BCMI
Brain-like Computing
&
Machine Intelligence

Model Assessment & Selection

Dept. Computer Science & Engineering,
Shanghai Jiao Tong University

Outline

- Bias, Variance and Model Complexity
- The Bias-Variance Decomposition
- Optimism of the Training Error Rate
- Estimates of In-Sample Prediction Error
- The Effective Number of Parameters
- The Bayesian Approach and BIC
- Minimum Description Length
- Vapnik-Chernovenkis Dimension
- Cross-Validation
- Bootstrap Methods

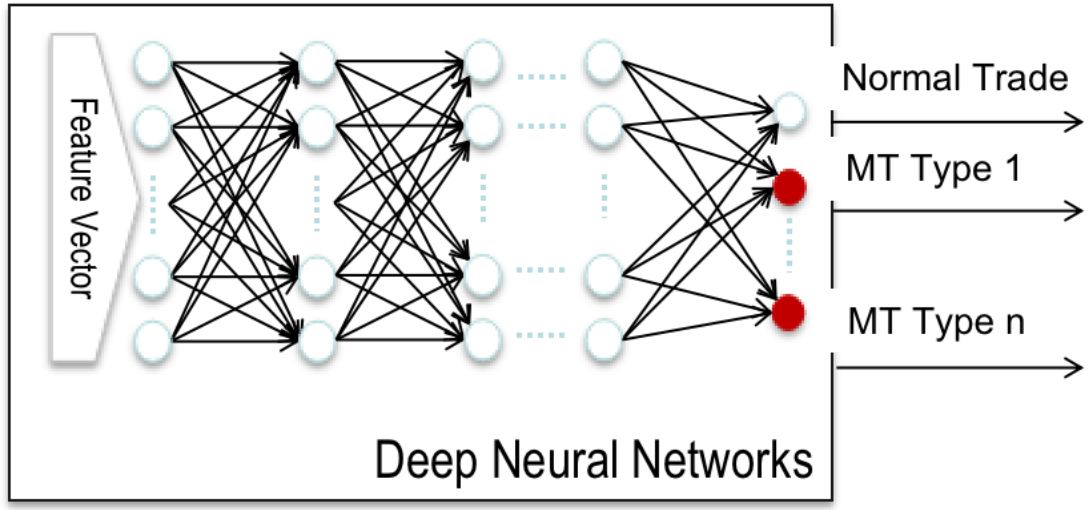
Objective of This Chapter

- To grasp the concept of model selection and assessment
- To derive criterions for model selection
  - In-sample error
- What are the most popular model selection criteria
  - AIC; BIC; MDL; VC
- CV for model selection

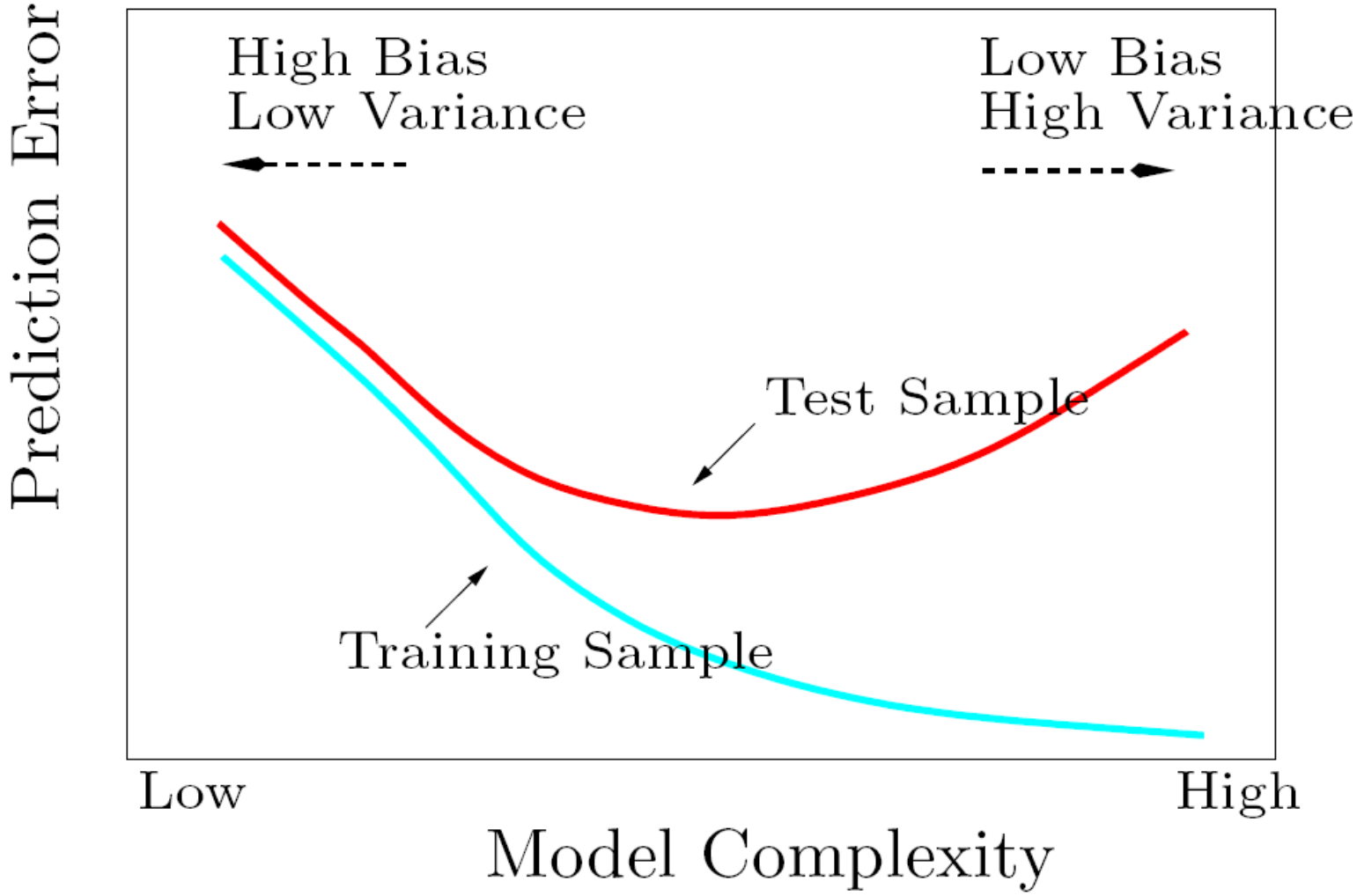
- Bootstrap method

Model Selection

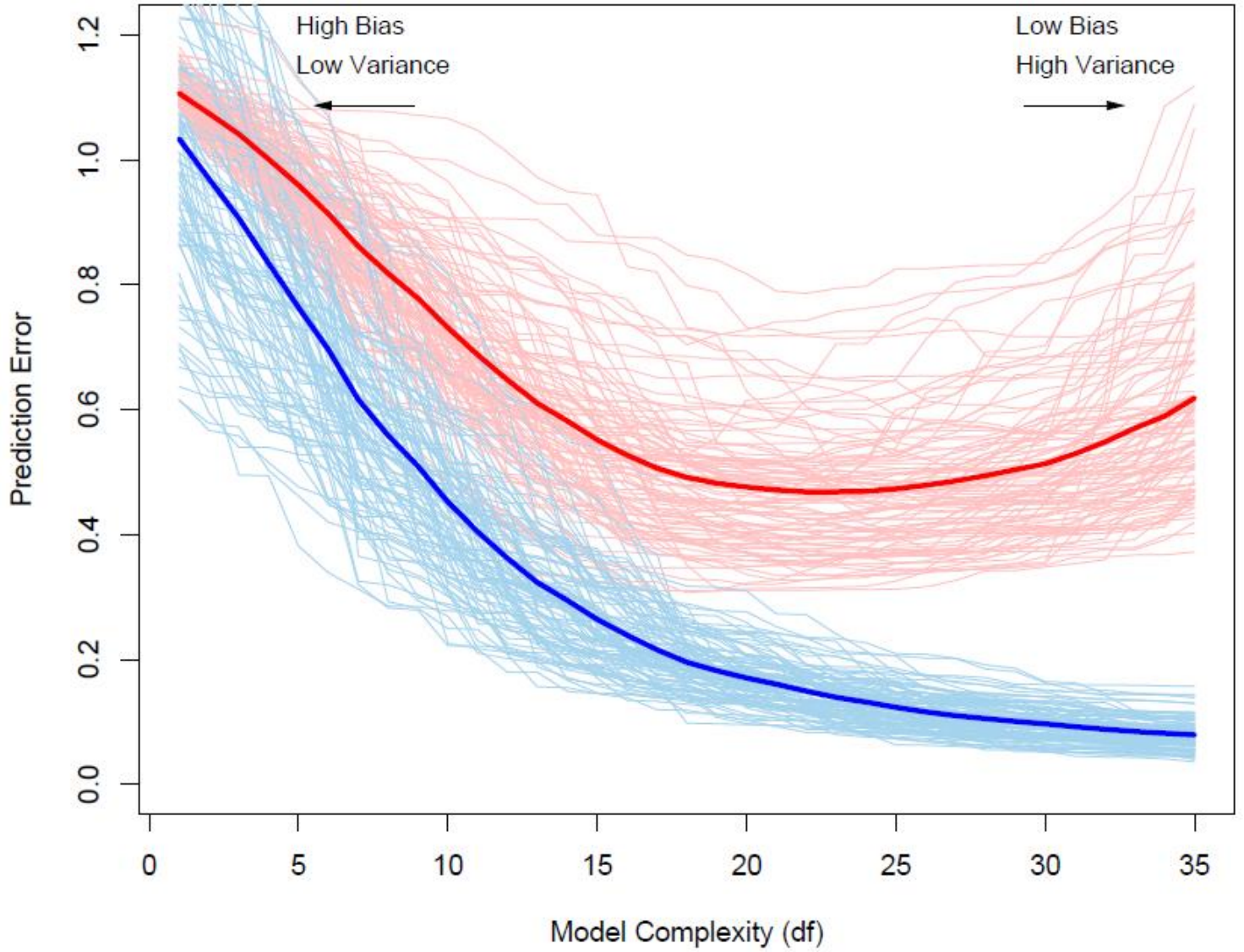
- To estimate the performance of different models in order to choose the best one.
- What is the parameters for model selection
- Example: CNN – parameter for model selection
  - The number of layers?
  - The number of neurons in each layer?
  - The activation function?
  - The size of convolution kernels?



Bias, Variance & Model Complexity



Bias, Variance & Model Complexity



Bias, Variance & Model Complexity

- The standard of model assessment : the generalization performance of a learning method
  - Model: $X \to Y;\ Y = f(X) + \varepsilon$
  - Prediction Model: $\hat{f}(X)$
  - Loss function:

$$L(Y, \hat{f}(X)) = \begin{cases} (Y - \hat{f}(X))^2 & \text{squared error} \\ |Y - \hat{f}(X)| & \text{absolute error} \end{cases}$$

Bias, Variance & Model Complexity

- Error: training error, generalization error

$$\text{Training error}: \ \overline{err} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(x_i))$$

$$\text{Generalization error}: \ Err = E\left[L(Y, \hat{f}(X))\right]$$

- Typical loss function:

$$0\text{-}1 \text{ loss} \quad L(G, \hat{G}(X)) = I(G \neq \hat{G}(X))$$

$$\log\text{-likelihood} \quad L(G, \hat{p}(X)) = -2\sum_{k=1}^{K} I(G = k)\log \hat{p}_k(X)$$

$$= -2\log \hat{p}_G(X)$$

Two Tasks in Model Selection

- Model selection:
  - estimating the performance of different models in order to choose the best one.
- Model assessment:
  - having chosen a final model, estimating its prediction error (generalization error) on new data.

Bias-Variance Decomposition

- Basic Model: $Y = f(X) + \varepsilon, \quad \varepsilon \sim N(0, \sigma_\varepsilon^2)$

- The expected prediction error of a regression fit $\hat{f}(X)$.

$$
\begin{aligned}
Err(x_0) &= E[(Y - \hat{f}(x_0))^2 \mid X = x_0] \\
&= \sigma_\varepsilon^2 + [E\hat{f}(x_0) - f(x_0)]^2 + E[E\hat{f}(x_0) - \hat{f}(x_0)]^2 \\
&= \sigma_\varepsilon^2 + Bias(\hat{f}(x_0))^2 + Var(\hat{f}(x_0)) \\
&= \text{Irreducible Error} + \text{Bias}^2 + \text{Variance}
\end{aligned}
$$

- The more complex the model, the lower the (squared) bias but the higher the variance.

Bias-Variance Decomposition

- For the k-NN regression fit the prediction error:

$$Err(x_0) = E[(Y - \hat{f}(x_0))^2 | X = x_0]$$

$$= \sigma_\varepsilon^2 + [f(x_0) - \frac{1}{k}\sum_{j=1}^{k} f(x_j)]^2 + \sigma_\varepsilon^2 / k$$

$$\hat{f}(x_0) = \frac{1}{k} \sum_{x_j \in N(x_0)} y_j,$$

- The in-sample error of the Linear Model

$$\frac{1}{N}\sum_{i=1}^{N} Err(x_i) = \sigma_\varepsilon^2 + \frac{1}{N}\sum_{i=1}^{N}[f(x_i) - E\hat{f}(x_i)]^2 + \frac{p}{N}\sigma_\varepsilon^2$$

  - The model complexity is directly related to the number of parameters p.

Bias-Variance Decomposition

- For ridge regression the square bias

$$\beta_* = \text{var}\max\ L(\beta) = E_x\left[\, f(x) - \beta^T x \right]^2$$

$$\frac{\partial L(\beta)}{\partial \beta} = 2E_x\left[(f(x) - \beta^T x)x \right] = 0$$

$$E_{x_0}\left[\, f(x_{x_0}) - E\hat{f}_\alpha(x_0) \right]^2 = E_{x_0}\left[\, f(x_0) - \beta_*^T x_0 \right]^2 + E_{x_0}\left[\, \beta_*^T x_0 - E\hat{\beta}_\alpha^T x_0 \right]^2$$

$$= [\text{Model Bias}]^2 + [\text{Estimation Bias}]^2$$

Bias-Variance Decomposition

- For ridge regression the square bias

$$\beta_* = \text{var max}\ \ L(\beta) = E_x\left[\, f(x) - \beta^T x \right]^2$$

$$\frac{\partial L(\beta)}{\partial \beta} = 2E_x\left[(f(x) - \beta^T x)x \right] = 0$$

$$E_{x_0}\left[(f(x_0) - \beta_*^T x_0)(\beta_*^T x_0 - E\left[\hat{\beta}_\alpha^T\right] x_0) \right]$$

$$= E_x\left[(f(x) - \beta_*^T x)(\beta_*^T x - E\left[\hat{\beta}_\alpha^T\right] x) \right]$$

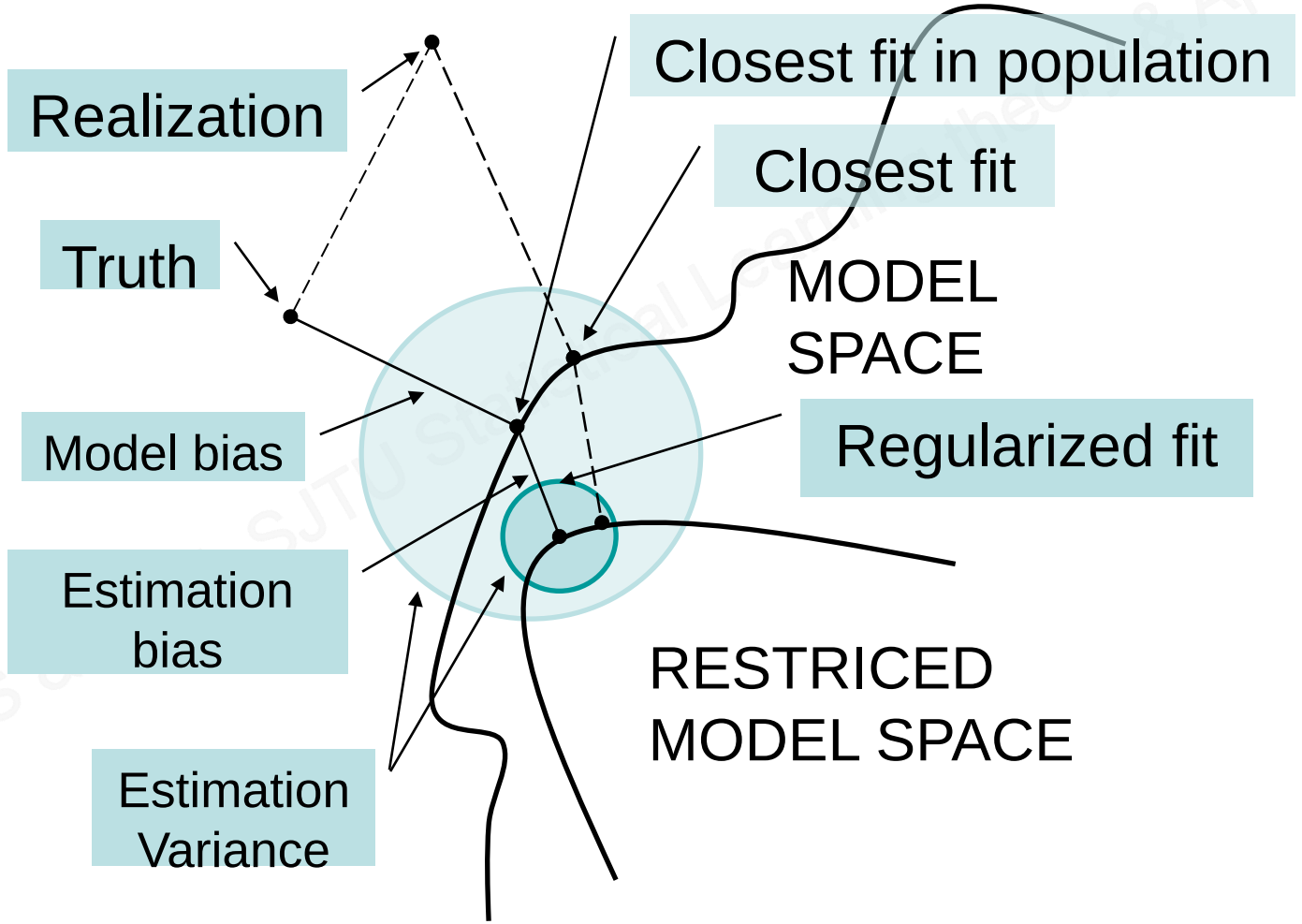
$$= \left(\beta_*^T - E\left[\hat{\beta}_\alpha^T\right]\right) E_x\left[(f(x) - \beta_*^T x)x \right] = 0$$

$$E_{x_0}\left[\, f(x_{x_0}) - E\hat{f}_\alpha(x_0) \right]^2 = E_{x_0}\left[\, f(x_0) - \beta_*^T x_0 \right]^2 + E_{x_0}\left[\, \beta_*^T x_0 - E\hat{\beta}_\alpha^T x_0 \right]^2$$

$$= [\text{Model Bias}]^2 + [\text{Estimation Bias}]^2$$

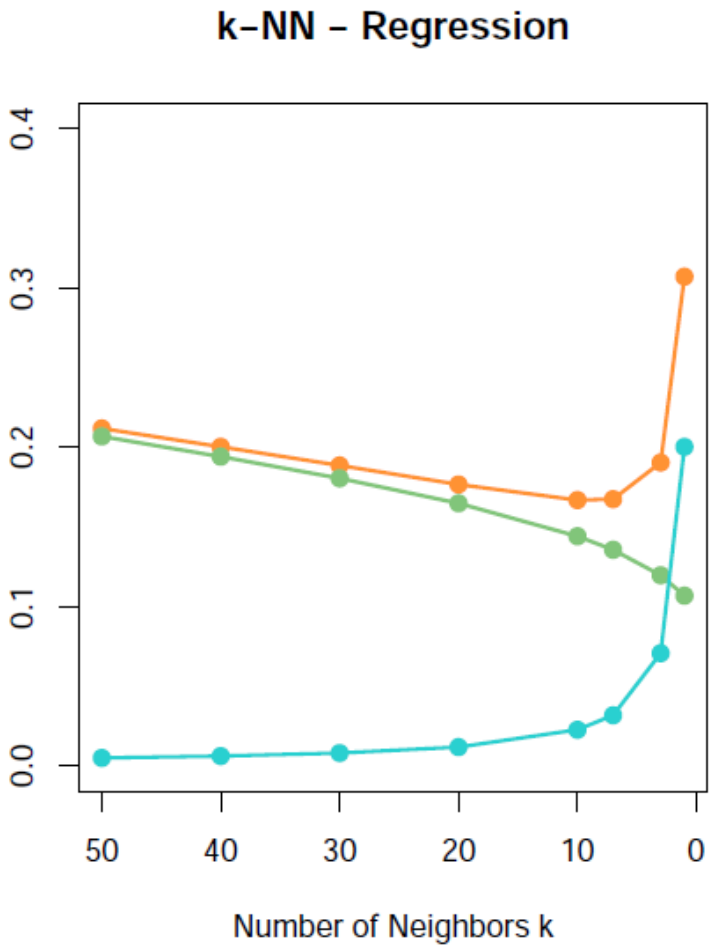
Bias-Variance Decomposition

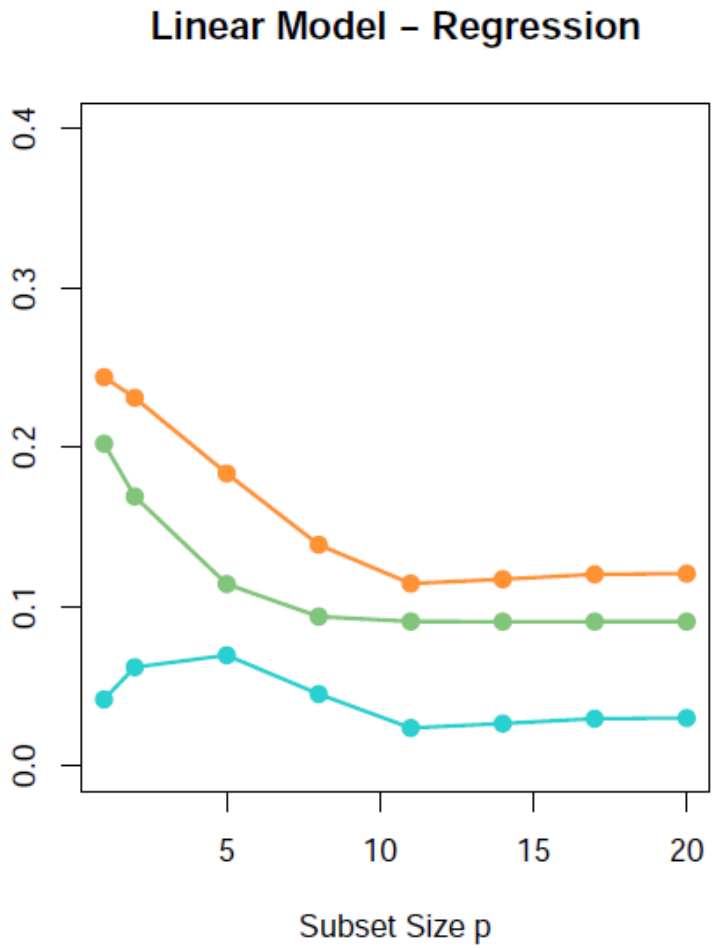
- Schematic of the behavior of bias and variance



Example: Bias–Variance Tradeoff

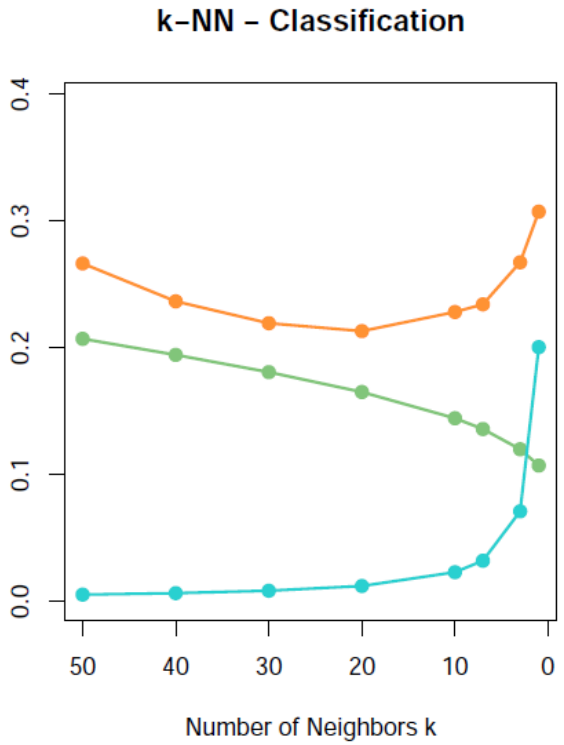
- There are 80 observations and 20 predictors, uniformly distributed in the hypercube $[0, 1]^{20}$.
- Left panels: Y is 0 if $X_1 \leq 1/2$, and 1 otherwise, and we apply k-nearest neighbors.
- Right panels: *Y is 1* if $\sum_{j=1}^{10} X_j \geq 5$, and *0* otherwise. We use best subset linear regression of size p.

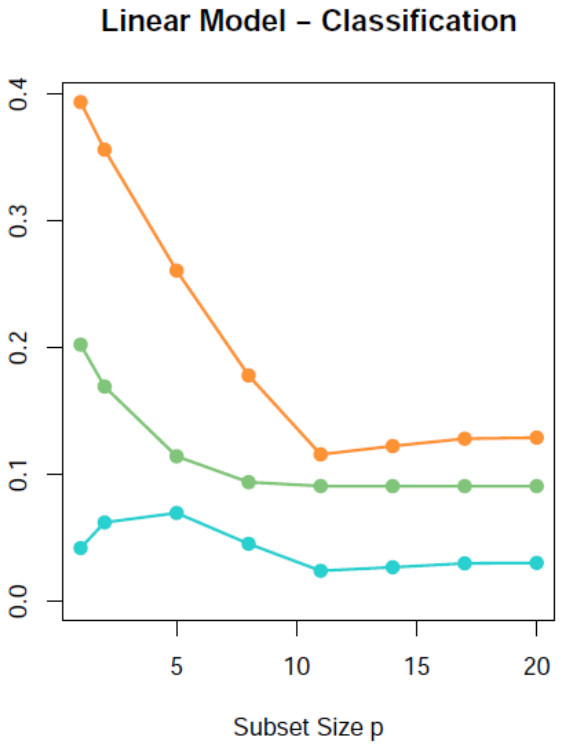




- Regression with squared error loss
  - The prediction error (red), squared bias (green) and variance (blue)
- Minima at about $k = 5$ for k-nearest neighbors, and $p \geq 10$ for the linear model.

- **Classification loss**
  - The prediction error (red), squared bias (green) and variance (blue)
- Prediction error is no longer the sum of squared bias and variance.
- For KNN, prediction error decreases or stays the same as the number of neighbors is increased to 20
- For the linear model classifier, the minimum occurs for p ≥ 10 as in regression, but the improvement over the p = 1 model is more dramatic.

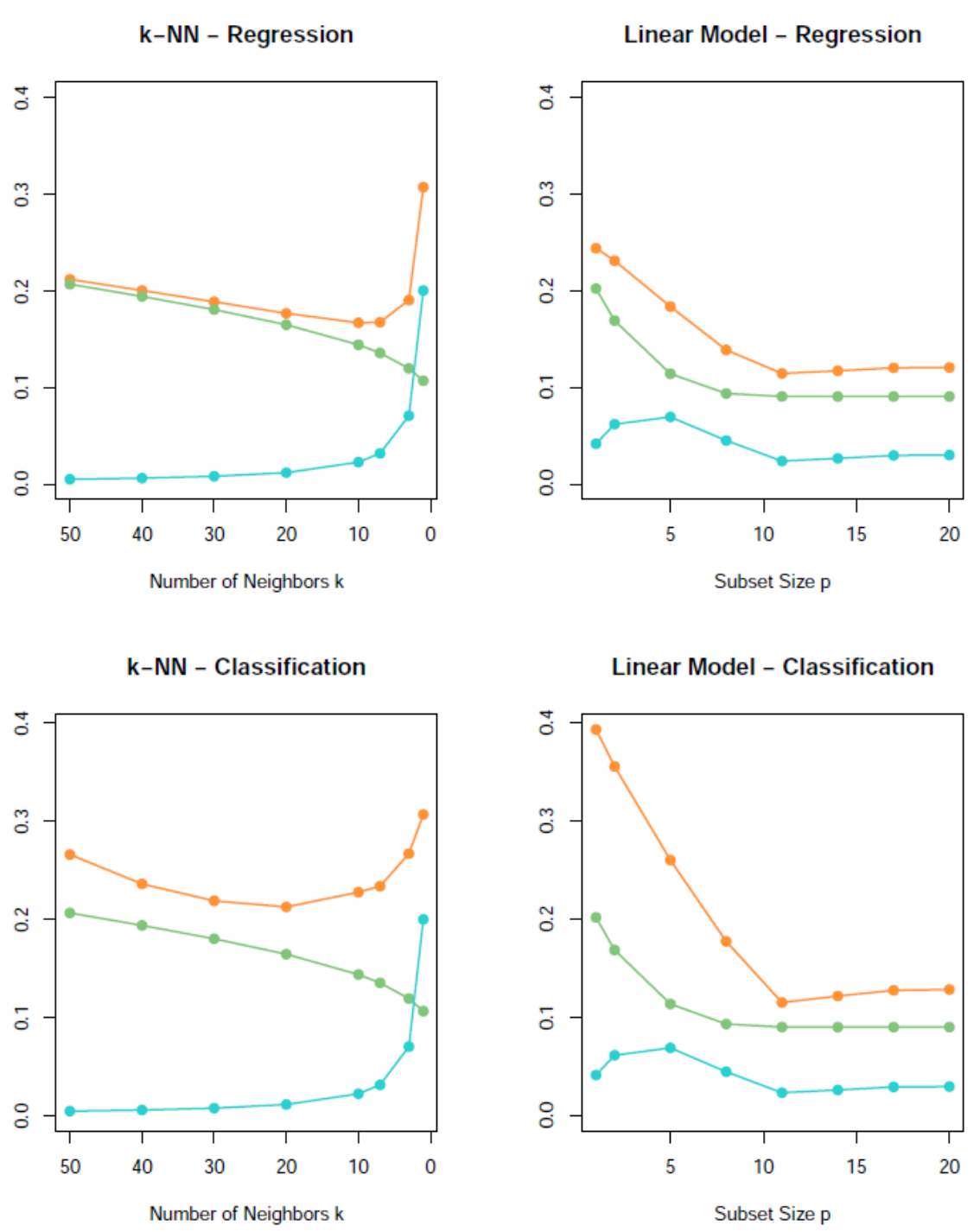




- The bias–variance tradeoff behaves differently for 0–1 loss than it does for squared error loss.

- Assume
  - The true probability of class 1 is 0.9
  - The expected value of our estimate is 0.6.

- The squared bias $(0.6-0.9)^2$ which is considerable
- The prediction error is zero since we make the correct decision.



Outline

- Bias, Variance and Model Complexity
- The Bias-Variance Decomposition
- Optimism of the Training Error Rate
- Estimates of In-Sample Prediction Error
- The Effective Number of Parameters
- The Bayesian Approach and BIC
- Minimum Description Length
- Vapnik-Chernovenkis Dimension
- Cross-Validation
- Bootstrap Methods

Optimism of the Training Error Rate

- Training Error < True Error

$$\text{Training Error} \quad \overline{err} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(x_i))$$

$$\text{True Error} \quad Err = E[L(Y, \hat{f}(X)]$$

- Err is extra-sample error
- The in-sample error

$$Err_{in} = \frac{1}{N}\sum_{i=1}^{N} E_{Y^{new}}\left[L(Y_i^{new}, \hat{f}(x_i)) | \mathbb{T} \right]$$

- Optimism:

$$op \equiv Err_{in} - \overline{err}$$

Optimism of the Training Error Rate

- The average optimism is the expectation of the optimism over training sets

$$\omega \equiv \mathrm{E}_y(op) \equiv \mathrm{E}_y(Err_{in} - \overline{err})$$

Training Error $\quad \overline{err} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(x_i))$

In sample error: $\quad Err_{in} = \frac{1}{N}\sum_{i=1}^{N} E_{Y^{new}}\left[L(Y_i^{new}, \hat{f}(x_i)) | \mathbb{T} \right]$

$$\omega = E_y(op) = E_y\left[Err_{in} - \overline{err}\right]$$

$$= \frac{1}{N}\sum_{i=1}^{N} E_{Y^{new}} E_y\left[(Y_i^{new} - \hat{f}(x_i))^2 - (y_i - \hat{f}(x_i))^2\right]$$

$$(Y_i^{new} - \hat{f}(x_i))^2 - (y_i - \hat{f}(x_i))^2$$

$$= (Y_i^{new})^2 - 2Y_i^{new}\hat{f}(x_i) - y_i^2 + 2y_i\hat{f}(x_i)$$

$$\hat{y}_i = \hat{f}(x_i) \qquad E_{Y^{new}}(Y_i^{new})^2 = E_y y_i^2$$

$$E_{Y^{new}} E_y\left[(Y_i^{new} - \hat{f}(x_i))^2 - (y_i - \hat{f}(x_i))^2\right]$$

$$= 2E[y_i, \hat{y}_i] - 2E\hat{y}_i Ey_i = 2Cov(\hat{y}_i, y_i)$$

Optimism of the Training Error Rate

- For squared error, 0-1, other loss function：

$$\omega=\mathrm{E}_y(op)=\frac{2}{N}\sum_{i=1}^{N}Cov(\hat{y}_i,y_i)$$

$$Err_{in}=E_y\left(\overline{err}\right)+\frac{2}{N}\sum_{i=1}^{N}Cov(\hat{y}_i,y_i)$$

- $\hat{y}_i$ is obtained by a linear fit with d inputs or basis function, a simplification is:

$$\sum_{i=1}^{N}\mathrm{cov}(\hat{y}_i,y_i)=d\sigma_\varepsilon^2$$

$$Err_{in}=E_y\left(\overline{err}\right)+2\frac{d}{N}\sigma_\varepsilon^2$$

- If the dimension / the number of Basis Functions increases, Optimism will increase too.
- If the number of training samples, Optimism will decrease

In-sample Prediction Error

- The general form of the in-sample estimates is

$$\hat{Err}_{in} = E_y[\overline{err}] + \hat{o}p$$

- d parameters are fit under Squared error loss

$$C_p \text{ statistic : } C_p = \overline{err} + 2\frac{d}{N}\hat{\sigma}_{\varepsilon}^2$$

- Use a log-likelihood function to estimate Err_{in}

$$N \to \infty, \quad -2E\left[\log \Pr_{\theta}(Y)\right] \approx -\frac{2}{N}E\left[loglik\right] + 2\frac{d}{N}$$

$$loglik = \sum_{i=1}^{N} \log \Pr_{\theta}(y_i)$$

  - This relationship introduces the Akaike Information Criterion

Akaike Information Criterion

- Akaike Information Criterion is a similar but more generally applicable estimate of Err_{in}
- A set of models $f_\alpha(x)$ with a turning parameter α :

$$\mathrm{AIC}(\alpha) = \overline{err}(\alpha) + 2\frac{d(\alpha)}{N}\hat{\sigma}_\varepsilon^2$$

$\overline{err}(\alpha)$:the training error; $d(\alpha)$: number of parameters

- $\mathrm{AIC}(\alpha)$ provides an estimate of the test error curve, and we find the turning parameter $\hat{\alpha}$ that minimizes it.

$$\left\{ f_{\hat{\alpha}}(x) \,|\, \hat{\alpha} : \min \mathrm{AIC}(\hat{\alpha}) \right\}$$

Akaike Information Criterion

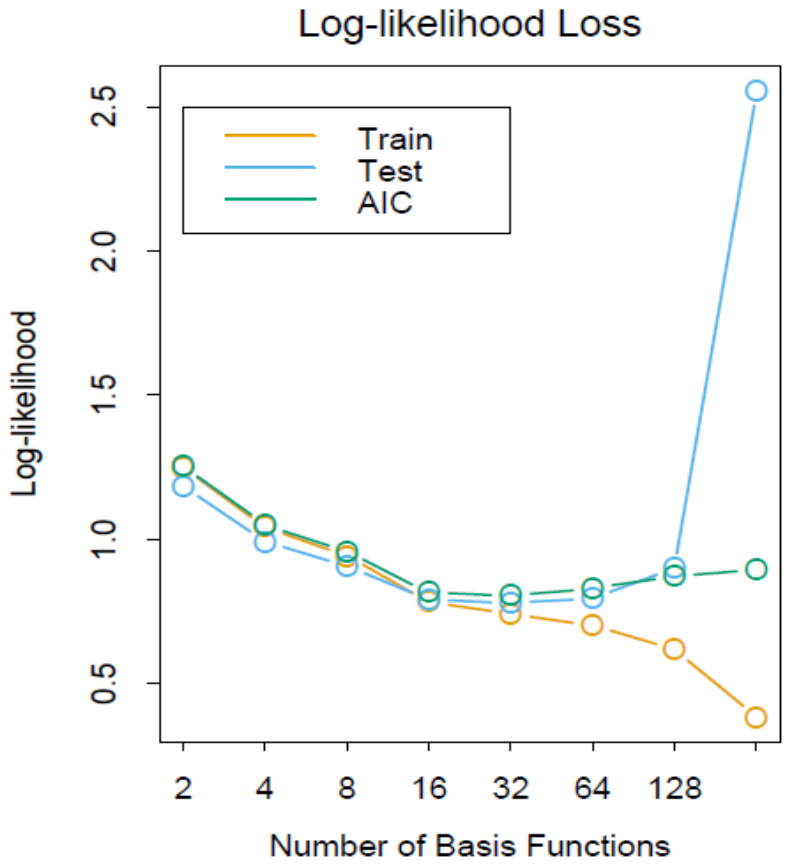
- For the logistic regression model, using the binomial log-likelihood.

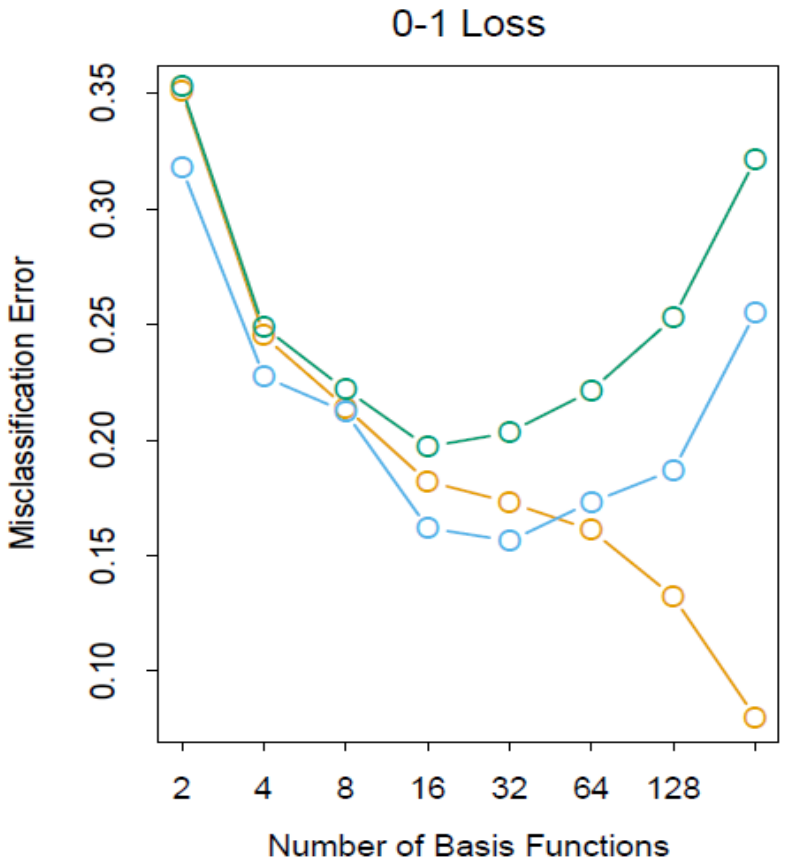
$$\mathrm{AIC} = -\frac{2}{N} E[\log lik] + 2\frac{d}{N}$$

- For Gaussian model the **AIC** statistic equals to the $\mathbf{C_p}$ statistic.

$$\mathrm{AIC} = C_p = \overline{err} + 2\frac{d}{N}\hat{\sigma}_\varepsilon^2$$

Phoneme Recognition (AIC)





- Data: Spoken Vowels

Outline

- Bias, Variance and Model Complexity
- The Bias-Variance Decomposition
- Optimism of the Training Error Rate
- Estimates of In-Sample Prediction Error
- The Effective Number of Parameters
- The Bayesian Approach and BIC
- Minimum Description Length
- Vapnik-Chernovenkis Dimension
- Cross-Validation
- Bootstrap Methods

Effective number of parameters

- A linear fitting method:

$$\hat{y} = Sy, \quad S \text{ is } N \times N \text{ matrix, depending on } x_i$$

- Effective number of parameters:

$$d(S) = \text{trace}(S)$$

  - If S is an orthogonal projection matrix onto a basis set spanned by M features, then:

$$\text{trace}(S) = M$$

  - $\text{trace}(S) = M$ is the correct quantity to replace d in the C_p statistic

Bayesian Approach & BIC

- The Bayesian Information Criterion (BIC)

$$\mathrm{BIC} = -2loglik + (\log N)d$$

- Gaussian model:
  - Variance σ_ε^2
  - then

$$-2loglik = C\sum_{i=1}^{N}(y_i - \hat{f}(x_i))^2 / \sigma_\varepsilon^2 = N \cdot \overline{err} / \sigma_\varepsilon^2$$

$$\mathrm{BIC} = \frac{N}{\sigma_\varepsilon^2}[\overline{err} + (\log N)\frac{d}{N}\sigma_\varepsilon^2]$$

  - BIC is proportional to $\mathrm{AIC}(\mathrm{C_p})$, 2 replaced by $\log N$
  - $N > e^2 \approx 7.4,$ BIC tends to choose simple model

Bayesian Model Selection

- BIC derived from Bayesian Model Selection
- Candidate models $\mathcal{M}_m$, model parameter θ_m and a prior distribution $\Pr(\theta_m \mid M_m)$
- Posterior probability:

$$\Pr(M_m \mid \mathrm{Z}) \propto \Pr(M_m)\Pr(\mathrm{Z} \mid M_m)$$
$$\propto \Pr(M_m)\int \Pr(\mathrm{Z} \mid \theta_m, M_m)\Pr(\theta_m \mid M_m)d\theta_m$$

- Z represents the training data $\{x_i, y_i\}_1^N$

Bayesian Model Selection

- Compare two models M_m and M_ℓ

$$\frac{\Pr(M_m \mid \mathrm{Z})}{\Pr(M_\ell \mid \mathrm{Z})} = \frac{\Pr(M_m)}{\Pr(M_\ell)} \frac{\Pr(\mathrm{Z} \mid M_m)}{\Pr(\mathrm{Z} \mid M_\ell)}$$

- If the odds(比率) are greater than 1, model m will be chosen, otherwise choose model ℓ
- Bayes Factor:

$$\mathrm{BF}(\mathrm{Z}) = \frac{\Pr(\mathrm{Z} \mid M_m)}{\Pr(\mathrm{Z} \mid M_\ell)}$$

  - The contribution of the data to the posterior odds

Bayesian Model Selection

- If the model prior Pr(M) is uniformly distributed,

$$\log \Pr(Z \mid M_m) = \log \Pr(Z \mid\mid \hat{\theta}_m, M_m) - \frac{d_m}{2} \log N + O(1)$$

where $\hat{\theta}_m$ is maximum likelihood estimator,

d_m is the model dimension.

The Loss Function:

$$-2 \log \Pr(Z \mid\mid \hat{\theta}_m, M_m)$$

Minimizing BIC is equivalent to maximizing posterior

Advantage: When the true model is included in the model family, the number of samples tends to infinite, BIC selects the true model with probability one

Minimum Description Length(MDL)

- Problem：Optimal Coding
  - Message: $z1$ $z2$ $z3$ $z4$
  - Coding： 0 10 110 111
  - Coding2： 110 10 111 0
- **Criterion：** Using the shortest coding length for the most frequent message.
- The probability of zi ： $Pr(z_i)$
- The coding length ： $l_i = -log\, Pr(z_i)$

Minimum Description Length(MDL)

$$E(\text{Length}) \geq -\sum \Pr(z_i)\log \Pr(z_i)$$

The equality holds if and only if $p_i = A^{-l_i}$。

$$\Pr(z_i) = 1/2;\ \ 1/4;\ \ 1/8;\ \ 1/8$$

Minimum Description Length(MDL)

Model: M; Parameter: θ; Input Output $Z = (X, y)$

Suppose the Conditional Probabilty function of y

$$p(y \mid \theta, M, X)$$

$$\text{length} = -\log \Pr(y \mid \theta, M, X) - \log \Pr(\theta \mid M)$$

$-\log \Pr(y \mid \theta, M, X)$: the average code length for transmitting the discrepancy between the model and actual target values

$-\log \Pr(\theta \mid M)$: the average code length for transmitting the model parameters

Minimum Description Length(MDL)

Assume $y \sim N(\theta, \sigma^2)$, and parameter $\theta \sim N(0,1)$

$$\text{Length} = \text{Const.} + log\,\sigma + \frac{(y-\theta)^2}{\sigma^2} + \frac{\theta^2}{2}$$

=> the smaller σ is, the shorter on average is the message length, since y is more concentrated around θ.

MDL Principle： Select a model, minimizing

$$\text{length} = -log\,Pr(y \mid \theta, M, x) - log\,Pr(\theta \mid M).$$

where model M is described by the parameter θ

Outline

- Bias, Variance and Model Complexity
- The Bias-Variance Decomposition
- Optimism of the Training Error Rate
- Estimates of In-Sample Prediction Error
- The Effective Number of Parameters
- The Bayesian Approach and BIC
- Minimum Description Length
- Vapnik-Chernovenkis Dimension
- Cross-Validation
- Bootstrap Methods

Vapnik-Chernovenkis Dimension

- Problem：How to select the model dimension d? which describes the model complexity
- VC dimension is a critical index describing the model complexity

$$\text{Function family:} \quad \{f(x,\alpha)\}, x \in IR^p$$

$$\alpha -- \text{parameter}, \ f -- \text{index function}$$

$$\text{Example}: \ \alpha = (\alpha_0, \ \alpha_1) \ , \ \text{linear function family}$$

$$f = I(\alpha_0 + \alpha^T{}_1 x > 0); \qquad f \text{ complexity: p} + 1$$

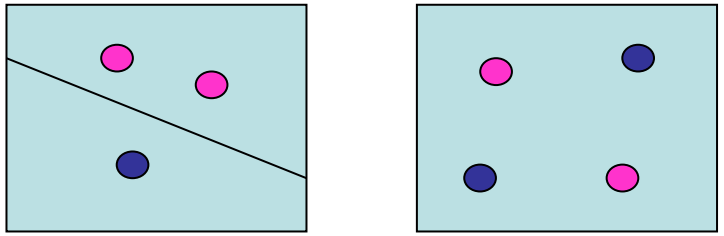
Another example $x \in IR, f(x,\alpha) = I(\sin\alpha \cdot x)$?

f has only one parameter。

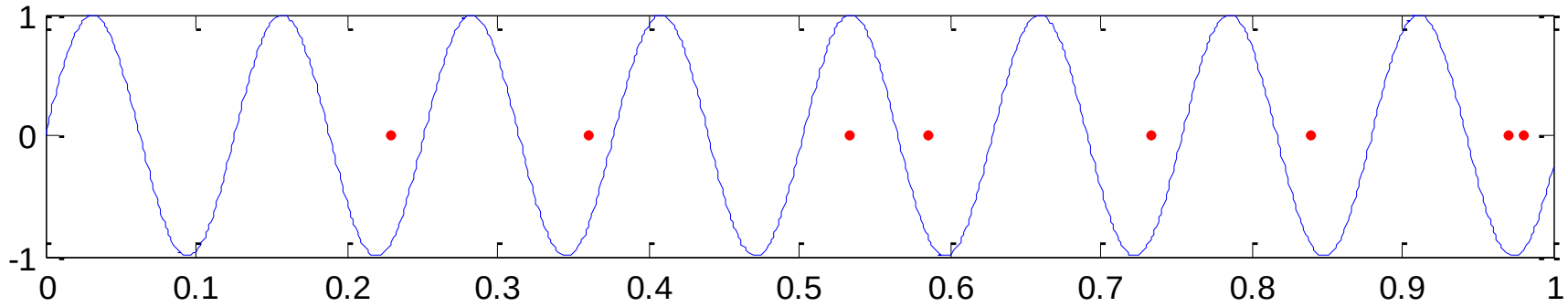
VC dimension

- The VC dimension of function family $\{f(x,\alpha)\}$ is defined to be the largest number *of points that can be* shattered *by members of*

$$\{f(x,\alpha)\}$$



- The VC dimension of 2d linear function family is 3.
- $\{\sin(\alpha x)\}$ its VC dimension is infinite.



VC Dimension

- The VC dimension of a class of real-valued functions $\{g(x,\alpha)\}$ is defined to be the VC dimension of the indicator class $\{I(g(x,\alpha)-\beta>0)\}$ where β takes values over the range of g.

- Assume $\{g(x,\alpha)\}$ has VC dimension h, the sample number *N*.

$$Err \le \overline{err} + \frac{\varepsilon}{2}\left(1+\sqrt{1+\frac{4\overline{err}}{\varepsilon}}\right) \quad \text{二类分类}$$

$$Err \le \frac{\overline{err}}{(1-c\sqrt{\varepsilon})_+};\text{回归}\quad \varepsilon = a_1\frac{h[\log(a_2 N/h+1)-\log(\eta/4)}{N}$$

$$0 < a_1 \le 4,\ 0 < a_2 \le 2$$

Cherkassky and Mulier (2007, pages 116–118)

Cross Validation

1	2	3	4	5
Training	Training	Testing Set	Training	Training

Cross Validation

- Denote the fitted function by $\hat{f}^{-k}(x)$ with removing k-th fold data. Then the cross-validation estimate of prediction error is

$$\mathrm{CV}(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_{i,}\, \hat{f}^{-k}(x_i))$$

- The case K = N, *leave-one-out* cross-validation.
- Given model family $f(x,\alpha)$ indexed by a tuning parameter α.
- $\hat{f}^{-k}(x,\alpha)$ ： fited with the kth part of the data removed.

$$\mathrm{CV}(\hat{f},\alpha) = \frac{1}{N}\sum_{i=1}^{N} L(y_{i,}\, \hat{f}^{-k}(x_i,\alpha))$$

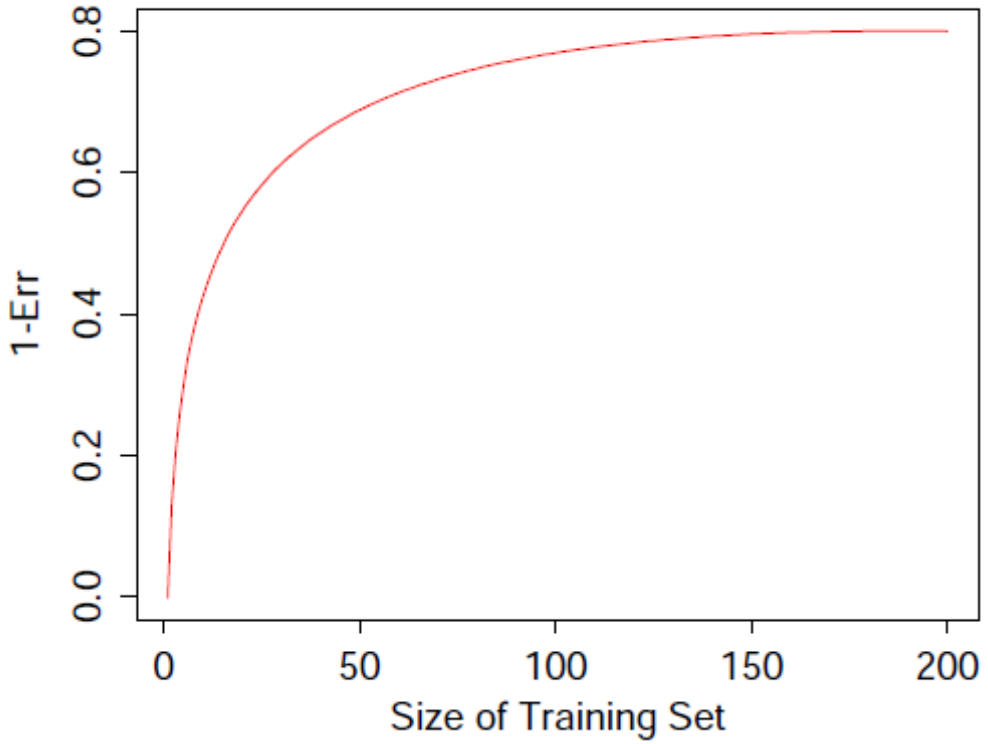


FIGURE 7.8. *Hypothetical learning curve for a classifier on a given task: a plot of* $1 - \text{Err}$ *versus the size of the training set* N*. With a dataset of* 200 *observations,* 5*-fold cross-validation would use training sets of size* 160*, which would behave much like the full set. However, with a dataset of* 50 *observations fivefold cross-validation would use training sets of size* 40*, and this would result in a considerable overestimate of prediction error.*

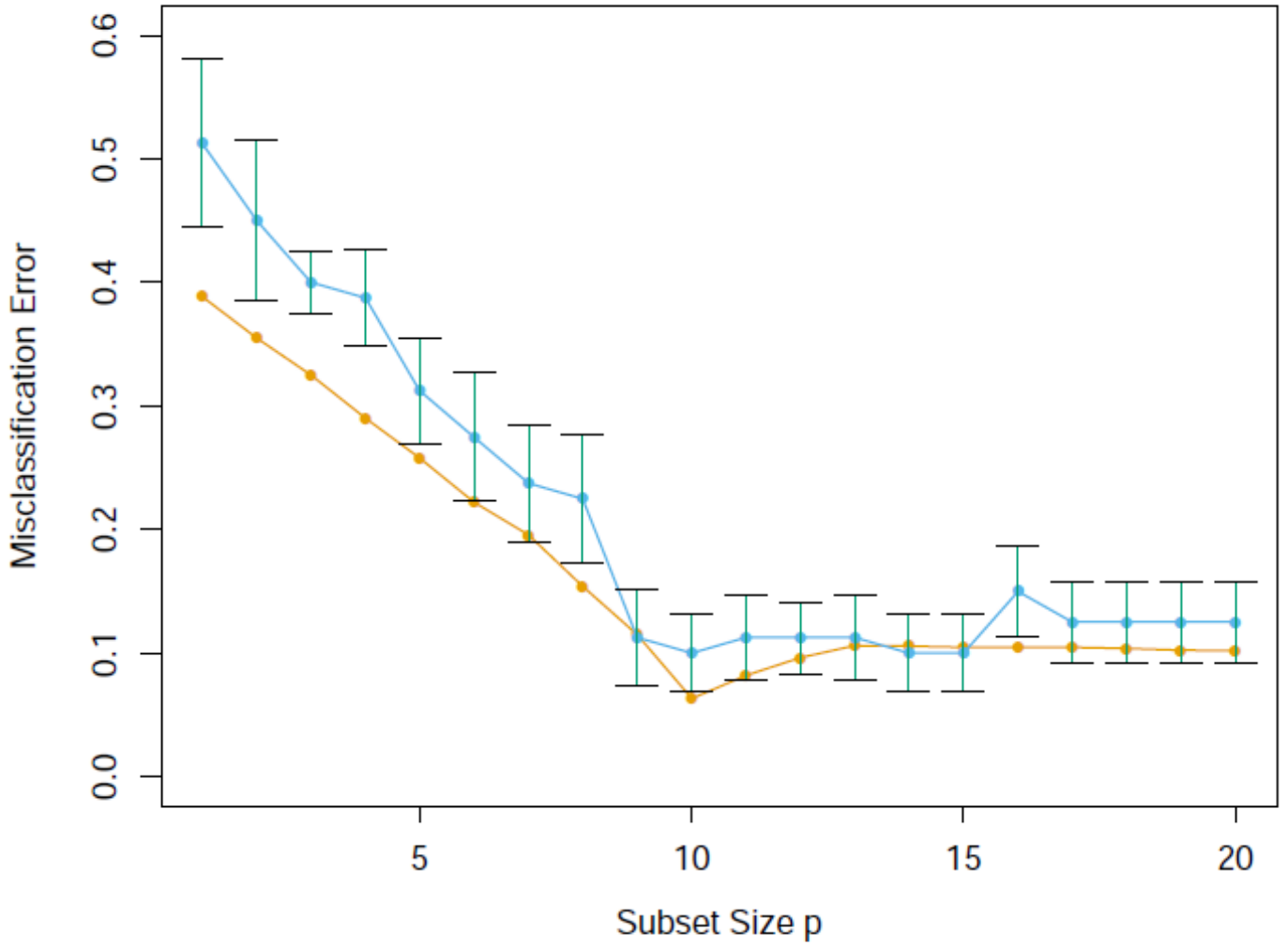


FIGURE 7.9. *Prediction error (orange) and tenfold cross-validation curve (blue) estimated from a single training set, from the scenario in the bottom right panel of Figure 7.3.*

Generalized cross-validation

- Linear fitting method：

$$\hat{y} = Sy$$

- For linear fitting methods,

$$\mathrm{CV}(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N}\left[y_i - \hat{f}^{-k}(x_i)\right]^2 = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{y_i - \hat{f}(x_i)}{1 - S_{ii}}\right]^2,$$

- where S_{ii} is the i-th diagonal element of S

$$\mathrm{GCV}(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{y_i - \hat{f}(x_i)}{1 - trace(S)/N}\right]^2.$$

Right Way to Do CV?

- A typical strategy for CV might be as follows:
  - Screen the predictors: find a subset of “good” predictors that show fairly strong (univariate) correlation with the class labels
  - Using just this subset of predictors, build a multivariate classifier.
  - Use cross-validation to estimate the unknown tuning parameters and to estimate the prediction error of the final model.
- Is this a correct application of cross-validation?

A Toy Model

- Consider a scenario with $N = 50$ samples in two equal-sized classes, and $p = 5000$ quantitative predictors (standard Gaussian) that are independent of the class labels.
- The true (test) error rate of any classifier is 50%.
- Step (1): the 100 predictors having highest correlation with the class labels, and then using a 1-nearest neighbor classifier, based on just these 100 predictors.
- Step (2): Over 50 simulations from this setting, the average CV error rate was 3%, far lower than the true error rate of 50%.

Right Way to Do CV?

- A typical strategy for CV might be as follows:
  - X Screen the predictors: find a subset of “good” predictors that show fairly strong (univariate) correlation with the class labels
  - Using just this subset of predictors, build a multivariate classifier.
  - Use cross-validation to estimate the unknown tuning parameters and to estimate the prediction error of the final model.

The Correct Way to Do CV

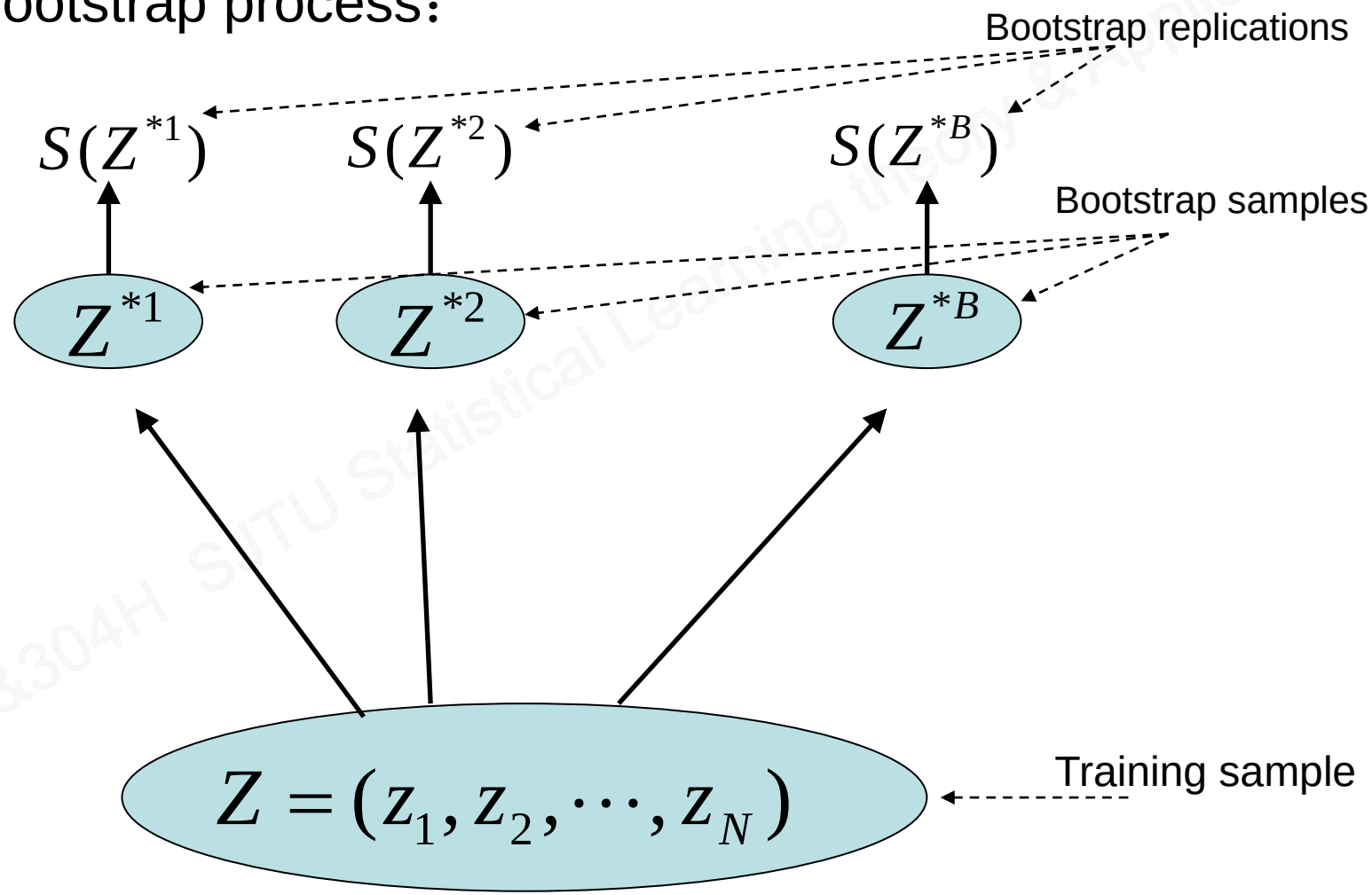
- A typical strategy for CV might be as follows:
  - Divide the samples into K cross-validation folds (groups) at random.
  - For each fold k = 1, 2, . . . ,K
    - Find a subset of “good” predictors that show fairly strong (univariate) correlation with the class labels, using all of the samples except those in fold k.
    - Using just this subset of predictors, build a multivariate classifier, using all of the samples except those in fold k.
    - Use the trained classifier to predict the class labels for the samples in fold k.

Outline

- Bias, Variance and Model Complexity
- The Bias-Variance Decomposition
- Optimism of the Training Error Rate
- Estimates of In-Sample Prediction Error
- The Effective Number of Parameters
- The Bayesian Approach and BIC
- Minimum Description Length
- Vapnik-Chernovenkis Dimension
- Cross-Validation
- Bootstrap Methods

Bootstrap Methods

- Schematic of the bootstrap process：



Bootstrap Methods

- Basic Idea：The basic idea is to randomly draw datasets with replacement from the training data, each sample the same size as the original training set。
- B times (B = 100 say), producing B bootstrap datasets

$$\hat{f}^{*b}(x_i) \text{ is the prediction function at } x_i$$

- The bootstrap error

$$Err_{boot} = \frac{1}{B}\frac{1}{N}\sum_{b=1}^{B}\sum_{i=1}^{N} L(y_i, \hat{f}^{*b}(x_i))$$

Summay

- Model selection and assessment
- How to derive model selection criterion
  - In-sample error
- What are the most popular model selection criteria
  - AIC; BIC; MDL; VC
- CV for model selection

- Bootstrap method

BCMI
Brain-like Computing
&
Machine Intelligence

The End!

SJTU