Régression polynomiale

Introduction à l'apprentissage automatique
Master Sciences Cognitives
Aix Marseille Université

Alexis Nasr

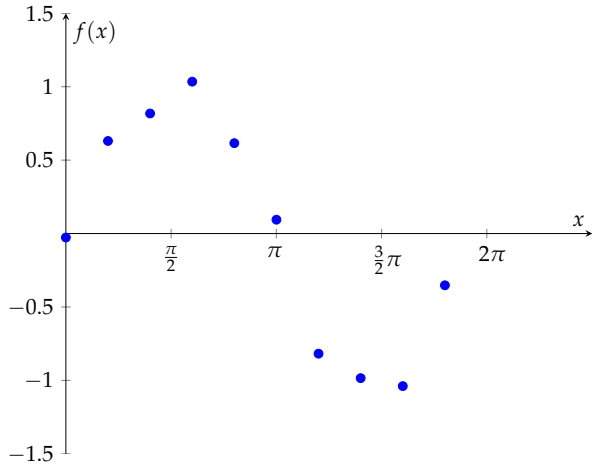
Motivations

- Introduire un problème classique d'apprentissage : la régression polynomiale, illustré sur un exemple simple.
- Introduire à travers ce problème de nombreux aspects de l'apprentissage artificiel.

Objectifs

- Comprendre le rôle de la **fonction d'erreur**.
- Introduire la notion de **minimisation** de la fonction d'erreur comme moyen de déterminer les paramètres d'un modèle à partir de données.
- Comprendre les rôles différents que jouent les **variables** et les **paramètres** dans une fonction de régression.
- Comprendre le problème du **sur-apprentissage**
- Introduire les notions de données d'**apprentissage** et de données de **test**.

Régression

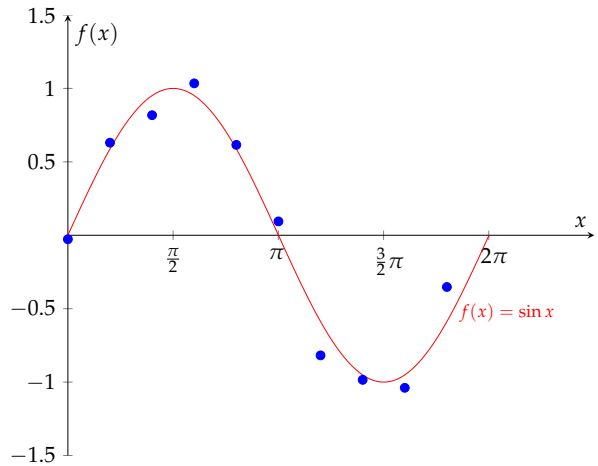


- On dispose de 10 points représentant la variation d'une quantité y en fonction de x.

$$\mathcal{D} = \{(x_i, y_i)\}_{i=0}^{10}$$

- On cherche à construire une fonction $f(x)$ telle que :
  - les valeurs $\hat{y}_i = f(x_i)$ soient les plus proches possibles de valeurs de y_i
  - f fasse des prédictions raisonnables sur des données non observées.
- Exemple emprunté à Christopher Bishop

Génération des données



- Les données ont été générées à partir de la fonction *sinus* en ajoutant du bruit (une petite valeur aléatoire).
- Idée : les données possèdent une certaine **régularité** mais sont entachées d'un **bruit** aléatoire.
- Situation **artificielle** : on connaît la fonction qui a permis de générer les données, ce qui n'est généralement pas le cas.

Approximation polynomiale

- On décide d'utiliser une fonction polynomiale :

$$\begin{aligned} f(x) &= w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \ldots + w_M x^M \\ &= \sum_{i=0}^{M} w_i x^i \end{aligned}$$

- M est appelé l'ordre du polynome.
- On représente les coefficients du polynome sous la forme du vecteur $\mathbf{w} = [w_0, w_1, \ldots, w_M]^T$.
- On ne connaît pas :
  - l'ordre du polynome
  - les valeurs des coefficients w_i
- On traite séparément et de manière différente ces deux inconnues.

Deux manières de voir un polynome

$$f(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \ldots + w_M x^M$$

- manière classique, x est inconnue et les coefficients $w_0 \ldots w_M$ sont connus.

$$f(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \ldots + w_M x^M$$

- autre manière de voir, x est connue et les coefficients sont inconnus.

$$f(\mathbf{w}) = w_0 + x w_1 + x^2 w_2 + x^3 w_3 + \ldots + x^M w_M$$

Deux manières de voir un polynome

$$f(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \ldots + w_M x^M$$

$$f(\mathbf{w}) = w_0 + x w_1 + x^2 w_2 + x^3 w_3 + \ldots + x^M w_M$$

- Les deux fonctions sont très différentes :
  - $f(x)$ est une **fonction polynomiale** d'une variable : x
  - $f(\mathbf{w})$ est une **fonction linéaire** de $M+1$ variables : $w_0 \ldots w_M$
- On note $f(x;\theta)$ pour distinguer les variables (ici x) et les paramètres (ici θ).
- Les deux fonctions s'écrivent :

$$f(x;\mathbf{w}) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \ldots + w_M x^M$$

$$f(\mathbf{w};x) = w_0 + x w_1 + x^2 w_2 + x^3 w_3 + \ldots + x^M w_M$$

Fonction d'erreur

- Etant donné une fonction particulière f, on cherche à calculer dans quelle mesure elle **décrit** bien les données observées.
- Pour cela, on définit une **fonction d'erreur** qui mesure la **différence** entre les prédictions et les données observées.
- Une fonction communément utilisée est la somme des carrés des différences entre les valeurs prédites et les valeurs réelles :

$$
\begin{aligned}
E(\mathbf{w}; \mathcal{D}) &= \frac{1}{2} \sum_{i=1}^{10} (f(\mathbf{w}; x_i) - y_i)^2 \\
&= \frac{1}{2} \sum_{i=1}^{10} (\sum_{j=0}^{M} w_j x_i^j - y_i)^2
\end{aligned}
$$

Minimisation de la fonction d'erreur

$$E(\mathbf{w};\mathcal{D}) = \frac{1}{2}\sum_{i=1}^{10}(\sum_{j=0}^{M} w_j x_i^j - y_i)^2$$

- Plus la valeur de $E(\mathbf{w};\mathcal{D})$ est petite, meilleure est la qualité du modèle.
- On aimerait trouver la valeur des paramètres qui **minimise** la fonction d'erreur :

$$\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathcal{W}} E(\mathbf{w};\mathcal{D})$$

- $\mathcal{W}$ est l'ensemble de tous les jeux de paramètres possibles.

Les notations min, max, arg min et arg max

- Une fonction $f(x)$ admet un **maximum** m en un point a si $m = f(a)$ et si, quel que soit x', $f(x')$ est inférieur ou égal à $f(a)$.
- On note :

$$m = \max_x f(x)$$

- On appelle **argmax** (argument du maximum) de la fonction $f(x)$, la valeur de x qui permet d'atteindre le maximum.
- On note :

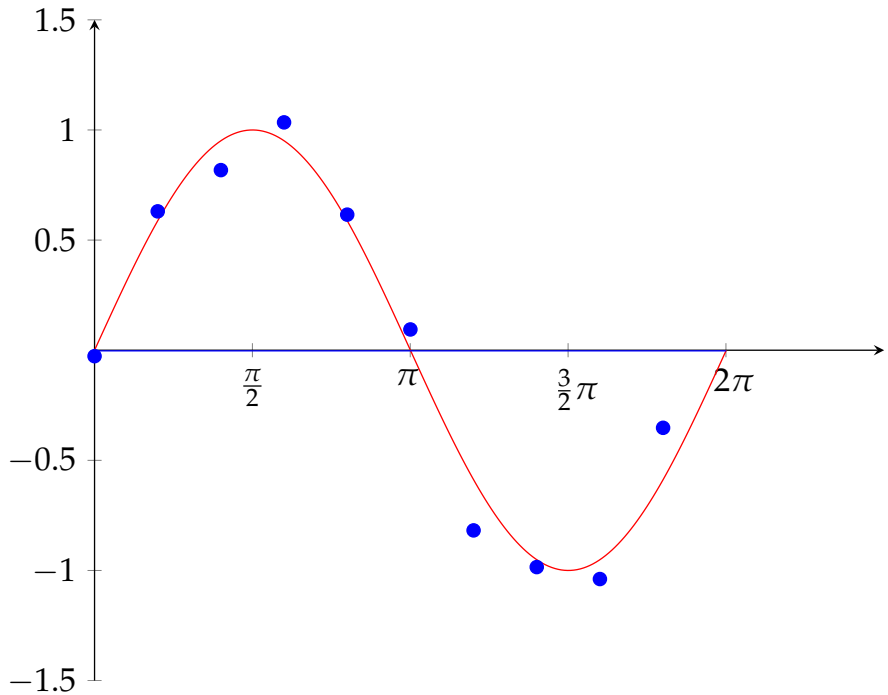
$$a = \arg\max_x f(x)$$

- On définit de manière identique min et arg min.

Optimisation

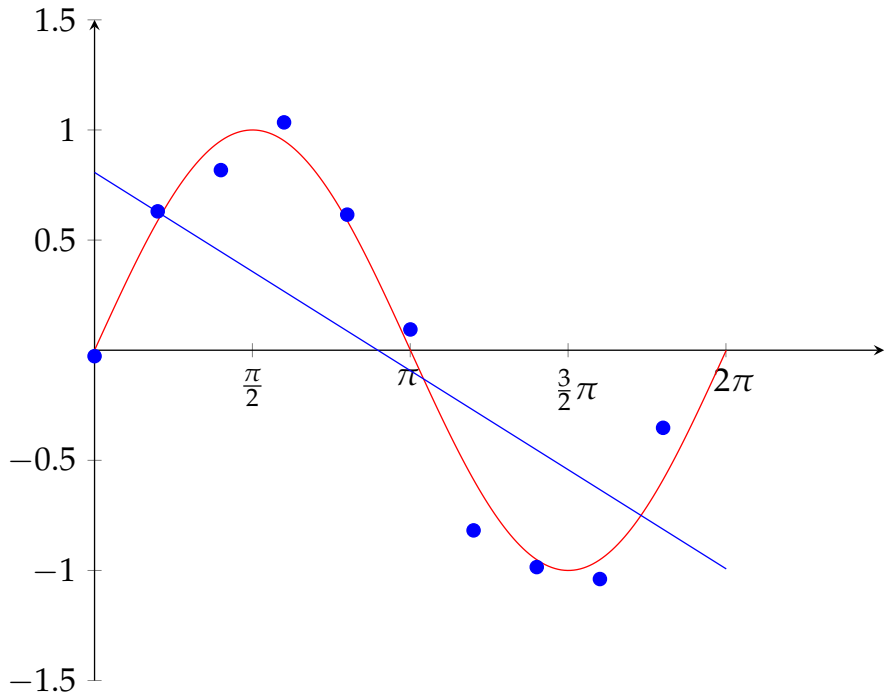
- L'optimisation est une branche des mathématiques cherchant à résoudre les problèmes qui consistent à **minimiser** ou **maximiser** une fonction sur un ensemble.
- L'optimisation joue un rôle important en apprentissage automatique en particulier pour minimiser la fonction d'erreur.
- On distingue les méthodes permettant de résoudre **analytiquement** ou **numériquement** le problème d'optimisation.
  - solution analytique : une ou plusieurs expressions mathématiques qui expriment la solution du problème.
  - solution numérique : approximation de la solution à l'aide de calculs, généralement effecutés par un ordinateur.
- Dans notre cas, il est possible (et assez facile) de trouver une solution analytique au problème.
- On verra comment faire à l'issue du cours sur le gradient.

$M = 0$



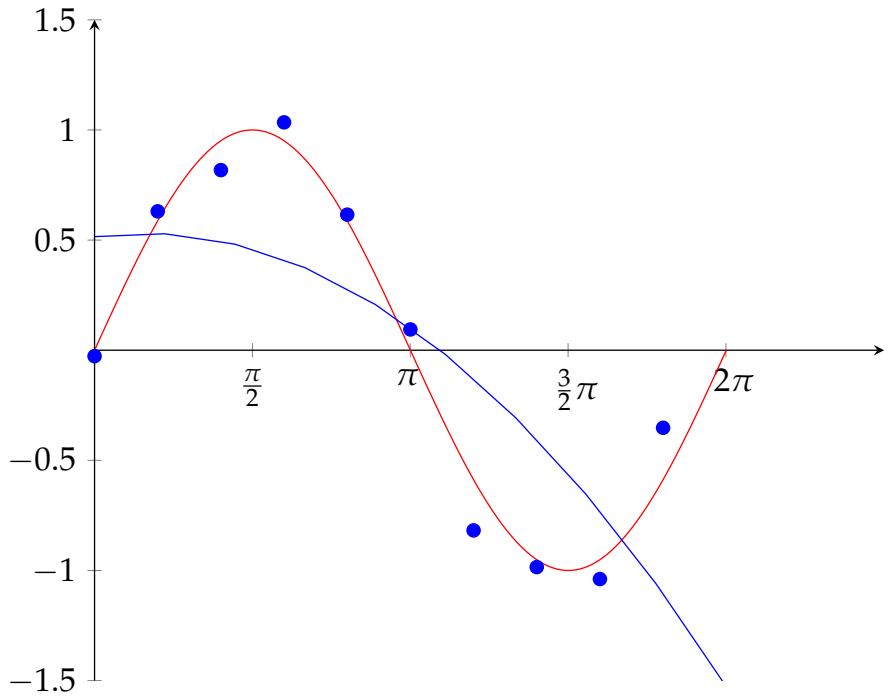
- $f(x) = -0.0028$
- $E(\mathbf{w}^*; \mathcal{D}) = 2.68$

$M = 1$



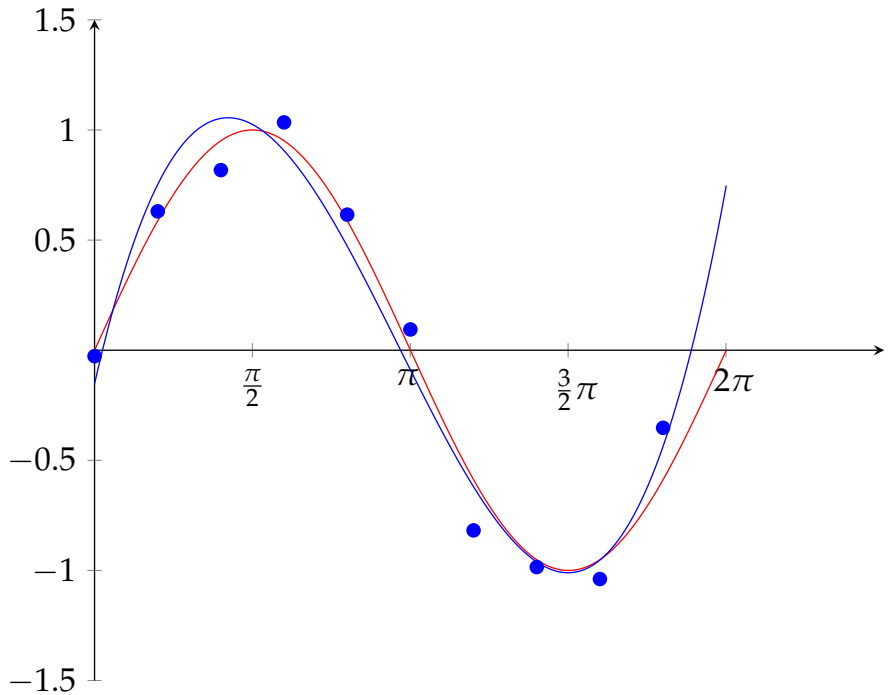
- $f(x) = 0.8072 - 0.2865x$
- $E(\mathbf{w}^*; \mathcal{D}) = 1.34$

$M = 2$



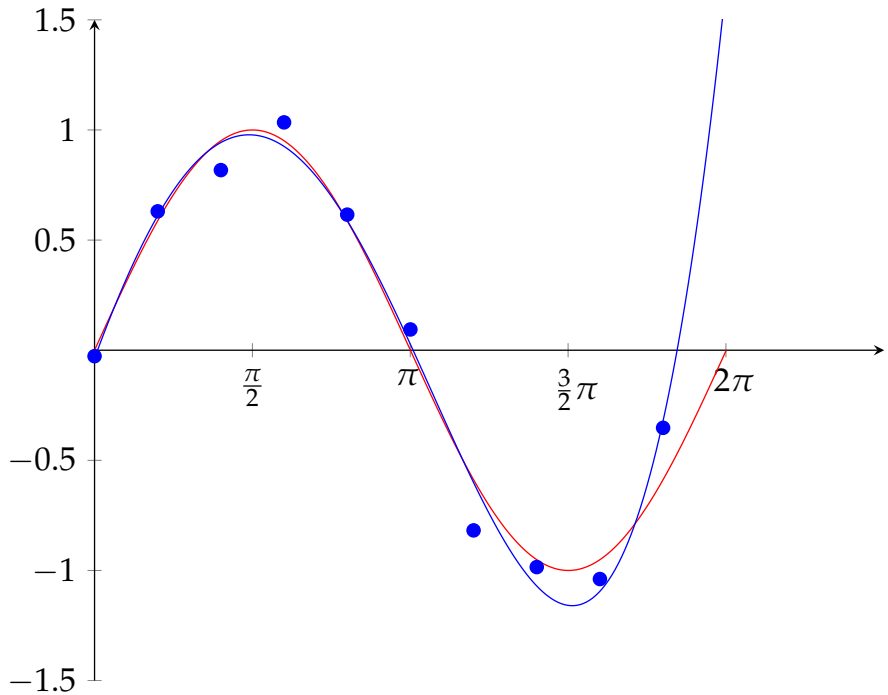
- $f(x) = 0.5156 + 0.0615x - 0.0615x^2$
- $E(\mathbf{w}^*; \mathcal{D}) = 1.19$

$M = 3$



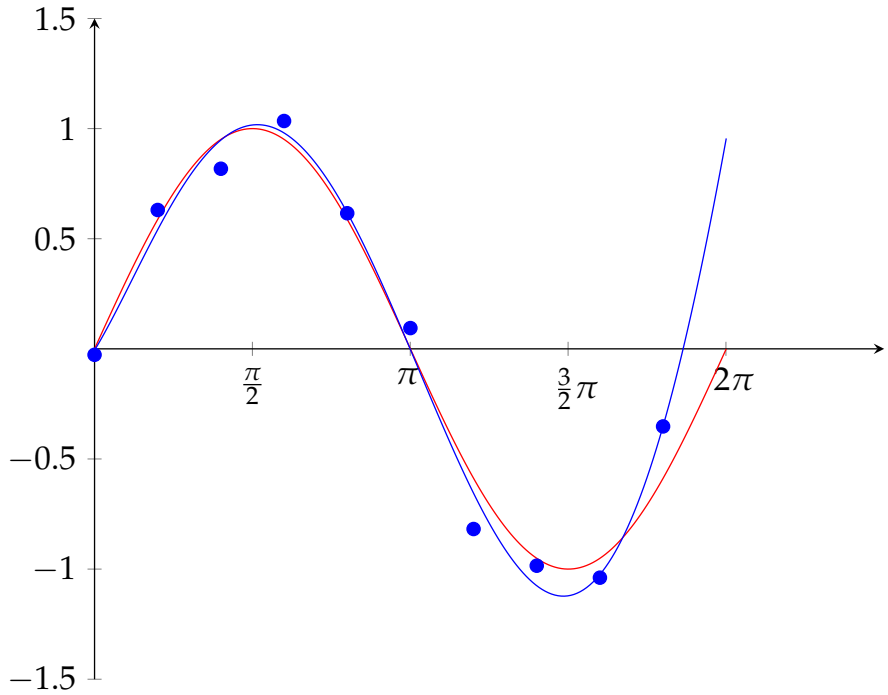
- $f(x) = -0.1523 + 2.0065x - 0.9680x^2 + 0.1068x^3$
- $E(\mathbf{w}^*; \mathcal{D}) = 0.105$

$M = 4$



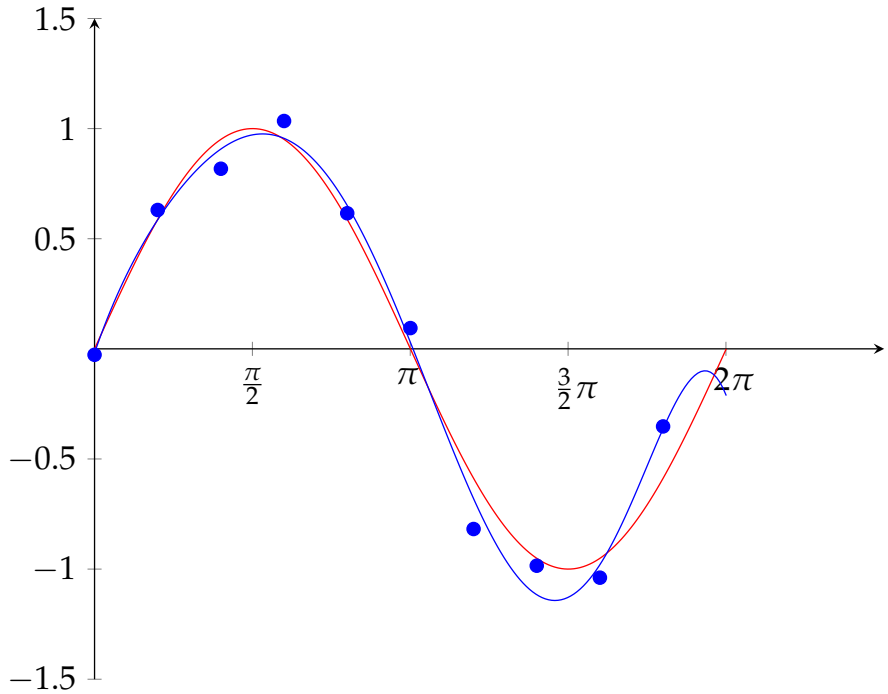
- $f(x) = -0.0364 + 1.2376x - 0.2814x^2 - 0.0878x^3 + 0.0172x^4$
- $E(\mathbf{w}^*; \mathcal{D}) = 0.046$

$M = 5$



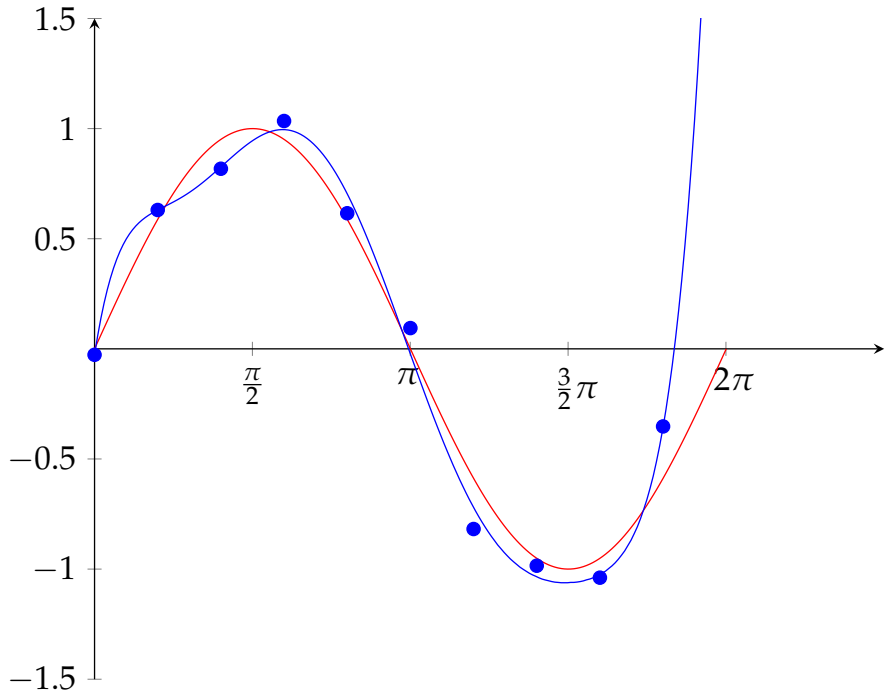
- $f(x) = -0.0065 + 0.7557x + 0.4289x^2 - 0.4429x^3 + 0.0891x^4 - 0.0050x^5$
- $E(\mathbf{w}^*; \mathcal{D}) = 0.036$

$M = 6$



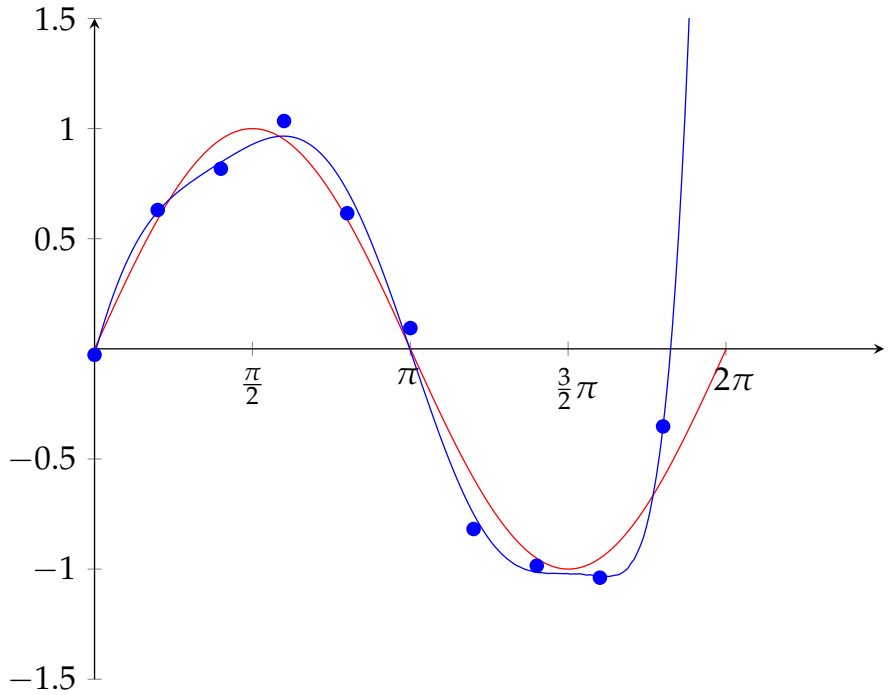
- $f(x) = -0.0187 + 1.3337x - 0.8319x^2 + 0.5167x^3 - 0.2387x^4 + 0.0464x^5 - 0.003x^6$
- $E(\mathbf{w}^*; \mathcal{D}) = 0.003$

$M = 7$



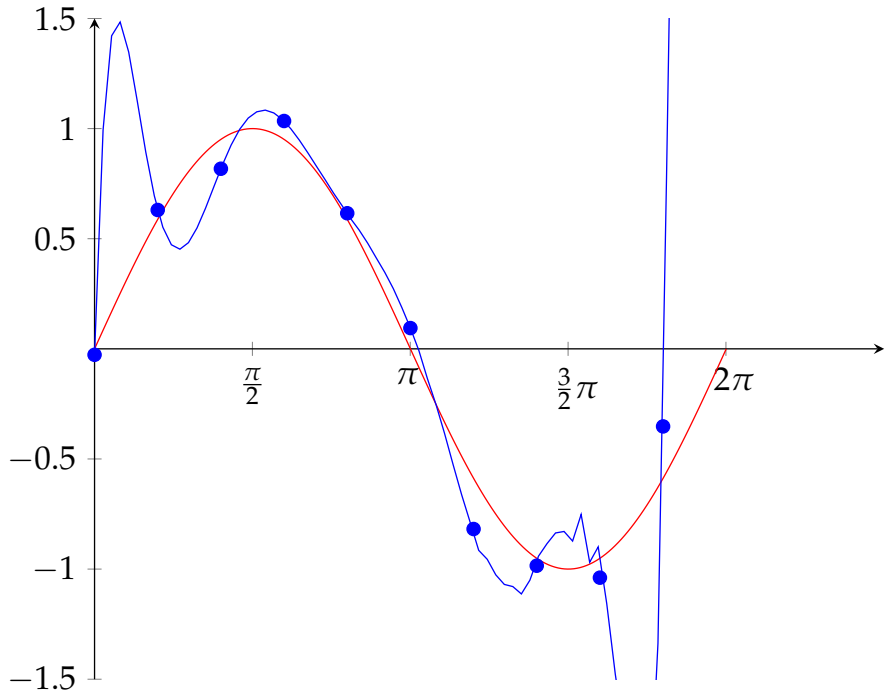
- $f(x) = -0.03 + 2.96x - 5.57x^2 + 5.49x^3 - 2.72x^4 + 0.69x^5 - 0.08x^6 + 0.004x^7$
- $E(\mathbf{w}^*; \mathcal{D}) = 0.0178$

$M = 8$



- $f(x) = -0.02 + 1.62x - 0.80x^2 - 0.87x^3 + 1.52x^4 - 0.87x^5 + 0.23x^6 - 0.03x^7 + 0.001x^8$
- $E(\mathbf{w}^*; \mathcal{D}) = 0.016$

$M = 9$



- $f(x) = -0.02 + 16.66x - 62.23x^2 + 96.98x^3 - 79.77x^4 + 38.29x^5 - 11.10x^6 + 1.91x^7 - 0.18x^8 + 0.007x^9$
- $E(\mathbf{w}^*; \mathcal{D}) = 1.52 \times 10^{-7}$

Que se passe-t-il ?

- Au fur et à mesure que l'on augmente l'ordre du polynome, on produit des solutions qui *collent* de mieux en mieux aux données.
- Pour $M = 9$ la fonction passe exactement par les 10 points.
- Du point de vue de la fonction d'erreur, la solution trouvée est parfaite.
- Mais la fonction est une mauvaise approximation de la fonction *sinus* sous-jacente.
- Cette situation est connue sous le nom de **sur-apprentissage**.
- La fonction f **généralise** très mal à partir des données.

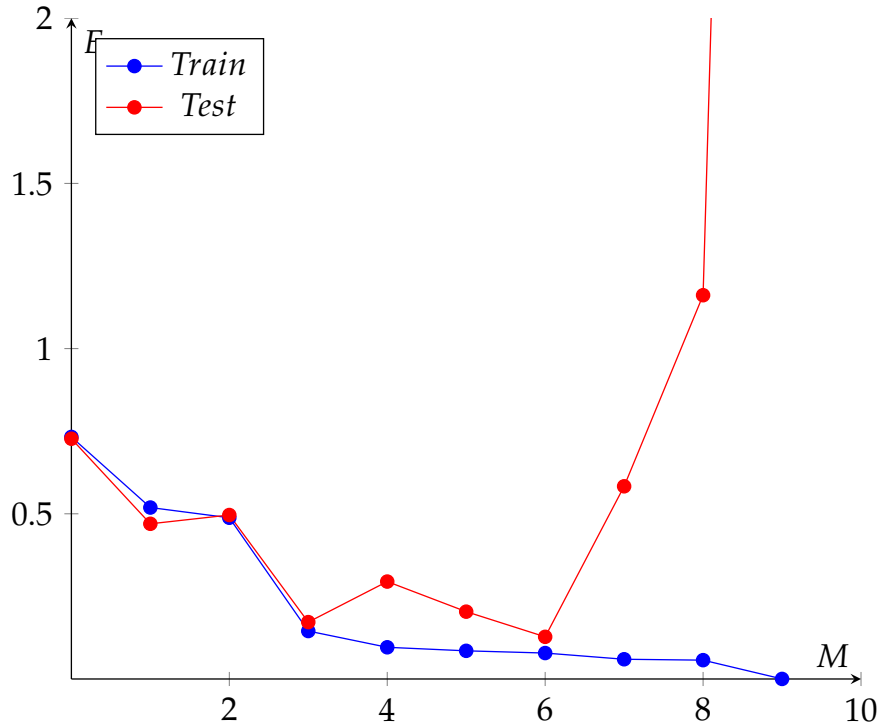
Généralisation

- Afin de connaître les performances en généralisation, on génère 100 nouveaux points.
- Ces données sont appelées **données de test**, elles ne sont pas utilisées pour déterminer $\mathbf{w}^*$
- Pour chaque valeur de M, on calcule l'erreur $E(\mathbf{w}^*; \mathcal{D})$ pour les données d'apprentissage et pour les données de test.
- On utilise généralement la racine de la moyenne des carrés (Root Mean Square) :

$$E_{RMS}(\mathbf{w}^*; \mathcal{D}) = \sqrt{\frac{2E(\mathbf{w}^*; \mathcal{D})}{N}}$$

- Cela permet de comparer l'erreur sur des jeux de données de tailles différentes.

Erreur sur les données d'apprentissage et sur les données de test



- pour $0 \leq M \leq 6$ l'erreur sur les données de test et sur les données d'apprentissage diminue.
- pour $M > 6$ l'erreur continue de diminuer sur les données d'apprentissage, mais augmente sur les données de test.

Régularisation

- On peut remarquer que les coefficients du polynome pour $M = 9$ ont tendance à être élevés :

  $f(x) =$

  $-0.02 + 16.66x - 62.23x^2 + 96.98x^3 - 79.77x^4 + 38.29x^5 - 11.10x^6 + 1.91x^7 - 0.18x^8 + 0.007x^9$
- Ceci s'explique par le fait que la fonctions doit prendre des virages serrés pour arriver à passer par tous les points.
- Pour effectuer des virages serrés, elle doit beaucoup croître à certains moments, et beaucoup décroître à d'autres.
- Il est possible de contraindre les coefficients à prendre des valeurs faibles, en introduisant un terme de **régularisation** dans la fonction d'erreur.

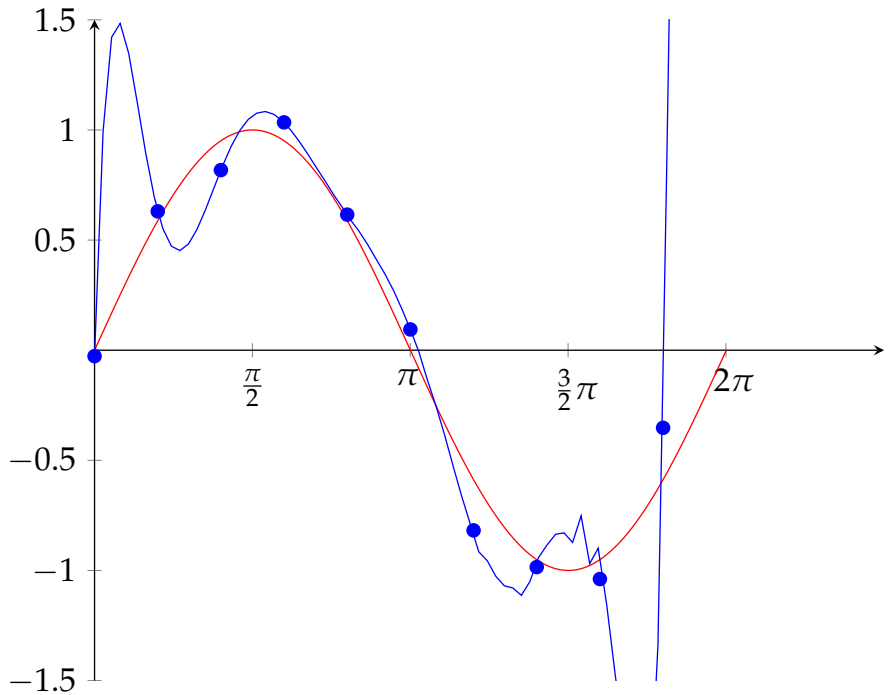
$$\tilde{E}(\mathbf{w}; \mathcal{D}) = \frac{1}{2} \sum_{i=1}^{10} (f(\mathbf{w}; x_i) - y_i)^2 + \frac{\lambda}{2} ||\mathbf{w}||^2$$

Régularisation

$$\tilde{E}(\mathbf{w};\mathcal{D}) = \frac{1}{2}\sum_{i=1}^{10}(f(\mathbf{w};x_i) - y_i)^2 + \frac{\lambda}{2}||\mathbf{w}||^2$$

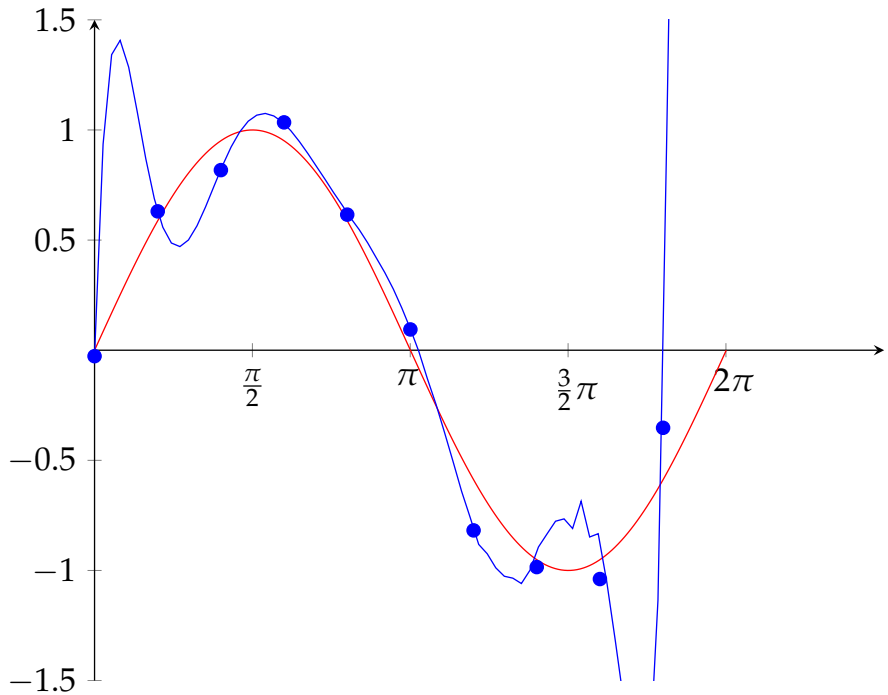
- Avec $||\mathbf{w}||^2 = w_0^2 + w_1^2 + \ldots + w_M^2$
- λ est un coefficient qui permet de contrôler le poids de la régularisation dans la fonction d'erreur.
- Plus λ est élevée, plus la régularisation est importante.
- En minimisant $\tilde{E}(\mathbf{w};\mathcal{D})$, on minimise l'erreur et la valeur des paramètres.

$M = 9\ \lambda = 0$



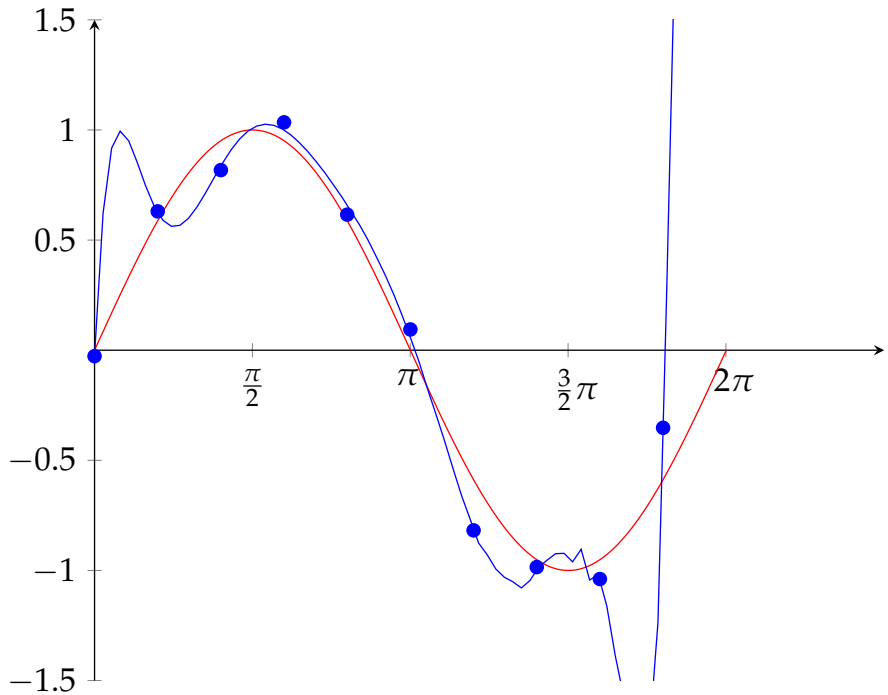
- $||\mathbf{w}||^2 = 21515$
- $E(\mathbf{w}^*, \mathcal{D}) = 1.52 \times 10^{-07}$

$M = 9\ \lambda = 10^{-7}$



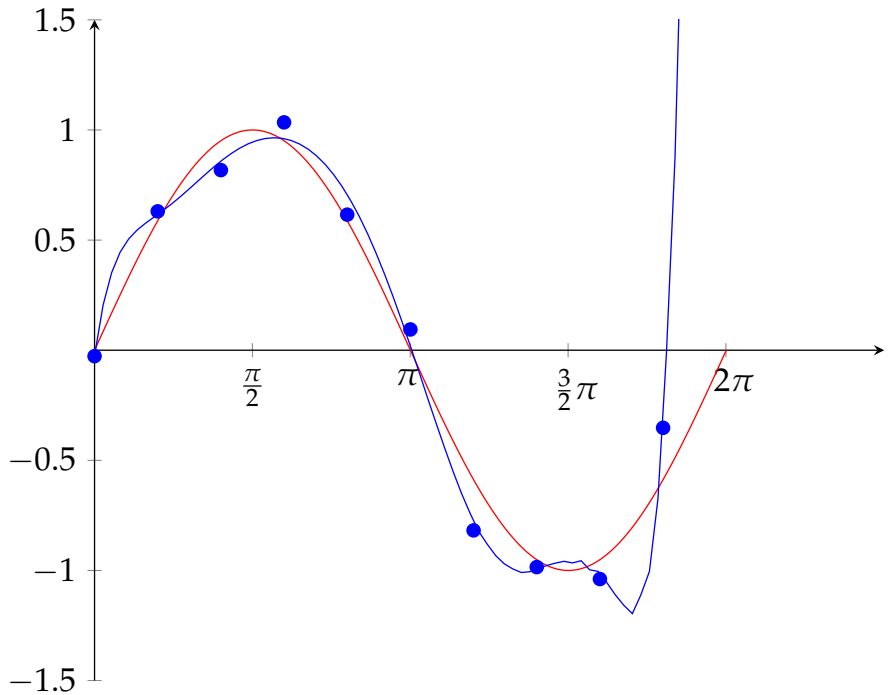
- $||\mathbf{w}||^2 = 18806$
- $E(\mathbf{w}^*, \mathcal{D}) = 7.09 \times 10^{-05}$

$M = 9\ \lambda = 10^{-6}$



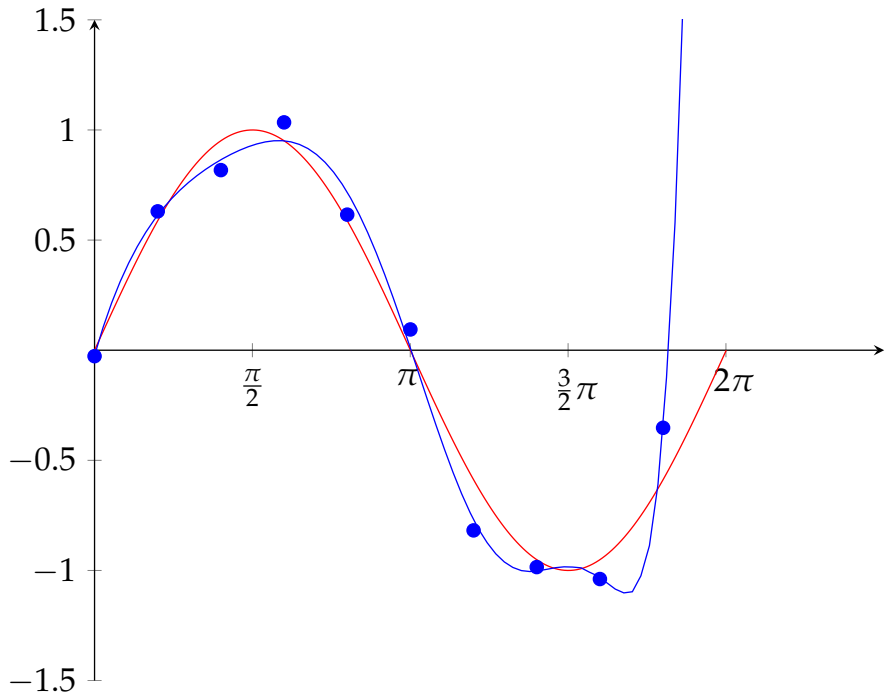
- $||\mathbf{w}||^2 = 7425$
- $E(\mathbf{w}^*, \mathcal{D}) = 0.00263$

$M = 9\ \lambda = 10^{-5}$



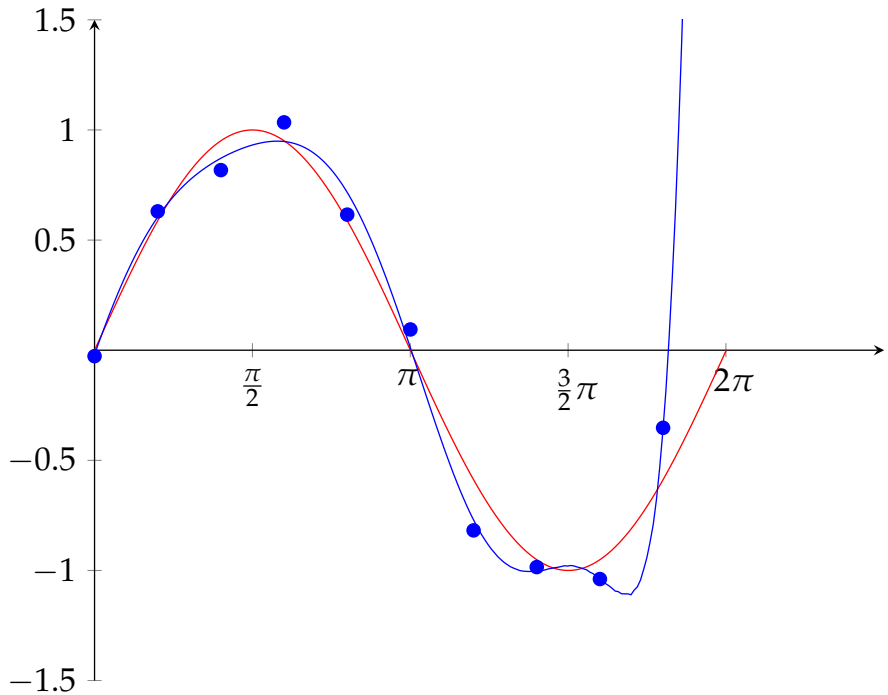
- $||\mathbf{w}||^2 = 336$
- $E(\mathbf{w}^*, \mathcal{D}) = 0.0117$

$M = 9\ \lambda = 10^{-4}$



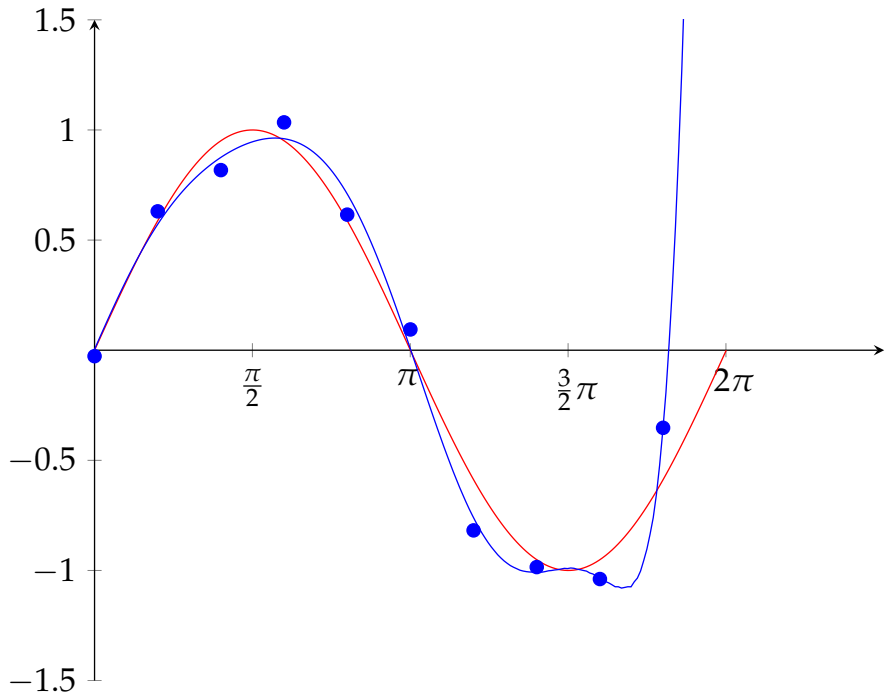
- $||\mathbf{w}||^2 = 6.67$
- $E(\mathbf{w}^*, \mathcal{D}) = 0.0149$

$M = 9\ \lambda = 10^{-3}$



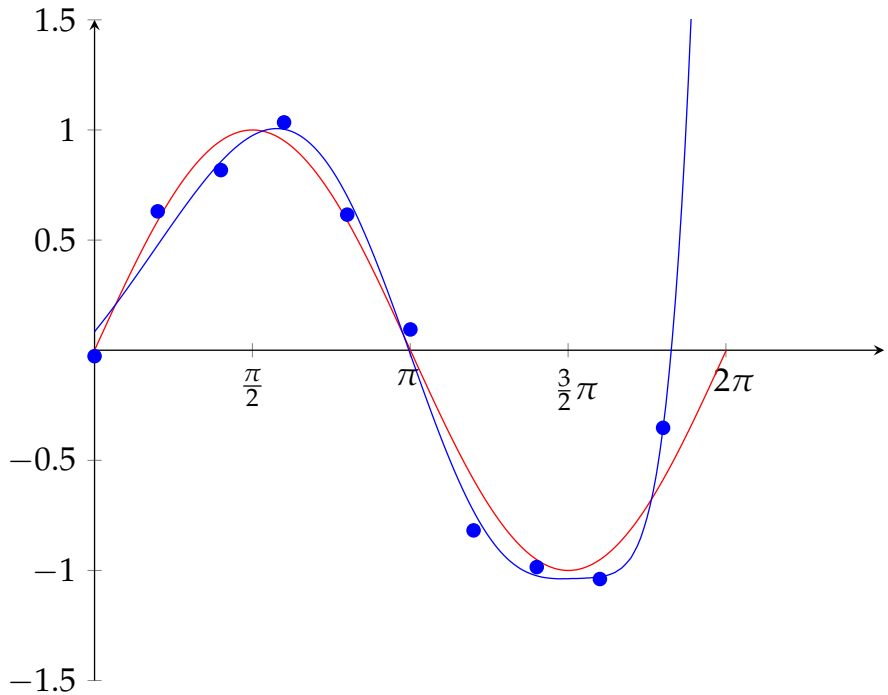
- $||\mathbf{w}||^2 = 1.99$
- $E(\mathbf{w}^*, \mathcal{D}) = 0.0154$

$M = 9\ \lambda = 10^{-2}$



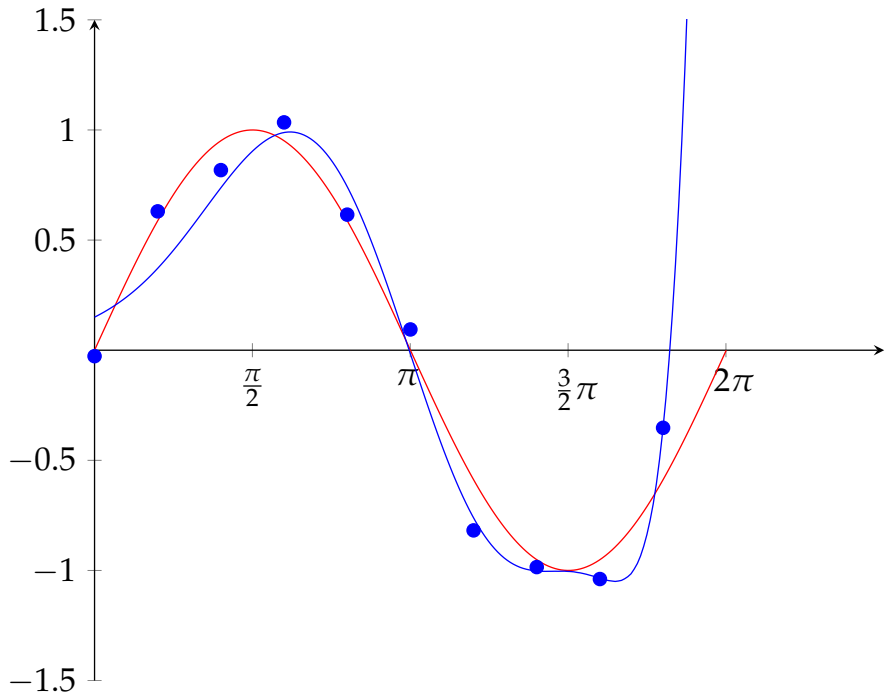
- $||\mathbf{w}||^2 = 1.31$
- $E(\mathbf{w}^*, \mathcal{D}) = 0.016$

$M = 9\ \lambda = 10^{-1}$



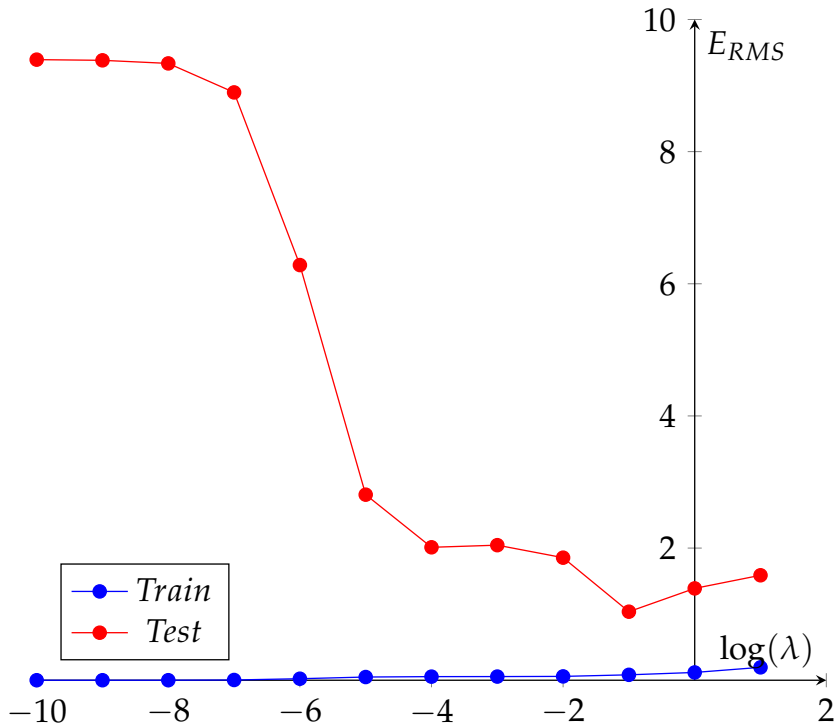
- $||\mathbf{w}||^2 = 0.35$
- $E(\mathbf{w}^*, \mathcal{D}) = 0.033$

$M = 9\ \lambda = 1$



- $||\mathbf{w}||^2 = 0.106$
- $E(\mathbf{w}^*, \mathcal{D}) = 0.069$

Influence de la régularisation



Sources

- Christopher Bishop, *Pattern recognition and machine learning* Springer, 2006.