Chapter 3
Regularization Framework

Contents

- Curse and Complexity of Dimensionality
- Function-Approximation and Characterization of Complexity
- Penalization
  - Parametric Penalties
  - Nonparametric penalties
- Model Selection (Complexity Control)
  - Analytic Model Selection Criteria
  - Model Selection via Resampling
  - Bias-Variance Trade-off

3.1 Curse and Complexity of Dimensionality (1)

- Curse of Dimensionality
  - "For high-dimensional functions it becomes difficult to collect enough samples to attain high density."

$$R_{\text{emp}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2$$

  - High-dimensional learning problems are more difficult in practice because low data density requires the user to specify stronger, more accurate constraints on the problem solution.

Curse and Complexity of Dimensionality (2)

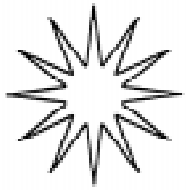


Figure 3.1 Conceptually, at least, high-dimensional data looks like a porcupine.

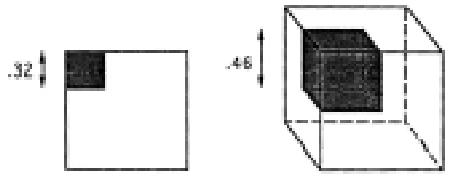


Figure 3.2 Both gray regions enclose 10% of the samples, but the edge length of the regions increases with increasing dimensionality.

Curse and Complexity of Dimensionality (3)

- Properties of High-dimensional Distributions
  1. *Sample sizes yielding the same density increase exponentially with dimension.*
  2. *A large radius is needed to enclose a fraction of the data points in a high-dimensional space.*
  3. *Almost every point is closer to an edge than to another point.*
  4. *Almost every point is an outlier in its own projection.*

Curse and Complexity of Dimensionality (4)

- Properties 1 and 2
  - difficulty in making local estimates for high-dimensional samples
- Properties 3 and 4
  - difficulty in predicting a response at a given point
  - Since any point will on average be closer to an edge than to the training data point, it will require extrapolation by the learning machine.

Curse and Complexity of Dimensionality (5)

- Kolmogorov's theorem
  - Any continuous function of multiple arguments can be written as a function of a single argument.

$$f(x_1, x_2, \cdots, x_d) = \sum_{j=1}^{2d+1} g_f \left(\sum_{i=1}^{k} \lambda_i \gamma_j (x_i) \right)$$

Curse and Complexity of Dimensionality (6)

- Conclusion
  - A function's dimensionality is not a good measure of its complexity since any high-dimensional function can be written as a decomposition of univariate functions.
  - High-dimensional functions have the potential to be more complex than low-dimensional functions.
  - There is a need to provide a characterization of a function's complexity that takes into account its smoothness and dimensionality.

3.2 Function Approximation

- Function Approximation
  - Representation of functions using some specified class of "basis" functions
    - Weierstrass Theorem (classical example)
      "For any function $f(x)$ and any positive ε, there exists a polynomial of degree m, $p_m(x)$, such that $\| f(x) - p_m(x) \| < \varepsilon$ for every x."
    - Two types of approximation-theory results relevant to problem of learning from data
      - universal approximation results
      - rate of convergence results

Universal Approximation Results (1)

- "Any function can be accurately approximated by another function from a given class."
- Universal approximators
  - Linear combination (General form)

    $$f_m(x, \mathbf{w}) = \sum_{i=1}^{m-1} w_i g_i(x)$$

  - Algebraic polynomials

    $$f_m(x, \mathbf{w}) = \sum_{i=0}^{m-1} w_i x^i$$

  - Trigonometric polynomials

    $$f_m(x, \mathbf{v}_m, \mathbf{w}_m) = \sum_{i=1}^{m-1} v_i \sin(ix) + \sum_{i=1}^{m-1} w_i \cos(ix) + w_0$$

Universal Approximation Results (2)

  - Multilayer networks

    $$f_m(\mathbf{x}, \mathbf{w}, V) = w_0 + \sum_{j=1}^{m} w_j g\left(v_{0j} + \sum_{i=1}^{d} x_i v_{ij}\right)$$

  - Local basis function networks

    $$f_m(x, \mathbf{v}, \mathbf{w}) = \sum_{i=0}^{m-1} w_i K_i\left(\frac{\|x - v_i\|}{\alpha}\right)$$

- The universal approximation property is a *necessary condition* for a set of approximating functions of the learning machine in the general formulation.

Rate-of-convergence Results (1)

- relate the accuracy of function approximation with some measure of the target function smoothness (complexity) and its dimensionality.
- Classical Approach to Characterization of a function's complexity
  1. Define the notion of complexity for a class of functions
  2. Specify a class of approximating functions of a learning machine
  3. Estimate the asymptotic rate of convergence, defined as the accuracy of approximating an arbitrary function (1) in the class (2)
     - Interest : how the rate of convergence depends on the dimensionality of the class of functions (1).

Classical Measures of Function's Complexity

- Numbers of continuous derivatives
  - approximation accuracy = $O(m^{-s/d})$
- Frequency content of a target function (signal)
- Properties of its Fourier transform

Rate-of-convergence Results (2)

- Two possible directions in dealing with high-dimensional problems with finite samples
  1. To adopt traditional global definition of complexity/smoothness (*fixed* number of continuous derivatives) but to impose suitable restrictions on function's smoothness in high dimensions.
     - Different complexity measures for various input dimensions (variable selection)

Rate-of-convergence Results (3)

2. To adopt new local measure of function's smoothness in high dimensions.
   - The function's potential for complexity is limited to some local regions.
   - The goal of learning is to locate regions as well as to estimate the function's local complexity in each region.

3. To measure the function's flexibility in terms of its ability to fit the finite data (VC-dimension)

3.3 Penalization

- A formation for controlling complexity of approximating functions to fit available finite data
- Risk for the regularization inductive principle
  - $R_{pen}(\omega) = R_{emp}(\omega) + \lambda\phi[\boldsymbol{f}(\mathbf{x}, \omega)]$
  - $R_{emp}(\omega)$: empirical risk
    - enforcing closeness of the approximating function to the data
  - $\phi[\boldsymbol{f}(\mathbf{x}, \omega)]$: penalty function
    - enforcing smoothness
    - smaller value for smooth functions, and larger value for non-smooth functions $\boldsymbol{f}(\mathbf{x}, \omega)$
  - λ : adjustment of the strength of the penalty criterion

Four Issues in Penalization

1. Class of approximating functions $f(\mathbf{x}, \omega)$
   - The usual choices are between a class of all continuous functions and a class of parametric functions.
2. Type of penalty function
   - parametric penalty function
     - measures the smoothness of a function indirectly by imposing constraints on the parameters of approximating functions
   - nonparametric penalty function
     - measures function smoothness directly based on the differential operators
3. Method for optimization (minimization of R_{pen})
   - For a given value λ, find $f_\lambda(x,w^*)$
4. Model for complexity control
   - Choice of regularization parameter λ

Parametric Penalties (1)

- Parametric Penalties
  - $\phi[f(\mathbf{x}, \mathbf{w}_m)] = \eta(\mathbf{w}_m)$
  - Two popular examples of penalty functions

$$\eta_r(w_m) = \sum_{i=0}^{m-1} w_i^2 \qquad \text{(ridge)}$$

$$\eta_s(w_m) = \sum_{i=0}^{m-1} I(w_i \neq 0) \qquad \text{(subset selection)}$$

  - Ridge penalty
    - encourages solutions that have small parameter values.
  - Subset penalty
    - encourages solutions that have a large number of parameters with zero values.

Parametric Penalties (2)

$$\eta_p(w_m) = \sum_{i=0}^{m-1} |w_m|^p \qquad \text{(bridge)}$$

$$\eta_q(w_m) = \sum_{i=0}^{m-1} \frac{(w_i/q)^2}{1+(w_i/q)^2} \qquad \text{(weight decay)}$$

- Bridge penalty
  - Ridge penalty when p=2
  - Subset selection, when p->0
- Weight decay penalty
  - Ridge penalty as q->∞
  - Subset selection penalty as q-> ∞

Nonparametric Penalties (1)

- Measure the smoothness of a function directly using a differential operator.
- Meaning of smoothness must be defined
  - It can be defined in terms of the wiggliness of a function measured in the frequency domain
  - The number of high-frequency components in the function indicate the function smoothness
    - smoothness can be measured by applying a high-pass filter and determining the signal output power.

$$\phi[f] = \int_{\Re^d} \frac{|\tilde{f}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})} d\mathbf{s}$$

    - tilde : Fourier transform
    - 1/G : transform function of a high-pass filter

Nonparametric Penalties (2)

- Under certain conditions on G, the functions that minimize the regularization risk correspond to commonly used class of basis functions.

$$R_{pen}(f) = \sum_{i=1}^{n} [f(\mathbf{x}_i) - y_i]^2 + \lambda\phi[f(\mathbf{x})]$$

  - Each different method for measuring complexity leads to a different set of approximating functions.
  - The selection of a class of functions for a learning machine implicitly defines a regularization procedure with a penalty function.

3.4 Model Selection (1)

- Task of choosing a model of optimal complexity for the given data
- The selection of appropriate penalty function $\phi[\boldsymbol{f}]$ and the value of regularization parameter λ should be made in such a way that an estimate found by minimizing functional $R_{\text{pen}}(\omega) = R_{\text{emp}}(\omega) + \lambda\phi[\boldsymbol{f}(\mathbf{x}, \omega)]$ provides minimum of the prediction risk.

Model Selection (2)

- Best penalty function $\phi[\boldsymbol{f}]$
  - $\phi[\boldsymbol{f}]$ should reflect properties of a target function, so that the penalty is small when $\boldsymbol{f}(\mathbf{x}, \omega^*)$ is close to the target function, and large otherwise.
  - Prior knowledge cannot completely determine the target function.
  - Under Bayesian paradigm, both $\phi[\boldsymbol{f}]$ and λ are chosen based on a priori knowledge, so by definition the observed data are not used for model selection.

Model Selection (3)

- To make learning machines “data-driven”
  - λ : selected by observed data
  - $\phi[\boldsymbol{f}]$: user-defined
- Task of model selection is to determine the value of λ such that minimization of the functional $R_{\text{pen}}(\omega) = R_{\text{emp}}(\omega) + \lambda\phi[\boldsymbol{f}(\mathbf{x}, \omega)]$ produces a solution $\boldsymbol{f}(\mathbf{x}, \omega^*)$ that has minimal prediction risk.
- How to estimate prediction risk from data
  - Use analytical results
  - Data resampling

3.4.1 Analytical Model Selection

- Use analytical estimates of the prediction risk
  - The form of estimates is dependent on the class of approximating functions.
  - The most commonly known criteria
    - apply to linear estimators for regression
    - A regression estimator is linear if

      $f_0(a\mathbf{y}' + b\mathbf{y}'' \mid \mathbf{X}) = af_1(\mathbf{y}' \mid \mathbf{X}) + bf_2(\mathbf{y}'' \mid \mathbf{X})$

      where $a \neq 0$, $b \neq 0$, f_0, f_1, f_2 are 3 estimators, $\mathbf{X}$ are predictor samples, $\mathbf{y}'$, $\mathbf{y}''$ are two response values.
    - The approximation can be written as $f(\mathbf{X}, \omega) = \mathbf{S}\mathbf{y}$

      where $\mathbf{S}$ is an $n \times n$ matrix that transforms the response values into estimates for each sample.

Linear Estimators for Regression (1)

- For linear estimators include two classes of functions
  - kernel smoothers

    $(\mathbf{S}_\alpha)_{ij} = K_\alpha(\mathbf{x}_i, \mathbf{x}_j), \qquad i = 1, \ldots, n, j = 1, \ldots, n$
  - For estimators linear in parameters

    $\mathbf{S} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$
- Consider the ridge regression risk functional.
  - For given λ, the solution which minimizes

    $R_{\text{ridge}}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{w} \cdot \mathbf{x}_i)^2 + \frac{\lambda}{n}(\mathbf{w} \cdot \mathbf{w})$

    is a linear estimator with the "hat" matrix

    $\mathbf{S}_\lambda = \mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T$
  - The number of degrees of freedom : $h_l = \text{trace}(\mathbf{S}_\lambda\mathbf{S}_\lambda^{\mathbf{T}})$

Linear Estimators for Regression (2)

- All known analytical model selection criteria can be written as

  $R(\omega) = r\left(\frac{h_l}{n}\right) R_{\text{emp}}$

  - r : penalization function
    - increasing function of the ratio of degrees of freedom h_l and the training sample size n
    - forms of r : $p = h_l / n$.
      - final prediction error (*fpe*, Akaike, 1970) $r(p) = (1 + p)(1 - p)^{-1}$
      - Schwartz' criteria (*sc*, 1978) $r(p, n) = \left(1 + \frac{\ln n}{2} p(1 - p)^{-1}\right)$
      - Generalized cross-validation (*gcv*, Craven and Wahba, 1979)

        $r(p) = (1 - p)^{-2}$
      - Shibata's model selector (*sms*, 1981)

        $r(p) = 1 + 2p$

3.4.2 Model Selection via Resampling (1)

- no assumptions on the statistics of the data or the type of a target function
- Basic approach
  - first estimate a model using a portion of the training data and then use the remaining samples to estimate the prediction risk for this model.
  - First portion of the data (n_l samples): *learning* set
  - Second portion ($n_v = n - n_l$): *validation* set

Model Selection via Resampling (2)

- Simplest Approach
  - split data into two portions
  - prediction risk

$$R(\omega) \cong R_v(\omega) = \frac{1}{n_v} \sum_{i=1}^{n_v} L(y_i, f_\lambda(\mathbf{x_i}, \omega^*))$$

  - Goal
    - find λ such that the corresponding model estimate $f_\lambda(\mathbf{x}, \omega^*)$ provides smallest prediction risk
  - true for large data sets

Model Selection via Resampling (3)

- Cross-validation
  - makes estimate invariant to a particular partitioning of samples
  - performs estimate for all $\binom{n}{n_l}$ possible partitions and
  - average them
- *k-fold cross validation*
  - practical approach
  - divide the data into k subsamples of roughly equal size $n_v = n/k$
  - leave-one-out cross validation if $n_v = 1$

k-fold Cross Validation

1. Divide the training data $\mathbf{Z}$ into k disjoint samples of roughly equal size, $\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_k$.
2. For each validation sample $\mathbf{Z}_i$ of size n/k,

   a. Use the remaining data, $\mathbf{Z}_l = \cup_{j \neq i} \mathbf{Z}_j$ to construct an estimate f_i.

   b. For the regression f_i, sum the empirical risk for the data $\mathbf{Z}_i$ "left out":

$$r_i = \frac{k}{n} \sum_{\mathbf{Z}_i} (f_i(\mathbf{x}) - y)^2$$

3. Compute the estimate for the prediction risk by averaging the empirical risk sums for $\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_k$:

$$R(\omega) \cong R_{cv}(\omega) = \frac{1}{k} \sum_{i=1}^{k} r_i$$

Model Selection via Resampling (4)

- Advantage
  - do not depend on assumptions about the statistics of the data or specific properties of approximating functions
- Disadvantage
  - computational effort
  - variability of estimates

3.4.3 Bias-Variance Trade-off

- effect of different values of λ
- For the regression learning problem using L_2 loss, the approximation error can be decomposed as the sum of two terms that distinguish the error due to estimation from finite samples (*variance*) and error due to mismatch between target function and approximating function (*bias*).

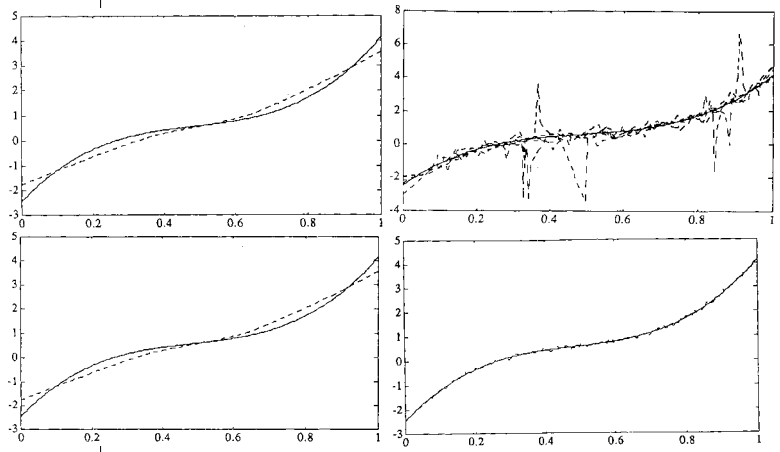
Bias and Variance (Example) (1)

- Artificial data of target functions

  $y = x + 20(x-0.5)^3 + (x-0.2)^2 + \varepsilon$

  where ε is zero mean gaussian, with variance 0.125
  - five data sets generated with 50 samples each.
- Procedure 1
  - Kernel smoothing (high smoothness)
  - kernel width 80% ⇒ 2 degrees of freedom
- Procedure 2
  - Kernel smoothing (high degree of complexity)
  - kernel width 10% ⇒ 10 degrees of freedom

Bias and Variance (2)



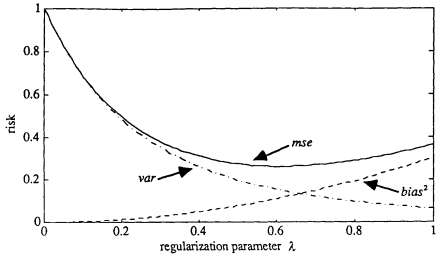
Mean Squared Error

- Mean Squared Error (mse)
  - approximation error between an estimate $f(\mathbf{x}, \omega)$ and the true function $g(\mathbf{x})$

$$E_n[(f(\mathbf{x},\omega)-g(\mathbf{x}))^2] = E_n[(f(\mathbf{x},\omega)-E_n[f(\mathbf{x},\omega)])^2] \quad \text{(variance)}$$

$$+ (g(\mathbf{x})-E_n[f(\mathbf{x},\omega)])^2 \quad \text{(bias squared)}$$

[Friedman, 1994]



Bias-Variance Dilemma

$$\text{mse}\big(f(\mathbf{x},\omega)\big) = \int E_n[(g(\mathbf{x}) - f(\mathbf{x},\omega))^2]\,p(\mathbf{x})d\mathbf{x}$$

$$\text{bias}^2\big(f(\mathbf{x},\omega)\big) = \int (g(\mathbf{x}) - E[f(\mathbf{x},\omega)])^2\,p(\mathbf{x})d\mathbf{x}$$

$$\text{var}\big(f(\mathbf{x},\omega)\big) = \int E[(f(\mathbf{x},\omega) - E_n[f(\mathbf{x},\omega)])^2]\,p(\mathbf{x})d\mathbf{x}$$

Bias and Variance (3)

- Useful for conceptual understanding
- Cannot be used for practical implementation of model selection
  - depend on the unknown sampling density $p(\mathbf{x})$
- In practice, model selection is performed using data resampling techniques to estimate the prediction risk without knowledge of the target function.

3.4.4 Example of Model Selection (1)

- Artifical data set of 25 samples generated according to $y = \sin^2(2\pi x)+\varepsilon$ where ε is zero mean gaussian with variance $\sigma^2=0.1$
- x had a uniform distribution on [0,1].
- Estimating data using the set of polynomial approximating functions

$$f_m(x,\mathbf{w}_m) = \sum_{i=0}^{m-1} w_i x^i$$

  - $m \le 10$
  - Task: choose the value of m that provides the lowest estimated expected risk

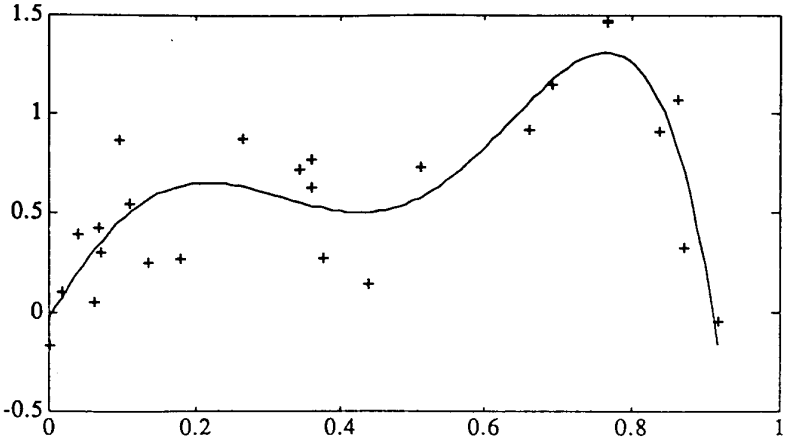
Example of Model Selection (2)

- Analytical Model Selection
  - Assume that 10 potential models, $f_m(x, \mathbf{w}_m)$, m = 1, …, 10.
  - The number of degrees of freedom: $h_l = m$
  - a polynomial with m = 6 provides the best estimated risk according to the fpe criteria

Table 3.1 Model selection using fpe for estimating prediction risk.

m	R_{emp}	Final Prediction Error $r(m/n)$	Estimated R via *fpe*
1	0.1892	1.0833	0.2049
2	0.1400	1.1739	0.1644
3	0.1230	1.2727	0.1565
4	0.1063	1.3810	0.1468
5	0.0531	1.5000	0.0797
6	0.0486	1.6316	0.0792
7	0.0485	1.7778	0.0863
8	0.0418	1.9412	0.0812
9	0.0417	2.1250	0.0886
10	0.0406	2.3333	0.0947

Example of Model Selection (3)



Example of Model Selection (4)

- Model Selection via Resampling
  - for each of candidate models, calculate the empirical risk estimate given by cross-validation.
  - 5-fold cross validation

Table 3.2 Validation sets for 5-fold cross-validation.

Validation Set	Samples from Training Set
$\mathbf{Z}_1$	$[(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5)]$
$\mathbf{Z}_2$	$[(x_6, y_6), (x_7, y_7), (x_8, y_8), (x_9, y_9), (x_{10}, y_{10})]$
$\mathbf{Z}_3$	$[(x_{11}, y_{11}), (x_{12}, y_{12}), (x_{13}, y_{13}), (x_{14}, y_{14}), (x_{15}, y_{15})]$
$\mathbf{Z}_4$	$[(x_{16}, y_{16}), (x_{17}, y_{17}), (x_{18}, y_{18}), (x_{19}, y_{19}), (x_{20}, y_{20})]$
$\mathbf{Z}_5$	$[(x_{21}, y_{21}), (x_{22}, y_{22}), (x_{23}, y_{23}), (x_{24}, y_{24}), (x_{25}, y_{25})]$

Calculation of the risk estimates via 5-fold cross-validation

Table 3.3 Calculation of the risk estimates via 5-fold cross-validation.

Polynomial Estimate of Degree m	Data to Construct Polynomial Estimate	Validation Set to Estimate Risk	Estimate of Expected Risk for Each Validation Set
$f_1(x)$	$[\mathbf{Z}_2, \mathbf{Z}_3, \mathbf{Z}_4, \mathbf{Z}_5]$	$\mathbf{Z}_1$	$r_1 = \frac{1}{5}\sum_{i=1}^{5}(f_1(x_i) - y_i)^2$
$f_2(x)$	$[\mathbf{Z}_1, \mathbf{Z}_3, \mathbf{Z}_4, \mathbf{Z}_5]$	$\mathbf{Z}_2$	$r_2 = \frac{1}{5}\sum_{i=6}^{10}(f_2(x_i) - y_i)^2$
$f_3(x)$	$[\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_4, \mathbf{Z}_5]$	$\mathbf{Z}_3$	$r_3 = \frac{1}{5}\sum_{i=11}^{15}(f_3(x_i) - y_i)^2$
$f_4(x)$	$[\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_3, \mathbf{Z}_5]$	$\mathbf{Z}_4$	$r_4 = \frac{1}{5}\sum_{i=16}^{20}(f_4(x_i) - y_i)^2$
$f_5(x)$	$[\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_3, \mathbf{Z}_4]$	$\mathbf{Z}_5$	$r_5 = \frac{1}{5}\sum_{i=21}^{25}(f_5(x_i) - y_i)^2$
	Risk estimate		$R_{cv}(m) = \frac{1}{5}\sum_{i=1}^{5} r_i$

Example of Model Selection (5)

Table 3.4 Prediction risk estimates found using cross-validaton.

m	Estimated R via Cross-Validation
1	0.2000
2	0.1782
3	0.1886
4	0.1535
5	0.0726
6	0.1152
7	0.1649
8	0.0967
9	0.0944
10	0.5337