Notes on Statistical and Machine Learning

Model Assessment and Selection

Chapter: *10* **Prepared by:** *Chenxi Zhou*

This note is prepared based on *Chapter 7, Model Assessment and Selection* in Hastie, Tibshirani, and Friedman (2009). In this chapter, we study

1. how to *assess* the performance, in particular, the generalized performance, of a model, and
2. how to use these assessment to *select* model.

I. Bias, Variance and Model Complexity

1. **Generalization Performance:** The *generalization performance* of a learning method relates to its prediction capability on an *independent test data*. It is important since it guides the choice of learning method or model and gives a measure of the quality of the ultimately chosen model.
2. **Basic Setup:**
   (a) a target variable Y (assumed to be *continuous* for the moment),
   (b) a vector of inputs X,
   (c) a prediction model $\hat{f}$ estimated from a training dataset $\mathcal{T}$,
   (d) a loss function $L(Y, \hat{f}(X))$, measuring errors between Y and $\hat{f}(X)$. Typical choices are
   - *Squared Error:* $L(Y, \hat{f}(X)) = (Y - \hat{f}(X))^2$;
   - *Absolute Error:* $L(Y, \hat{f}(X)) = |Y - \hat{f}(X)|$.
3. **Test/Generalization Error:** *Test error* or *generalization error* is the prediction error over an independent test sample
$$\mathrm{Err}_{\mathcal{T}} := \mathbb{E}[L(Y, \hat{f}(X)) \,|\, \mathcal{T}] \tag{1}$$
where both X and Y are drawn randomly from their joint distribution (population).
Remark. In (1), the training set $\mathcal{T}$ is fixed, and test error refers to the error for this *specific* training set.
4. **Expected Prediction Error:** The *expected prediction error* (or *expected test error*) is
$$\mathrm{Err} := \mathbb{E}[L(Y, \hat{f}(X))] = \mathbb{E}[\mathrm{Err}_{\mathcal{T}}] \tag{2}$$

Remark 1. This expectation averages over *everything* that is random, including the randomness in the training set $\mathcal{T}$ that produced $\hat{f}$.

Remark 2. Estimation of $\mathrm{Err}_{\mathcal{T}}$ is the *goal*, but Err is more amenable to statistical analysis.

5. **Training Error:** *Training error* is the average loss over the training sample

$$\overline{\mathrm{err}} := \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{f}(\mathbf{x}_i)). \tag{3}$$

Remark. If the model we build is getting increasingly complex, we use the training dataset more to exploit the complicated underlying structures. Then,

(a) there is a tendency of lowering the bias and increasing the variance. There is some *intermediate* model complexity that gives the minimum expected test error;

(b) training error consistently decreases with model complexity and can drop to 0 if one increases the model complexity sufficiently. However, a model with 0 training error is *overfit* to the training set and is typically generalized *poorly*.

6. **Analogous Results for Categorical Response Variables:** Suppose the response variable is qualitative or categorical, denoted by G, that takes on one of W values in $\mathcal{W} := \{1, \cdots, W\}$. We model the probabilities

$$p_w(X) := \mathbb{P}(G = w \,|\, X),$$

and classify X according to the following rule

$$\widehat{G}(X) = \arg\max_{w \in \mathcal{W}} \hat{p}_w(X).$$

Some typical choices of loss functions are

- *0-1 loss:*

$$L(G, \widehat{G}(X)) = \mathbb{1}(G \neq \widehat{G}(X));$$

- $-2\times$*Log-likelihood loss:*

$$\begin{aligned} L(Y, \hat{p}(X)) &= -2\sum_{w=1}^{W} \mathbb{1}(G = w) \log \hat{p}_w(X) \\ &= -2 \log \hat{p}_G(X). \end{aligned}$$

The quantity "$-2\times$log-likelihood" is sometimes referred to the *deviance*.

For categorical variables, similar to before,

(a) the *test error*, $\text{Err}_{\mathcal{T}}$, is defined to be

$$\text{Err}_{\mathcal{T}} = \mathbb{E}\big[L(G, \widehat{G}(X)) \,|\, \mathcal{T}\big],$$

the population misclassification error of the classifier trained on $\mathcal{T}$;

(b) the *expected prediction error*, Err, is the expected misclassification error

$$\text{Err} = \mathbb{E}[\text{Err}_{\mathcal{T}}];$$

(c) the *training error* is

$$\overline{\text{err}} = \frac{1}{n}\sum_{i=1}^{n} L(g_i, \widehat{G}(\mathbf{x}_i)),$$

where $\{(\mathbf{x}_i, g_i)\}_{i=1}^n$ is the training dataset and $g_i \in \mathcal{W}$ is the observed class label for the i-th observation for all $i = 1, \cdots, n$.

7. **Log-Likelihood as a Loss Function:** The log-likelihood function can be used a loss function for general response density functions, such as Poisson, gamma, exponential, log-normal and so on. If $\mathbb{P}_{\theta(X)}$ is the density function of Y, where $\theta(X)$ is the parameter depending on the predictor X, then the loss function is

$$L(Y, \theta(X)) = -2 \cdot \log \mathbb{P}_{\theta(X)}(Y).$$

Remark. The "-2" in the front is to make the log-likelihood loss for the Gaussian distribution match the squared-error loss function.

8. **Model Selection vs. Model Assessment:** Typically, the model has a tuning parameter(s) α that affects the model complexity, and the predictions can be written as $\hat{f}_\alpha(\mathbf{x})$. We wish to determine the value of α minimizing the errors.

In this procedure, there are two separate goals:

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

If we have a sufficiently rich dataset, one can randomly split it into *three* parts:

- *Training Set:* used to fit the models;
- *Validation Set:* used to estimate prediction error for model selection;
- *Test Set:* used for the assessment of the generalization error of the final chosen model. This set should be brought out *only* at the end of the data analysis.

A typical split for three parts might be 50% for training, 25% each for validation and testing.

II. The Bias-Variance Decomposition

1. **The Bias-Variance Decomposition — General Case:** Assume that the model is of the form

$$Y = f(X) + \varepsilon,$$

where we assume $\mathbb{E}[\varepsilon] = 0$ and $\text{Var}[\varepsilon] = \sigma_\varepsilon^2$. Then, the *expected prediction error* of a regression fit $\hat{f}$ at $X = \mathbf{x}_0$ using the squared-error loss is derived as follows:

$$\begin{aligned}
\text{Err}(\mathbf{x}_0) &= \mathbb{E}\big[(Y - \hat{f}(\mathbf{x}_0))^2 \,\big|\, X = \mathbf{x}_0\big] \\
&= \mathbb{E}\big[(Y - f(\mathbf{x}_0) + f(\mathbf{x}_0) - \hat{f}(\mathbf{x}_0))^2 \,\big|\, X = \mathbf{x}_0\big] \\
&= \mathbb{E}\big[(Y - f(\mathbf{x}_0))^2 \,\big|\, X = \mathbf{x}_0\big] - 2\,\mathbb{E}\big[(Y - f(\mathbf{x}_0))(f(\mathbf{x}_0) - \hat{f}(\mathbf{x}_0)) \,\big|\, X = \mathbf{x}_0\big] \\
&\quad + \mathbb{E}\big[(f(\mathbf{x}_0) - \hat{f}(\mathbf{x}_0))^2 \,\big|\, X = \mathbf{x}_0\big] \\
&= \sigma_\varepsilon^2 + 0 + \mathbb{E}\big[(f(\mathbf{x}_0) - \hat{f}(\mathbf{x}_0))^2 \,\big|\, X = \mathbf{x}_0\big] \\
&= \sigma_\varepsilon^2 + \mathbb{E}\big[(f(\mathbf{x}_0) - \mathbb{E}[\hat{f}(\mathbf{x}_0)] + \mathbb{E}[\hat{f}(\mathbf{x}_0)] - \hat{f}(\mathbf{x}_0))^2 \,\big|\, X = \mathbf{x}_0\big] \\
&= \sigma_\varepsilon^2 + \mathbb{E}\big[(f(\mathbf{x}_0) - \mathbb{E}[\hat{f}(\mathbf{x}_0)])^2 \,\big|\, X = \mathbf{x}_0\big] + \mathbb{E}\big[(\mathbb{E}[\hat{f}(\mathbf{x}_0)] - \hat{f}(\mathbf{x}_0))^2 \,\big|\, X = \mathbf{x}_0\big] \\
&= \text{Irreducible Error} + \text{Bias}^2(\hat{f}(\mathbf{x}_0)) + \text{Var}[\hat{f}(\mathbf{x}_0)].
\end{aligned}$$

Analysis of the three terms:

- The *first term* is the variance of the target around its true mean $f(\mathbf{x}_0)$, and can not be avoided no matter how well we estimate $f(\mathbf{x}_0)$, unless $\sigma_\varepsilon^2 = 0$;
- The *second term* is the squared bias, the amount by which the average of our estimate $\hat{f}(\mathbf{x}_0)$ differs from the true mean $f(\mathbf{x}_0)$;
- The *last term* is the variance, the expected squared deviation of $\hat{f}(\mathbf{x}_0)$ around its mean.

Remark. Typically, the more complex we make the model, the *lower* the (squared) bias but the *higher* the variance.

2. **The Bias-Variance Decomposition for k-Nearest Neighbor Regression:** For the k-nearest-neighbor regression fit, the bias-variance decomposition has the form

$$\begin{aligned}
\text{Err}(\mathbf{x}_0) &= \mathbb{E}\big[(Y - \hat{f}_k(\mathbf{x}_0))^2 \,|\, X = \mathbf{x}_0\big] \\
&= \sigma_\varepsilon^2 + \left[f(\mathbf{x}_0) - \frac{1}{k}\sum_{l=1}^{k} f(\mathbf{x}_{(l)})\right]^2 + \frac{\sigma_\varepsilon^2}{k},
\end{aligned}$$

where we assume the training inputs $\mathbf{x}_i$ are fixed and the randomness arises from y_i's.

Remark. The number of neighbors k is *inversely* related to the model complexity.

- For small k, the estimate $\hat{f}(\mathbf{x})$ can potentially adapt itself better to the underlying $f(\mathbf{x})$; i.e., a smaller bias;

- As one increases k, the bias will typically increase and the variance decreases.

3. **The Bias-Variance Decomposition for Linear Regression Model:** Recall the linear model fit is $\hat{f}(\mathbf{x}) = \mathbf{x}^\top \widehat{\boldsymbol{\beta}}$, where $\widehat{\boldsymbol{\beta}} \in \mathbb{R}^p$ is the parameter estimated by the least squares method. Then,

$$\begin{aligned}\mathrm{Err}(\mathbf{x}_0) &= \mathbb{E}\big[(Y - \hat{f}(\mathbf{x}_0))^2 \,|\, X = \mathbf{x}_0\big] \\ &= \sigma_\varepsilon^2 + \mathbb{E}\big[f(\mathbf{x}_0) - \mathbb{E}[\hat{f}(\mathbf{x}_0)]\big]^2 + \sigma_\varepsilon^2 \cdot \|\mathbf{h}(\mathbf{x}_0)\|_2^2, \end{aligned} \tag{4}$$

where we assume the design matrix $\mathbf{X}$ is of the full rank and $\mathbf{h}(\mathbf{x}_0) = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}_0$.

We show $\mathrm{Var}[\hat{f}(\mathbf{x}_0)] = \sigma_\varepsilon^2 \cdot \|\mathbf{h}(\mathbf{x}_0)\|^2$. Recall that the least squares regression fit in the full rank case is of the form

$$\hat{f}(\mathbf{x}_0) = \mathbf{x}_0^\top \widehat{\boldsymbol{\beta}} = \mathbf{x}_0^\top (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}.$$

Then, its variance is

$$\begin{aligned}\mathrm{Var}[\hat{f}(\mathbf{x}_0)] &= \mathrm{Var}[\mathbf{x}_0^\top (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}] \\ &= \mathbf{x}_0^\top (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top \,\mathrm{Var}[\mathbf{Y}](\mathbf{x}_0^\top (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top)^\top \\ &= \mathbf{x}_0^\top (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top \cdot \sigma_\varepsilon^2\mathbf{I} \cdot \mathbf{X}(\mathbf{X}^\top\mathbf{X})\mathbf{x}_0 \\ &= \sigma_\varepsilon^2 \cdot \|\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}_0\|_2^2 \\ &= \sigma_\varepsilon^2 \cdot \|\mathbf{h}(\mathbf{x}_0)\|_2^2.\end{aligned}$$

Replacing $\mathbf{x}_0$ by $\mathbf{x}_i$ for all $i = 1, \cdots, n$ and taking the average, we have

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^n \mathrm{Var}[\hat{f}(\mathbf{x}_i)] &= \frac{1}{n}\sum_{i=1}^n \sigma_\varepsilon^2 \|\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}_i\|^2 \\ &= \frac{\sigma_\varepsilon^2}{n}\sum_{i=1}^n \mathbf{x}_i^\top (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}_i \\ &= \frac{\sigma_\varepsilon^2}{n}\sum_{i=1}^n \mathbf{x}_i^\top (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}_i \\ &= \frac{\sigma_\varepsilon^2}{n}\sum_{i=1}^n \mathrm{trace}(\mathbf{x}_i^\top (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}_i) \\ &= \frac{\sigma_\varepsilon^2}{n}\sum_{i=1}^n \mathrm{trace}(\mathbf{x}_i\mathbf{x}_i^\top (\mathbf{X}^\top\mathbf{X})^{-1}) \\ &= \frac{\sigma_\varepsilon^2}{n}\,\mathrm{trace}\left(\left(\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^\top\right)(\mathbf{X}^\top\mathbf{X})^{-1}\right) \\ &= \frac{\sigma_\varepsilon^2}{n}\,\mathrm{trace}\big((\mathbf{X}^\top\mathbf{X})(\mathbf{X}^\top\mathbf{X})^{-1}\big) \\ &= \frac{\sigma_\varepsilon^2}{n}\,\mathrm{trace}(\mathbf{I}_p) \\ &= \frac{p\sigma_\varepsilon^2}{n}.\end{aligned}$$

Therefore, the *in-sample error* is

$$\overline{\text{err}} = \frac{1}{n}\sum_{i=1}^{n}\text{Err}(\mathbf{x}_i) = \sigma_\varepsilon^2 + \frac{1}{n}\sum_{i=1}^{n}\big(f(\mathbf{x}_i) - \mathbb{E}[\hat{f}(\mathbf{x}_i)]\big)^2 + \frac{p}{n}\sigma_\varepsilon^2.$$

Notice that the complexity is directly related to the number of parameters p.

4. **The Bias-Variance Decomposition for Ridge Regression:** Consider the ridge regression where we solve the following optimization problem

$$\underset{\boldsymbol{\beta}}{\text{minimize}}\left\{(\mathbf{Y}-\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^\top\boldsymbol{\beta}\right\},$$

where $\lambda > 0$ is the tuning parameter and we assume there is no intercept term. The minimizer to the preceding optimization problem, denoted by $\widehat{\boldsymbol{\beta}}_\lambda$, is

$$\widehat{\boldsymbol{\beta}}_\lambda = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^\top\mathbf{Y}.$$

Let the prediction function be $\hat{f}_\lambda(\mathbf{x}) = \mathbf{x}^\top\widehat{\boldsymbol{\beta}}_\lambda$.

The test error has the same form as in (4), but the bias and variance components are different:

- *Bias:* Let $\boldsymbol{\beta}^*$ be the parameters of the best-fitting linear approximation to f, i.e.,

$$\boldsymbol{\beta}^* := \arg\min_{\boldsymbol{\beta}} \mathbb{E}\big[(f(X) - X^\top\boldsymbol{\beta})^2\big],$$

  where the expectation is taken with respect to the distribution of the input variables X. Then, the average squared bias is

$$\begin{aligned}&\mathbb{E}_{\mathbf{x}_0}\big[(f(\mathbf{x}_0) - \mathbb{E}[\hat{f}_\lambda(\mathbf{x}_0)])^2\big]\\ =\ &\mathbb{E}_{\mathbf{x}_0}\big[(f(\mathbf{x}_0) - \mathbf{x}_0^\top\boldsymbol{\beta}^*)^2\big] + \mathbb{E}_{\mathbf{x}_0}\big[(\mathbf{x}_0^\top\boldsymbol{\beta}^* - \mathbb{E}[\mathbf{x}_0^\top\widehat{\boldsymbol{\beta}}_\lambda])^2\big]\\ =\ &\text{Ave[Model Bias]}^2 + \text{Ave[Estimation Bias]}^2.\end{aligned}$$

  - The first term is the *average squared model bias*, the error between the best-fitting linear approximation and the true function;
  - The second term is the *average squared estimation bias*, the error between the average estimate $\mathbb{E}[\mathbf{x}_0^\top\widehat{\boldsymbol{\beta}}_\lambda]$ and the best-fitting linear approximation.

  Remark.

  (a) The model bias can only be reduced by *enlarging* the class of linear models to a richer collection of models, by including *interactions* and *transformations of the variables* in the model.

  (b) For linear models fit by ordinary least squares, the *estimation bias* is zero. For restricted fits (such as ridge regression) it is positive, and we trade it off with the benefits of a reduced variance.

- *Variance:* For the *variance* component, $\mathbf{h}$ function above becomes

$$\widetilde{\mathbf{h}}(\mathbf{x}_0) := \mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{x}_0,$$

and, hence,

$$\mathrm{Var}[\hat{f}(\mathbf{x}_0)] = \sigma_\varepsilon^2\|\widetilde{\mathbf{h}}(\mathbf{x}_0)\|_2^2 = \sigma_\varepsilon^2\mathbf{x}_0^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{x}_0.$$

Remark. The discussion above about the finer decomposition is *not* restricted to ridge regression but can be generalized to any restricted fits.

III. Optimism of the Training Error Rate

1. **Generalization Error:** Given a training set $\mathcal{T} := \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the *generalization error* of a model $\hat{f}$ is

$$\mathrm{Err}_\mathcal{T} := \mathbb{E}_{X^0,Y^0}\big[L(Y^0, \hat{f}(X^0)) \,|\, \mathcal{T}\big], \tag{5}$$

where the training dataset $\mathcal{T}$ here is considered to be *fixed* and (X^0, Y^0) is a new test data point drawn from the joint distribution of the data.

2. **Expected Prediction Error:** Based on the definition of generalization error above, the *expected prediction error* is obtained by averaging over the training set $\mathcal{T}$, that is,

$$\mathrm{Err} := \mathbb{E}_\mathcal{T}\big[\mathbb{E}_{X^0,Y^0}[L(Y^0, \hat{f}(X^0)) \,|\, \mathcal{T}]\big]. \tag{6}$$

Remark 1. The quantity Err is more amenable to statistical analysis.

Remark 2. It turns out that most methods effectively estimate the Err rather than $\mathrm{Err}_\mathcal{T}$.

3. **Training Error:** The *training error* is defined to be

$$\overline{\mathrm{err}} := \frac{1}{n}\sum_{i=1}^n L(y_i, \hat{f}(\mathbf{x}_i)). \tag{7}$$

Remark. Typically, $\overline{\mathrm{err}} \le \mathrm{Err}_\mathcal{T}$, since the same data $\mathcal{T}$ is used to fit the method and assess its error, and the model typically adapts to the training set, resulting in an *overly optimistic* estimate of the generalization error $\mathrm{Err}_\mathcal{T}$.

4. **Extra- and In- Sample Error:** The quantity $\mathrm{Err}_\mathcal{T}$ can be thought of as *extra-sample error* since the new test point (X^0, Y^0) does *not* coincide the training input vectors. The *in-sample error* is defined to be

$$\mathrm{Err}_{\mathrm{in}} = \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\mathbf{Y}^0}\big[L(Y_i^0, \hat{f}(\mathbf{x}_i)) \,|\, \mathcal{T}\big], \tag{8}$$

where $\mathbf{Y}^0 := (Y_1^0, \cdots, Y_n^0)^\top \in \mathbb{R}^n$ indicates that we observe n *new* response values at each of the training points $\mathbf{x}_i$, for all $i = 1, \cdots, n$.

5. **Optimism:** The *optimism* is defined to be the difference between $\mathrm{Err}_{\mathrm{in}}$ and the training error $\overline{\mathrm{err}}$,

$$\mathrm{op} := \mathrm{Err}_{\mathrm{in}} - \overline{\mathrm{err}}.$$

Remark. The optimism is typically positive as $\overline{\mathrm{err}}$ is usually biased downward as an estimate of prediction error.

6. **Average Optimism:** The *average optimism* is defined to be the expectation of the optimism over the training sets

$$\omega := \mathbb{E}_{\mathbf{Y}}[\mathrm{op}], \tag{9}$$

where $\mathbf{Y} := (y_1, y_2, \cdots, y_n)^\top \in \mathbb{R}^n$ and the predictors in the training dataset are viewed as fixed and the expectation is taken over the *training set outcome values*.

7. **Average Optimism for Squared Error Loss:** When the loss function is the squared error loss function, we have

$$\mathrm{Err}_{\mathrm{in}} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\mathbf{Y}^0}[(Y_i^0 - \hat{f}(\mathbf{x}_i))^2], \qquad \text{and} \qquad \overline{\mathrm{err}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(\mathbf{x}_i))^2.$$

We show that the average optimism is

$$\omega = \frac{2}{n}\sum_{i=1}^{n} \mathrm{Cov}(y_i, \hat{y}_i). \tag{10}$$

where Cov indicates the covariance.

First note the following

$$\begin{aligned}
\mathrm{Err}_{\mathrm{in}} &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\mathbf{Y}^0}\left[(Y_i^0 - f(\mathbf{x}_i) + f(\mathbf{x}_i) - \mathbb{E}[\hat{f}(\mathbf{x}_i)] + \mathbb{E}[\hat{f}(\mathbf{x}_i)] - \hat{f}(\mathbf{x}_i))^2\right] \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\mathbf{Y}^0}\Big[(Y_i^0 - f(\mathbf{x}_i))^2 + (f(\mathbf{x}_i) - \mathbb{E}[\hat{f}(\mathbf{x}_i)])^2 + (\mathbb{E}[\hat{f}(\mathbf{x}_i)] - \hat{f}(\mathbf{x}_i))^2 \\
&\qquad + 2(Y_i^0 - f(\mathbf{x}_i))(f(\mathbf{x}_i) - \mathbb{E}[\hat{f}(\mathbf{x}_i)]) + 2(Y_i^0 - f(\mathbf{x}_i))(\mathbb{E}[\hat{f}(\mathbf{x}_i)] - \hat{f}(\mathbf{x}_i)) \\
&\qquad\qquad + 2(f(\mathbf{x}_i) - \mathbb{E}[\hat{f}(\mathbf{x}_i)])(\mathbb{E}[\hat{f}(\mathbf{x}_i)] - \hat{f}(\mathbf{x}_i))\Big] \\
&= \frac{1}{n}\sum_{i=1}^{n} \Big[\sigma_\varepsilon^2 + (f(\mathbf{x}_i) - \mathbb{E}[\hat{f}(\mathbf{x}_i)])^2 + (\mathbb{E}[\hat{f}(\mathbf{x}_i)] - \hat{f}(\mathbf{x}_i))^2 \\
&\qquad + 2(f(\mathbf{x}_i) - \mathbb{E}[\hat{f}(\mathbf{x}_i)])(\mathbb{E}[\hat{f}(\mathbf{x}_i)] - \hat{f}(\mathbf{x}_i))\Big].
\end{aligned}$$

Also, note the following

$$\begin{aligned}
\overline{\text{err}} &= \frac{1}{n}\sum_{i=1}^{n}\big(y_i - f(\mathbf{x}_i) + f(\mathbf{x}_i) - \mathbb{E}[\hat{f}(\mathbf{x}_i)] + \mathbb{E}[\hat{f}(\mathbf{x}_i)] - \hat{f}(\mathbf{x}_i)\big)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\Big[(y_i - f(\mathbf{x}_i))^2 + (f(\mathbf{x}_i) - \mathbb{E}[\hat{f}(\mathbf{x}_i)])^2 + (\mathbb{E}[\hat{f}(\mathbf{x}_i)] - \hat{f}(\mathbf{x}_i))^2 \\
&\qquad + 2(y_i - f(\mathbf{x}_i))(f(\mathbf{x}_i) - \mathbb{E}[\hat{f}(\mathbf{x}_i)]) + 2(y_i - f(\mathbf{x}_i))(\mathbb{E}[\hat{f}(\mathbf{x}_i)] - \hat{f}(\mathbf{x}_i)) \\
&\qquad\qquad + 2(f(\mathbf{x}_i) - \mathbb{E}[\hat{f}(\mathbf{x}_i)])(\mathbb{E}[\hat{f}(\mathbf{x}_i)] - \hat{f}(\mathbf{x}_i))^2\Big] \\
&= \frac{1}{n}\sum_{i=1}^{n}\Big[\varepsilon_i^2 + (f(\mathbf{x}_i) - \mathbb{E}[\hat{f}(\mathbf{x}_i)])^2 + (\mathbb{E}[\hat{f}(\mathbf{x}_i)] - \hat{f}(\mathbf{x}_i))^2 \\
&\qquad + 2\varepsilon_i(f(\mathbf{x}_i) - \mathbb{E}[\hat{f}(\mathbf{x}_i)]) + 2\varepsilon_i(\mathbb{E}[\hat{f}(\mathbf{x}_i)] - \hat{f}(\mathbf{x}_i)) \\
&\qquad\qquad + 2(f(\mathbf{x}_i) - \mathbb{E}[\hat{f}(\mathbf{x}_i)])(\mathbb{E}[\hat{f}(\mathbf{x}_i)] - \hat{f}(\mathbf{x}_i))^2\Big].
\end{aligned}$$

Therefore, the optimism is given by

$$\begin{aligned}
\text{op} &= \text{Err}_{\text{in}} - \overline{\text{err}} \\
&= \sigma_\varepsilon^2 - \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i^2 - \frac{2}{n}\sum_{i=1}^{n}\varepsilon_i(f(\mathbf{x}_i) - \mathbb{E}[\hat{f}(\mathbf{x}_i)]) - \frac{2}{n}\sum_{i=1}^{n}\varepsilon_i(\mathbb{E}[\hat{f}(\mathbf{x}_i)] - \hat{f}(\mathbf{x}_i)).
\end{aligned}$$

As a consequence, the average optimism is

$$\begin{aligned}
\mathbb{E}_{\mathbf{Y}}[\text{op}] &= -\frac{2}{n}\sum_{i=1}^{n}\mathbb{E}_{\mathbf{Y}}\big[\varepsilon_i(\mathbb{E}[\hat{f}(\mathbf{x}_i)] - \hat{f}(\mathbf{x}_i))\big] \\
&= \frac{2}{n}\sum_{i=1}^{n}\mathbb{E}_{\mathbf{Y}}\big[(y_i - \mathbb{E}[y_i])(\hat{f}(\mathbf{x}_i)) - \mathbb{E}[\hat{f}(\mathbf{x}_i)]\big] \\
&= \frac{2}{n}\sum_{i=1}^{n}\text{Cov}(y_i, \hat{y}_i).
\end{aligned}$$

Remark 1. For 0-1 and some other loss functions, we also have the average optimism is given by (10).

Remark 2. Equation (10) suggests that the amount by which $\overline{\text{err}}$ underestimates the true error depends on how strongly y_i affects its own prediction.

8. **Relationship Among In-sample Error, Training Error and Average Optimism:** Combining all results above, we have

$$\mathbb{E}_{\mathbf{Y}}[\text{Err}_{\text{in}}] = \mathbb{E}_{\mathbf{Y}}[\overline{\text{err}}] + \frac{2}{n}\sum_{i=1}^{n}\text{Cov}(\hat{y}_i, y_i). \tag{11}$$

Remark. Suppose we work with linear regression and $\hat{y}_i$ is obtained by a linear fit with p inputs or basis functions, for example, for the additive error model $Y = f(X) + \varepsilon$, we have

$$\sum_{i=1}^{n} \text{Cov}(\hat{y}_i, y_i) = p\sigma_\varepsilon^2,$$

and (11) simplfies to

$$\mathbb{E}_{\mathbf{Y}}[\text{Err}_{\text{in}}] = \mathbb{E}_{\mathbf{Y}}[\overline{\text{err}}] + \frac{2p}{n}\sigma_\varepsilon^2. \tag{12}$$

This result says that the optimism increases linearly with the number of inputs or basis functions, p, and decreases with the training sample size.

9. **Approaches to Estimation of the Prediction Error:**
   (a) *Approach 1:* Estimate the optimism and add it to the training error $\overline{\text{err}}$. Examples include C_p, AIC, BIC;
   (b) *Approach 2:* Estimate the extra-sample error Err directly. Examples include cross-validation and bootstrap.

IV. Estimates of In-Sample Prediction Error

1. **In-Sample Error Estimate:** The general form of the in-sample error estimate is

$$\widehat{\text{Err}}_{\text{in}} = \overline{\text{err}} + \hat{\omega},$$

where $\hat{\omega}$ is an estimate of the average optimism.

2. **Mallow's C_p Statistic:** Based on the expression (12), where p parameters are fit under the squared error loss function, we obtain the C_p statistic as

$$C_p = \overline{\text{err}} + \frac{2p}{n}\hat{\sigma}_\varepsilon^2, \tag{13}$$

where $\hat{\sigma}_\varepsilon^2$ is an estimate of the variance.

3. **Akaike Information Criterion (AIC):** The *Akaike Information Criterion* (AIC) is a more general estimate of Err_{in} when a log-likelihood loss function is used.
   (a) *Theoretical Basis:* The development of AIC relies on the asymptotic result

$$-2\,\mathbb{E}[\log \mathbb{P}_{\hat{\theta}}(Y)] \approx -\frac{2}{n}\,\mathbb{E}[\text{log-likelihood}] + \frac{2p}{n},$$

where $\mathbb{P}_\theta$ is a family of densities for Y containing the "true" density of it, $\hat{\theta}$ is the maximum-likelihood estimate of θ, and "log-likelihood" is the maximized log-likelihood given by

$$\text{log-likelihood} = \sum_{i=1}^{n} \log \mathbb{P}_{\hat{\theta}}(y_i).$$

(b) *Examples:*

- For the *logistic regression* using the binomial log-likelihood, we have

$$\text{AIC} = -\frac{2}{n}\text{log-likelihood} + \frac{2p}{n}. \tag{14}$$

- For the *Gaussian regression* with variance $\sigma_\varepsilon^2 = \hat{\sigma}_\varepsilon^2$ assumed to be *known*, the AIC statistics is equivalent to C_p.

(c) *Using AIC for Model Selection:* To use the AIC for model selection, we choose the one with the *smallest* AIC over the set of models considered.

4. **AIC for Model Selection in General Setting:** For nonlinear and more complex models, we need to replace p appearing in AIC by some measure of model complexity. Given a family of models f_α indexed by the tuning parameter α, we let $\overline{\text{err}}(\alpha)$ be the training error and $p(\alpha)$ be the number of parameters for each model. Then, the AIC, depending on α, is defined to be

$$\text{AIC}(\alpha) = \overline{\text{err}}(\alpha) + \frac{2p(\alpha)}{n}\hat{\sigma}_\varepsilon^2, \tag{15}$$

which provides an estimate of the *test error curve*. We find the tuning parameter $\hat{\alpha}$ that minimizes $\text{AIC}(\alpha)$ and the final model is $f_{\hat{\alpha}}$.

5. **A Remark:** The formula

$$\frac{2}{n}\sum_{i=1}^{n}\text{Cov}(\hat{y}_i, y_i) = \frac{2p}{n}\sigma_\varepsilon^2$$

exactly holds for linear models with additive errors and squared error loss, and *approximately* for linear models and log-likelihoods.

In particular, it does *not* hold in general for the 0-1 loss function.

V. The Effective Number of Parameters

1. **Overview:** We generalize the concept of the "number of parameters", especially in the context of regularized model fitting.

2. **Notation:** We let

$$\mathbf{Y} = \begin{pmatrix} y_1 & y_2 & \cdots & y_n \end{pmatrix}^\top \in \mathbb{R}^n$$

to be the vector of responses and

$$\widehat{\mathbf{Y}} = \begin{pmatrix} \hat{y}_1 & \hat{y}_2 & \cdots & \hat{y}_n \end{pmatrix}^\top \in \mathbb{R}^n$$

be the vector of the fitted values.

3. **Linear Fitting Method:** A method is said to a *linear fitting method* if we can write

$$\widehat{\mathbf{Y}} = \mathbf{S}\mathbf{Y},$$

where $\mathbf{S} \in \mathbb{R}^{n \times n}$ is a matrix depending *not* on the input vectors $\mathbf{x}_i$'s but *not* on y_i's.

4. **Effective Number of Parameters for Linear Fitting Methods:** The *effective number of parameters* for linear fitting methods is defined to be

$$\mathrm{df}(\mathbf{S}) = \mathrm{trace}(\mathbf{S}) = \sum_{i=1}^{n} \mathbf{S}_{i,i},$$

where $\mathbf{S}_{i,i}$ is the i-th diagonal element of $\mathbf{S}$. The quantity df is also known as the *effective degrees-of-freedom*.

5. **Examples:**

- If $\mathbf{S}$ is an *orthogonal-projection matrix* onto a basis set spanned by M features, then $\mathrm{trace}(\mathbf{S}) = M$;
- If $\mathbf{Y}$ arises from an additive-error model

$$Y = f(X) + \varepsilon, \qquad \text{where } \mathrm{Var}[\varepsilon] = \sigma_\varepsilon^2,$$

then

$$\sum_{i=1}^{n} \mathrm{Cov}(\hat{y}_i, y_i) = \mathrm{trace}(\mathbf{S})\sigma_\varepsilon^2,$$

which motivates the more general definition

$$\mathrm{df}(\widehat{\mathbf{Y}}) = \frac{1}{\sigma_\varepsilon^2} \sum_{i=1}^{n} \mathrm{Cov}(\hat{y}_i, y_i);$$

- In the context of neural networks where we minimize an error function $R(\mathbf{w})$ with the weight decay penalty $\alpha \mathbf{w}^\top \mathbf{w} = \alpha \sum_{m=1}^{M} w_m^2$, the effective number of parameters is of the form

$$\mathrm{df}(\alpha) = \sum_{m=1}^{M} \frac{\theta_m}{\theta_m + \alpha},$$

where θ_m's are the eigenvalues of $\dfrac{\partial^2 R(\mathbf{w})}{\partial \mathbf{w} \partial \mathbf{w}^\top}$.

VI. The Bayesian Approach and BIC

1. **Introduction to BIC:** The *Bayesian Information Criterion (BIC)* is applicable when the fitting is carried out by maximization of a log-likelihood function, which has the general form

$$\text{BIC} = -2 \times \text{log-likelihood} + p \log n. \tag{16}$$

2. **BIC under the Gaussian Model:** Under the Gaussian model and the assumption that σ_ε^2 is known, we have

$$\begin{aligned} -2 \times \text{log-likelihood} &= \frac{1}{\sigma_\varepsilon^2} \sum_{i=1}^{n} (y_i - \hat{f}(\mathbf{x}_i))^2 \qquad \text{up to a constant} \\ &= \frac{n}{\sigma_\varepsilon^2} \overline{\text{err}}. \end{aligned}$$

   Hence, we can write

$$\text{BIC} = \frac{n}{\sigma_\varepsilon^2} \left[\overline{\text{err}} + (\log n) \frac{p}{n} \sigma_\varepsilon^2 \right], \tag{17}$$

   which is proportional to the AIC with the factor 2 replaced by $\log n$. Therefore, BIC tends to penalize complex models *more heavily* and gives preference to simpler models.

3. **Derivation of BIC:** The BIC is derived from a Bayesian approach to model selection. Assume that

   - we have a set of candidate models $\mathcal{M}_m$, for $m = 1, \cdots, M$, with the corresponding model parameters θ_m, and we wish to choose a best model from them; and
   - we have a prior distribution $\mathbb{P}(\theta_m \,|\, \mathcal{M}_m)$ for the parameters in each model $\mathcal{M}_m$.

   Then, letting $\mathbf{Z} := \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be the training data and applying Bayes Theorem, the posterior probability of a given model is

$$\begin{aligned} \mathbb{P}(\mathcal{M}_m \,|\, \mathbf{Z}) &\propto \mathbb{P}(\mathcal{M}_m) \cdot \mathbb{P}(\mathbf{Z} \,|\, \mathcal{M}_m) \\ &\propto \mathbb{P}(\mathcal{M}_m) \int \mathbb{P}(\mathbf{Z} \,|\, \mathcal{M}_m, \theta_m)(\theta_m \,|\, \mathcal{M}_m) \, \mathrm{d}\theta_m. \end{aligned}$$

   To compare two distinct models $\mathcal{M}_m$ and $\mathcal{M}_l$, with $m \neq l$, we take the ratio of their posterior probabilities

$$\mathcal{R} := \frac{\mathbb{P}(\mathcal{M}_m \,|\, \mathbf{Z})}{\mathbb{P}(\mathcal{M}_l \,|\, \mathbf{Z})} = \frac{\mathbb{P}(\mathcal{M}_m)}{\mathbb{P}(\mathcal{M}_l)} \cdot \frac{\mathbb{P}(\mathbf{Z} \,|\, \mathcal{M}_m)}{\mathbb{P}(\mathbf{Z} \,|\, \mathcal{M}_l)}. \tag{18}$$

   We choose $\mathcal{M}_m$ if $\mathcal{R} > 1$ and $\mathcal{M}_l$ otherwise.

4. **Bayes Factor:** The quantity appearing in (18)

$$\mathrm{BF}(\mathbf{Z}) := \frac{\mathbb{P}(\mathbf{Z} \,|\, \mathcal{M}_m)}{\mathbb{P}(\mathbf{Z} \,|\, \mathcal{M}_l)} \tag{19}$$

is called the *Bayes Factor*, which is interpreted as "the contribution of the data toward the posterior odds".

5. **Laplace Approximation of $\mathbb{P}(\mathbf{Z} \,|\, \mathcal{M}_m)$:** Typically, one assumes that the prior probability distribution over the models $\{\mathcal{M}_m\}_{m=1}^M$ is uniform so that $\mathbb{P}(\mathcal{M}_m) = \frac{1}{M}$ is constant for all $m = 1, 2, \cdots, M$. And one needs to approximate $\mathbb{P}(\mathbf{Z} \,|\, \mathcal{M}_m)$. One way to perform this approximation is the *Laplace approximation*

$$\log \mathbb{P}(\mathbf{Z} \,|\, \mathcal{M}_m) = \log(\mathbf{Z} \,|\, \hat{\theta}_m, \mathcal{M}_m) - \frac{p_m}{2} \log n + \mathcal{O}(1), \tag{20}$$

where $\hat{\theta}_m$ is a maximum likelihood estimate and p_m is the number of free parameters in model $\mathcal{M}_m$.

If we let the loss function be

$$-2 \times \log \mathbb{P}(\mathbf{Z} \,|\, \hat{\theta}_m, \mathcal{M}_m),$$

the resulting quantity is equivalent to (16).

6. **BIC for Model Selection:** We choose the model $\mathcal{M}_{m^*}$ that gives the smallest BIC among all $\mathcal{M}_1, \cdots, \mathcal{M}_M$.
From (20), we see that choosing the model with the *minimum BIC* is equivalent to choosing the model with the largest (approximate) posterior probability.

7. **Posterior Probabilities for $\{\mathcal{M}_m\}_{m=1}^M$:** If one is able to compute the BIC criterion for a set of M models, denoted by BIC_m, for all $m = 1, \cdots, M$, one can estimate the posterior probability of model $\mathcal{M}_m$ as

$$\frac{\exp(-\frac{1}{2}\mathrm{BIC}_m)}{\sum_{l=1}^m \exp(-\frac{1}{2}\mathrm{BIC}_l)}.$$

8. **A Comparison between AIC and BIC:**

   - For model selection purposes, there is no clear choice between AIC and BIC;
   - BIC is asymptotically consistent as a selection criterion in the sense that "*given a family of models, including the true model, the probability that BIC will select the correct model approaches one as the sample size* $n \to \infty$", and AIC does *not* have this property and tends to choose too complex models as $n \to \infty$;
   - For finite samples, BIC tends to choose models that are *too* simple, due to the heavy penalty on complexity.

VII. Cross-validation

1. **Overview:** *Cross-validation* directly estimates the *expected extra-sample error*

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

the average generalization error when $\hat{f}$ is applied to an independent test sample from the joint distribution of X and Y.

2. **K-Fold Cross-Validation:**

   (a) *Motivation:* Ideally, when we have sufficient data, we set aside a validation set and use it to assess the performance of the prediction model. Since data are often scarce, this is *not* usually possible.

   (b) *Main Idea:* The K-fold cross-validation uses part of data to fit the model and a different set to assess the model.

   (c) *Procedure:*

   - We split the dataset into K roughly equal-sized parts. K-fold cross-validation uses $(K-1)$ parts, excluding the k-th part, to fit the model and the remaining k-th part to test it. With the remaining part of data, we calculate the prediction error of the fitted model. We do this for $k = 1, 2, \cdots, K$ and combine the K estimates of the prediction error;
   - Let $\kappa : \{1, \cdots, n\} \to \{1, \cdots, K\}$ be an indexing function that indicates the partition to which observation i is allocated by *randomization*. Let $\hat{f}^{-k}$ denote the fitted function, computed without the k-th part of data. The cross-validation estimate of the prediction error is

$$\text{CV}(\hat{f}) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \hat{f}^{-\kappa(i)}(\mathbf{x}_i)). \tag{21}$$

   (d) *Remarks:*

   - Typical choices of K are 5 or 10.
   - The case when $K = n$ is known as *leave-one-out cross-validation*, and in this case, $\kappa(i) = i$.

3. **Using K-Fold CV for Model Selection:** Consider a set of models $\{f_\alpha\}_\alpha$ with $\mathbf{x}$ being the predictors and α being the tuning parameter. Let $\hat{f}_\alpha^{-k}$ be the α-th model fit with the k-th part of the data removed. Then, the following quantity

$$\text{CV}(\hat{f}_\alpha) := \frac{1}{n} \sum_{i=1}^{n} L(y_i, \hat{f}_\alpha^{-\kappa(i)}(\mathbf{x}_i)) \tag{22}$$

provides an estimate of the *test error curve* (as a function of α). We choose the value of α that minimizes $\text{CV}(\hat{f}_\alpha)$, say $\hat{\alpha}$. The final model is $f_{\hat{\alpha}}$.

4. Discussion on Choice of K in K-Fold Cross-Validation:

(a) With $\underline{K = n}$,

i. the cross-validation estimator is approximately unbiased for the true *expected predictor error*, but can have *high variance*;

ii. the computation burden is considerable.

(b) With a smaller value of K, say $K = 5$ or 10,

i. cross-validation has *lower variance* but may have *large bias*, which depends on how the performance of the learning method varies with the size of the training set;

ii. if the learning curve (plotting cross-validation error against size of training set) has a considerable slope at the given training set size, 5- or 10-fold cross-validation may overestimate the true prediction error.

Conclusion: 5- or 10-fold cross-validation is recommended.

5. Generalized Cross-Validation: *Generalized cross-validation (GCV)* provides a convenient approximation to leave-one-out cross-validation for *linear fitting* under the *squared-error loss function*.

(a) *Review of linear fitting methods:* A linear fitting method is the one for which we can write

$$\widehat{\mathbf{Y}} = \mathbf{S}\mathbf{Y},$$

where $\mathbf{Y} := (y_1, \cdots, y_n)^\top \in \mathbb{R}^n$ is the vector of the observed response variable, $\widehat{\mathbf{Y}} := (\hat{y}_1, \cdots, \hat{y}_n)^\top \in \mathbb{R}^n$ is the vector of the fitted values, and $\mathbf{S} \in \mathbb{R}^{n \times n}$.

(b) *Leave-one-out cross-validation relationship:* For many linear fitting methods, including the least squares method and cubic smoothing splines, we have

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}^{-i}(\mathbf{x}_i))^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}(\mathbf{x}_i)}{1 - \mathbf{S}_{i,i}}\right)^2,$$

where $\mathbf{S}_{i,i}$ is the i-th diagonal element of the matrix $\mathbf{S}$.

(c) *GCV approximation:* The *GCV approximation* is defined to be

$$\mathrm{GCV}(\hat{f}) := \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}(\mathbf{x}_i)}{1 - \mathrm{trace}(\mathbf{S})/n}\right)^2, \tag{23}$$

where the quantity $\mathrm{trace}(\mathbf{S})$ is the effective number of parameters.

(d) *Comparison between GCV and CV:* The GCV has the computing advantage in the settings where the trace of $\mathbf{S}$ is easier to compute than the individual elements of $\mathbf{S}$.

(e) *Connection between GCV and C_p and AIC:* Using the approximation $\frac{1}{(1-x)^2} \approx 1+2x$, we have

$$\begin{aligned}\text{GCV} &\approx \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(\mathbf{x}_i))^2\left(1+\frac{2\,\text{trace}(\mathbf{S})}{n}\right)\\ &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(\mathbf{x}_i))^2 + \frac{2\,\text{trace}(\mathbf{S})}{n}\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(\mathbf{x}_i))^2\right). \qquad (24)\end{aligned}$$

In (24), the first term $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(\mathbf{x}_i))^2$ is the training error, $\overline{\text{err}}$, when the squared-error loss function is used. In the second term, $\hat{\sigma}_\varepsilon^2 \approx \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(\mathbf{x}_i))^2$, and $\text{trace}(\mathbf{S}) = p$. Combining all pieces together, we see

$$\text{GCV} \approx C_p.$$

Assume the random error term has a Gaussian distribution and $\text{Var}[\varepsilon] = \sigma_\varepsilon^2$ is known, we have $\text{GCV} \approx \sigma_\varepsilon^2 \times \text{AIC}$.

6. **Correct Way of Doing Cross-Validation in a Multistep Modeling Procedures:** In a multistep modeling procedure, cross-validation must be applied to the *entire* sequence of modeling steps; in particular, samples must be "left out" **before** any selection or filtering steps are applied.

VIII. Bootstrap Methods

1. **Overview:** *Bootstrap method* is a general tool for assessing statistical accuracy, and can be use to estimate extra-sample prediction error. It can estimate well the *expected prediction error*.

2. **Setup:** Let $\mathbf{Z} := \{z_i\}_{i=1}^n$ denote the training sets, where $z_i := (\mathbf{x}_i, y_i)$ for all $i = 1, \cdots, n$.

3. **Procedures:**

   (a) *Randomly* draw datasets *with replacement* from $\mathbf{Z}$, each sample the same size as the original training set;

   (b) Repeat the preceding step for B times, producing B bootstrap datasets;

   (c) Refit the model to each of the bootstrap datasets, and examine the behavior of the fits over the B replications;

   (d) Suppose $S(\mathbf{Z})$ is some quantity of interest computed from the data $\mathbf{Z}$. Compute $S(\mathbf{Z}^b)$ for all $b = 1, \cdots, B$, where $\mathbf{Z}^b$ denotes the b-th bootstrapped sample. We can use $S(\mathbf{Z}^1), S(\mathbf{Z}^2), \cdots, S(\mathbf{Z}^B)$ to assess the statistical accuracy of $S(\mathbf{Z})$.

4. **Example:** Suppose we want to estimate the variance of $S(\mathbf{Z})$ using the bootstrap. The bootstrap estimate of the variance of $S(\mathbf{Z})$ is

$$\widehat{\text{Var}}[S(\mathbf{Z})] = \frac{1}{B-1}\sum_{b=1}^{B}\big(S(\mathbf{Z}^b) - \bar{S}^*\big)^2,$$

where $\bar{S}^* := \frac{1}{B}\sum_{b=1}^{B} S(\mathbf{Z}^b)$.

Remark. The quantity $\widehat{\text{Var}}[S(\mathbf{Z})]$ can be thought of as a *Monte-Carlo estimate* of the variance of $S(\mathbf{Z})$ under sampling from the empirical distribution function $\widehat{F}$ for the data $(z_1, \cdots, z_n)$.

5. **Estimation of Prediction Error Using the Bootstrap — Method 1:** Fit the model in question on a set of bootstrap samples. Let $\hat{f}^{*b}(\mathbf{x}_i)$ be the predicted value at $\mathbf{x}_i$, from the model fitted to the b-th bootstrapped dataset. The estimate of the prediction error is

$$\widehat{\text{Err}}_{\text{boot}} := \frac{1}{B}\frac{1}{n}\sum_{b=1}^{B}\sum_{i=1}^{n} L(y_i, \hat{f}^{*b}(\mathbf{x}_i)). \tag{25}$$

Comments: This is a *bad* estimate. Note that the training samples (the bootstrapped datasets) and the test samples (the original training dataset) have observations in common. This overlap can make overfit predictions look unrealistically good.

6. **Estimation of Prediction Error Using the Bootstrap — Method 2:** For each observation, we only keep track of predictions from bootstrapped samples *not* containing that observation. The *leave-one-out bootstrap estimate of prediction error* is

$$\widehat{\text{Err}}^{(1)}_{\text{boot}} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|C^{-i}|}\sum_{b\in C^{-i}} L(y_i, \hat{f}^{*b}(\mathbf{x}_i)), \tag{26}$$

where C^{-i} is the set of indices of the bootstrap samples b that do *not* contain the observation i, and $|C^{-i}|$ is the number of such samples.

Remark. In computing $\widehat{\text{Err}}^{(1)}_{\text{boot}}$, we either

- have to choose B large enough to ensure that all of the $|C^{-i}|$ are greater than zero, or
- just leave out the terms in (26) corresponding to $|C^{-i}|$'s that are zero.

7. **"0.632 Estimator":**

   (a) *Motivation:* The leave-one out bootstrap estimate of the prediction error $\widehat{\text{Err}}^{(1)}_{\text{boot}}$ has the *training-set-size bias* problem discussed in the cross-validation. The average number of distinct observations in each bootstrap sample is about $0.632 \times n$,

where

$$\begin{aligned}\mathbb{P}\Big(\text{ observation } i \in \text{ bootstrap samples }\Big) &= 1 - \left(1 - \frac{1}{n}\right)^n \\ &\approx 1 - e^{-1} \\ &\approx 0.632.\end{aligned}$$

If the learning curve has considerable slope at sample size roughly $n/2$, the leave-one out bootstrap estimate will be biased *upward* as an estimate of the true error.

(b) *"0.632 Estimator":* The "*0.632 estimator*" is designed to alleviate this upward bias and is defined to be

$$\widehat{\text{Err}}_{\text{boot}}^{(0.632)} = 0.368 \times \overline{\text{err}} + 0.632 \times \widehat{\text{Err}}_{\text{boot}}^{(1)}. \tag{27}$$

The *main idea* is that the "0.632 estimator" pulls the leave-one out bootstrap estimate down toward the training error rate, and hence reduces its upward bias.

8. No-information Error Rate: The *no-information error rate* is defined to be the error rate of our prediction rule as if the inputs and the outputs were independent. We denote it by γ.

(a) *Estimation:* An estimate of γ can be obtained by evaluating the prediction rule on *all* possible combinations of targets y_i and predictors $\mathbf{x}_{i'}$

$$\hat{\gamma} := \frac{1}{n^2}\sum_{i=1}^{n}\sum_{i'=1}^{n} L(y_i, f(\mathbf{x}_{i'})). \tag{28}$$

(b) *Example — Binary classification problem:* Consider a binary classification problem with class labels being $\{+1, -1\}$ and the loss function being the 0-1 loss. Let $\hat{p}_1$ be the observed proportion of responses equaling 1, and $\hat{q}$ be the observed proportion of predictions $f(\mathbf{x}_{i'})$ equal to 1. Then, we show

$$\hat{\gamma} = \hat{p}(1-\hat{q}) + (1-\hat{p})\hat{q}.$$

Note the following

$$\begin{aligned}\hat{\gamma} &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{i'=1}^{n} L\big(y_i, \hat{f}(\mathbf{x}_{i'})\big) \\ &= \frac{1}{n^2}\sum_{i'=1}^{n}\sum_{\{i\,|\,y_i=+1\}} \mathbb{1}\big(\hat{f}(\mathbf{x}_{i'}) \neq +1\big) + \frac{1}{n^2}\sum_{i'=1}^{n}\sum_{\{i\,|\,y_i=-1\}} \mathbb{1}\big(\hat{f}(\mathbf{x}_{i'}) \neq -1\big) \\ &= \frac{1}{n^2}\big|\{i\,|\,y_i=+1\}\big|\sum_{i'=1}^{n} \mathbb{1}\big(\hat{f}(\mathbf{x}_{i'}) \neq +1\big) + \frac{1}{n^2}\big|\{i\,|\,y_i=-1\}\big|\sum_{i'=1}^{n} \mathbb{1}\big(\hat{f}(\mathbf{x}_{i'}) \neq -1\big) \\ &= \frac{1}{n^2}\big|\{i\,|\,y_i=+1\}\big| \cdot \big|\{i'\,|\,\hat{f}(\mathbf{x}_{i'}) = -1\}\big| + \frac{1}{n^2}\big|\{i\,|\,y_i=-1\}\big| \cdot \big|\{i'\,|\,\hat{f}(\mathbf{x}_{i'}) = +1\}\big| \\ &= \hat{p}(1-\hat{q}) + (1-\hat{p})\hat{q}.\end{aligned}$$

9. **Relative Overfitting Rate:** The *relative overfitting rate* is defined to be

$$\widehat{R} := \frac{\widehat{\text{Err}}^{(1)}_{\text{boot}} - \overline{\text{err}}}{\hat{\gamma} - \overline{\text{err}}}. \tag{29}$$

Note that $\widehat{R}$ ranges from

- 0, if there is no overfitting and $\widehat{\text{Err}}^{(1)}_{\text{boot}} = \overline{\text{err}}$, to
- 1, if the overfitting is equal to $\hat{\gamma} - \overline{\text{err}}$.

10. **"0.632+ Estimator":** The "*0.632+ estimator*" is defined to be

$$\widehat{\text{Err}}^{(0.632+)}_{\text{boot}} = (1 - \hat{w}) \times \overline{\text{err}} + \hat{w} \times \widehat{\text{Err}}^{(1)}_{\text{boot}}, \tag{30}$$

where

$$\hat{w} = \frac{0.632}{1 - 0.367\widehat{R}}.$$

(a) *Analysis of $\hat{w}$ and $\widehat{\text{Err}}^{(0.632+)}_{\text{boot}}$:* Note that the weight $\hat{w}$ ranges from 0.632 if $\widehat{R} = 0$ to 1 if $\widehat{R} = 1$; and $\widehat{\text{Err}}^{(0.632+)}_{\text{boot}}$ ranges from $\widehat{\text{Err}}^{(0.632)}_{\text{boot}}$ to $\widehat{\text{Err}}^{(1)}_{\text{boot}}$.

(b) *General Idea of $\widehat{\text{Err}}^{(0.632+)}_{\text{boot}}$:* The quantity $\widehat{\text{Err}}^{(0.632+)}_{\text{boot}}$ produces a compromise between the leave-one-out bootstrap and the training error rate that depends on the amount of overfitting.

References

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning*. Vol. 1. Springer Series in Statistics. New York, NY, USA: Springer New York Inc.