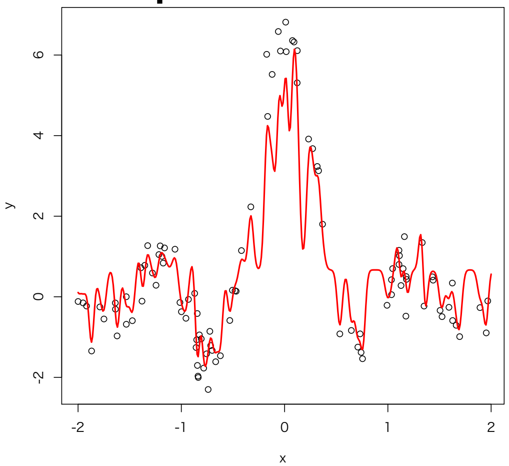
1 Regression Analysis

Training data: $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n), \quad \boldsymbol{x}_i \in \mathbb{R}^d, \; y_i \in \mathbb{R}$.

$$y = \underbrace{f(\boldsymbol{x})}_{\text{function}} + \underbrace{\varepsilon}_{\text{r.v.: error}} \longrightarrow \text{estimate } f(\boldsymbol{x})$$

- Simple model is not good to learn complex data structure.
  $\longrightarrow$ Complex model is desirable

- Too complex models $\Longrightarrow$ overfitting



Overfit to data
$\longrightarrow$ low prediction accuracy

- It is crucial to tuning the model complexity properly.

Linear Regression Model

- Map $\boldsymbol{x}$ to high-dimensional space:

$$\boldsymbol{x} \longmapsto \boldsymbol{\phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \ldots, \phi_D(\boldsymbol{x}))^T$$

  $\phi_k(\boldsymbol{x})$: (non-linear) basis functions

- Linear Regression Model:

$$f(\boldsymbol{x}) = \sum_{k=1}^{D} a_k \phi_k(\boldsymbol{x}) = \boldsymbol{a}^T \boldsymbol{\phi}(\boldsymbol{x})$$

Estimate the coefficient $\boldsymbol{a}$ from training data.

- Choose functions $\boldsymbol{\phi}(\boldsymbol{x})$ having a "nice" property
  $\longrightarrow$ the computation is tractable.

- To avoid overfitting, regularization and cross validation are useful.

— Kernel Regression Analysis —

- least square method with kernel-based modeling

Estimation for Linear Regression Models

- Least Square Method (LSM):

$$\sum_{i=1}^{n}(y_i - \boldsymbol{\phi}(\boldsymbol{x}_i)^T\boldsymbol{a})^2 = \|\boldsymbol{y} - \boldsymbol{\Phi}^T\boldsymbol{a}\|^2 \ \rightarrow \ \text{minimize w.r.t. } \boldsymbol{a},$$

where $\boldsymbol{\Phi} = (\boldsymbol{\phi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\phi}(\boldsymbol{x}_n)) \in \mathbb{R}^{D\times n}, \quad \boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n.$

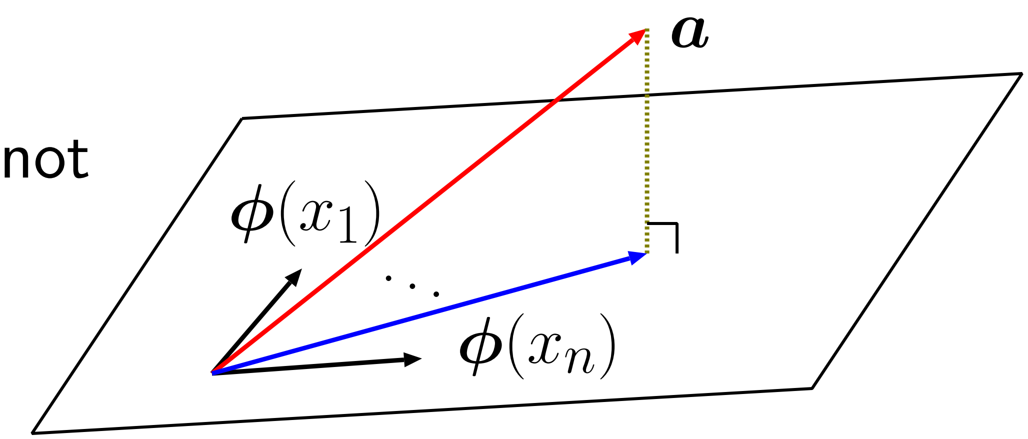
- $\operatorname{rank}\boldsymbol{\Phi} = D \ \Rightarrow \ \widehat{\boldsymbol{a}} = (\boldsymbol{\Phi}\boldsymbol{\Phi}^T)^{-1}\boldsymbol{\Phi}\boldsymbol{y}.$

Another expression of the solution:

$$\min_{\boldsymbol{a}\in\mathbb{R}^D} \sum_{i=1}^{n}(y_i - \boldsymbol{\phi}(\boldsymbol{x}_i)^T\boldsymbol{a})^2$$

- the solution lies on $\mathrm{span}\{\boldsymbol{\phi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\phi}(\boldsymbol{x}_n)\}$.

Orthogonal component does not affect the square error.



- $\boldsymbol{a} = \sum_{j=1}^{n} \beta_j \boldsymbol{\phi}(\boldsymbol{x}_j) = \Phi\boldsymbol{\beta}, \quad \boldsymbol{\beta} \in \mathbb{R}^n$.

$$\sum_{i=1}^{n}(y_i - \boldsymbol{\phi}(\boldsymbol{x}_i)^T\boldsymbol{a})^2 = \|\boldsymbol{y} - \boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{\beta}\|^2 \longrightarrow \min_{\boldsymbol{\beta}}$$

Optimality conditions : $\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{\Phi}^T \underbrace{\boldsymbol{\Phi}\widehat{\boldsymbol{\beta}}}_{\widehat{\boldsymbol{a}}} = \boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{y}$

Define n by n matrix $K = (K_{ij})$ as

$$K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) \stackrel{\textsf{def}}{=} \boldsymbol{\phi}(\boldsymbol{x}_i)^T\boldsymbol{\phi}(\boldsymbol{x}_j) \in \mathbb{R},$$

$$\Longrightarrow \quad K = \boldsymbol{\Phi}^T\boldsymbol{\Phi}$$

- $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\phi}(\boldsymbol{x}')$ is called kernel function
- K: Gram matrix

(the rigorous definition is given later)

- Optimality condition:

$$\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{\Phi}\widehat{\boldsymbol{\beta}} = \boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{y} \iff K^2\widehat{\boldsymbol{\beta}} = K\boldsymbol{y}$$

$$\implies \text{calculate } \widehat{\boldsymbol{\beta}} = (\widehat{\beta}_1, \ldots, \widehat{\beta}_n)^T$$

- Estimated regression function: $\widehat{f}(\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x})^T\widehat{\boldsymbol{a}}$.

$$\widehat{f}(\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x})^T \underbrace{\sum_{i=1}^n \boldsymbol{\phi}(\boldsymbol{x}_i)\widehat{\beta}_i}_{\widehat{\boldsymbol{a}}} = \sum_{i=1}^n k(\boldsymbol{x}, \boldsymbol{x}_i)\widehat{\beta}_i$$

- kernel function $k(\boldsymbol{x}, \boldsymbol{x}') \implies$ estimator $\widehat{f}(\boldsymbol{x})$

Examples of kernel functions: $\boldsymbol{x} \in \mathbb{R}^d \longmapsto \boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^D$.

- linear kernel: $D = d$.

$$k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^T \boldsymbol{x}', \quad (\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{x})$$

Model: $y = \boldsymbol{a}^T \boldsymbol{\phi}(\boldsymbol{x}) + \varepsilon = \boldsymbol{a}^T \boldsymbol{x} + \varepsilon$

- Polynomial kernel of degree $\ell \in \mathbb{N}$: $D = \frac{(\ell+d)!}{\ell!\, d!}$

$$k(\boldsymbol{x}, \boldsymbol{x}') = (1 + \boldsymbol{x}^T \boldsymbol{x}')^\ell,$$

Model: $y = \boldsymbol{a}^T \boldsymbol{\phi}(\boldsymbol{x}) + \varepsilon$.

$\boldsymbol{\phi}(\boldsymbol{x})$: all monominals of degree $\leq \ell$.

For $d = 2,\ \ell = 2$ and $\boldsymbol{x} = (x_1, x_2)^T \in \mathbb{R}^2$,

$$\begin{aligned}
\boldsymbol{\phi}(\boldsymbol{x}) &= (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)^T. \\
\boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\phi}(\boldsymbol{z}) &= 1 + 2x_1z_1 + 2x_2z_2 + x_1^2z_1^2 + x_2^2z_2^2 + 2x_1x_2z_1z_2 \\
&= (1 + x_1z_1 + x_2z_2)^2
\end{aligned}$$

- Gaussian kernel: $D = \infty$.

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\{-\sigma \cdot \|\boldsymbol{x} - \boldsymbol{x}'\|^2\}, \quad \sigma > 0$$

For $d = 1$, $\sigma = 1$ and $x \in \mathbb{R}$,

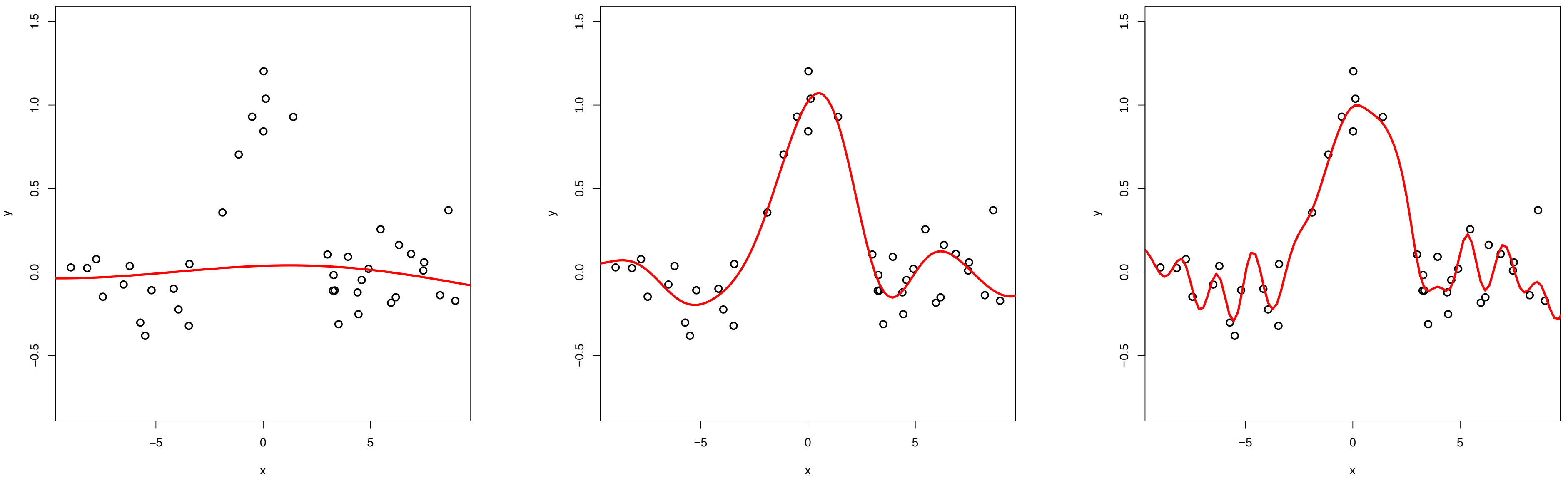
$$\boldsymbol{\phi}(x) = (\phi_0(x), \phi_1(x), \phi_2(x), \cdots)^T, \qquad \phi_j(x) = \frac{x^j e^{-x^2/2}}{\sqrt{j!}}, \quad x \in \mathbb{R}$$

Overfiting to data

- simple model: hard to deal with complex data
  $\longrightarrow$ use the model with many parameters
- model with too many parameters does not work.
  overfitting to data.

degree of freedom

small medium large



Regularization: tune the degree of freedom

large model & appropriate constraint

linear regression model : $\quad y = \underline{\boldsymbol{a}^T \boldsymbol{\phi}(\boldsymbol{x}) + b} + \varepsilon$

ex. $\boldsymbol{\phi}(x) = (x, x^2, x^3, \ldots, x^{100})$, $\boldsymbol{\phi}(\boldsymbol{x})$ of Gaussian kernel, etc.
data: $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$.

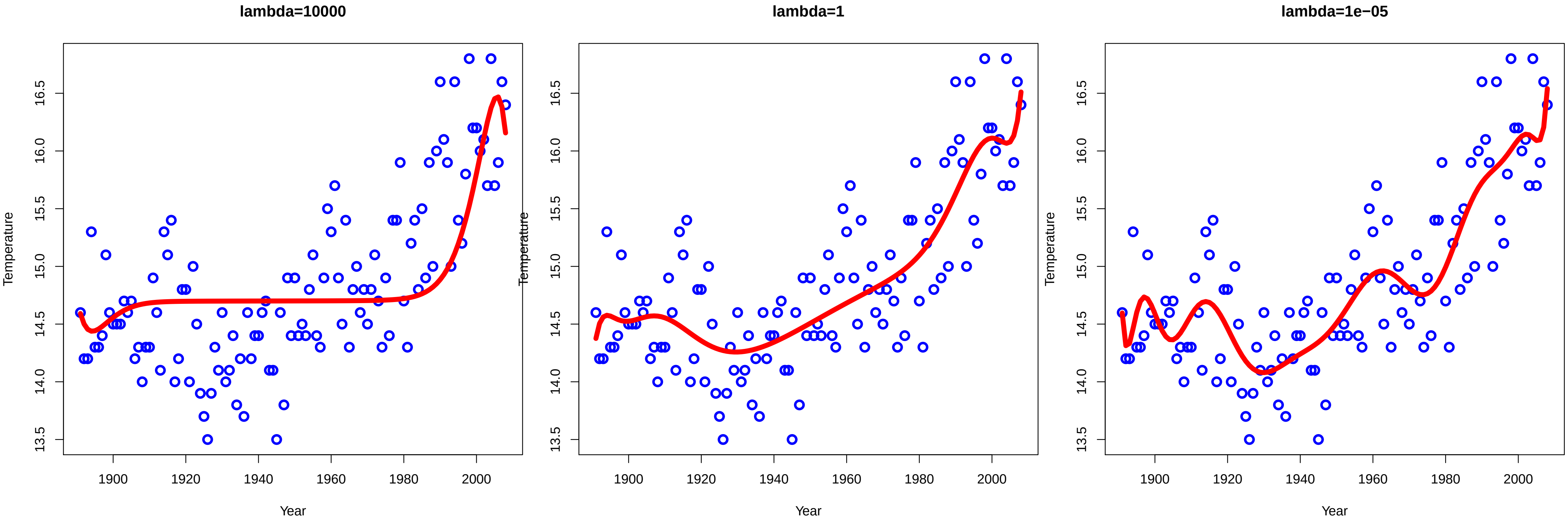
$$\min_{\boldsymbol{a},\, b} \sum_{i=1}^{n} (y_i - (\boldsymbol{\phi}(\boldsymbol{x}_i)^T \boldsymbol{a} + b))^2 + \underbrace{\lambda \|\boldsymbol{a}\|^2}_{\text{regularization term}} \qquad \text{(Ridge regression)}$$

$\Longrightarrow$ opt. sol. $\widehat{\boldsymbol{a}}$, $\widehat{b}$. $\quad \widehat{f}(x) = \widehat{\boldsymbol{a}}^T \boldsymbol{\phi}(\boldsymbol{x}) + \widehat{b}$

$$\min_{\boldsymbol{a} \in \mathbb{R}^D, b \in \mathbb{R}} \|\boldsymbol{y} - \Phi^T \boldsymbol{a} - b\mathbf{1}\|^2 + \lambda \|\boldsymbol{a}\|^2, \quad \Phi = (\boldsymbol{\phi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\phi}(\boldsymbol{x}_n))$$

regularization parameter $\lambda > 0$.

λ: large λ: medium λ: small



small $\longleftarrow$ degree of freedom $\longrightarrow$ large

Kernel representation of Ridge regression:

$$\min_{\boldsymbol{a},b} \|\boldsymbol{y} - \Phi^T \boldsymbol{a} - b\mathbf{1}\|^2 + \lambda \|\boldsymbol{a}\|^2, \quad \Phi = (\boldsymbol{\phi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\phi}(\boldsymbol{x}_n)).$$

In the same way as the standard LMS, the optimal $\widehat{\boldsymbol{a}}$ lies on the subspace $\mathrm{span}\{\boldsymbol{\phi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\phi}(\boldsymbol{x}_n)\}$.

Substitute $\boldsymbol{a} = \sum_{i=1}^{n} \boldsymbol{\phi}(\boldsymbol{x}_i)\beta_i = \boldsymbol{\Phi}\boldsymbol{\beta}$, then for $K = \boldsymbol{\Phi}^T\boldsymbol{\Phi}$,

$$\begin{aligned}\|\boldsymbol{y} - \Phi^T \boldsymbol{a} - b\mathbf{1}\|^2 + \lambda \|\boldsymbol{a}\|^2 &= \|\boldsymbol{y} - \boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{\beta} - b\mathbf{1}\|^2 + \lambda \boldsymbol{\beta}^T \boldsymbol{\Phi}^T \boldsymbol{\Phi}\boldsymbol{\beta} \\ &= \|\boldsymbol{y} - K\boldsymbol{\beta} - b\mathbf{1}\|^2 + \lambda \boldsymbol{\beta}^T K \boldsymbol{\beta} \ \longrightarrow \ \min_{\boldsymbol{\beta},b}\end{aligned}$$

Kernel-Ridge Regression

- Optimality condition

$$\min_{\boldsymbol{\beta},b} \|\boldsymbol{y} - K\boldsymbol{\beta} - b\mathbf{1}\|^2 + \lambda\boldsymbol{\beta}^T K\boldsymbol{\beta}$$

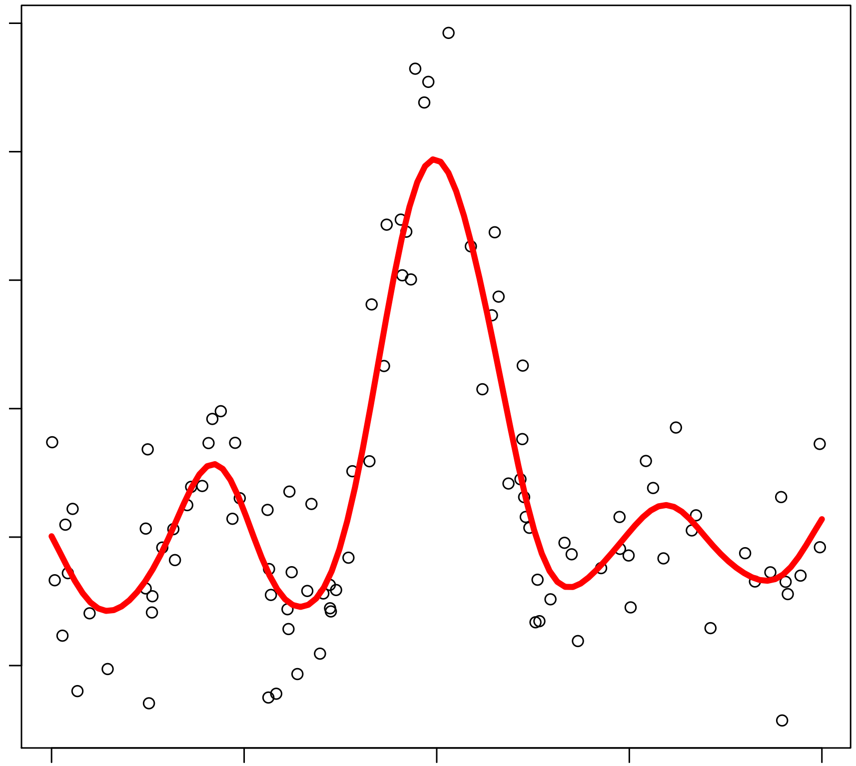
$$\Longrightarrow \quad \begin{pmatrix} K+\lambda I & \mathbf{1} \\ \mathbf{1}^T K & n \end{pmatrix} \begin{pmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{b} \end{pmatrix} = \begin{pmatrix} \boldsymbol{y} \\ \mathbf{1}^T\boldsymbol{y} \end{pmatrix}$$

- estimated regression function:

$$\widehat{f}(\boldsymbol{x}) = \boldsymbol{\phi}(x)^T \sum_{i=1}^{n} \boldsymbol{\phi}(\boldsymbol{x}_i)\widehat{\beta}_i + \widehat{b} = \sum_{i=1}^{n} k(\boldsymbol{x},\boldsymbol{x}_i)\widehat{\beta}_i + \widehat{b}$$

Plot: estimated regression function

- kernel width: $\sigma = 3$
- regularization par.: $\lambda = 1$



— Model Selection —

How to choose regularization parameter λ ?

- Training error and Test error
- Cross Validation for model parameter tuning

Kernel-Ridge Regression

$$\text{Gaussian kernel}: \quad k(\boldsymbol{x}, \boldsymbol{x}') = \exp\{-\sigma \cdot \|\boldsymbol{x} - \boldsymbol{x}'\|^2\}$$

We need to determine the following model parameters:

- Regularization par.: λ
- kernel parameter: σ

How to choose λ and σ ?

Note. For the polynomial kernel, we need to determine λ and the degree ℓ.

Training error and Test error

- training data: $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n) \sim_{i.i.d.} p(\boldsymbol{x}, y)$
- estimated regression function $\widehat{f}(x)$

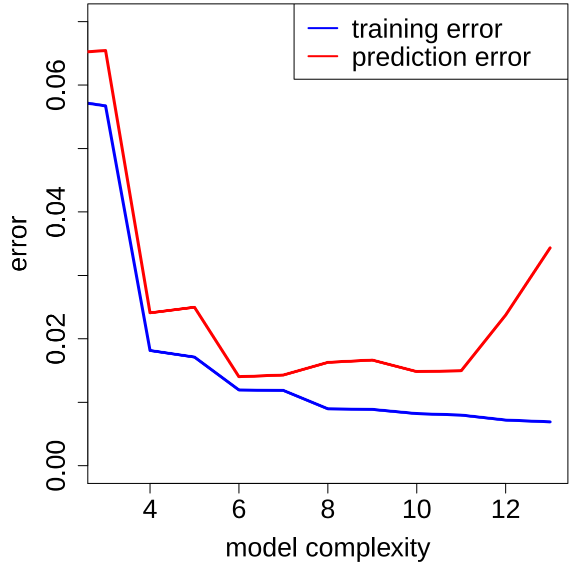
$$\text{training error of } \widehat{f}(\boldsymbol{x}) : \quad \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{f}(\boldsymbol{x}_i) - y_i\right)^2 \quad \text{(calculated from data)}$$

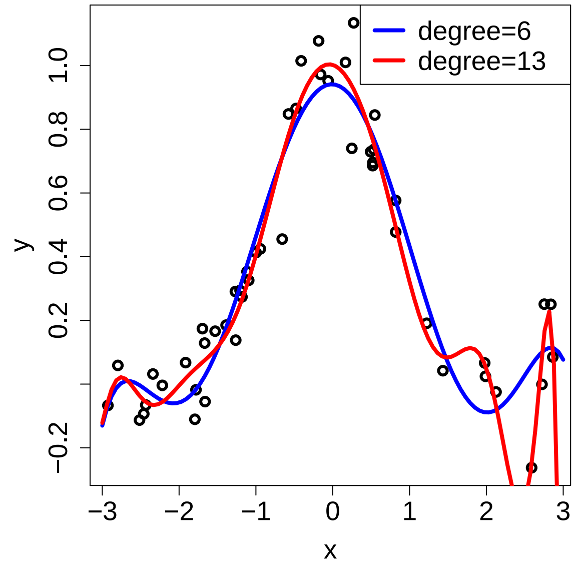
$$\text{test error of } \widehat{f}(\boldsymbol{x}) : \quad \mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\left(\widehat{f}(\boldsymbol{x}) - y\right)^2\right] \quad (P \text{ is unknown})$$

Purpose of Regression Analysis

Find $\widehat{f}(\boldsymbol{x})$ that achieves a small (or minimum) test error.

Polynomial regression





overfitting

- For high degree polynomial models, we have
  - small training error
  - large test error
  - overfitting: large gap between training error and test error.

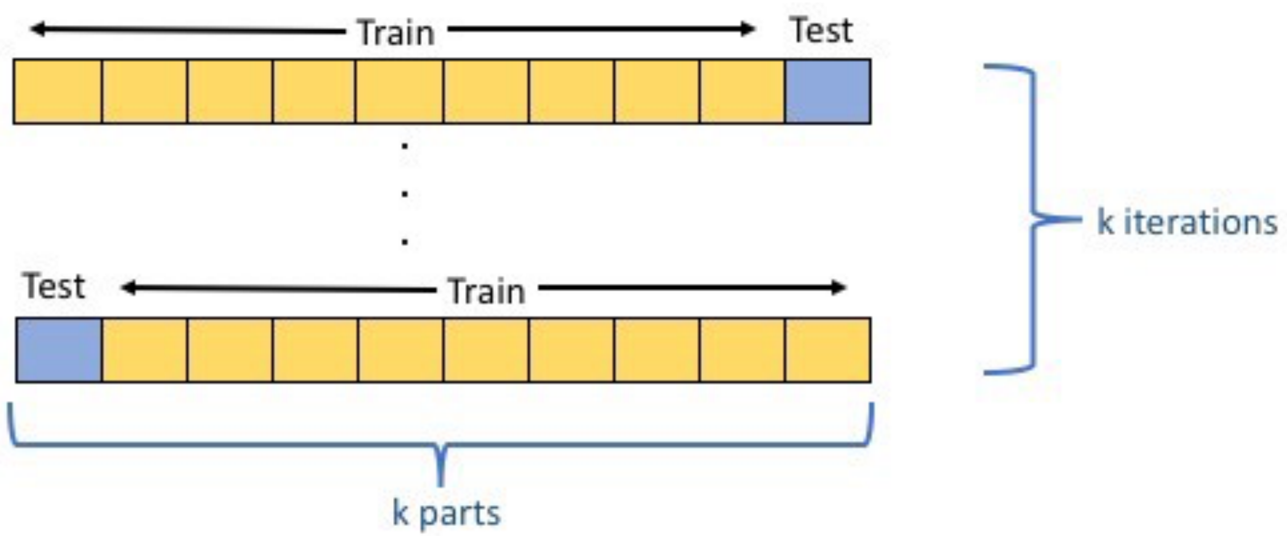
appropriate model complexity is required.

Cross validation: estimator of test error

K-fold Cross Validation Method

Fixing a model parameter, say λ and σ, execute the following procedure.

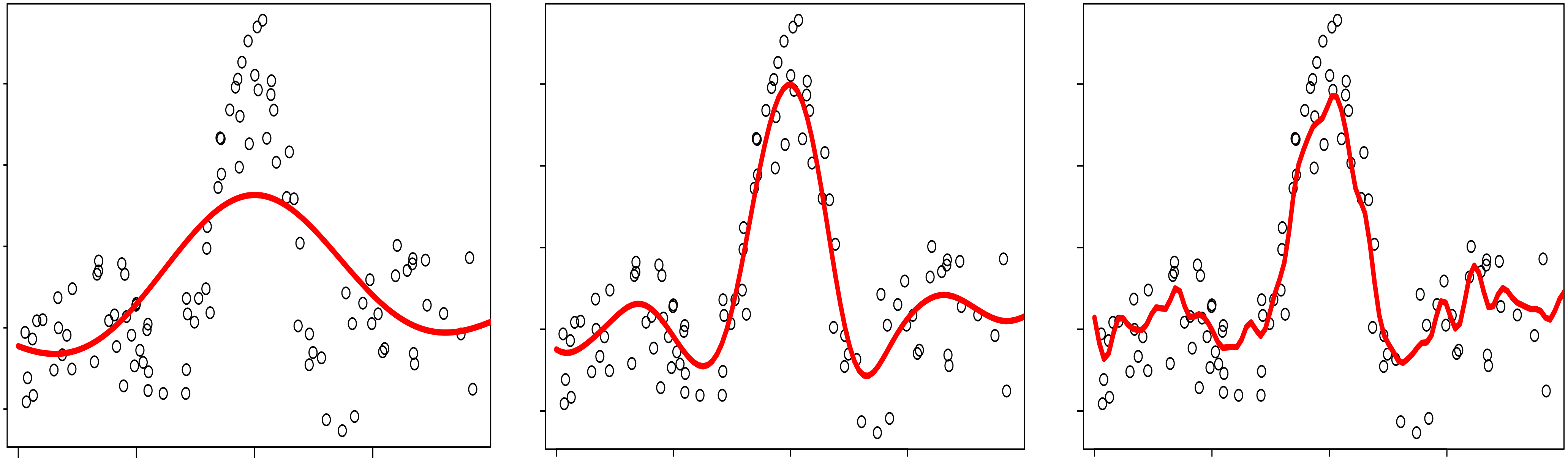
1. Divide the training data into k parts.
2. Use $k-1$ of the parts for training, and 1 for testing.
3. Repeat the procedure k times, rotating the test set.
4. Calculate an expected performance metric (mean square error/test error rate) based on the results across the iterations



https://medium.com/@mtterribile/understanding-cross-validations-purpose-53490faf6a86

Example: Kernel regression with Gaussian kernel

- kernel par. $\sigma > 0$ is determined by K-cv.
- regularization par. $\lambda > 0$ is fixed to 1.



$(\sigma, \lambda) = (0.1, 1)$ $\quad$ $(\sigma, \lambda) = (2.5, 1)$ $\quad$ $(\sigma, \lambda) = (30, 1)$

Median Heuristics for Gaussian Kernel

Gaussian kernel:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\{-\sigma\|\boldsymbol{x} - \boldsymbol{x}'\|^2\}$$

- For computational stability, choose σ such that

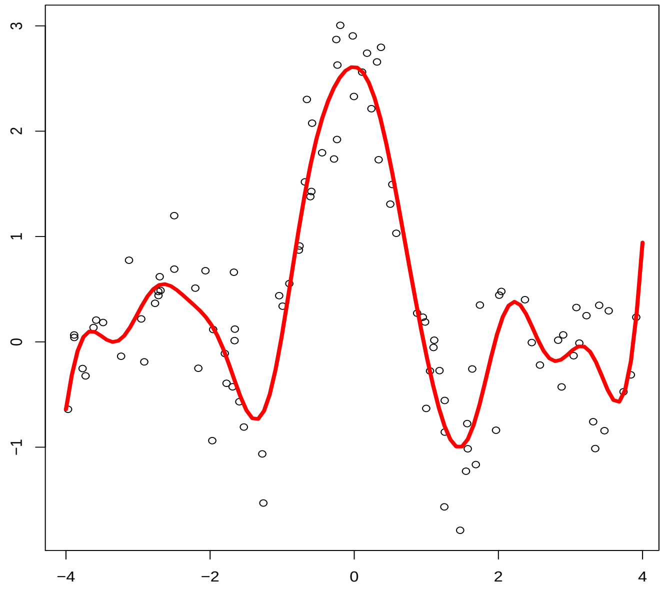
$$\sigma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \ \text{ takes values around } 1.$$

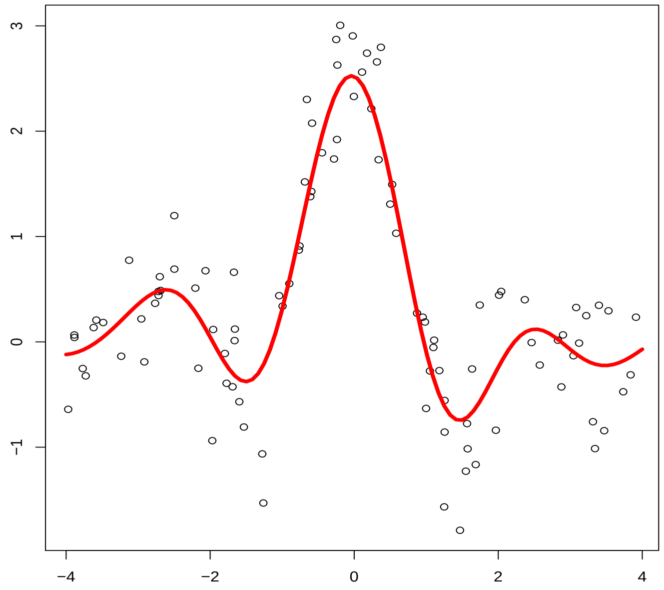
$$\sigma \ \longleftarrow \ \frac{1}{\texttt{median}\{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \,|\, i < j\}}$$

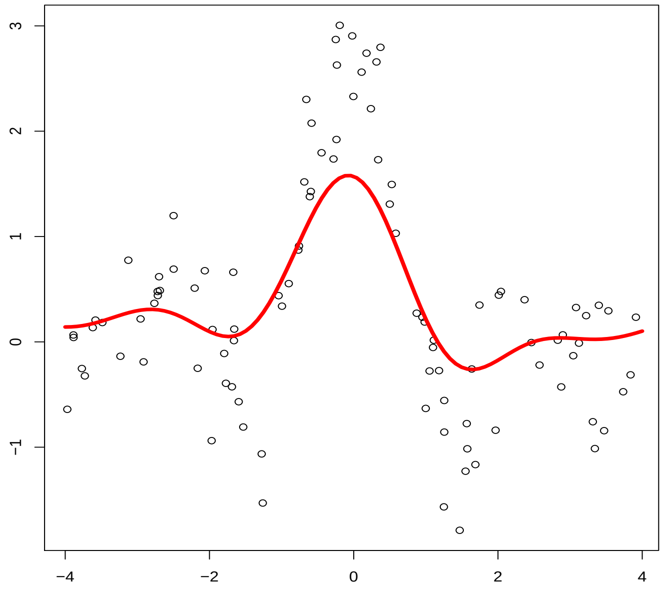
Example

- kernel parameter σ is determined by the heuristics.

- regularization par. λ: K-cv







$(\texttt{sigma}, \lambda) = (0.98, 0.1^5)$ $(\texttt{sigma}, \lambda) = (0.98, 0.78)$ $(\texttt{sigma}, \lambda) = (0.98, 10)$