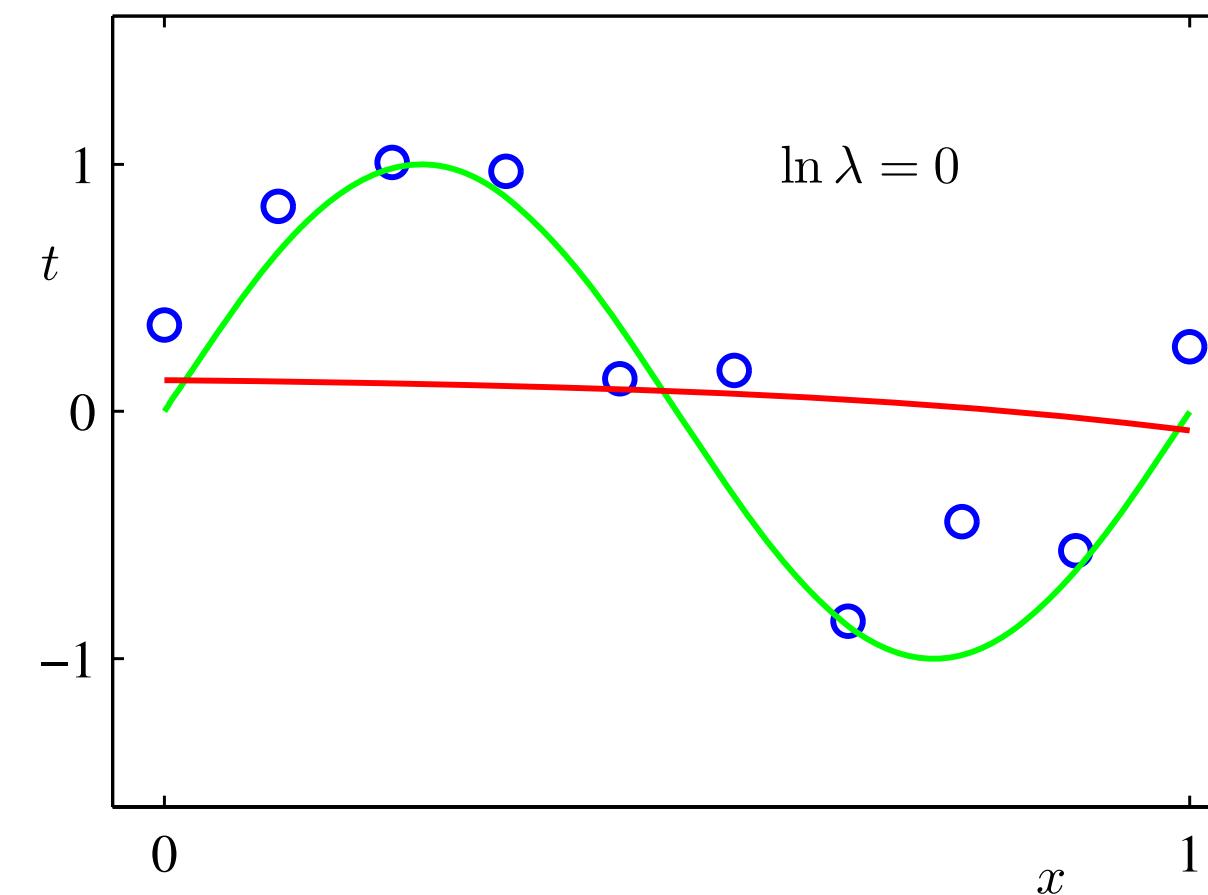
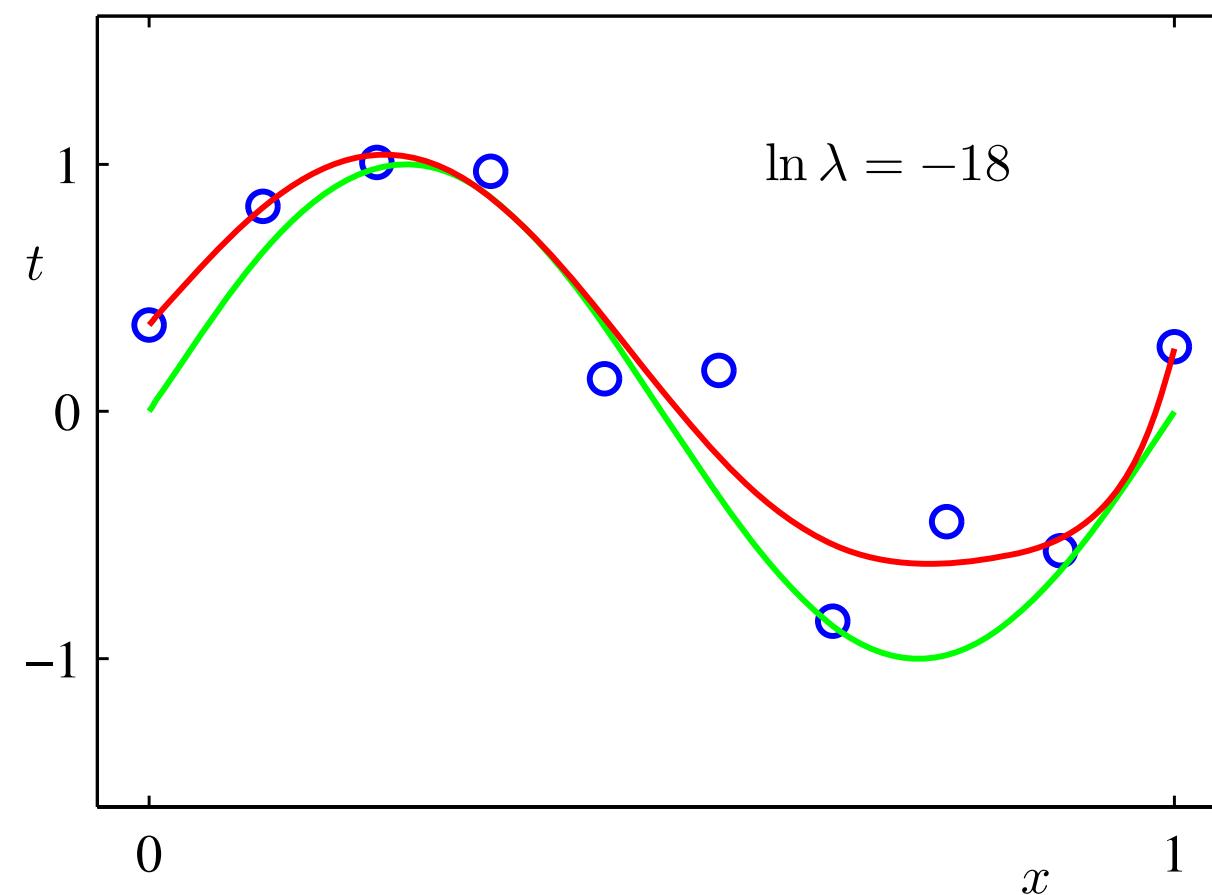
- Valeurs des paramètres w^* pour différents M , sans régularisation

| | $M = 0$ | $M = 1$ | $M = 6$ | $M = 9$ |
|---------|---------|---------|---------|-------------|
| 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^* | | | | 640042.26 |
| w_6^* | | | | -1061800.52 |
| w_7^* | | | | 1042400.18 |
| w_8^* | | | | -557682.99 |
| w_9^* | | | | 125201.43 |

EXEMPLE: RÉGRESSION

Sujets: régularisation

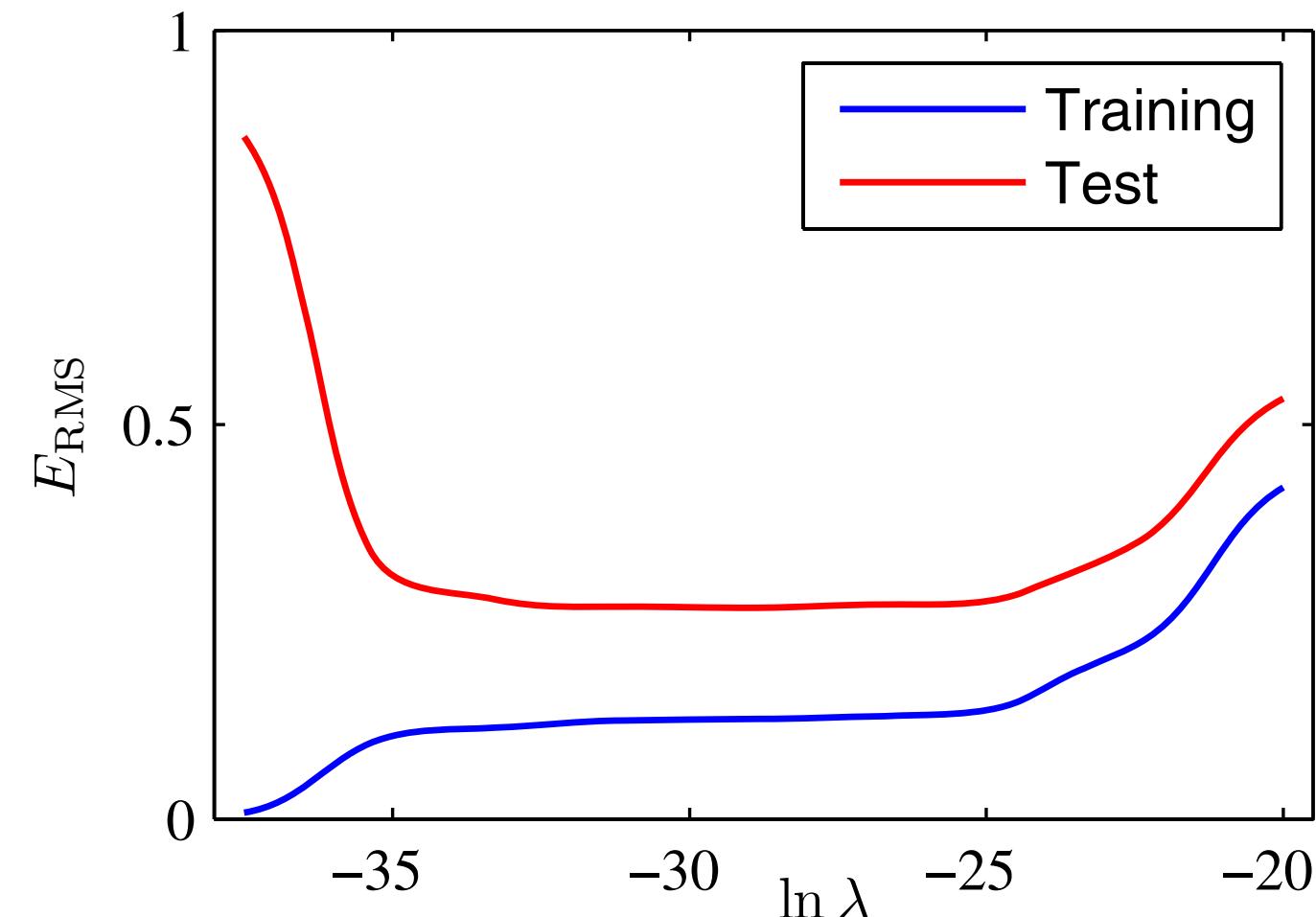
- Plus la régularisation est forte, moins le modèle sera flexible (donc il aura moins de capacité)



EXEMPLE: RÉGRESSION

Sujets: régularisation

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



Apprentissage automatique

Concepts fondamentaux - sélection de modèle

SÉLECTION DE MODÈLE

Sujets: hyper-paramètres

- Soit l'algorithme d'apprentissage qui optimise

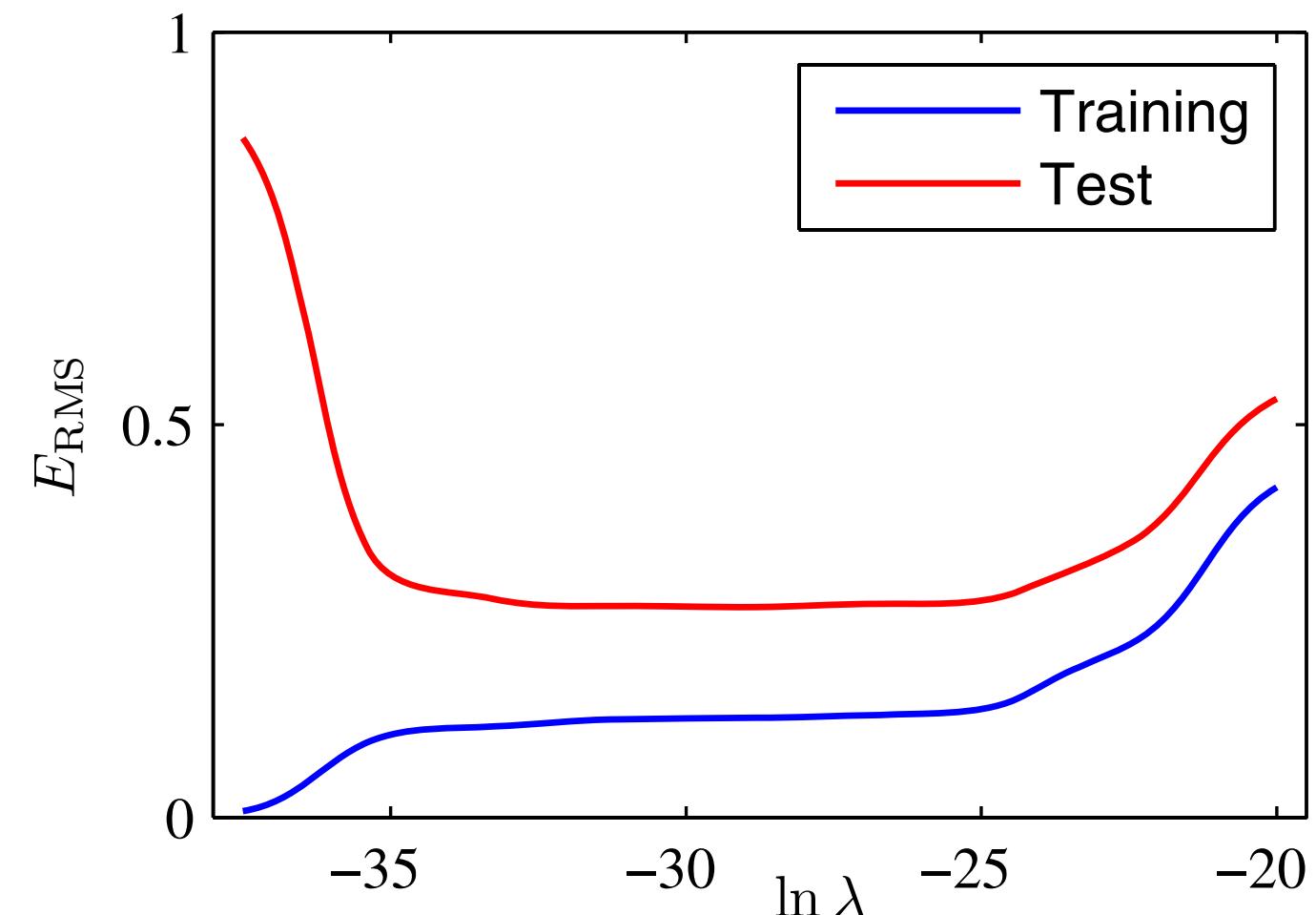
$$\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

- On appelle M et λ des **hyper-paramètres**
 - ils doivent être déterminés avant l'entraînement
- Comment choisir la valeur de ces hyper-paramètres ?
 - on appelle cela de la **sélection de modèle**

SÉLECTION DE MODÈLE

Sujets: hyper-paramètres

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



SÉLECTION DE MODÈLE

Sujets: ensemble de validation

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

SÉLECTION DE MODÈLE

Sujets: sélection de modèle

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

SÉLECTION DE MODÈLE

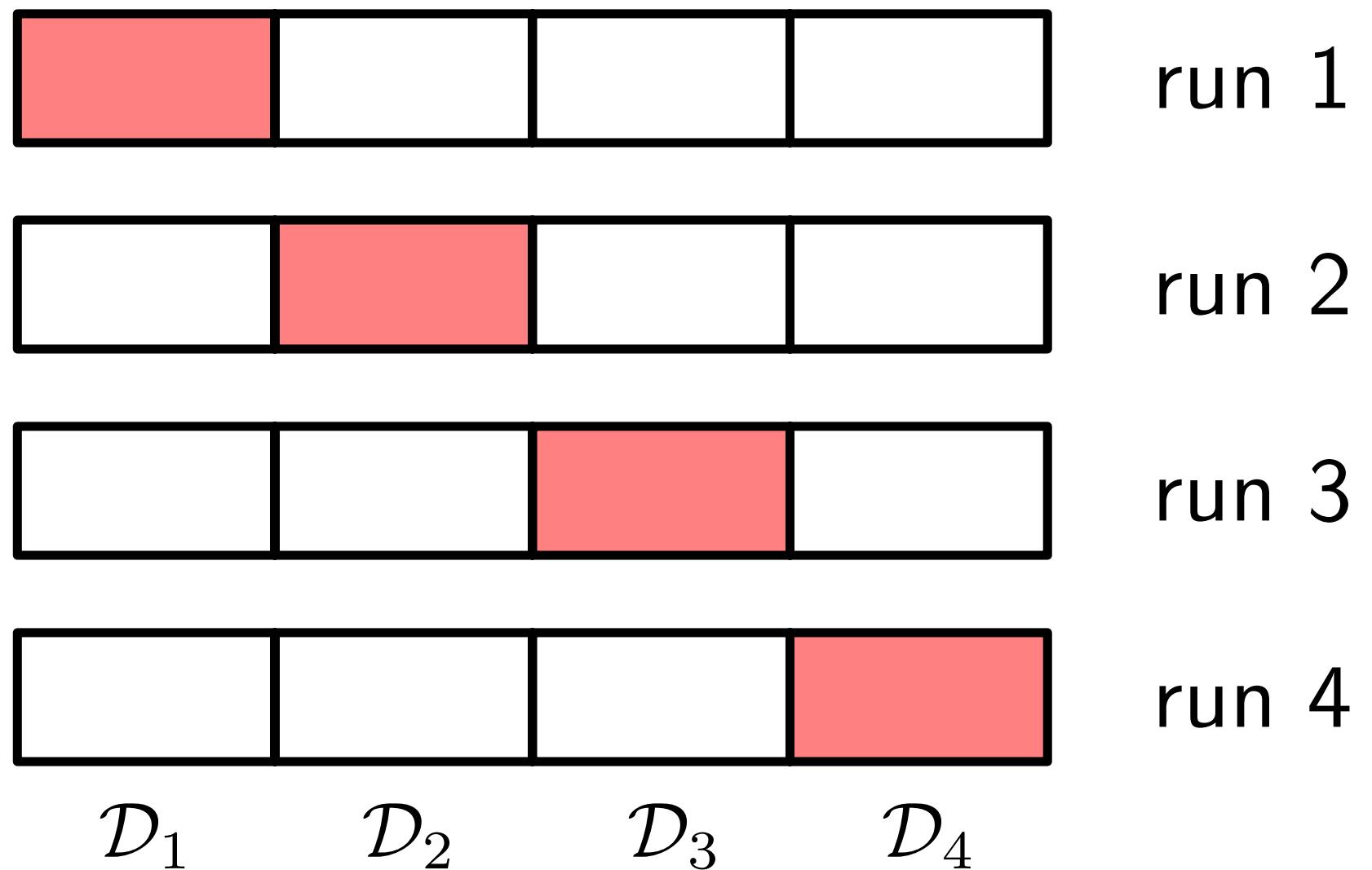
Sujets: *S-fold cross-validation*

- 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 $\mathcal{D}_{\text{valid}}$

SÉLECTION DE MODÈLE

Sujets: S -fold cross-validation

- Exemple : $S = 4$



SÉLECTION DE MODÈLE

Sujets: *S-fold cross-validation, leave-one-out*

- Sélection de modèle avec *S-fold cross-validation*
 - pour $s = 1 \dots S$
 - pour chaque valeur d'hyper-paramètres à comparer
 - ✓ obtenir un modèle entraîné à partir de $\mathcal{D}_{\text{train}} = \mathcal{D} \setminus \mathcal{D}_s$
 - ✓ évaluer la performance du modèle sur $\mathcal{D}_{\text{valid}} = \mathcal{D}_s$
 - retourner la valeur des hyper-paramètres ayant la meilleure performance **moyenne** sur les ensembles $\mathcal{D}_{\text{valid}}$
- Si $S = N$, on parle alors de méthode **leave-one-out**

SÉLECTION DE MODÈLE

Sujets: recherche sur une grille

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

```
>>> M = [1,2]
>>> lba = [0,1e-6,1e-3]
>>> hypers = [ [ (m,l) for m in M ] for l in lba ]
>>> print hypers
[(1, 0), (2, 0)], [(1, 1e-06), (2, 1e-06)], [(1, 0.001),
(2, 0.001)]]
```

Apprentissage automatique

Concepts fondamentaux - malédiction de la dimensionnalité

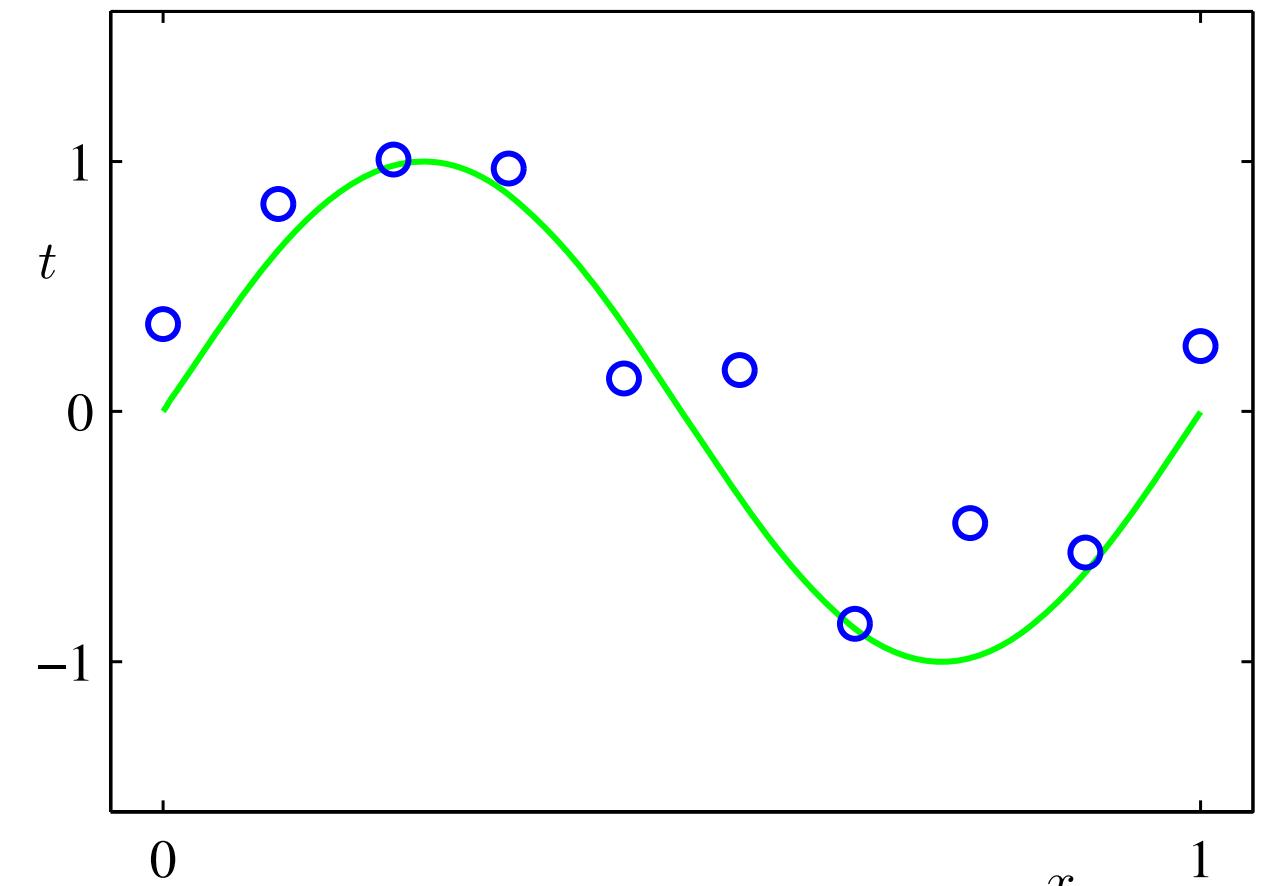
EXEMPLE: RÉGRESSION

Sujets: régression polynomiale

- Jusqu'à maintenant, on a considéré un problème où les entrées vivent sur 1 dimension

$$\begin{aligned}y(x, \mathbf{w}) &= w_0 + w_1 x + w_2 x^2 + \dots + w_M x^M \\&= \sum_{j=0}^M w_j x^j\end{aligned}$$

- Qu'arrive-t-il si on augmente le nombre de dimensions ?



MALÉDICTION DE LA DIMENSIONNALITÉ

Sujets: nombre de paramètres

- Notre modèle de régression aura plus de paramètres

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^D w_i x_i + \sum_{i=1}^D \sum_{j=1}^D w_{ij} x_i x_j + \sum_{i=1}^D \sum_{j=1}^D \sum_{k=1}^D w_{ijk} x_i x_j x_k$$

- pour $M = 3$, on a maintenant $1 + D + D^2 + D^3$ paramètres
- De façon générale, augmente selon $O(D^M)$!
- pour $D=100, M=3$, on a déjà plus d'un million de paramètres

MALÉDICTION DE LA DIMENSIONNALITÉ

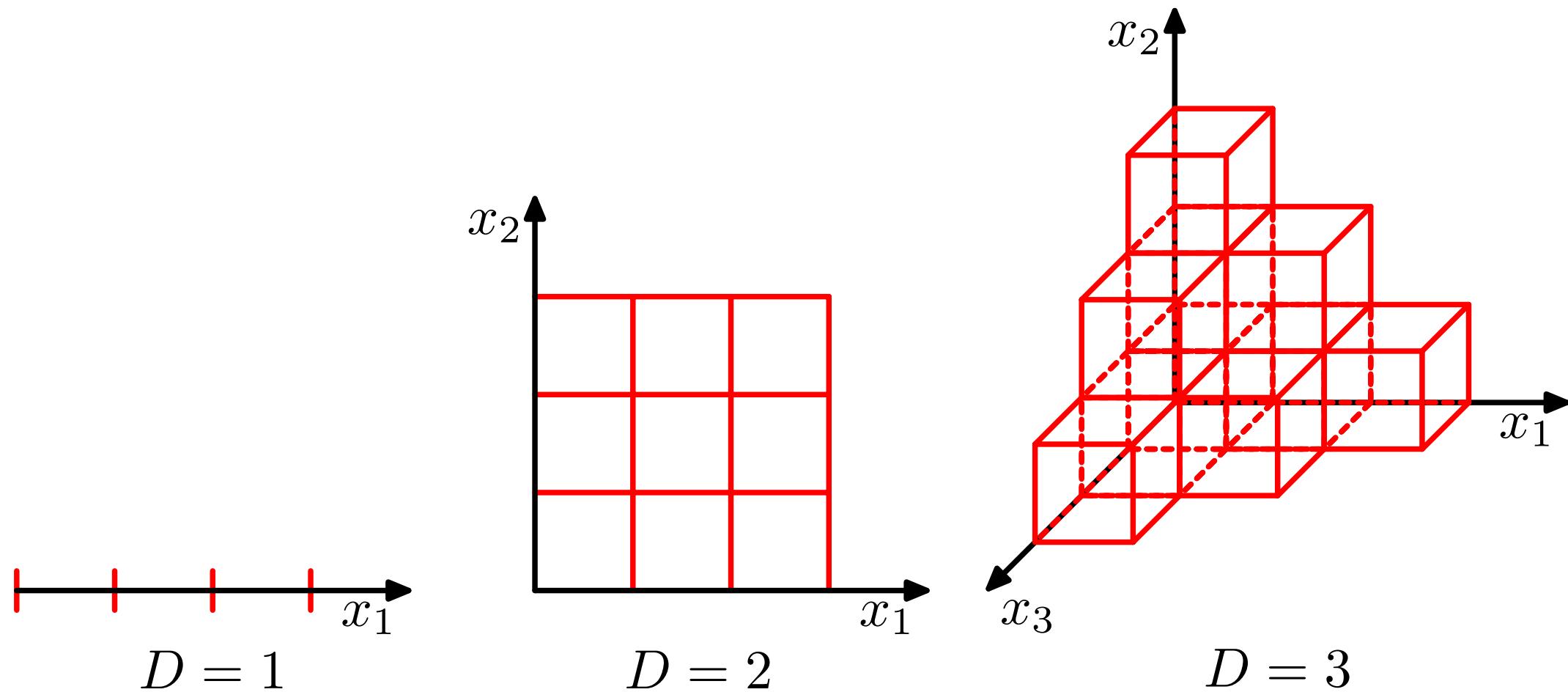
Sujets: nombre d'exemples pour bien généraliser

- Comment garantir qu'on généralise bien à une entrée x ?
 - avoir des entrées similaires dans l'ensemble d'entraînement
- Imaginons qu'on divise également l'espace d'entrée en région (hypercubes)
 - on aimerait avoir un exemple d'entraînement dans chaque région
 - qu'arrive-t-il au nombre de régions, lorsque D augmente ?

MALÉDICTION DE LA DIMENSIONNALITÉ

Sujets: nombre d'exemples pour bien généraliser

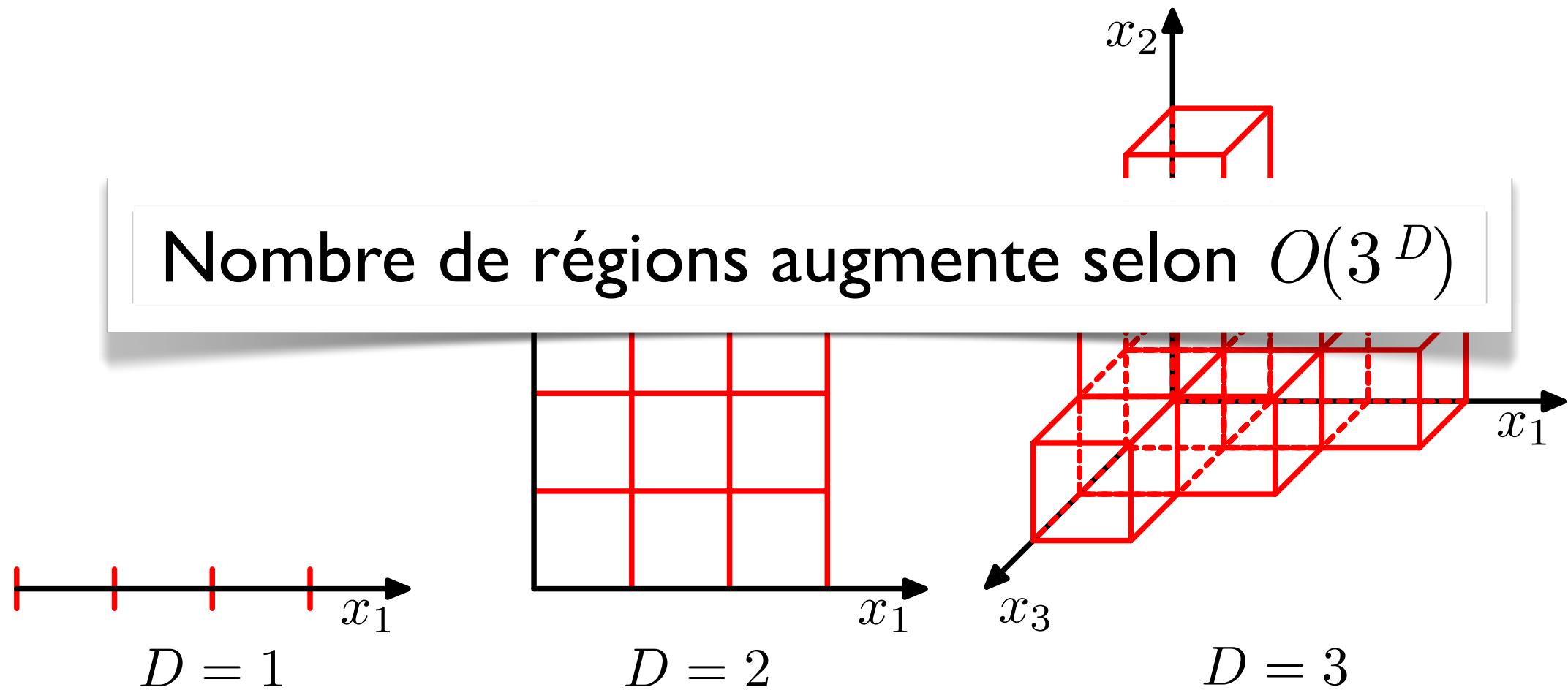
- Comment garantir qu'on généralise bien à une entrée x ?
 - avoir des entrées similaires dans l'ensemble d'entraînement



MALÉDICTION DE LA DIMENSIONNALITÉ

Sujets: nombre d'exemples pour bien généraliser

- Comment garantir qu'on généralise bien à une entrée x ?
 - avoir des entrées similaires dans l'ensemble d'entraînement



MALÉDICTION DE LA DIMENSIONNALITÉ

Sujets: malédiction de la dimensionnalité

- La difficulté à bien généraliser peut donc potentiellement augmenter **exponentiellement** avec la dimensionnalité D des entrées
- Cette observation est appelée la **malédiction de la dimensionnalité**
- Nécessite le design de modèles / algorithmes appropriés pour chaque problème
 - on cherche des modèles / algorithmes qui vont bien exploiter les données à notre disposition

Apprentissage automatique

Concepts fondamentaux - résumé

APPRENTISSAGE AUTOMATIQUE

Sujets: algorithme d'apprentissage

- Un algorithme d'apprentissage
 - entraîne un **modèle** à partir d'un **ensemble d'entraînement**, pouvant faire des prédictions sur de nouvelles données
 - a des **hyper-paramètres** qui contrôlent la **capacité** du modèle entraîné, choisis à l'aide d'une procédure de **sélection de modèle**
 - mesure sa performance de **généralisation** sur un **ensemble de test** (selon une fonction d'erreur qui peut être différente de la **perte d'entraînement**)

APPRENTISSAGE AUTOMATIQUE

Sujets: algorithme d'apprentissage

- Un algorithme d'apprentissage
 - aura une meilleure performance de généralisation si la quantité de données d'entraînement augmente
 - peut souffrir de **sous-apprentissage** (pas assez de capacité) ou **sur-apprentissage** (trop de capacité)
 - sera plus ou moins victime de la malédiction de la dimensionnalité