13
Estimator and Model Selection Using Cross-Validation

Iván Díaz

CONTENTS

13.1 Introduction ............................................. 223
13.2 Classification of Chemicals ................................ 227
13.3 Estimation of the Prediction Error ...................... 227
13.3.1 Cross-Validation with Correlated Units .......... 228
13.4 Discrete Model Selection .................................. 228
13.5 Model Stacking and Super Learning .................... 231
13.5.1 Assessing the Prediction Error of a Stacked Predictor 232
13.6 Asymptotic Results for the Cross-Validated Estimator Selector and the Cross-Validated Risk ................. 233
13.6.1 Asymptotic Optimality of the Cross-Validation Selector for Prediction ........................................... 233
13.6.2 Asymptotic Inference for the Cross-Validated Risk Estimator .................................................... 235
Appendix A: R Code ........................................ 237
Appendix B: Description of the Variables of the Example in Section 13.2 .............................................. 238
References .................................................. 239

13.1 Introduction

Suppose we observe an independent and identically distributed sample \( Z_1, \ldots, Z_n \) of random variables with distribution \( P_0 \), and assume that \( P_0 \) is an element of a statistical model \( M \). Suppose also that nothing is known about \( P_0 \), so that \( M \) represents the nonparametric model. Statistical and machine learning are concerned with drawing inferences about target parameters \( \eta_0 \) of the distribution \( P_0 \). The types of parameters discussed in this chapter are typically functions \( \eta_0 \) of \( z \), which can be represented as the minimizer of the expectation of a loss function. Specifically, we consider parameters that may be defined as

\[
\eta_0 = \arg \min_{\eta \in \mathcal{F}} \int L(z, \eta) \, dP_0(z),
\]

(13.1)

where \( \mathcal{F} \) is a space of functions of \( z \) and \( L \) is a loss function of interest. The choice of space \( \mathcal{F} \) and loss function \( L \) explicitly defines the estimation problem. Below we discuss some examples of parameters that may be defined as in Equation 13.1.

Example 1. Prediction. Consider a random vector \( Z = (X, Y) \), where \( Y \) is an outcome of interest and \( X \) is a \( p \)-dimensional vector of covariates. A key problem in
statistics and artificial intelligence is that of predicting the values of the random variable $Y$, given $X = x$ is observed. In this case, the parameter of interest $\eta_0$ may be defined as Equation 13.1, where $\mathcal{F}$ is an appropriately defined space of functions of $x$ and $L$ is an appropriately chosen loss function.

The space $\mathcal{F}$ must satisfy constraints imposed on the parameter $\eta_0$. For example, if the range of $Y$ in the interval $(a, b)$, so is the range of $\eta_0(x)$ and the functions $\eta(x)$ in $\mathcal{F}$ must also have range $(a, b)$. Analogously, if $Y$ is binary, $\mathcal{F}$ must be the space of functions taking values on $[0, 1]$.

Common choices for the loss function are the squared error and the absolute error:

$$L(z, \eta) = \begin{cases} (y - \eta(x))^2 \\ |y - \eta(x)| \end{cases}$$

The mean squared-error loss function yields $\eta_0(x)$, which is equal to the regression function $E(Y|X = x) = \int y dP_0(y|X = x)$, whereas the absolute error loss function yields $\eta_0(x)$, which is equal to the conditional median of $Y$ given $X = x$.

Example 2. **Conditional Density Estimation.** As in the previous example, assume that $Y$ is a continuous outcome and $X$ is a $p$-dimensional vector of covariates, but we are now interested in estimating the density of $Y$ conditional on $X = x$, denoted by $\eta_0(y|x)$. In this case, $\mathcal{F}$ is the space of all nonnegative functions $\eta$ of $y$ and $x$ satisfying $\int \eta(y|x)dy = 1$. A common choice for the loss function is the negative log-likelihood:

$$L(z, \eta) = -\log \eta(y|x)$$

If $Y$ is a binary random variable, its conditional density is uniquely identified by the conditional expectation. Thus, prediction and density estimation are equivalent problems. For binary $Y$, both the negative log-likelihood and the mean squared error are appropriate loss functions.

Example 3. **Estimation of the Dose–Response Curve.** In addition to $Y$ and $X$, consider a continuous treatment variable $T$, and assume that we are interested in estimating the function $\eta_0(t) = E\{E(Y|T = t, X)\}$, where the outer expectation is taken with respect to the marginal distribution of $X$ with $T = t$ fixed. Under certain untestable assumptions (cf. positivity and no unmeasured confounders), $\eta_0(t)$ can be interpreted as the expected outcome that would be observed in a hypothetical world in which $P(T = t) = 1$. This interpretation gives $\eta_0(t)$ the name dose–response curve.

The dose–response curve may also be defined as $\eta_0 = \arg\min_{\eta \in \mathcal{F}} \int L(z, \eta) dP_0(z)$, where $\mathcal{F}$ is an appropriately defined space of functions of $t$. A possible choice for the loss function is

$$L(z, \eta) = \frac{(y - \eta(t))^2}{g_0(t|x)}$$

where $g_0(t|x)$ denotes the conditional density of $T$ given $X = x$, evaluated at $T = t$. Although other choices of loss function are available, this choice suffices for illustration purposes. In particular, note that, compared to the previous problems, this loss function depends on the generally unknown density $g_0(t|x)$. In this example, $g_0$ is a so-called nuisance parameter: a parameter whose
estimation is necessary but which is not the target parameter itself. In this case, we favor the notation $L_{g_0}(z, \eta)$, to acknowledge the dependence of $L$ on an unknown parameter $g_0$.

The most common problem in data analysis applications is the prediction problem described in Example 1. Throughout this chapter we focus on this example to illustrate all the relevant concepts, treating two different but related problems. In the first problem, estimator or model selection, the aim is to find an optimal estimator among a given class of estimators, where optimality is with respect to the minimal prediction risk. In the second problem, estimator performance assessment, the aim is to estimate the risk as a measure of generalization error of the estimator, that is, a measure of the ability of the estimator to correctly predict the units out of the sample. These two problems are discussed in more detail in Sections 13.6.1 and 13.6.2, respectively.

We discuss a model selection approach involving the following steps:

Step 1. Define the parameter of interest using formulation (Equation 13.1). In particular, this involves choosing a loss function of interest.

Step 2. Propose a finite collection $\mathcal{L} = \{\hat{\eta}_k : k = 1, \ldots, K_n\}$ of candidate estimators for $\eta$. We call this collection a library.

For illustration, consider Example 1. Each candidate estimator $\hat{\eta}_k$ is seen as an algorithm that takes a training sample $\mathcal{T} = \{Z_i : i = 1, \ldots, n\}$ as an input and outputs as an estimated function $\hat{\eta}_k(x)$. The literature in machine and statistical learning provides us with a wealth of algorithms that may be used in this step. Examples include algorithms based on regression trees (e.g., random forests [RFs], classification and regression trees, Bayesian classification and regression trees), algorithms based on smoothing (e.g., generalized additive models, local polynomial regression, multivariate adaptive regression splines), and others (e.g., support vector machines [SVM] and neural networks). Note that we have allowed the number of algorithms $K_n$ to be indexed by $n$.

Step 3. Construct an estimate $\hat{R}(\hat{\eta}_k)$ of the risk

$$R_0(\hat{\eta}_k) = \int L(z, \hat{\eta}_k) dP_0(z)$$

and define the estimator of $\eta_0$ as $\hat{\eta}_k$, where

$$\hat{k} = \arg\min_{k \in \{1, \ldots, K_n\}} \hat{R}(\hat{\eta}_k)$$

The second step of this process relies on the statistical and machine learning literature to build a rich library of candidate estimators. Fortunately, outcome prediction has been extensively studied and a large variety of estimation algorithms exist. However, other problems such as dose–response curve estimation have not enjoyed as much attention from the data science community. In such problems, the proposal of a rich library of candidate estimators may be a more difficult task.

Construction of the estimator $\hat{R}(\hat{\eta}_k)$ in the third step poses some challenges, because the dataset $\mathcal{T}$ must be used both to train the algorithm $\hat{\eta}_k$ and to assess its performance. As an example of these challenges, consider a naïve risk estimator, given by the empirical risk $\frac{1}{n} \sum_{i=1}^{n} L(\eta_k, Z_i)$, which suffers from various drawbacks. First, unless the candidates in the library $\mathcal{L}$ consider very small spaces $\mathcal{F}$ (e.g., logistic regression or exponential families in
regression and density estimation), a selector based on the empirical risk may result in highly variable and possibly ill-defined estimators (cf. overfitting and curse of dimensionality). In addition, in terms of prediction assessment, the empirical risk estimator is usually an underestimate. This is a well-recognized problem arising as a consequence of evaluating the performance of the prediction algorithm on the same data that were used to train it. Other classical model selection approaches such as Akaike’s information criterion or the Bayesian information criterion also fail to account for the fact that prediction methods are usually data driven [1].

As a solution to the above problems, statistical and machine learning methods have shifted towards data-splitting techniques. The main idea is to split the data so that the training and assessment datasets are independent. Some of the most well-known data-splitting techniques are as follows:

- **Multifold cross-validation.** A number of folds $J$ is chosen, and the data are randomly partitioned in $J$ subsamples. Each of $J$ times, a different subsample is left out, and the algorithm is trained in the remaining portion of the sample. Units in the left-out subsample are posteriorly used to assess the performance of the algorithm through computation of the loss function. These individual performance measures are averaged across units and subsamples to construct the risk estimate.

- **Leave-one-out cross-validation.** This is multifold cross-validation with $J = n$. The algorithm is trained $n$ times. Each time, unit $i \in \{1, \ldots, n\}$ is excluded from the training dataset, and it is used to assess the performance of the algorithm. The performance measures (loss function) across units $i$ are averaged to construct the risk estimate. This method has various practical and theoretical disadvantages. First, it can be computationally intensive because the estimator needs to be trained $n$ times. Second, the theoretical results presented in Section 13.6 below require that the size of the validation sample converges to infinity, and therefore do not cover leave-one-out cross-validation.

- **Bootstrap resampling.** Draw a number $J$ of bootstrap samples (samples of size $n$ drawn with replacement). Train the algorithm in each bootstrap sample, and assess its performance in the left-out portion of the data (i.e., the data that were not included in the bootstrap sample). Average the performance measures across units and bootstrap samples to obtain an estimate of the risk. For large sample sizes, the proportion of units in the validation set may be approximated as $\left(1 - 1/n\right)^n \approx e^{-1} \approx 0.368$.

- **Monte Carlo resampling.** For some predetermined sample size $m$, draw $J$ subsamples of size $m$ without replacement. Train the algorithm $J$ times using only the $m$ units in each subsample, and assess its performance using the left-out $n - m$ units. Average the performance measures across units and samples to obtain an estimate of the risk.

For conciseness of presentation, we focus on multifold cross-validation and refer to it simply as cross-validation. The remaining of the chapter is organized as follows. In Section 13.2, we briefly describe an applied example that will serve to illustrate the methods. In Section 13.3, we describe estimation of the risk $R_0$ of an estimator $\hat{\eta}$ using cross-validation. Section 13.4 describes a model selection algorithm using a library $\mathcal{L}$ with finite number of candidates. In Section 13.5, we discuss an estimator in which the library $\mathcal{L}$ is allowed to have infinite candidates indexed by an Euclidean parameter. In Section 13.6, we present two key theoretical results: (1) a convergence result guaranteeing that the cross-validation selector is guaranteed to choose the algorithm in the library that is closest to the true value $\eta_0$, where the notion of distance is given by the chosen loss function, as the sample size $n$ increases; and (2) asymptotic linearity result that may be used to compute confidence intervals and test hypotheses about the risk of a given algorithm. We illustrate most concepts using the SuperLearner library in R, and provide the code in Appendix A at the end of the chapter.
13.2 Classification of Chemicals

Throughout the chapter we use the quantitative structure-activity relationship (QSAR) biodegradation dataset to illustrate the methods in a binary regression problem, using the library Super Learner in R. The QSAR dataset was built in the Milano Chemometrics and QSAR Research Group, and is available online at the UC Irvine Machine Learning Repository [3]. The dataset contains measures on 41 biodegradation experimental values of 1055 chemicals, collected from the webpage of the National Institute of Technology and Evaluation of Japan. The aim is to develop classification models to discriminate ready (356) and not ready (699) biodegradable molecules. A description of the variables may be found in Appendix B.

13.3 Estimation of the Prediction Error

In this section, we consider a single estimator \( \hat{\eta} \) and drop the index \( k \) from the notation. Estimates \( \hat{\eta} \) of \( \eta_0 \) are usually obtained by training an algorithm on an i.i.d. sample \( Z_1, \ldots, Z_n \) of the random variable \( Z \). The objective of this section is to construct an estimate of the prediction error of \( \hat{\eta} \). To do this, we view the estimator \( \hat{\eta} \) as an algorithm that takes as input a given dataset and outputs a predictive function \( \hat{\eta}(x) \) that may be evaluated at any value \( x \).

Consider the following cross-validation setup. Let \( V_1, \ldots, V_J \) denote a random partition of the index set \( \{1, \ldots, n\} \) into \( J \) validation sets of approximately the same size, that is, \( V_j \subset \{1, \ldots, n\} \); \( \bigcup_{j=1}^J V_j = \{1, \ldots, n\} \); and \( V_j \cap V_j' = \emptyset \). In addition, for each \( j \), the associated training sample is given by \( T_j = \{1, \ldots, n\} \setminus V_j \). Denote by \( \hat{\eta}_{T_j} \) the prediction function obtained by training the algorithm using only data in the sample \( T_j \). The cross-validated prediction error of an estimated function \( \hat{\eta} \) is defined as

\[
\hat{R}(\hat{\eta}) = \frac{1}{J} \sum_{j=1}^J \frac{1}{|V_j|} \sum_{i \in V_j} L(Z_i, \hat{\eta}_{T_j})
\]

(13.2)

Figure 13.1 provides a schematic explanation of the steps involved in the cross-validated estimation of the risk assuming \( n = 15 \) and \( J = 5 \). The data are assumed ordered at random.
First, the algorithm is trained $J$ times, using only data in $T_j$ each time. Then, for each $j$, the performance of $\hat{\eta}_{T_j}$ is evaluated on all the units $i$ in the left-out portion of the sample, $V_j$. Lastly, the performance measures $L(Z_i, \hat{\eta}_{T_j})$ are averaged across units in $V_j$, and across sample splits $j$, giving $\hat{R}(\hat{\eta})$.

Intuitively, $\hat{R}(\hat{\eta})$ is a good measure of the risk of $\hat{\eta}$ because, unlike the empirical risk, it incorporates the randomness in $\hat{\eta}$ by considering a sample $\hat{\eta}_{T_j} : j = 1, \ldots, J$, instead of a fixed $\hat{\eta}$ trained using all the sample. In addition, the performance of the estimator is assessed based on data outside the training set, therefore providing an honest assessment of predictive power.

### 13.3.1 Cross-Validation with Correlated Units

In various applications, the data may not be assumed independently distributed. Such is the case, for example, of longitudinal studies in the medical sciences in which the objective is to predict the health status of a patient at a given time point, conditional on his health status and covariates in the previous time point. Specifically, consider an observation unit (e.g., a patient) $i$, with $t = 1, \ldots, m_i$ measurements for each unit (e.g., index $t$ may represent geographical locations, measurement times, etc.). Denote by $X_i = (X_{i1}, X_{i2}, \ldots, X_{im_i})$ the covariate vector and by $Y_i = (Y_{i1}, Y_{i2}, \ldots, Y_{im_i})$ the outcome vector for unit $i$. Each $X_{it}$ may be a covariate vector itself, containing observations recorded at time $t$ but also at previous times, for example at the most recent past $t - 1$. In this case, the correct assumption is that $Z_i = (X_i, Y_i) : i = 1, \ldots, n$ are an i.i.d sample of a random variable $Z \sim P_0$. The predictive function, given by $\eta_0(x, t) = E(Y_i | X_i = x)$, may be estimated using the same type of predictive algorithms discussed above, adding to the explanatory variables a time variable containing the index $t$.

However, note that these data are independent across the index $i$, but not the index $t$. As a result, the optimality properties of cross-validation presented in Section 13.6 will only hold if cross-validation is performed on the index set $\{i : i = 1, \ldots, n\}$. This is an important clarification to prevent cross-validation users from naively cross-validating on the index set $\{(i, t) : i = 1, \ldots, n; t = 1, \ldots, m_i\}$, that is, once a unit $i$ is chosen to belong to a validation dataset $V_j$, all its observations $(i, t) : t = 1, \ldots, m_i$ must also be in $V_j$.

### 13.4 Discrete Model Selection

An important step of the model selection approach outlined in the introduction of the chapter involves proposing a collection of estimation algorithms $L = \{\hat{\eta}_k : j = 1, \ldots, K_n\}$. Some of these algorithms may be based on a subject-matter expert’s knowledge (e.g., previous studies suggesting particular functional forms, knowledge of the physical nature of a phenomenon, etc.), some may be flexible data-adaptive methods (of which the statistical and machine learning literature have developed a large variety involving varying degrees of flexibility and computational complexity), and some other algorithms may represent a researcher’s favorite prediction tool, or simply standard practice. For example, in a regression problem, it may be known to a subject-matter expert that the relation between a given predictor and the outcome is logarithmic. In that case, a parametric model including a logarithmic term may be included in the library.

Our aim is to construct a candidate selector based on predictive power. This aim may be achieved by defining the estimator as the candidate in the library with the smallest the prediction error estimate $\hat{R}(\hat{\eta}_k)$. That is, the selector is defined as

$$\hat{k} = \arg\min_k \hat{R}(\hat{\eta}_k)$$

(13.3)
which is referred to as the discrete super learner in [5]. Its corresponding predictive function is given by $\hat{\eta}_{DSL} = \hat{\eta}_k$, that is, the predictive function of the algorithm with the smallest prediction risk.

For example, consider the problem described in Section 13.2, where we consider the following library of candidate estimators (we give only brief and intuitive descriptions of each algorithm; complete discussions may be found in most modern statistical learning textbooks).

- **Logistic regression with main terms (generalized linear model [GLM])**. This prediction method represents standard practice in fields such as epidemiology. The main idea is to estimate the coefficients $\beta_j$ in the parametrization

$$\logit \eta_0(x) = \beta_0 + \sum_{j=1}^{41} \beta_j x_j$$

where $\logit(p) = \log(p/(1-p))$.

- **Generalized additive model (GAM)**. The main idea of a GAM is to extend the GLM by considering a model where the predictors enter nonlinearly in the equation. That is, consider the following model:

$$\logit \eta_0(x) = \beta_0 + \sum_{j=1}^{41} \beta_j f_j(x_j)$$

where the functions $f_j$ are assumed smooth but otherwise unspecified and are typically estimated using smoothing techniques such as cubic splines or local polynomial regression.

- **SVM**. In its simplest form (linear), an SVM is a classifier that separates the $X$ space using a hyperplane, in a way such that the points $Y$ in each of two categories $\{0, 1\}$ are maximally separated. The intuition behind the prediction method is best explained with a figure; a toy example for two-dimensional $X$ and binary $Y$ with perfect separation is presented in Figure 13.2.

- **RF**. This is a prediction algorithm that runs many classification trees. To obtain the probability $\hat{\eta}(x)$, the vector $x$ is run down each of the trees in the forests, and the

![Figure 13.2](image-url)

**FIGURE 13.2**

SVM illustration. The shape of the points defines the categories, the solid and dotted lines represent the SVM maximum separation margin. The dashed line provides suboptimal separation.
classification results are averaged. Each tree is grown by running a classification tree on $q \ll p$ randomly selected variables on a bootstrap sample from the original data. Each tree is grown to the largest extent possible.

- **Multivariate adaptive regression splines (MARS).** The predictor is built as

$$\eta_0(x) = \beta_0 + \sum_{s=1}^{S} \beta_j B_s(x)$$

where each $B_s$ is a basis function and $S$ is a tuning parameter. The basis functions take the form of hinge functions $\max(0, x_j - c_j)$ or $\max(0, c_j - x_j)$, where $c_j$ is a tuning parameter called knot. Basis functions may also take the form of the product of two or more hinge functions. The MARS predictor is built in a forward–backward fashion, by adding and deleting terms to minimize the sum of squared errors.

- **Mean.** For comparison, we add the most na"ive prediction algorithm: $\hat{\eta}(x) = \bar{Y}$.

The cross-validated risk $\hat{R}(\hat{\eta}_k)$ of each of these algorithms applied to our illustrating example is presented in Table 13.4, and may be computed using the SuperLearner package in R (code available in Appendix A), which uses the mean squared-error function. These estimated risks indicate that the best predictor for this dataset, among the candidates studied, is the RF.

Figure 13.3 shows the receiver operating characteristic (ROC) curves computed with cross-validated and non-cross-validated predictions. The area under the curve (AUC) is also

<table>
<thead>
<tr>
<th>TABLE 13.1</th>
<th>Cross-validated risk for each algorithm applied to classification of chemicals.</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLM</td>
<td>0.1003</td>
</tr>
<tr>
<td>GAM</td>
<td>0.0985</td>
</tr>
<tr>
<td>SVM</td>
<td>0.1021</td>
</tr>
<tr>
<td>RF</td>
<td>0.0933</td>
</tr>
<tr>
<td>MARS</td>
<td>0.1154</td>
</tr>
<tr>
<td>Mean</td>
<td>0.224</td>
</tr>
</tbody>
</table>

**FIGURE 13.3**

ROC curve and AUC for each prediction method. (a) Complete sample prediction. (b) Cross-validated prediction.
displayed as a performance measure: an area of 1 indicates perfect prediction, whereas an area of 0.5 indicates that the predictor is not better than a coin toss. This graph illustrates the importance of using cross-validation. Consider first the ROC curves in Figure 13.3a. In this graph, the RF seems to perfectly fit the data, arising suspicion for overfitting and poor generalization power. Meanwhile, MARS seems to be the best performing algorithm. Figure 13.3b shows the ROC computed with cross-validated predictions. Note that, according to Figure 13.3b, the conclusions obtained from Figure 13.3a are wrong. The RF has the best generalization power, whereas the generalization power of the MARS is the poorest among all the algorithms considered.

13.5 Model Stacking and Super Learning

In the previous section, we described an algorithm that may be used to select a unique estimator from a discrete library of candidate algorithms. In this section, we improve discrete model selection by using an ensemble model that combines the predictions of all the models in the discrete candidate library. The main idea behind the super learner (or more generally a stacked predictor) is to use a parametrization $\hat{\eta}_\alpha$ to combine the predictors $\hat{\eta}_k$. Examples of this parametrization include

$$\hat{\eta}_\alpha(z) = \begin{cases} \sum_{k=1}^{K} \alpha_k \hat{\eta}_k(z) \\ \expit \left( \sum_{k=1}^{K} \alpha_k \logit(\hat{\eta}_k(z)) \right) \end{cases}$$

where expit is its inverse of the logit function. The first parametrization is often referred to as a linear model, whereas the second one is referred to as a logistic model, and is often used for binary regression. In the linear model, one may decide to restrict the values $\alpha_k$ to the set of positive weights summing up to one. This constraint on the $\alpha$ parameters would be particularly relevant for estimation of a conditional density function, because it would guarantee that the final estimate is a proper density function.

For simplicity, we focus on parametric models for the ensemble model. In principle, however, any prediction method could be used to combine the predictors in the discrete library.

In this formulation, the optimal selection problem may be equivalently framed as selection of the optimal weights $\alpha$ in a library of candidate estimators given by $\mathcal{L} = \{\hat{\eta}_\alpha(z) : \alpha\}$. It can be seen that the library used for the discrete super learner of the previous section is a subset of this library where $\alpha_k \in \{0, 1\}$. Selection of a candidate estimator in this library is carried out in a similar fashion to discrete model selection, but the optimization is performed in an infinite set as opposed to a finite one. The super learning predictive function is thus defined as $\hat{\eta}_{SL} = \hat{\eta}_\hat{\alpha}$, where the selector equals

$$\hat{\alpha} = \arg \min_{\alpha} \hat{R}(\hat{\eta}_\alpha) \quad (13.4)$$

The optimization in Equation 13.4 is often a convex optimization problem that can be readily solved using standard software. In particular, computing $\hat{\alpha}$ involves only one
additional step compared to the discrete selection algorithm of the previous section. As an example, consider the regression problem with a continuous outcome and note that

\[
L(Z_i, \hat{\eta}_{\alpha, T_j}) = (Y_i - \alpha_k \hat{\eta}_{k, T_j}(X_i))^2
\]

where we have introduced a set of variables \( H_k : k = 1, \ldots, K_n \), computed, for unit \( i \), as \( H_{k,i} = \hat{\eta}_{\alpha, T_j}(X_{k,i}) \). The cross-validated risk is equal to

\[
\hat{R}(\hat{\eta}) = \frac{1}{J} \sum_{j=1}^{J} \frac{1}{|V_j|} \sum_{i \in V_j} L(Z_i, \hat{\eta}_{\alpha, T_j})
\]

Thus, computation of \( \alpha \) in Equation 13.4 is reduced to computation of the ordinary least-squares estimator of a regression of \( Y \) with predictors \( H_k : k = 1, \ldots, K_n \). Note that the variables \( H_k \) must also be computed to evaluate the discrete super learner of the previous section. As a result, computation of the super learner involves running only one additional regression, compared to the discrete super learner. This optimization problem under a constraint on the parameter space such as restricting the values \( \alpha_k \) to the set of positive weights summing up to one is implemented in the SuperLearner package in R. In our chemical classification example, the weights of each algorithm in the super learner are given in Table 13.2.

Note that the RF gets more than half the weight in the predictor, in agreement with the results from the discrete super learner of the previous section.

### 13.5.1 Assessing the Prediction Error of a Stacked Predictor

As with any prediction method, the considerations presented in Section 13.3 about over fitting apply to a stacked predictor such as the super learner. Consequently, cross-validation provides the correct tool to assess the performance of the super learner. The cross-validated prediction error is thus given by

\[
\hat{R}(\hat{\eta}_{SL}) = \frac{1}{J} \sum_{j=1}^{J} \frac{1}{|V_j|} \sum_{i \in V_j} L(Z_i, \hat{\eta}_{SL, T_j})
\]

where \( \hat{\eta}_{SL, T_j} \) is the super learner trained using only data in \( T_j \). Note that training \( \hat{\eta}_{SL, T_j} \) involves further splitting of \( T_j \) in validation and training subsamples, which results in larger computation times for the risk of the super learner. In our chemical classification data, we obtain the results presented in Table 13.3, using the function \texttt{CV.SuperLearner} from the package \texttt{SuperLearner} (code in Appendix A).

### TABLE 13.2

Weights of each prediction algorithm in the chemical classification data.

<table>
<thead>
<tr>
<th>GLM</th>
<th>GAM</th>
<th>SVM</th>
<th>RF</th>
<th>MARS</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>0.2641</td>
<td>0.0109</td>
<td>0.5514</td>
<td>0.1736</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
TABLE 13.3
Cross-validated risk for each algorithm, including the super learner, applied to classification of chemicals.

<table>
<thead>
<tr>
<th>SL</th>
<th>DSL</th>
<th>GLM</th>
<th>GAM</th>
<th>SVM</th>
<th>RF</th>
<th>MARS</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0887</td>
<td>0.0931</td>
<td>0.0991</td>
<td>0.0976</td>
<td>0.1022</td>
<td>0.0931</td>
<td>0.1094</td>
<td>0.2239</td>
</tr>
</tbody>
</table>

These estimated risks are slightly different from those in Table 13.4. This randomness is caused by the different validation splits used in each analysis, and generally do not cause an important amount of variability. For certain datasets and cross-validation splits, it is possible that the super learner is outperformed by the discrete super learner (or, equivalently, by one of the candidates in the library). In such cases, the difference between both is expected to vanish as the sample size increases, as we argue in Section 13.6 below. In our applications, we have never observed the difference to be practically relevant in favor of the discrete super learner.

13.6 Asymptotic Results for the Cross-Validated Estimator Selector and the Cross-Validated Risk

In this section, we discuss the conditions for obtaining asymptotic optimality of the cross-validation selector. The benchmark used is an optimal selector that uses the true, unknown data generating distribution $P_0$ to assess the risk of the predictor. We present two different results, both originally proven in [2]. First, we discuss the conditions under which the cross-validation selector converges to the optimal benchmark. Second, we present asymptotic formulas for standard errors and confidence intervals for the cross-validated risk estimate $\hat{R}(\hat{\eta})$, based on the asymptotic linearity of the cross-validated risk.

We focus on asymptotic optimality of the discrete super learner, but the results also apply to the super learner with parametric family for the ensemble learner. To start, let us define some relevant quantities:

- The **conditional risk** is the risk of an estimator assessed using the true distribution $P_0$ (or, equivalently, using an infinite validation dataset):
  \[
  R_0(\hat{\eta}_k) = \frac{1}{J} \sum_{j=1}^{J} \int L(z, \hat{\eta}_k, T_j) dP_0(z)
  \]

- The **optimal risk** is the conditional risk achieved with the true function $\eta_0$ and is defined as
  \[
  R_0(\eta_0) = \int L(z, \eta_0) dP_0(z)
  \]

- The **oracle selector** is the estimation algorithm that would be selected using the conditional risk:
  \[
  \tilde{k} = \arg \min_k R_0(\hat{\eta}_k)
  \]

13.6.1 Asymptotic Optimality of the Cross-Validation Selector for Prediction

Theorem 1 of 2 establishes that, under the following conditions:

i. The loss function is the quadratic loss function $L(Z, \eta) = (Y - \eta(X))^2,$
ii. \( P(|Y| \leq M) = 1 \) for some \( M < \infty \),

iii. \( P(\sup_{k,X} |\tilde{\eta}_k(X)| \leq M) = 1 \),

iv. \( (\log(K_n)) / (n(\hat{R}_0(\tilde{\eta}_k) - R_0(\eta_0))) \xrightarrow{P} 0 \) as \( n \to \infty \), where \( \xrightarrow{P} \) denotes convergence in probability,

we have that

\[
\frac{R_0(\tilde{\eta}_k) - R_0(\eta_0)}{R_0(\tilde{\eta}_k) - R_0(\eta_0)} \xrightarrow{P} 1 \quad \text{and} \quad \frac{E[R_0(\tilde{\eta}_k)] - R_0(\eta_0)}{E[R_0(\tilde{\eta}_k)] - R_0(\eta_0)} \to 1
\]

The proof of the result may be consulted in the original research article, along other useful convergences and inequalities.

This is a very important result establishing the asymptotic optimality of cross-validation model selection for prediction, in the sense that a selector based on cross-validation is asymptotically equivalent to an optimal oracle selector based on the risk under the true, unknown data generating distribution \( P_0 \).

In practice, assumptions (ii) and (iii) imply that the outcome must be bounded (i.e., it may not take on arbitrarily large values), and, perhaps most importantly, that each of the predictors of the library must respect its bounds. In addition, assumption (iv) is satisfied if the risk difference \( R_0(\tilde{\eta}_k) - R_0(\eta_0) \) is bounded in probability and the ratio \( \log(K_n)/n \) converges to zero in probability.

We now illustrate the implication of these asymptotic results with a simulated example.

**Simulation setup:** Let \( X = (X_1, X_2, X_3) \) and \( Y \) be drawn from the joint distribution implied by the conditional distributions

\[
\begin{align*}
X_1 &\sim N(0, 1) \\
X_2|X_1 &\sim N(\sin(X_1), 1) \\
X_3|X_2, X_1 &\sim N(X_1X_2, |X_1|) \\
Y|X_3, X_2, X_1 &\sim Ber\{\expit(\cos(X_1)|X_2| + X_3^2X_1 - \exp(X_3))\}
\end{align*}
\]

where \( Ber\{p\} \) is the Bernoulli distribution with probability \( p \). Consider three of the algorithms of Section 13.4: the GLM, GAM, and MARS, and the sample sizes \( n = 100, 500, 1,000, 5,000, 10,000 \). In addition, consider a validation sample of size \( n = 10^7 \), used to approximate the true risks \( R_0(\tilde{\eta}_k) \) (i.e., approximate integrals with respect to the measure \( P_0 \)). For each sample size, we draw 500 samples. We then use each of these samples to compute the selector \( \hat{k} \) and use the validation sample of size \( 10^7 \) to approximate \( R_0(\tilde{\eta}_k) : k = 1, 2, 3 \), which includes \( R_0(\tilde{\eta}_k) \). We proceed to compute \( R_0(\tilde{\eta}_k) \) as \( \min_k R_0(\tilde{\eta}_k) \), and average the results across 500 samples to approximate \( E[R_0(\tilde{\eta}_k)] \) and \( E[R_0(\tilde{\eta}_k)] \).

**Simulation results:** Figure 13.4 shows the ratios of the expected risk differences

\[
\frac{E[R_0(\tilde{\eta}_k)] - R_0(\eta_0)}{E[R_0(\tilde{\eta}_k)] - R_0(\eta_0)}
\]

which, as predicted by theory, can be seen converging to 1.

In addition, Table 13.4 shows an approximation to the joint distribution of \( \hat{k} \) and \( \tilde{k} \) for different sample sizes in the simulation described above. Each entry of the table is the probability that a given algorithm is chosen as \( k \) and \( \tilde{k} \). These probabilities are approximated by computing \( k \) and \( \tilde{k} \) for each of the 500 samples and counting the number of times that each algorithm is selected. Note that both selectors converge to the MARS as the sample size increases.
FIGURE 13.4
Risk difference ratio in Equation 13.5 for different sample sizes

TABLE 13.4
Simulated joint distribution of \( \hat{k} \) and \( \hat{k} \) for different sample sizes.

<table>
<thead>
<tr>
<th>( n )</th>
<th>( \hat{k} )</th>
<th>MARS</th>
<th>GAM</th>
<th>GLM</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.004</td>
<td>0.080</td>
<td>0.008</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>0.142</td>
<td>0.200</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>1,000</td>
<td>0.762</td>
<td>0.064</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>5,000</td>
<td>0.996</td>
<td>0.002</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>10,000</td>
<td>0.998</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
</tbody>
</table>

13.6.2 Asymptotic Inference for the Cross-Validated Risk Estimator

In addition to selecting an optimal algorithm, estimation of the risk may be an objective in itself. We approach this problem as a statistical estimation problem and consider the asymptotic distribution of the statistics involved so that standard errors and confidence intervals for the risk estimate may be computed. For the cross-validated risk estimator discussed, such standard errors and confidence intervals may be based on the Gaussian asymptotic distribution of the estimator. For a general estimator \( \hat{\eta} \), Dudoit and van der
Laan [2] showed that

\[ \hat{R}(\hat{\eta}) - R_0(\hat{\eta}) = \frac{1}{n} \sum_{i=1}^{n} (L(Z_i, \eta^*) - R_0(\eta^*)) + o_P(1/\sqrt{n}) \]  

(13.6)

where \( \eta^* \) is a fixed limiting function of the algorithm \( \hat{\eta} \), in the sense that

\[ \sqrt{n} ||L(z, \hat{\eta}) - L(z, \eta^*)||_{P_0} = o_P(1) \]  

(13.7)

assuming that this limit exists. Here we have denoted \( ||f||_{P_0}^2 = \int f^2(z) dP_0(z) \), for a function \( f \) of \( z \).

In particular, Equation 13.6 and the central limit theorem imply that

\[ \sqrt{n}(\hat{R}(\hat{\eta}) - R_0(\hat{\eta})) \overset{d}{\to} N(0, \sigma^2) \]  

(13.8)

where \( \sigma^2 = \text{var}\{L(Z, \eta^*) - R_0(\eta^*)\} \). This result may be used to construct a \((1 - \alpha)100\%\) confidence level interval for the conditional (on the sample) parameter \( R_0(\hat{\eta}) \) as

\[ \hat{R}(\hat{\eta}) \pm z_{\alpha/2} \hat{\sigma} / \sqrt{n} \]

and \( z_{\alpha} \) is such that \( P(X > z_{\alpha}) = \alpha \), for a standard Gaussian random variable \( X \). In general, this cross-validated estimator of the variance should be preferred to its non-cross-validated version, as the latter may not be appropriate to use with data-adaptive estimators \( \hat{\eta} \). In our working example, the estimated standard error \( \hat{\sigma}_k / \sqrt{n} \) associated to the risks of Table 13.3 are given in Table 13.5.

The asymptotic linearity result given in Equation 13.6 has other additional useful implications. For example, the joint distribution of the vector \( \hat{R} = (\hat{R}(\hat{\eta}_1), \ldots, \hat{R}(\hat{\eta}_K)) \) of estimated risks may be obtained as

\[ \sqrt{n}(\hat{R} - R_0) \overset{d}{\to} N(0, \Sigma) \]

where \( \hat{R}_0 = (R_0(\hat{\eta}_1), \ldots, R_0(\hat{\eta}_K)) \) and \( \Sigma \) is a matrix with \((k,l)\)th element equal to

\[ \sigma_{kl} = \text{cov}\{L(Z, \eta^*_k), L(Z, \eta^*_l)\} \]

An important application of this result is testing whether two predictors have the same conditional risk \( R_0(\hat{\eta}) \). For example, define \( \theta = R_0(\hat{\eta}_{\text{SL}}) - R_0(\hat{\eta}_{\text{DSL}}) \) and assume that we want to test the hypothesis

\[ H_0 : \theta = 0 \quad \text{vs} \quad H_1 : \theta \neq 0 \]

### TABLE 13.5

Estimated risks and standard deviations in the chemical classification example.

<table>
<thead>
<tr>
<th></th>
<th>SL</th>
<th>DSL</th>
<th>GLM</th>
<th>GAM</th>
<th>SVM</th>
<th>RF</th>
<th>MARS</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{R}(\hat{\eta}_k) )</td>
<td>0.0887</td>
<td>0.0931</td>
<td>0.0991</td>
<td>0.0976</td>
<td>0.1022</td>
<td>0.0931</td>
<td>0.1094</td>
<td>0.2239</td>
</tr>
<tr>
<td>( \hat{\sigma}_k )</td>
<td>0.0108</td>
<td>0.0105</td>
<td>0.0114</td>
<td>0.0115</td>
<td>0.0107</td>
<td>0.0105</td>
<td>0.0126</td>
<td>0.0001</td>
</tr>
</tbody>
</table>
of whether the super learner and the discrete super learner provide the same conditional risk. According to Equation 13.6, the statistic $T = \hat{R}(\hat{\eta}_{SL}) - \hat{R}(\hat{\eta}_{DSL})$ satisfies

$$\sqrt{n}(T - \theta) \overset{d}{\to} N(0, \tau^2)$$

where

$$\tau^2 = \text{var}\{L(Z, \eta^*_{SL}) - L(Z, \eta^*_{DSL})\}$$

Here $\eta^*_{SL}$ and $\eta^*_{DSL}$ are the limits of the super learner and the discrete super learner defined as in Equation 13.7.

The hypothesis $H_0$ may be rejected at the $\alpha$ level if $\sqrt{n}|T|/\hat{\tau} > z_{\alpha}$, where $\hat{\tau}^2$ is the cross-validated estimate of the variance $\tau^2$,

$$\hat{\tau}^2 = \frac{1}{J} \sum_{j=1}^{J} \frac{1}{|V_j|} \sum_{i \in V_j} (L(Z_i, \hat{\eta}_{SL,T_j}) - L(Z_i, \hat{\eta}_{DSL,T_j}) - \hat{R}(\hat{\eta}_{SL}) + \hat{R}(\hat{\eta}_{DSL}))^2$$

In our chemical classification example, the statistic $\sqrt{n}|T|/\hat{\tau}$ equals 2.1841, which yields a rejection of the hypothesis of equality of the conditional risks at level 0.05. The conditional (on the sample) 95% confidence interval for the risk difference $\theta$ is equal to $(-0.0083, -0.0004)$.

---

**Appendix A: R Code**

```R
## read data
## prepare outcome and covariates
X <- data[, -42]
Y <- data[, 42]
Y <- (Y=='RB')*1
## load package and create library
require('SuperLearner')
L <- c('SL.glm', 'SL.gam', 'SL.svm', 'SL.randomForest', 'SL.earth', 'SL.mean')
## super learner
SL <- SuperLearner(Y, X, SL.library = L, family=binomial(), cvControl=list(V=20))
## cross-validation of the super learner
CVSL <- CV.SuperLearner(Y, X, SL.library = L, family=binomial())
## estimate of the standard deviation of the risk of the discrete super learner
RDSL <- var(CVSL$discreteSL.predict - summary(CVSL)$Table[2,2])
## estimate of the standard deviation of the risk of the super learner
RSL <- var(CVSL$SL.predict - summary(CVSL)$Table[1,2])
## estimate of the standard deviation of the risk of the candidates in teh library
RS <- diag(var(t(t(CVSL$library.predict) - summary(CVSL)$Table[-c(1,2),2])))
## statistic for hypothesis test
T <- -diff(summary(CVSL)$Table[1:2,2])
tau <- sqrt(var(CVSL$SL.predict - CVSL$discreteSL.predict))
## test
abs(T) > qnorm(0.975) * tau / sqrt(dim(data)[1])
```

Appendix B: Description of the Variables of the Example in Section 13.2

The 42 variables in the dataset are, in order of appearance:

1. SpMax L: Leading eigenvalue from Laplace matrix
2. J Dz(e): Balaban-like index from Barysz matrix weighted by Sanderson electronegativity
3. nHM: Number of heavy atoms
4. F01[N-N]: Frequency of N-N at topological distance 1
5. F04[C-N]: Frequency of C-N at topological distance 4
6. NssssC: Number of atoms of type sssssC
7. nCb-: Number of substituted benzene C(sp2)
8. C%: Percentage of C atoms
9. nCp: Number of terminal primary C(sp3)
10. nO: Number of oxygen atoms
11. F03[C-N]: Frequency of C-N at topological distance 3
12. SdssC: Sum of dssC E-states
13. HyWi B(m): Hyper–Wiener-like index (log function) from Burden matrix weighted by mass
14. LOC: Lopping centric index
15. SM6 L: Spectral moment of order 6 from Laplace matrix
16. F03[C-O]: Frequency of C-O at topological distance 3
17. Me: Mean atomic Sanderson electronegativity (scaled on carbon atom)
18. Mi: Mean first ionization potential (scaled on carbon atom)
19. nN-N: Number of N hydrazines
20. nArNO2: Number of nitro groups (aromatic)
21. nCRX3: Number of CRX3
22. SpPosA B(p): Normalized spectral positive sum from Burden matrix weighted by polarizability
23. nCIR: Number of circuits
24. B01[C-Br]: Presence/absence of C-Br at topological distance 1
25. B03[C-Cl]: Presence/absence of C-Cl at topological distance 3
27. $\text{SpMax}_A$: Leading eigenvalue from adjacency matrix (Lovasz–Pelikan index)
28. $\text{Psi}_{1d}$: Intrinsic state pseudoconnectivity index-type 1d
29. $\text{B04[C-Br]}$: Presence/absence of C-Br at topological distance 4
30. $\text{SdO}$: Sum of dO E-states
31. $\text{TI2}_L$: Second Mohar index from Laplace matrix
32. $nCrt$: Number of ring tertiary C(sp3)
33. $\text{C-026}$: R–CX–R
34. $\text{F02[C-N]}$: Frequency of C-N at topological distance 2
35. $n\text{HDOn}$: Number of donor atoms for H-bonds (N and O)
36. $\text{SpMax}_B(m)$: Leading eigenvalue from Burden matrix weighted by mass
37. $\text{Psi}_{S_A}$: Intrinsic state pseudoconnectivity index-type S average
38. $nN$: Number of nitrogen atoms
39. $\text{SM6}_B(m)$: Spectral moment of order 6 from Burden matrix weighted by mass
40. $n\text{ArCOOR}$: Number of esters (aromatic)
41. $nX$: Number of halogen atoms
42. experimental class: ready biodegradable (RB) and not ready biodegradable (NRB)

References