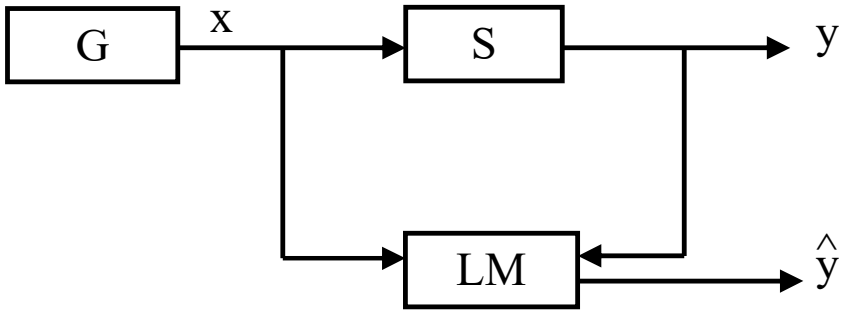
Statistical Learning Theory & Methods

Lecturer: Liqing Zhang

Dept. Computer Science & Engineering
Shanghai Jiao Tong University

Function Estimation Model



- **Key concepts:** $F(x,y)$, an i.i.d. k-sample on F, functions $f(x,\alpha)$ and the equivalent representation of each f using its index α

The Problem of Risk Minimization

- The **loss functional** (*L, Q*)
  - the error of a given function on a given example

$$L:(x,y,f_\alpha) \mapsto L(y,f(x,\alpha))$$
$$Q:(z,\alpha) \mapsto L(z_y,f(z_x,\alpha))$$

- The **risk functional** (*R*)
  - the expected loss of a given function on an example drawn from $F(x,y)$
  - the (usual concept of) generalisation error of a given function

$$R(\alpha)=\int Q(z,\alpha)dF(z)$$

The Problem of Risk Minimization

- Three Main Learning Problems
  - Pattern Recognition:

$$y \in \{0,1\} \text{ and } L(y, f(x,\alpha)) = \mathbf{1}[y = f(x,\alpha)]$$

  - Regression Estimation:

$$y \in \Re \text{ and } L(y, f(x,\alpha)) = (y - f(x,\alpha))^2$$

  - Density Estimation:

$$y \in [0,1] \text{ and } L(p(x,\alpha)) = -\log p(x,\alpha)$$

General Formulation

- The Goal of Learning
  - Given an i.i.d. k-sample $z_1,\ldots, z_k$ drawn from a fixed distribution $F(z)$
  - For a function class' loss functionals $Q(z, \alpha)$, with α in Λ
  - To **minimise the risk**, finding a function α^*

$$\alpha^* = \arg\min_{\alpha \in \Lambda} R(\alpha)$$

General Formulation

- The Empirical Risk Minimization (ERM) Inductive Principle
  - Define the **empirical risk** (sample/training error):

$$R_{\text{emp}}(\alpha) = \frac{1}{k}\sum_{i=1}^{k} Q(z_i, \alpha)$$

  - Define the empirical risk minimiser:

$$\alpha_k = \arg\min_{\alpha \in \Lambda} R_{\text{emp}}(\alpha)$$

  - ERM approximates $Q(z, \alpha^*)$ with $Q(z, \alpha_k)$ the R_{emp} minimiser…that is ERM approximates α^* with α_k
  - Least-squares and Maximum-likelihood are realisations of ERM

Key Points in SL

- **Curse of dimensionality**
  - How many samples are necessary for estimating one dimensional pdf in general
  - How many samples are good for 10-d pdf estimation
  - Pseudo orthogonality for any two high-dimensional vectors
- **Regression function**

$$EPE(f) = E[Y - f(X)]^2$$

极小解为：

$$f(x) = E\ (\mathrm{Y} \mid X = x)$$

Overview of the Course

- Introduction
- Overview of Supervised Learning
- Linear Method for Regression and Classification
- Basis Expansions and Regularization
- Kernel Methods
- Model Selections and Inference
- Support Vector Machine
- Unsupervised Learning
- Latent Dirichlet Allocation
- Deep Networks and Regularization

Key Points in SL

- Curse of dimensionality
  - How many samples are necessary for estimating one dimensional pdf in general
  - How many samples are good for 10-d pdf estimation
  - Pseudo orthogonality for any two high-dimensional vectors
- Regression function

$$EPE(f) = E[Y - f(X)]^2$$

极小解为：

$$f(x) = E\ (\mathrm{Y} \mid X = x)$$

Error Decomposition

$$
\begin{aligned}
EPE(x_0) &= E_T[f(x_0) - \hat{y}_0]^2 \\
&= E_T[f(x_0) - E[\hat{y}_0] + E[\hat{y}_0] - \hat{y}_0]^2 \\
&= E_T[\hat{y}_0 - E_T(\hat{y}_0)]^2 + E_T[E(\hat{y}_0) - f(x_0)]^2 \\
&\quad + 2E\{[\hat{y}_0 - E_T(\hat{y}_0)][E(\hat{y}_0) - f(x_0)]\} \\
&= Var_T(\hat{y}_0) + Bias^2(\hat{y}_0)
\end{aligned}
$$

Key Points in Linear Regression

- Model is $f(x_i) = \beta_0 + \sum_{j=1}^{P-1} x_{ij}\beta_j$
- or $\mathbf{F} = \mathbf{X}\beta$

Solution is $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T y$

$\hat{y} = \mathbf{X}\hat{\beta}$

Also $Var(\hat{\beta}) = (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2$

Key Points in Linear Regr.

- Solution is $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T y$

$$\hat{y} = \mathbf{X}\hat{\beta}$$

Also $Var(\hat{\beta}) = (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2$

- How to formulate the regularization?
  - Data
  - Model
  - Bootstrap

Shrinkage methods

Ridge regression

- The ridge estimator is defined by

$$\hat{\beta}^{ridge} = \arg\min(\mathrm{Y} - \mathbf{X}\beta)^T (Y - \mathbf{X}\beta) + \lambda\beta^T\beta$$

- Equivalently,

$$\hat{\beta}^{ridge} = \arg\min(\mathrm{Y} - \mathbf{X}\beta)^T (\mathrm{Y} - \mathbf{X}\beta)$$

$$\text{subject to } \sum \beta_j^2 \le s$$

Ridge regression

- Ridge solution:

$$\hat{\beta}_\lambda = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathrm{Y}$$

- Singular value Decomposition:

$$\mathbf{X}=\mathbf{U}\mathbf{D}\mathbf{V}^T; \quad \mathbf{D} \text{ is a diagonal matrix with}$$

$$d_1 \geq d_2 \geq d_3 \geq \ldots \geq d_p \geq 0$$

- For Ordinary Regression

$$\mathbf{X}\hat{\beta}^{\mathrm{ls}}=\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathrm{Y}=\mathbf{U}\mathbf{U}^T\mathrm{Y}$$

Ridge regression

- Ridge solution:

$$\hat{\beta}_\lambda = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathrm{Y}$$

- Singular value Decomposition:

$$\mathbf{X}=\mathbf{U}\mathbf{D}\mathbf{V}^T; \quad \mathbf{D} = \mathrm{diagonal}(d_1, d_2, ..., d_p)$$

- For Ridge Regression

$$\mathbf{X}\hat{\beta}^{\mathrm{ridge}} = \mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathrm{I})^{-1}\mathbf{X}^T\mathbf{Y}$$

$$= \mathbf{U}D\left(\mathbf{D}\mathbf{D} + \lambda\mathbf{I}\right)^{-1} D\mathbf{U}^T\mathbf{Y} = \sum_{j=1}^{p} \mathbf{u}_j \frac{d_j^2}{d_j^2 + \lambda} \mathbf{u}_j^T\mathbf{Y}$$

Q&A

- To prove that the norm of regression solution $\hat{\beta}=\left(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y}$ will not increase as λ increases.

证明思路：利用数据奇异值分解

$$\mathbf{X}=\mathrm{U}DV^T,\quad \mathbf{X}^T\mathbf{X}=VD^TDV^T;\quad \mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}=V\left(D^TD+\lambda\mathbf{I}\right)V^T$$

$$\left\|\hat{\beta}\right\|^2=\hat{\beta}^T\hat{\beta}=\mathbf{y}^T\mathbf{X}\left(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}\right)^{-1}\left(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y}$$

$$=\mathbf{y}^TU\left[D\left(D^TD+\lambda\mathbf{I}\right)^{-2}D^T\right]U^T\mathbf{y}$$

The Lasso

- The lasso (least absolute shrinkage and selection operator) is a shrinkage method like ridge, but acts in a nonlinear manner on the outcome y.
- The lasso is defined by

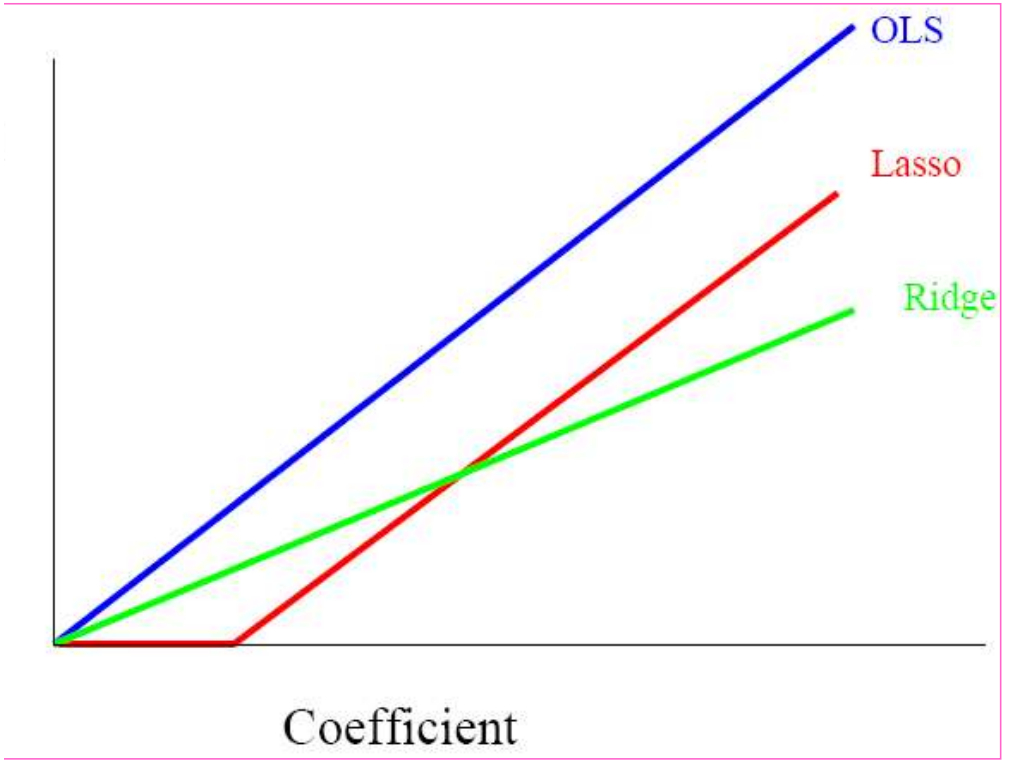
$$\hat{\beta}^{lasso} = \arg\min(\mathrm{Y} - \mathbf{X}\beta)^T(\mathrm{Y} - \mathbf{X}\beta)$$

$$\text{subject to } \sum_{j=1}^{p}\left|\beta_j\right| \le t$$

- No constraint on β_0

The Lasso

- The parameter t should be adaptively chosen to minimize an estimate of expected, using say cross-validation
- *Ridge vs Lasso:* if inputs are orthogonal,
  - ridge *multiplies* least squares coefficients by a constant < 1,
  - lasso *translates* them towards zero by a constant, truncating at zero.



Linear Classification

- Linear and Quadratic Discriminant Functions
- Reduced Rank Linear Discriminant Analysis
- Logistic Regression
- Separating Hyperplanes

Linear Discriminant Analysis

- According to the Bayes optimal classification mentioned in chapter 2, the posteriors is needed.

  post probability : $\Pr(G \mid X)$

  Assume:

  $f_k(x)$ —— condition-density of $\mathbf{X}$ in class $G=k$.

  π_k —— prior probability of class k, with $\sum_{k=1}^{K} \pi_k = 1$

  Bayes theorem give us the discriminant:

$$\Pr(G = k \mid X = x) = \frac{f_k(x)\pi_k}{\sum_{l=1}^{K} f_l(x)\pi_l}$$

Linear Discriminant Analysis

- Multivariate Gaussian density:

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}$$

- Comparing *pdfs* of two classes k and l, assume $\Sigma_k = \Sigma, \forall k$

$$\log \frac{\Pr(G=k \mid X=x)}{\Pr(G=l \mid X=x)} = \log \frac{f_k(x)}{f_l(x)} + \log \frac{\pi_k}{\pi_l}$$

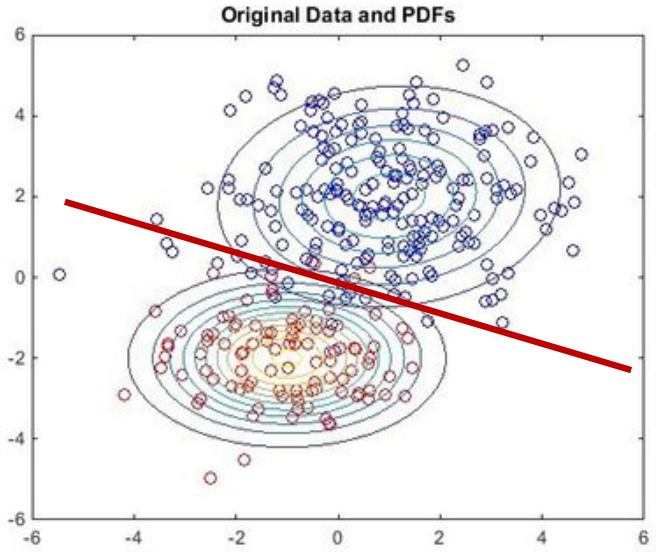
$$= \log \frac{\pi_k}{\pi_l} - \frac{1}{2}(\mu_k + \mu_l)\Sigma^{-1}(\mu_k - \mu_l)$$

$$+ x^T \Sigma^{-1}(\mu_k - \mu_l)$$

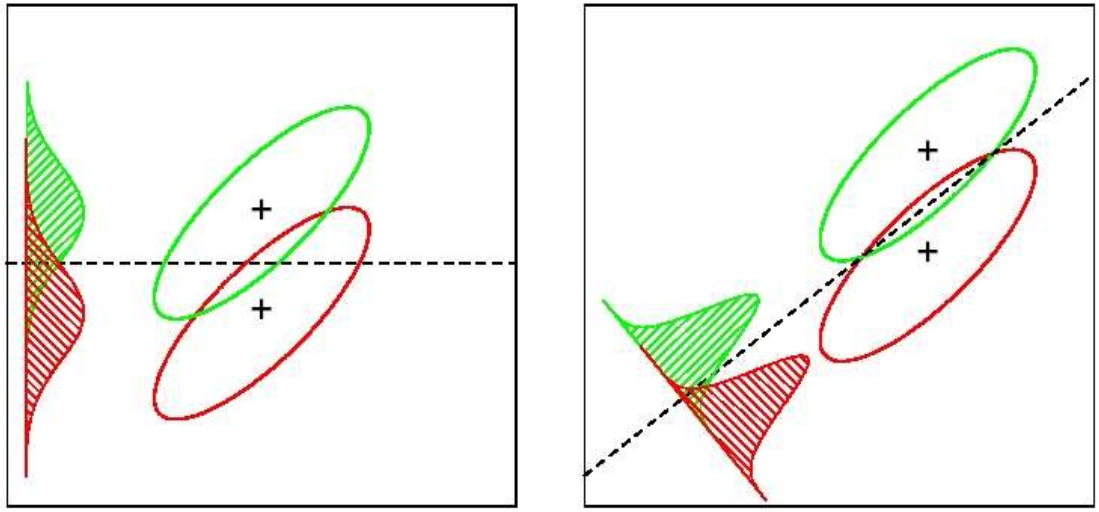
Q&A

- 关于LDA的分界面：
  - 对于线性判别分析，两类协方差相同，第一类样本数N1大约第二类样本数，分界面里哪一类中心更近？

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k \Sigma^{-1} \mu_k + \log \pi_k$$



Reduced Rank LDA



- Although the line joining the centroids defines the direction of greatest centroid spread, the projected data overlap because of the covariance (left panel).
- The discriminant direction minimizes this overlap for Gaussian data (right panel).

Reduced Rank LDA

- Let $\hat{\Sigma} = UDU^T$
- Let $X^* = D^{-1/2}U^T X$ $\quad$ i.e. $\hat{\Sigma}^{-1/2} X$
  $\hat{U}_K^* = D^{-1/2}U^T \hat{U}_K$ $\quad$ $\hat{\Sigma}^{-1/2} U_K$
- LDA: $\delta_K(x) = \frac{1}{2}\left\|x^* - \hat{\mu}_K^*\right\|^2 - \log \hat{\pi}_K$
  - Closest centroid in sphered space(apart from $-\log \hat{\pi}_K$)
- Can project data onto K-1 dim subspace spanned by $\hat{U}_1^*, \cdots, \hat{U}_{K-1}^*$, and lose nothing!
- Can project even lower dim using principal components of $\hat{U}_k^*$, $k=1, \ldots, M(<K)$.

Logistic Regression vs LDA

$$\log\frac{\Pr(g=k|X=x)}{\Pr(g=K|X=x)}=\log\frac{\pi_k}{\pi_K}-\frac{1}{2}(\mu_k+\mu_K)^T\Sigma^{-1}(\mu_k-\mu_K)+x^T\Sigma^{-1}(\mu_k-\mu_K)=\alpha_{k0}+\alpha_k^T x$$

Same form as Logistic Regression

$$LR:\Pr(X,g=k)=\Pr(X)\Pr(g=k|X=x)$$

Conditional Likelihood

Probability

$$\Pr(g=k|X=x)=\frac{\exp(\beta_{k,0}+\beta_k^T x)}{1+\sum_{l=1}^{K-1}\exp(\beta_{l,0}+\beta_l^T x)}$$

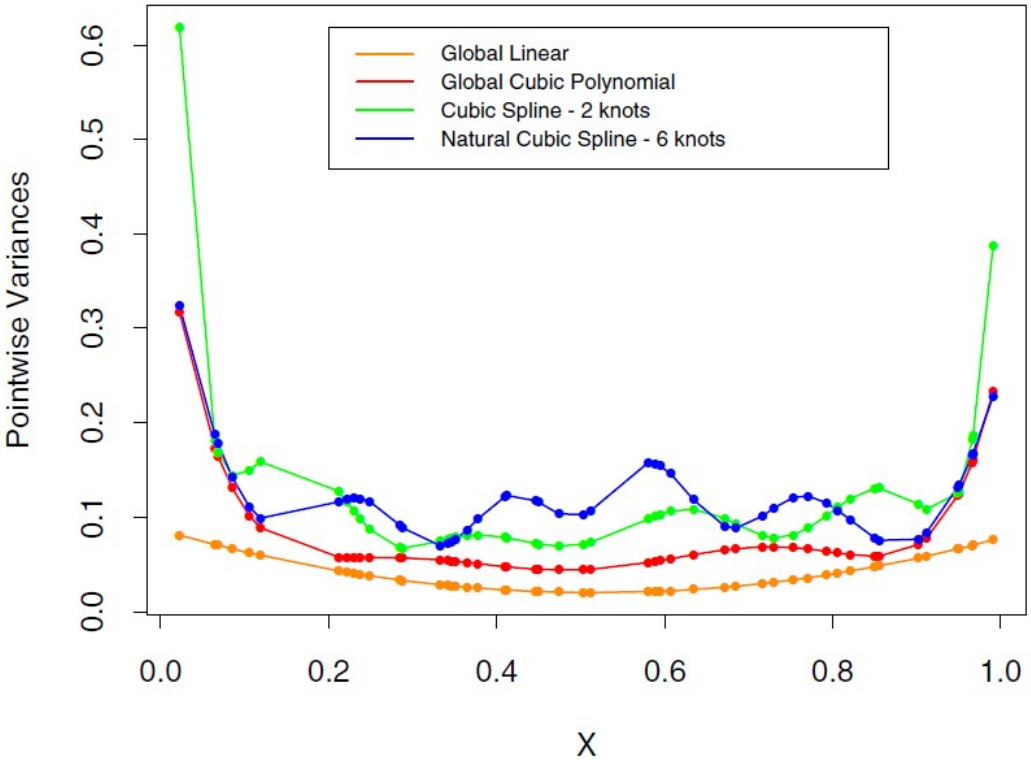
$$LDA:\Pr(X,g=k)=\phi(X;\mu_k,\Sigma)\pi_k$$

$$\Pr(X)=\sum_{k=1}^{K}\pi_k\phi(X;\mu_k,\Sigma)$$

Overview of the Course

- Introduction
- Overview of Supervised Learning
- Linear Method for Regression and Classification
- Basis Expansions and Regularization
- Kernel Methods
- Model Selections and Inference
- Support Vector Machine
- Unsupervised Learning
- Latent Dirichlet Allocation
- Deep Networks and Regularization

Boundary Effect in Variances



$$f(x) = \sum_{k=1}^{N} \beta_k h_k(x) = \beta^{\mathrm{T}} h(x)$$

$$\mathrm{var}(\hat{f}(x)) = h(x)^T (H^T H)^{-1} h(x) \sigma^2$$

*

Smoothing Splines

- Base on the spline basis method: $f(x) = \sum_{k=1}^{N} \beta_k h_k(x)$

$y = \sum_{k=1}^{m} \beta_k h_k(x) + \varepsilon$, ε is the noise.

- Minimize the penalized residual sum of squares

$$RSS(f, \lambda) = \sum_{i=1}^{N} \{y_i - f(x_i)\}^2 + \lambda \int \{f''(t)\}^2 dt$$

λ is a fixed smoothing parameter

$\lambda = 0$: f can be any function that interpolates the data

$\lambda = \infty$: the simple least squares line fit

Smoothing Splines

- The solution is a natural spline: $f(x) = \sum_{j=1}^{N} N_j(x)\theta_j$
- Then the criterion reduces to:

$$RSS(\theta, \lambda) = (y - N\theta)^T (y - N\theta) + \lambda \theta^T \Omega_N \theta$$

  - where $N = \{N_j(x_i)\}; \quad \Omega_{Nij} = \int N_i''(t) N_j''(t) dt$

- So the solution:

$$\hat{\theta} = (N^T N + \lambda \Omega_N)^{-1} N^T y$$

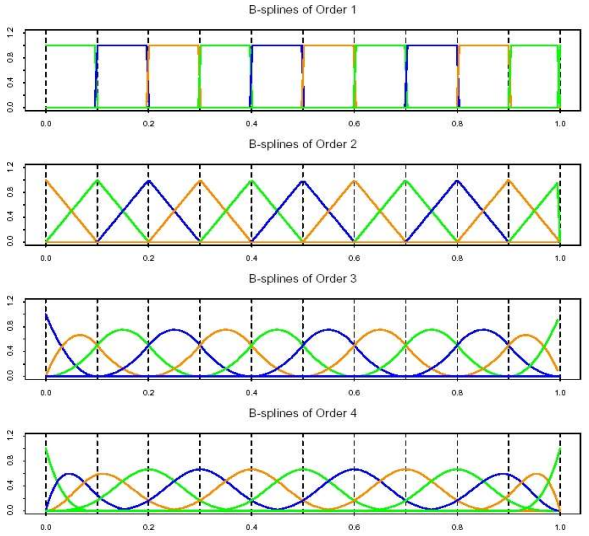
- The fitted smoothing spline:

$$\hat{f}(x) = \sum_{j=1}^{N} N_j(x)\hat{\theta}_j$$

Some Key Points

- The B-spline have local support; they are nonzero on an interval spanned by M+1 knots.

$$f(x) = \sum_{k=1}^{N} \beta_k h_k(x)$$



- Minimize

$$RSS(f,\lambda) = \sum_{i=1}^{N} \{y_i - f(x_i)\}^2 + \lambda \int \{f''(t)\}^2 dt$$

$$\hat{f} = N(N^T N + \lambda \Omega_N)^{-1} N^T y = S_\lambda y$$

$$df_\lambda = trace(S_\lambda)$$

- Effective degrees of freedom

Bias-Variance Tradeoff

- The integrated squared prediction error (EPE) combines both bias and variance in a single summary:

$$EPE(\hat{f}_\lambda) = E(Y - \hat{f}_\lambda(X))^2$$

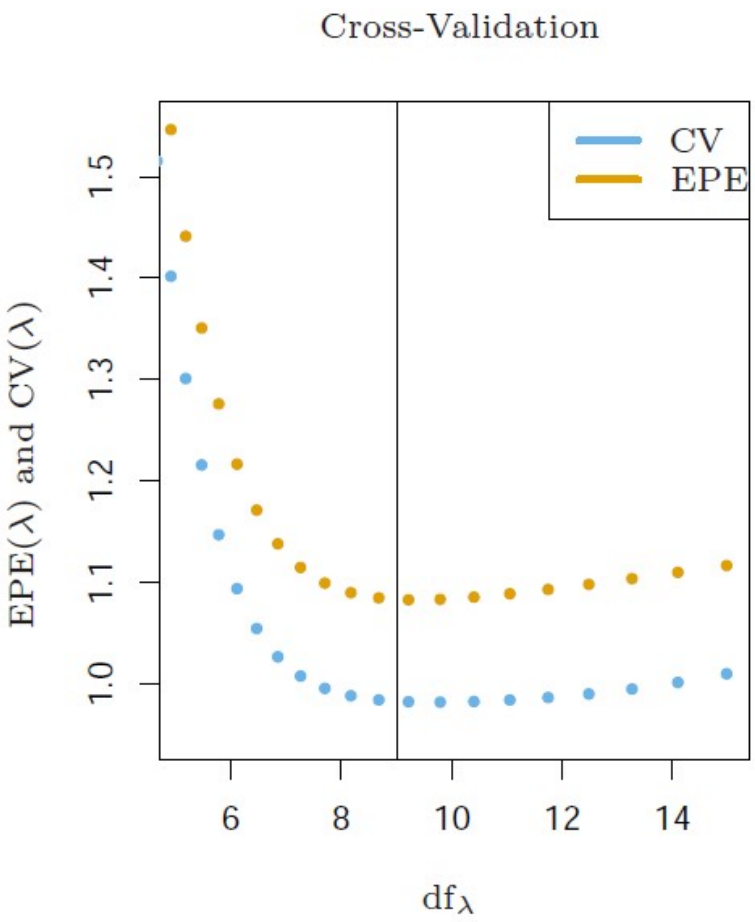
$$= Var(Y) + E\left[Bias^2(\hat{f}_\lambda(X)) + Var(\hat{f}_\lambda(X)) \right]$$

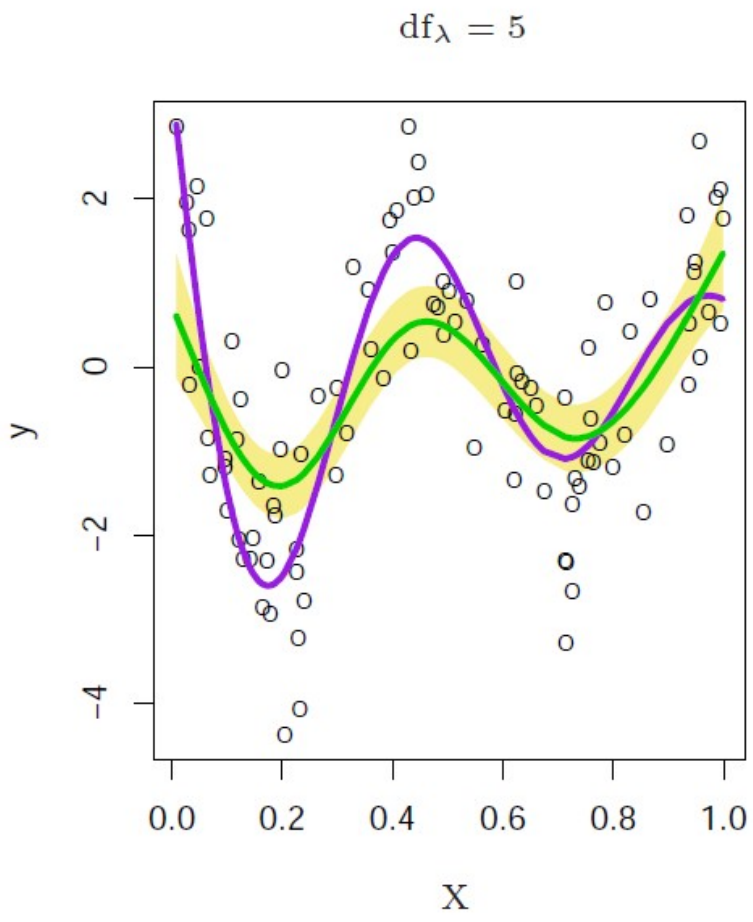
$$= \sigma^2 + MSE(\hat{f}_\lambda)$$

- **N fold (leave one)** cross-validation:

$$CV(\hat{f}_\lambda) = \sum_{i=1}^{N} (y - \hat{f}_\lambda^{-i}(x_i))^2 = \sum_{i=1}^{N} \left(\frac{y_i - \hat{f}_\lambda(x_i)}{1 - S_\lambda(i,i)} \right)^2$$

Bias-Variance Tradeoff





The EPE and CV curves have the a similar shape.
And, overall the CV curve is approximately unbiased as an estimate of the EPE curve

Reproducing Kernel Hilbert space

- A regularization problems has the form:

$$\min_{f \in H}\left[\sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda J(f)\right] \qquad J(f) = \int \frac{\left|\widetilde{f}(s)\right|^2}{\widetilde{G}(s)} ds$$

  - $L(y_i, f(x_i))$ is a loss-function.
  - *J(f)* is a penalty functional, and H is a space of functions on which *J(f)* is defined.

- The solution

$$f(x) = \sum_{k=1}^{K} \alpha_k \phi_k(X) + \sum_{i=1}^{N} \theta_i G(X - x_i)$$

  - ϕ_k span the null space of the penalty functional ***J***

Spaces of Functions Generated by Kernel

- Important subclass are generated by the positive kernel $K(x,y)$.
- The corresponding space of functions H_k is called reproducing kernel Hilbert space.
- Suppose that K has an eigen-expansion

$$K(x,y)=\sum_{i=1}^{\infty}\gamma_i\phi_i(x)\phi_i(y),\quad \gamma_i\geq 0,\sum_{i=1}^{\infty}\gamma_i^2<\infty$$

- Elements of H have an expansion

$$f(x)=\sum_{i=1}^{\infty}c_i\phi_i(x),\quad \|f\|_{H_k}^2\doteq\sum_{i=1}^{\infty}c_i^2/\gamma_i<\infty$$

*

Spaces of Functions Generated by Kernel

- Reproducing properties of kernel function

$$< K(\bullet, x_i), f >_{H_K} = f(x_i), \quad < K(\bullet, x_i), K(\bullet, x_j) >_{H_K} = K(x_i, x_j)$$

$$f(x) = \sum_{i=1}^{\infty} c_i \phi_i(x), \quad \|f\|_{H_k}^2 \doteq \sum_{i=1}^{\infty} c_i^2 / \gamma_i < \infty$$

Key Points in the Basis Expansion

- Good representation of function spaces
  - Easy to implement (efficient in space and time)
  - Good for generalization
  - Easy to select good models
- Good parameter for model selection
  - Effective degrees of freedom
  - CV for Model selection
- Reproducing Kernel Hilbert Space
  - Polynomial Kernel
- Spline & Wavelet

Overview of the Course

- Introduction
- Overview of Supervised Learning
- Linear Method for Regression and Classification
- Basis Expansions and Regularization
- Kernel Methods
- Model Selections and Inference
- Support Vector Machine
- Unsupervised Learning
- Latent Dirichlet Allocation
- Deep Networks and Regularization

Local Linear Regression

- Locally weighted linear regression make a first-order correction
- Separate weighted least squares at each target point x_0:

$$\min_{\alpha(x_0),\beta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i)[y_i - \alpha(x_0) - \beta(x_0)x_i]^2$$

- The estimate: $\hat{f}(x_0) = \hat{\alpha}(x_0) + \hat{\beta}(x_0)x_0$
- $b(x)^T=(1,x)$; B: Nx2 regression matrix with i-th row $b(x)^T$;

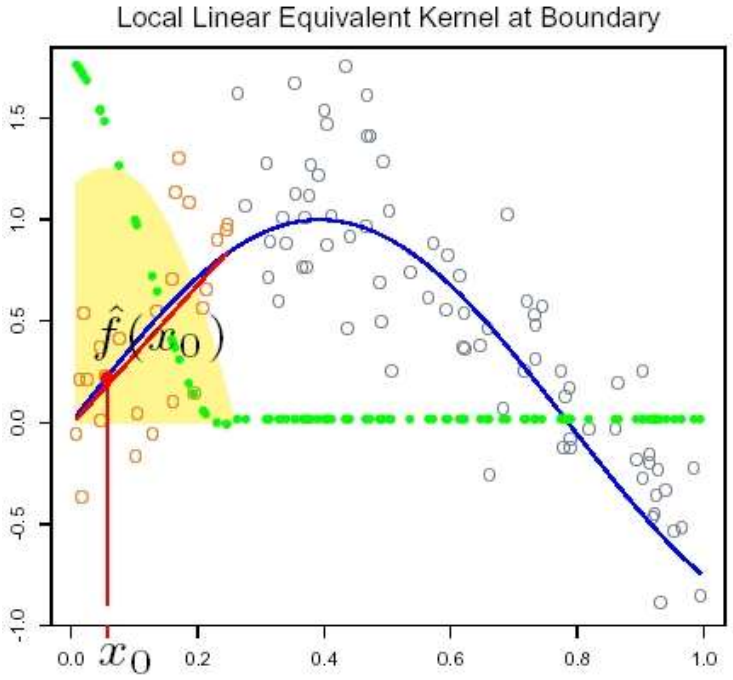
$$W_{N\times N}(x_0) = diag\big(K_\lambda(x_0, x_i)\big), i = 1, \ldots, N$$

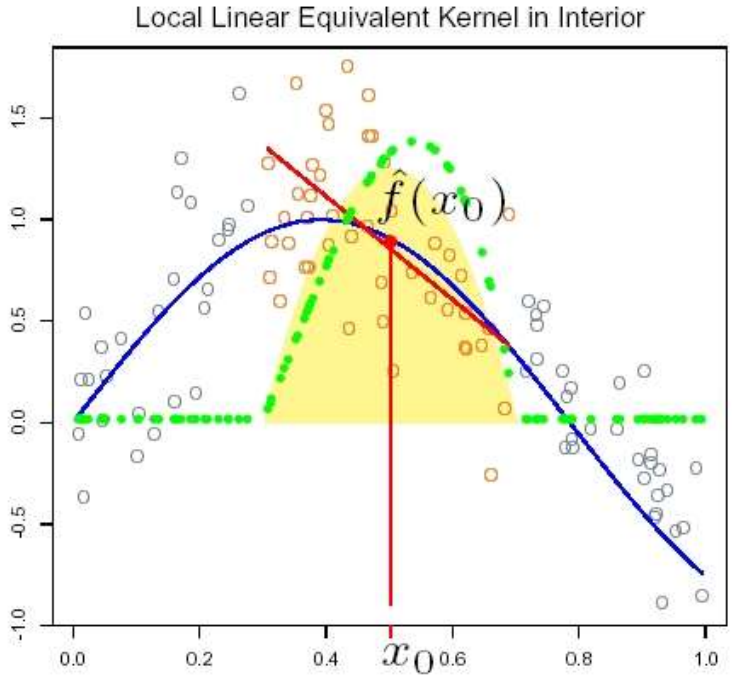
$$\hat{f}(x_0) = b(x_0)^T (B^T W(x_0) B)^{-1} B^T W(x_0) y = \sum_{i=1}^{N} l_i(x_0) y_i$$

*

Local Linear Regression

- The weights $l_i(x_0)$ combine the weighting kernel $K_\lambda(x_0,\cdot)$ and the least squares operations——Equivalent Kernel





Local Linear Regression

- The expansion for $E\hat{f}(x_0)$, using the linearity of local regression and a series expansion of the true function f around x_0

$$E\hat{f}(x_0) = \sum_{i=1}^{N} l_i(x_0) f(x_i) = f(x_0)\sum_{i=1}^{N} l_i(x_0) + f'(x_0)\sum_{i=1}^{N}(x_i - x_0) l_i(x_0)$$
$$+ \frac{f''(x_0)}{2}\sum_{i=1}^{N}(x_i - x_0)^2 l_i(x_0) + R$$

$$\sum_{i=1}^{N} l_i(x_0) = 1, \quad \sum_{i=1}^{N}(x_i - x_0) l_i(x_0) = 0$$

For local regression

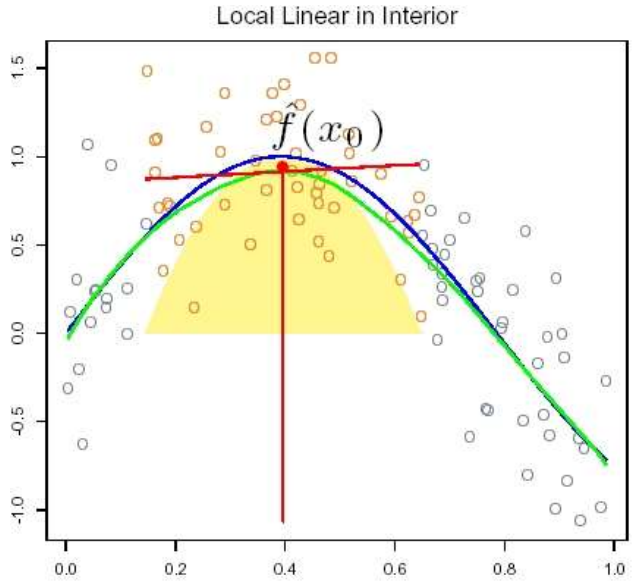
- The bias $E\hat{f}(x_0) - f(x_0)$ depends only on quadratic and higher-order terms in the expansion of f.

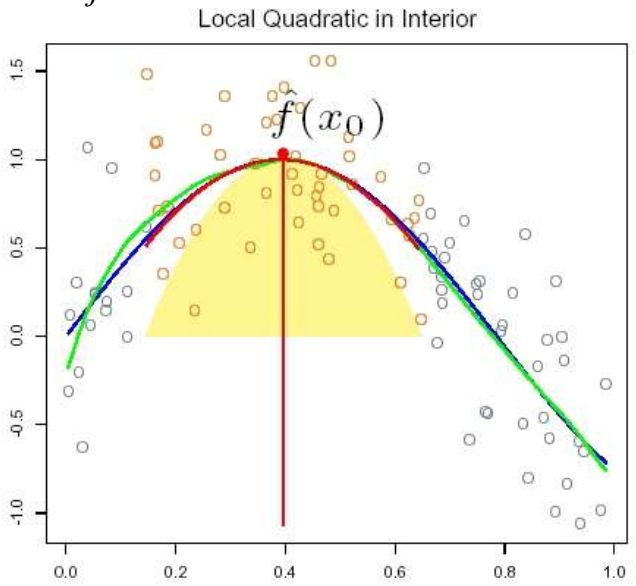
Local Polynomial Regression

- Fit local polynomial fits of any degree d

$$\min_{\alpha(x_0),\beta_j(x_0),j=1,\ldots,d} \sum_{i=1}^{N} K_\lambda(x_0,x_i)\left[y_i-\alpha(x_0)-\sum_{j=1}^{d}\beta_j(x_0)x_i^j\right]^2$$

$$\hat{f}(x_0)=\hat{\alpha}(x_0)+\sum_{j=1}^{d}\hat{\beta}_j(x_0)x_0^j$$

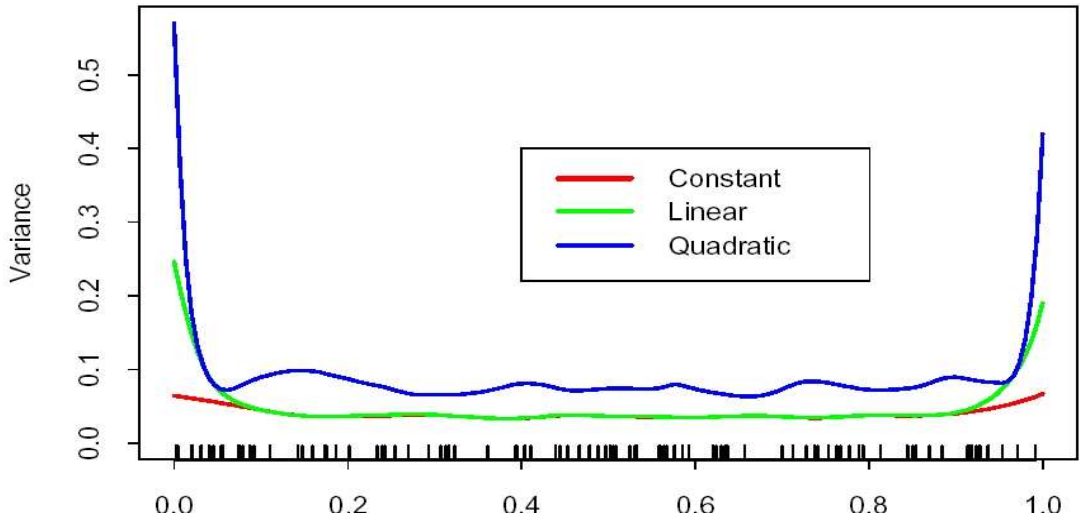




Local Polynomial Regression

- Bias only have components of degree d+1 and higher.
- The reduction for bias costs the increased variance.

$$\mathrm{var}(\hat{f}(x_0)) = \sigma^2 \|l(x_0)\|^2, \boxed{\|l(x_0)\| \text{ increases with } d}$$



Summary

- To understand why increasing the order of polynomials causes the increase of model variance
- How to kernel method to implement local regression
- Probability density estimation by kernel methods
- EM algorithm for estimating GMM

Overview of the Course

- Introduction
- Overview of Supervised Learning
- Linear Method for Regression and Classification
- Basis Expansions and Regularization
- Kernel Methods
- Model Selections and Inference
- Support Vector Machine
- Unsupervised Learning
- Latent Dirichlet Allocation
- Deep Networks and Regularization

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

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.

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

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

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

$$= \frac{1}{N}\sum_{i=1}^{N}\left[2E[y_i, \hat{y}_i] - 2E\hat{y}_i Ey_i\right] = \frac{2}{N}\sum_{i=1}^{N} Cov(\hat{y}_i, y_i)$$

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.

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

Bayesian Approach & BIC

- The Bayesian Information Criterion (BIC)

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

- Gaussian model:
  - Variance σ_ε^2
  - then

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

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

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

*

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

- 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.

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

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.

The EM Algorithm

- Gaussian Mixture Model

$$f(x) = \sum_{m=1}^{M} \alpha_m \phi(x; \mu_m, \Sigma_m)$$

- EM algorithm for 2 Gaussian mixtures
  - Given $x_1, x_2, \ldots, x_n$, log-likelihood:

$$l(y, \theta) = \sum_{i=1}^{N} \log\left[\alpha\phi_{\theta_1}(x_i) + (1-\alpha)\phi_{\theta_2}(x_i)\right]$$

  - Suppose we observe Latent Binary

Bad

$$L(x, z, \theta) = \sum_{\substack{i=1 \\ z_i=1}}^{N} \log\left[\alpha\phi_{\theta_1}(x_i)\right] + \sum_{\substack{i=1 \\ z_i=0}}^{N} \log\left[(1-\alpha)\phi_{\theta_2}(x_i)\right]$$

z such that $z = 1 \Rightarrow x \sim \phi_{\theta_1}, \quad z = 0 \Rightarrow x \sim \phi_{\theta_2}$

Good

The EM Algorithm in General

- Start with initial params $\hat{\theta}$
- Expectation Step: at the j-th step compute

$$Q(\theta', \hat{\theta}^{(j)}) = \mathrm{E}(\ell_0(\theta', \mathbf{T}) \mid \mathbf{Z}, \hat{\theta}^{(j)})$$

  as a function of the dummy argument θ'
- Maximization Step: Determine the new params by maximizing $\hat{\theta}^{(j+1)}$

$$Q(\theta', \hat{\theta}^{(j)})$$

- Iterate 2 and 3 until convergence