Pattern Information Processing:<sup>73</sup> Model Selection by Cross-Validation

> Masashi Sugiyama (Department of Computer Science)

Contact: W8E-505 <u>sugi@cs.titech.ac.jp</u> http://sugiyama-www.cs.titech.ac.jp/~sugi/

### **Model Parameters**

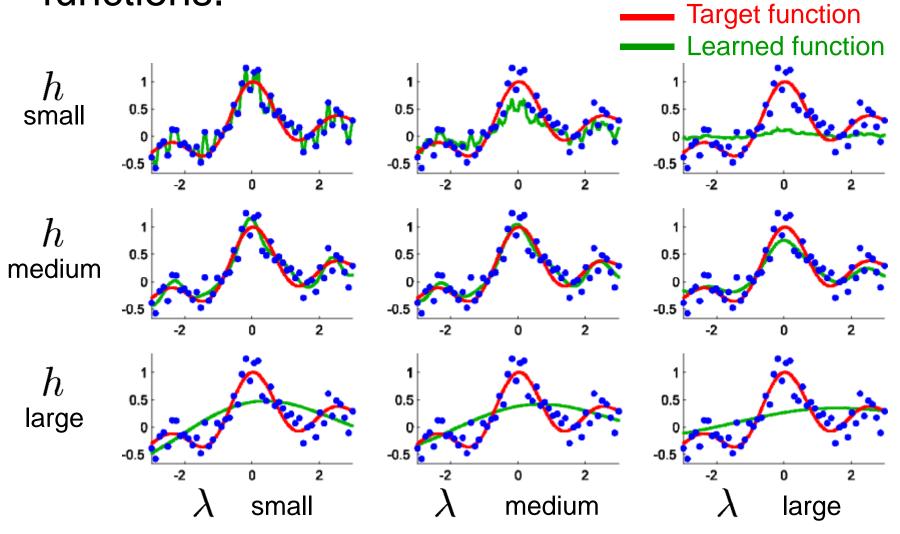
- In the process of parameter learning, we fixed model parameters.
- For example, quadratically constrained leastsquares with a Gaussian kernel model:
  - Gaussian width: h (> 0)
  - Regularization parameter:  $\lambda \ (\geq 0)$

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^{b}} \left[ \sum_{i=1}^{n} \left( f_{\boldsymbol{\alpha}}(\boldsymbol{x}_{i}) - y_{i} \right)^{2} + \lambda \|\boldsymbol{\alpha}\|^{2} \right]$$

$$f_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|^2}{2h^2}\right)$$

## **Different Model Parameters**

Model parameters strongly affect learned functions.



## Determining Model Parameters<sup>76</sup>

We want to determine the model parameters so that the generalization error (expected test error) is minimized.

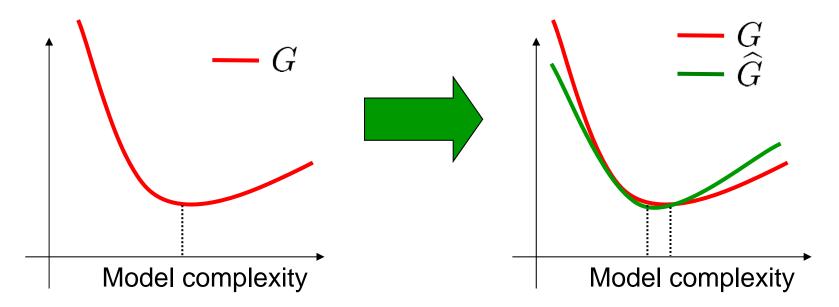
$$G = \int_{\mathcal{D}} \left( \hat{f}(t) - f(t) \right)^2 q(t) dt$$
$$t \sim q(t)$$

However, f(x) is unknown so the generalization error is not accessible.
q(x) may also be unknown.

## Generalization Error Estimation<sup>77</sup>

$$G = \int_{\mathcal{D}} \left( \hat{f}(\boldsymbol{t}) - f(\boldsymbol{t}) \right)^2 q(\boldsymbol{t}) d\boldsymbol{t}$$

Instead, we use a generalization error estimate.



## **Model Selection**

$$\min_{\mathcal{M}} G = \int_{\mathcal{D}} \left( \hat{f}(\boldsymbol{t}) - f(\boldsymbol{t}) \right)^2 q(\boldsymbol{t}) d\boldsymbol{t}$$

- 1. Prepare a set of model candidates.  $\{\mathcal{M} \mid \mathcal{M} = (h, \lambda)\}$
- 2. Estimate generalization error for each model.  $\widehat{G}(\mathcal{M})$
- 3. Choose the one that minimizes the estimated generalization error.

$$\widehat{\mathcal{M}} = \operatorname*{argmin}_{\mathcal{M}} \widehat{G}(\mathcal{M})$$

## **Extra-Sample Method**

79

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

Idea: Test the prediction performance of the learned function using the extra example.

$$\widehat{G}_{extra} = \left(\widehat{f}(\mathbf{x}') - y'\right)^2$$
$$\widehat{f} \longleftarrow \{(\mathbf{x}_i, y_i)\}_{i=1}^n$$

Extra-Sample Method (cont.)  
Suppose 
$$(x', y')$$
 satisfies:  $\mathbb{E}_{\epsilon'}[\epsilon'] = 0$   
 $x' \sim q(x)$   $\mathbb{E}_{\epsilon'}[\epsilon'^2] = \sigma^2$   
 $y' = f(x') + \epsilon'$   $\mathbb{E}_{\epsilon'}[\epsilon'\epsilon_i] = 0, \forall i$   
 $\mathbb{E}_{\epsilon'}$ :Expectation over noise  $\epsilon'$   
 $\widehat{G}_{extra}$  is unbiased w.r.t.  $x'$  and  $\epsilon'$  (up to  $\sigma^2$ ):  
 $\mathbb{E}_{x'}\mathbb{E}_{\epsilon'}[\widehat{G}_{extra}] = G + \sigma^2$ 

Proof: 
$$\mathbb{E}_{\boldsymbol{x}'}\mathbb{E}_{\epsilon'}\left(\hat{f}(\boldsymbol{x}') - f(\boldsymbol{x}') - \epsilon'\right)^2$$
  

$$= \mathbb{E}_{\boldsymbol{x}'}\mathbb{E}_{\epsilon'}\left[(\hat{f}(\boldsymbol{x}') - f(\boldsymbol{x}'))^2 - 2\epsilon'(\hat{f}(\boldsymbol{x}') - f(\boldsymbol{x}')) + \epsilon'^2\right]$$

$$= G + \sigma^2$$

## Extra-Sample Method (cont.)<sup>81</sup>

$$\widehat{G}_{extra} = \left(\widehat{f}(\boldsymbol{x}') - \boldsymbol{y}'\right)^2$$
$$\widehat{f} \longleftarrow \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$$

\$\heta\_{extra}\$ may be used for model selection.
 However, in practice, such an extra example is not available (or if we have it, it should be included in the original training set!).

### Holdout Method

- Idea: Use one of the training samples as an extra sample
  - Train a learning machine using  $\{(x_i, y_i)\}_{i \neq j}$  $\hat{f}_j(x) \leftarrow \{(x_i, y_i)\}_{i \neq j}$
  - Test its prediction performance using the holdout sample  $(x_j, y_j)$ :

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

# Holdout Method (cont.)

83

### Suppose $\{(x_i, y_i)\}_{i=1}^n$ satisfies:

$$\begin{aligned} \boldsymbol{x}_i \stackrel{i.i.d.}{\sim} q(\boldsymbol{x}) & \mathbb{E}_{\epsilon_j} [\epsilon_i] = 0 \\ y_i &= f(\boldsymbol{x}_i) + \epsilon_i \\ & \mathbb{E}_{\epsilon_i} \mathbb{E}_{\epsilon_j} [\epsilon_i \epsilon_j] = \begin{cases} \sigma^2 & (i = j) \\ 0 & (i \neq j) \end{cases} \end{aligned}$$

Holdout method is almost unbiased w.r.t.  $x_j, \epsilon_j$ :

$$\mathbb{E}_{\boldsymbol{x}_j} \mathbb{E}_{\epsilon_j} [\widehat{G}_j] = G_j + \sigma^2 \approx G + \sigma^2$$

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

 $\widehat{f}_j(\boldsymbol{x}) \approx \widehat{f}(\boldsymbol{x})$  if *n* is large

However,  $\widehat{G}_j$  is heavily affected by the choice of the holdout sample  $(x_j, y_j)$ .

## Leave-One-Out Cross-Validation<sup>84</sup>

Idea: Repeat the holdout procedure for all combinations and output the average.

$$\widehat{G}_{LOOCV} = \frac{1}{n} \sum_{j=1}^{n} \widehat{G}_j$$

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

LOOCV is almost unbiased w.r.t.  $\{x_i, \epsilon_i\}_{i=1}^n$ :

$$\mathbb{E}_{\{\boldsymbol{x}_i\}_{i=1}^n} \mathbb{E}_{\{\epsilon_i\}_{i=1}^n} [\widehat{G}_{LOOCV}]$$
  
$$\approx \mathbb{E}_{\{\boldsymbol{x}_i\}_{i=1}^n} \mathbb{E}_{\{\epsilon_i\}_{i=1}^n} [G] + \sigma^2$$

### k-Fold Cross-Validation

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

$$\widehat{G}_{kCV} = \frac{1}{k} \sum_{j=1}^{k} \widehat{G}_{\mathcal{T}_{j}}$$
$$\widehat{G}_{\mathcal{T}_{j}} = \frac{1}{|\mathcal{T}_{j}|} \sum_{i \in \mathcal{T}_{j}} \left( \widehat{f}_{\mathcal{T}_{j}}(\boldsymbol{x}_{i}) - y_{i} \right)^{2}$$
$$\widehat{f}_{\mathcal{T}_{j}}(\boldsymbol{x}) \leftarrow \{(\boldsymbol{x}_{i}, y_{i}) \mid i \notin \mathcal{T}_{j}\}$$

k-fold is easier to compute and more stable than leave-one-out.

## Advantages of CV

86

Wide applicability: Almost unbiasedness of LOOCV holds for (virtually) any learning methods

Practical usefulness: CV has been 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

Number of folds:

It is often recommended to use k = 5, 10. However, how to optimally choose k is still open.

## Closed Form of LOOCV<sup>88</sup>

$$f_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \sum_{i=1}^{b} \alpha_{i} \varphi_{i}(\boldsymbol{x}) \qquad \min_{\boldsymbol{\alpha} \in \mathbb{R}^{b}} \left[ \sum_{i=1}^{n} \left( f_{\boldsymbol{\alpha}}(\boldsymbol{x}_{i}) - y_{i} \right)^{2} + \lambda \|\boldsymbol{\alpha}\|^{2} \right]$$

For a linear model trained by quadratically constrained least-squares, the LOOCV score can be expressed as

$$\widehat{G}_{LOOCV} = \frac{1}{n} \|\widetilde{\boldsymbol{H}}^{-1} \boldsymbol{H} \boldsymbol{y}\|^2$$

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

H :same diagonal as H but zero for off-diagonal

### Homework

 Prove the closed-form expression of leaveone-out cross-validation score for a linear model with quadratically constraint least-

squares: 
$$\widehat{G}_{LOOCV} = rac{1}{n} \|\widetilde{m{H}}^{-1} m{H} m{y}\|^2$$

#### Hint: Express $\hat{\alpha}_j$ in terms of $\hat{\alpha}$

- $\hat{\alpha}_j$ : Learned parameter without the j-th sample
- $\hat{\alpha}$  : Learned parameter with all samples.
- Key formula:

$$(U - uu^{\top})^{-1} = U^{-1} + \frac{U^{-1}uu^{\top}U^{-1}}{1 - u^{\top}U^{-1}u}$$

# Homework (cont.)

90

2. For your own toy 1-dimensional data, perform simulations using

- Gaussian kernel models
- Quadratically-constrained least-squares learning and optimize
  - Width of Gaussian kernel
  - Regularization parameter

based on cross-validation. Analyze the results when changing

- Target function
- Number of samples
- Noise level