Model Parameters

- In the process of parameter learning, we fixed model parameters.
- Regularization learning with Gaussian kernel model
  - Gaussian width: $c\ (> 0)$
  - Regularization parameter: $\lambda\ (\geq 0)$

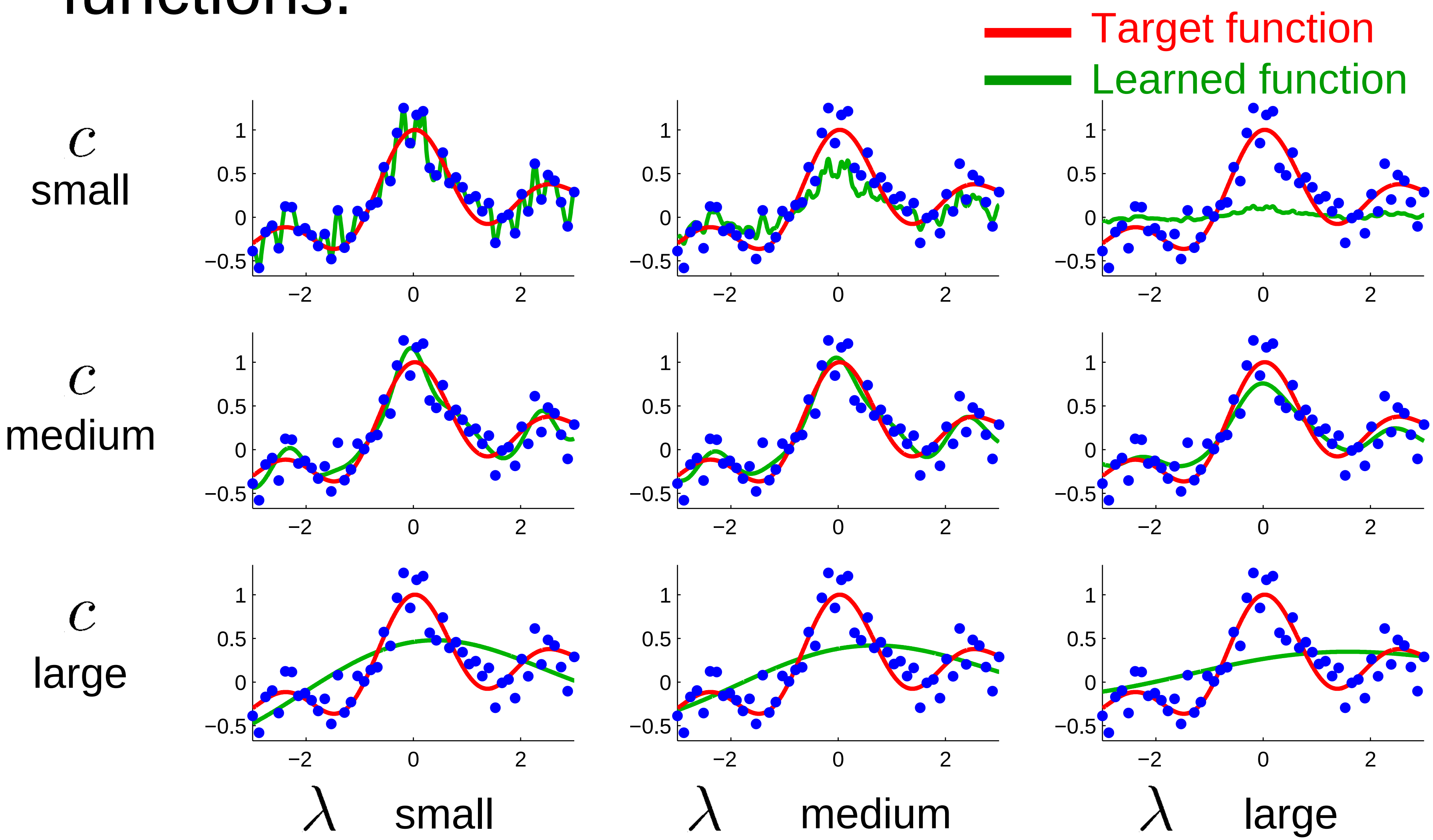
$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i)$$

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2c^2}\right)$$

$$J_{reg}(\boldsymbol{\alpha}) = J_{LS}(\boldsymbol{\alpha}) + \lambda\|\boldsymbol{\alpha}\|^2$$

Different Model Parameters

- Model parameters strongly affect learned functions.



Determining Model Parameters

- We want to determine the model parameters so that the generalization error is minimized.

$$J = \int_{\mathcal{D}} \left(\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x}) \right)^2 p(\boldsymbol{x}) d\boldsymbol{x}$$

- However, $f(\boldsymbol{x})$ and $p(\boldsymbol{x})$ are unknown so the generalization error is not accessible.

Model Selection

- Prepare a set of model candidates.

$$\{\mathcal{M}_i \mid \mathcal{M}_i = (c_i, \lambda_i)\}$$

- Estimate generalization error for each model.

$$\hat{J}(\mathcal{M}_i)$$

- Choose the one with minimum estimated generalization error.

$$\hat{\mathcal{M}} = \operatorname*{argmin}_{\mathcal{M}_i} \hat{J}(\mathcal{M}_i)$$

Estimating Generalization Error

- Suppose we have an extra example $(\boldsymbol{x}', y')$ in addition to $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$.

$$y' = f(\boldsymbol{x}') + \epsilon'$$

- Test the learned function using the extra example.

$$\hat{J}_{extra} = \left(\hat{f}(\boldsymbol{x}') - y'\right)^2$$

- $\hat{J}$ is unbiased w.r.t. $\boldsymbol{x}'$ and ϵ' (except σ^2).

$$\mathbb{E}_{\boldsymbol{x}'}\mathbb{E}_{\epsilon'}\hat{J}_{extra} = J + \sigma^2$$

- However, in practice, we do not have such an extra example $(\boldsymbol{x}', y')$.

Holdout Method

- Divide training examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ into $\{(\boldsymbol{x}_i, y_i)\}_{i \neq j}$ and $(\boldsymbol{x}_j, y_j)$

- Train a learning machine using $\{(\boldsymbol{x}_i, y_i)\}_{i \neq j}$

$$\hat{f}_j(\boldsymbol{x}) \longleftarrow \{(\boldsymbol{x}_i, y_i)\}_{i \neq j}$$

- Test it using the holdout sample $(\boldsymbol{x}_j, y_j)$

$$\hat{J}_j = \left(\hat{f}_j(\boldsymbol{x}_j) - y_j\right)^2$$

Almost Unbiasedness of Holdout

- Holdout method is unbiased w.r.t. $\{\boldsymbol{x}_i, \epsilon_i\}_{i=1}^n$

$$\begin{aligned}\boxed{\mathbb{E}_{\{\boldsymbol{x}_i\}_{i=1}^n}\mathbb{E}_{\{\epsilon_i\}_{i=1}^n}\hat{J}_i &= \mathbb{E}_{\{\boldsymbol{x}_i\}_{i\neq j}}\mathbb{E}_{\{\epsilon_i\}_{i\neq j}}J_j + \sigma^2 \\ &\approx \mathbb{E}_{\{\boldsymbol{x}_i\}_{i=1}^n}\mathbb{E}_{\{\epsilon_i\}_{i=1}^n}J + \sigma^2}\end{aligned}$$

$$J_j = \int_{\mathcal{D}}\left(\hat{f}_j(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2 p(\boldsymbol{x})d\boldsymbol{x}$$

- However, $\hat{J}_j$ is heavily affected by a deviation of a single example $(\boldsymbol{x}_j, y_j)$.

Leave-One-Out Cross-Validation

- Repeat the holdout procedure for all combinations and output the average.

for $j = 1, 2, \ldots, n$

$$\hat{f}_j(\boldsymbol{x}) \longleftarrow \{(\boldsymbol{x}_i, y_i)\}_{i \neq j}$$

$$\hat{J}_j = \left(\hat{f}_j(\boldsymbol{x}_j) - y_j\right)^2$$

end

output $\hat{J}_{LOOCV} = \frac{1}{n}\sum_{j=1}^{n} \hat{J}_j$

- LOOCV is almost unbiased w.r.t. $\{\boldsymbol{x}_i, \epsilon_i\}_{i=1}^n$

$$\mathbb{E}_{\{\boldsymbol{x}_i\}_{i=1}^n}\mathbb{E}_{\{\epsilon_i\}_{i=1}^n}\hat{J}_{LOOCV} \approx \mathbb{E}_{\{\boldsymbol{x}_i\}_{i=1}^n}\mathbb{E}_{\{\epsilon_i\}_{i=1}^n} J + \sigma^2$$

k-fold Cross-Validation

- Randomly split training examples into k disjoint subsets $\{\mathcal{T}_j\}_{j=1}^{k}$.

for $i = 1, 2, \ldots, k$

$$\hat{f}_j(\boldsymbol{x}) \longleftarrow \{(\boldsymbol{x}_i, y_i) \mid i \notin \mathcal{T}_j\}$$

$$\hat{J}_j = \frac{1}{|\mathcal{T}_j|} \sum_{i \in \mathcal{T}_j} \left(\hat{f}_j(\boldsymbol{x}_i) - y_i \right)^2$$

end

output $\hat{J}_{kCV} = \frac{1}{k} \sum_{j=1}^{k} \hat{J}_j$

Advantages of CV

- Wide applicability: Almost unbiasedness of LOOCV holds for (virtually) any learning methods
- Practical usefulness: CV is shown to work very well in many practical applications

Disadvantages of CV

- Computationally expensive: It requires repeating training of models with different subsets of training samples
- Input independence: Almost unbiasedness holds w.r.t. the expectation over both training input points and output noise, although training input points are specifically given.
- Number of folds: It is often recommended to use $k = 5, 10$. However, how to choose k is still open.

Closed Form of LOOCV

- Linear model

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{p} \alpha_i \varphi_i(\boldsymbol{x})$$

- Quadratically constrained least-squares

$$J_{QCLS}(\boldsymbol{\alpha}) = J_{LS}(\boldsymbol{\alpha}) + \lambda \|\boldsymbol{\alpha}\|^2$$

$$\boxed{\hat{J}_{LOOCV} = \frac{1}{n} \|\widetilde{\boldsymbol{H}}^{-1} \boldsymbol{H} \boldsymbol{y}\|^2}$$

$$\boldsymbol{H} = \boldsymbol{I} - \boldsymbol{X} \boldsymbol{L}_{QCLS} \qquad \boldsymbol{L}_{QCLS} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top$$

$\widetilde{\boldsymbol{H}}$:same diagonal as $\boldsymbol{H}$ but zero for off-diagonal