Course 3

Multivariate regression

Christophe Eloy

Outline

- Linear regression with one variable
- Linear regression with multiple variables
- Data normalization
- Bias-variance tradeoff
- Regularization

Univariate linear regression

- Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$
- Parameters: $\theta = [\theta_0, \theta_1]^T$
- Cost function (least squares): $J(\theta_0, \theta_1) = \frac{1}{2N} \sum_{i=1}^{N} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$
- Goal: find $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Gradient descent algorithm

- Iterative procedure

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

$$\theta := \theta - \alpha \nabla_\theta J(\theta)$$

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{N} \sum_{i=1}^{N} \left(h_\theta(x^{(i)}) - y^{(i)} \right)$$

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{N} \sum_{i=1}^{N} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x^{(i)}$$

- Learning parameter: α

- Full batch vs. stochastic vs. mini-batch gradient descent

Multivariate linear regression

- Hypothesis: $h_\theta(x) = \theta \cdot x$ with $x = [1, x_1 \cdots x_n]^T$
- Parameters: $\theta = [\theta_0 \cdots \theta_n]^T$
- Cost function (least squares): $J(\theta) = \frac{1}{2N} \sum_{i=1}^{N} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$
- Goal: find $\min_\theta J(\theta)$
- Gradient descent: $\theta := \theta - \alpha \nabla_\theta J(\theta)$
- Normal equation: $\theta = \left(X^T \cdot X \right)^{-1} \cdot X^T \cdot Y$

Data normalization

- Z normalization

$$\hat{x}_j^{(i)} = \frac{x_j^{(i)} - \langle x_j \rangle}{\sigma_j}, \text{ with } \sigma_j^2 = \mathrm{Var}(x_j)$$

such that $\langle \hat{x}_j \rangle = 0$ and $\mathrm{Var}(\hat{x}_j) = 1$

- Min-max normalization

$$\tilde{x}_j^{(i)} = \frac{x_j^{(i)} - \min(x_j)}{\max(x_j) - \min(x_j)}$$

such that $0 \leq \tilde{x}_j^{(i)} \leq 1$

Bias-variance tradeoff

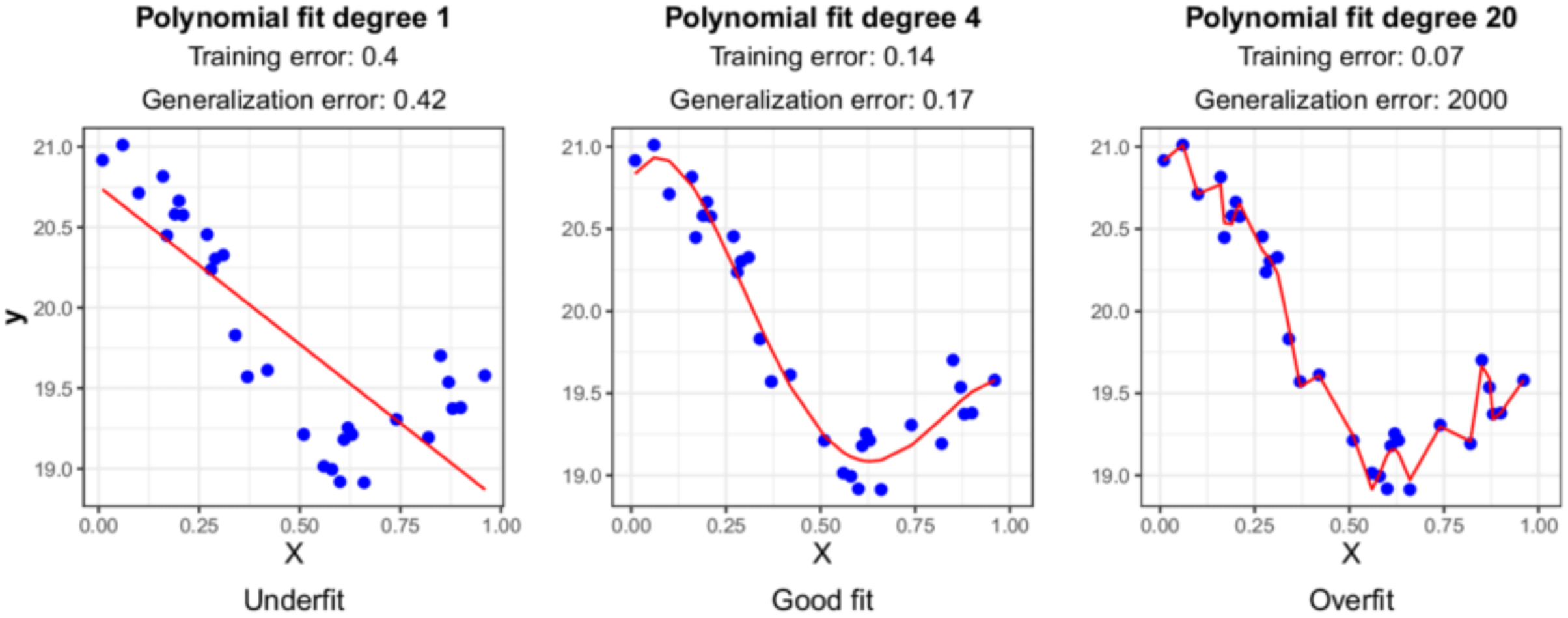
- Function: $y = f(x) + \varepsilon$
- Hypothesis: $h_\theta(x)$
- Cost/error: $J(\theta) = \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$
- Expected error: $\mathbb{E}(J) = Noise + Bias^2 + Var$

$$Noise = \frac{1}{2}\sigma^2$$

$$Bias^2 = \frac{1}{2}\left(f(x^{(i)}) - \mathbb{E}\left[h_\theta(x^{(i)})\right]\right)^2$$

$$Var = \frac{1}{2}\mathbb{E}\left[\left(h_\theta(x^{(i)}) - \mathbb{E}\left(h_\theta(x^{(i)})\right)\right)^2\right]$$

Problem of overfitting



- To avoid overfitting, one method is regularization

Badillo et al. (2020). 10.1002/cpt.1796

Regularization

- New cost function: $J(\boldsymbol{\theta}) = \frac{1}{2N}\left[\sum_{i=1}^{N}\left(h_\theta(\boldsymbol{x}^{(i)}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$
- Calculation of the gradient

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{N}\sum_{i=1}^{N}\left(h_\theta(\boldsymbol{x}^{(i)}) - y^{(i)}\right)x_0^{(i)}$$

$$\frac{\partial J}{\partial \theta_j} = \frac{1}{N}\sum_{i=1}^{N}\left(h_\theta(\boldsymbol{x}^{(i)}) - y^{(i)}\right)x_j^{(i)} + \frac{\lambda}{N}\theta_j$$

- Normal equation $\theta = \left(X^T \cdot X + \lambda \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}\right)^{-1} \cdot X^T \cdot Y$