Direct Learning: Linear Rule for Regression Problem
We consider approximating the core function in the prediction rule to be a parametric function of feature variables.

For continuous $Y$, under the squared loss function, the Bayes rule, $f^*(x) = E[Y|X = x]$, is assumed to be a linear function of $x$.

For binary $Y$, under the 0-1 loss, the Bayes rule is the sign of $P(Y = 1|X = x) - 1/2$, so we assume $\logit P(Y = 1|X = 1)$ to be a linear function of $x$.

Thus, learning the optimal prediction rule reduces to estimating the parameters in the assumed model using empirical data.
Advantage of parametric or even linear rules

- They are computationally simple, and easy to be interpreted.
- The parameters are often informative about associated feature variables so can be used for feature selection.
- Variability in estimating parametric rules tends to be smaller than non-parametric rules.
- Although a linear rule may not be the Bayes rule, its empirical performance in terms of prediction is usually satisfactory especially when the number of feature variables is high. For example, in deep neural network, each layer is some simple and parametric prediction function.
- Nonlinearity can be introduced if we transform feature variables or include interactions among feature variables.
Linear rule under a squared loss

- Training data: \((X_1, Y_1), \ldots, (X_n, Y_n)\), each \(X_i\) with dimension \(p\), \(X_i = (X_{i1}, \ldots, X_{ip})^T\).
- Learning the optimal linear rule is equivalent to a least-squared problem:
  \[
  \sum_{i=1}^{n} (Y_i - \beta_0 - \sum_{j=1}^{p} X_{ij} \beta_j)^2.
  \]
- The optimal rule is \(\hat{\beta}_0 + x^T \hat{\beta}\), where
  \[
  (\hat{\beta}_0, \hat{\beta})^T = (X^T X)^{-1} X^T Y
  \]
  \(X = (X_1^T, \ldots, X_n^T)^T\) is the feature variable matrix, and \(Y = (Y_1, \ldots, Y_n)^T\) is the outcome vector.
- For any future subject, the prediction is
  \[
  x^T (\hat{\beta}_0, \hat{\beta})^T = x^T (X^T X)^{-1} X^T Y.
  \]
- Inference for \(\hat{\beta}\) well studied for Gaussian linear model.
Analysis of prostate cancer data

TABLE 3.1. Correlations of predictors in the prostate cancer data.

<table>
<thead>
<tr>
<th></th>
<th>lcavol</th>
<th>lweight</th>
<th>age</th>
<th>lbph</th>
<th>svi</th>
<th>lcp</th>
<th>gleason</th>
</tr>
</thead>
<tbody>
<tr>
<td>lcavol</td>
<td>0.300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lweight</td>
<td>0.286</td>
<td>0.317</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>age</td>
<td>0.063</td>
<td>0.437</td>
<td>0.287</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lbph</td>
<td>0.593</td>
<td>0.181</td>
<td>0.129</td>
<td>-0.139</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>svi</td>
<td>0.692</td>
<td>0.157</td>
<td>0.173</td>
<td>-0.089</td>
<td>0.671</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lcp</td>
<td>0.426</td>
<td>0.024</td>
<td>0.366</td>
<td>0.033</td>
<td>0.307</td>
<td>0.476</td>
<td></td>
</tr>
<tr>
<td>gleason</td>
<td>0.483</td>
<td>0.074</td>
<td>0.276</td>
<td>-0.030</td>
<td>0.481</td>
<td>0.663</td>
<td>0.757</td>
</tr>
</tbody>
</table>

TABLE 3.2. Linear model fit to the prostate cancer data. The Z score is the coefficient divided by its standard error (3.12). Roughly a Z score larger than two in absolute value is significantly nonzero at the p = 0.05 level.

<table>
<thead>
<tr>
<th>Term</th>
<th>Coefficient</th>
<th>Std. Error</th>
<th>Z Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>2.46</td>
<td>0.09</td>
<td>27.60</td>
</tr>
<tr>
<td>lcavol</td>
<td>0.68</td>
<td>0.13</td>
<td>5.37</td>
</tr>
<tr>
<td>lweight</td>
<td>0.26</td>
<td>0.10</td>
<td>2.75</td>
</tr>
<tr>
<td>age</td>
<td>-0.14</td>
<td>0.10</td>
<td>-1.40</td>
</tr>
<tr>
<td>lbph</td>
<td>0.21</td>
<td>0.10</td>
<td>2.06</td>
</tr>
<tr>
<td>svi</td>
<td>0.31</td>
<td>0.12</td>
<td>2.47</td>
</tr>
<tr>
<td>lcp</td>
<td>-0.29</td>
<td>0.15</td>
<td>-1.87</td>
</tr>
<tr>
<td>gleason</td>
<td>-0.02</td>
<td>0.15</td>
<td>-0.15</td>
</tr>
<tr>
<td>pgg45</td>
<td>0.27</td>
<td>0.15</td>
<td>1.74</td>
</tr>
</tbody>
</table>
Improved linear rule incorporating feature selection

- A linear rule with more features has smaller prediction bias with a price of increased variability.
- Neither large bias or high variability is good for prediction since

\[
E[(\hat{f}(X) - f(X))^2] = E[(\hat{f}(X) - E_{data}[\hat{f}(X)])^2] \\
+ E[(f(X) - E_{data}[\hat{f}(X)])^2].
\]

- If only a small number of feature variables are really related to \( Y \), identifying them and including them in the rule can substantially improve the prediction, especially when the feature dimension is high or much larger than \( n \).
- This leads to a vast number of work for feature (variable) selection in statistical literature.

Donglin Zeng, Department of Biostatistics, University of North Carolina
An intuitive way is to perform an exhaustive search over all subsets of \( \{X_1, \ldots, X_p\} \) to identify the best subset optimizing a given criterion related to prediction performance.

The procedure is as follows:

- for a given subset of size \( k \in \{0, 1, 2, \ldots, p\} \), we use some efficient algorithm (e.g., the leaps and bounds procedure, Furnival and Wilson, 1974) to identify the best subset of size \( k \), \( C_k \), that gives the smallest residual sum-of-squares.
- we then select \( k^* \) such that \( C_{k^*} \) minimizes a given criterion.
- This search is computationally intensive and becomes infeasible when \( p > 40 \).
FIGURE 3.5. All possible subset models for the prostate cancer example. At each subset size is shown the residual sum-of-squares for each model of that size.
Suboptimal subset selection

- There are some suboptimal subset selection approaches but the computationally efficient.
- Forward-stepwise selection: it is a greedy algorithm to search along an optimal sequence of models with increasing sizes.
- Backward-stepwise selection: it starts from the full model and sequentially eliminates one variable from the model.
- Stepwise selection adds or deletes one variable based on certain significance testing so it is a locally optimal search.
Comparing different subset selection

**FIGURE 3.6.** Comparison of four subset-selection techniques on a simulated linear regression problem $Y = X^T \beta + \varepsilon$. There are $N = 300$ observations on $p = 31$ standard Gaussian variables, with pairwise correlations all equal to 0.85. For 10 of the variables, the coefficients are drawn at random from a $N(0, 0.4)$ distribution; the rest are zero. The noise $\varepsilon \sim N(0, 0.25)$, resulting in a signal-to-noise ratio of 0.64. Results are averaged over 50 simulations. Shown is the mean-squared error of the estimated coefficient $\hat{\beta}(k)$ at each step from the true $\beta$. 

Donglin Zeng, Department of Biostatistics, University of North Carolina
Shrinkage methods

- Subset selection is computationally intensive and there is high variability when model sizes change.
- Shrinkage methods provide more a smooth search procedure to identify the “best” models.
- Such methods are often carried out via regularization/penalization.
The estimate for $\beta$ is obtained by minimizing
\[
\sum_{i=1}^{n}(Y_i - \beta_0 - X_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2,
\]
resulting in
\[
\hat{\beta} = (X^T X + \lambda I)^{-1} X^T Y.
\]
That is, we impose a $L_2$-penalty to further shrink coefficients towards zero. When $X^T X = I$,
\[
\hat{\beta} = \hat{\beta}_{LSE}/(1 + \lambda)
\]
so the prediction based on $\hat{\beta}^T x$ is less variable than $\hat{\beta}_{LSE}^T x$.

The optimization is equivalent to
\[
\min \sum_{i=1}^{n}(Y_i - \beta_0 - X_i^T \beta)^2, \quad \text{subject to } \sum_{j=1}^{p} \beta_j^2 \leq C.
\]

$\lambda$ or $C$ is a tuning parameter.
Prostate cancer example (ridge regression)

**FIGURE 3.8.** Profiles of ridge coefficients for the prostate cancer example, as the tuning parameter $\lambda$ is varied. Coefficients are plotted versus $df(\lambda)$, the effective degrees of freedom. A vertical line is drawn at $df = 5.0$, the value chosen by cross-validation.
The lasso minimizes
\[
\frac{1}{2} \sum_{i=1}^{n} (Y_i - \beta_0 - X_i^T \beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j|
\]
so the regularization is a $L_1$-penalty for $\beta$.

The objective function is convex. The optimization can be solved using quadratic programming.

This is equivalent to solving
\[
\min (\beta - \hat{\beta}_{LSE})^T X^T X (\beta - \hat{\beta}_{LSE}) + \lambda \sum_{j=1}^{p} |\beta_j|.
\]
So when $X^T X = I$, we obtain an explicit solution
\[
\hat{\beta}_j = \text{sign}(\hat{\beta}_{LSE,j})(|\hat{\beta}_{LSE,j}| - \lambda)_+.
\]
One efficient algorithm called Least Angle Regression (LAR) algorithm can be used to solve the problem and give the whole solution path when $\lambda$ varies from 0 to $\infty$.

Alternatively, coordinate-descent algorithm (shooting algorithm) is proven to be effective with a large number of feature variables: we update one coefficient at one time and iterate across all coefficients.

**Homework:** Write down the detail for the coordinate-descent algorithm.
LAR algorithm

Algorithm 3.2 Least Angle Regression.

1. Standardize the predictors to have mean zero and unit norm. Start with the residual $r = y - \bar{y}$, $\beta_1, \beta_2, \ldots, \beta_p = 0$.

2. Find the predictor $x_j$ most correlated with $r$.

3. Move $\beta_j$ from 0 towards its least-squares coefficient $\langle x_j, r \rangle$, until some other competitor $x_k$ has as much correlation with the current residual as does $x_j$.

4. Move $\beta_j$ and $\beta_k$ in the direction defined by their joint least squares coefficient of the current residual on $(x_j, x_k)$, until some other competitor $x_l$ has as much correlation with the current residual.

5. Continue in this way until all $p$ predictors have been entered. After $\min(N - 1, p)$ steps, we arrive at the full least-squares solution.

Algorithm 3.2a Least Angle Regression: Lasso Modification.

4a. If a non-zero coefficient hits zero, drop its variable from the active set of variables and recompute the current joint least squares direction.
Prostate cancer example (Lasso)

**FIGURE 3.10.** Profiles of lasso coefficients, as the tuning parameter $t$ is varied. Coefficients are plotted versus $s = t / \sum_j |\hat{\beta}_j|$. A vertical line is drawn at $s = 0.36$, the value chosen by cross-validation. Compare Figure 3.8 on page 65; the lasso profiles hit zero, while those for ridge do not. The profiles are piece-wise linear, and so are computed only at the points displayed; see Section 3.4.4 for details.
Comparison among shrinkage methods

**TABLE 3.4.** Estimators of $\beta_j$ in the case of orthonormal columns of $X$. $M$ and $\lambda$ are constants chosen by the corresponding techniques; sign denotes the sign of its argument ($\ast$,$\ast$), and $x_0$ denotes “positive part” of $x$. Below the table, estimators are shown by broken red lines. The 45° line in gray shows the unrestricted estimate for reference.

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best subset (size $M$)</td>
<td>$\hat{\beta}_j I(\vert \hat{\beta}<em>j \vert \geq \vert \hat{\beta}</em>{(M)} \vert)$</td>
</tr>
<tr>
<td>Ridge</td>
<td>$\beta_j / (1 + \lambda)$</td>
</tr>
<tr>
<td>Lasso</td>
<td>$\text{sign}(\hat{\beta}_j)(\vert \hat{\beta}<em>j \vert - \lambda)</em>+$</td>
</tr>
</tbody>
</table>

**FIGURE 3.11.** Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions $\vert \beta \vert + \vert \beta \vert \leq \lambda$ and $\beta^2 + \beta^2 \leq \lambda^2$, respectively, while the red ellipses are the contours of the least squares error function.
Comparison among shrinkage methods in prostate cancer data

<table>
<thead>
<tr>
<th>Term</th>
<th>LS</th>
<th>Best Subset</th>
<th>Ridge</th>
<th>Lasso</th>
<th>PCR</th>
<th>PLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>2.465</td>
<td>2.477</td>
<td>2.452</td>
<td>2.468</td>
<td>2.497</td>
<td>2.452</td>
</tr>
<tr>
<td>lcavol</td>
<td>0.680</td>
<td>0.740</td>
<td>0.420</td>
<td>0.533</td>
<td>0.543</td>
<td>0.419</td>
</tr>
<tr>
<td>lwght</td>
<td>0.263</td>
<td>0.316</td>
<td>0.238</td>
<td>0.169</td>
<td>0.289</td>
<td>0.344</td>
</tr>
<tr>
<td>age</td>
<td>-0.141</td>
<td>-0.046</td>
<td>-0.152</td>
<td>-0.026</td>
<td>2.14</td>
<td>0.220</td>
</tr>
<tr>
<td>lbph</td>
<td>0.210</td>
<td>0.162</td>
<td>0.002</td>
<td>0.214</td>
<td>0.220</td>
<td>0.243</td>
</tr>
<tr>
<td>svi</td>
<td>0.305</td>
<td>0.227</td>
<td>0.094</td>
<td>0.315</td>
<td>0.243</td>
<td>0.079</td>
</tr>
<tr>
<td>lcp</td>
<td>-0.288</td>
<td>0.000</td>
<td>-0.051</td>
<td>0.079</td>
<td>0.111</td>
<td>0.084</td>
</tr>
<tr>
<td>gleason</td>
<td>-0.021</td>
<td>0.040</td>
<td>0.232</td>
<td>0.011</td>
<td>0.133</td>
<td>0.084</td>
</tr>
<tr>
<td>pgg45</td>
<td>0.267</td>
<td>0.133</td>
<td>-0.056</td>
<td>0.084</td>
<td>0.152</td>
<td></td>
</tr>
</tbody>
</table>

**TABLE 3.3. Estimated coefficients and test error results, for different subset and shrinkage methods applied to the prostate data. The blank entries correspond to variables omitted.**
Structural shrinkage methods

If we have some prior knowledge or purpose to have a specific structure on the prediction rule, structural shrinkage methods can force the estimated rule to be approximate to this structure.

For example, if we desire the prediction rule to depend on a small number of feature variables, $L_1$-penalty in Lasso can automatically eliminate non-important features in the estimated rule.

What are other possible structures?

- Group structure (genes on the same regularization pathway, image features from the same region and the same modality) so the feature variables in the same group should be retained if any of them is associated with the outcome.

- Network structure (feature variables are networked with each other such as brain network among different regions due to either structure or functional connectivity) so the effects from the strongly connected feature variables may present the similar effects on the outcome.

Donglin Zeng, Department of Biostatistics, University of North Carolina
Structural regularization

- Group Lasso for regularizing group structures ($X_l$ denotes the features from the $l$th group):

$$\min \sum_{i=1}^{n} (Y_i - \beta_0 - \sum_{l=1}^{L} X_i^T \beta_l)^2 + \lambda \sum_{l=1}^{L} \sqrt{p_l} \| \beta_l \|_{L_2}$$

- Feature similarity using Laplacian regularization
  - define a Laplacian eigenmap matrix $D$ as

$$D = \text{diag}(d_1, \ldots, d_p) - (w_{jk})_{k,j=1}^{p},$$

where $d_j = \sum_{k=1}^{p} w_{jk}$ (degree in an undirected graph), and $(w_{jk})$ is the similarity matrix (edge weight in a graph)

- a Laplacian regularization uses a penalty given by

$$\lambda \beta^T D \beta = \lambda \sum_{j,k=1}^{p} w_{jk} \| \beta_j - \beta_k \|^2.$$  

- this regularization encourages similar coefficients of two variables (nodes) with a high similarity.
Sparsity shrinkage method: SCAD

- Such methods aim to produce a truly sparse prediction rule by using less shrinkage for large coefficients (note Lasso shrinks all the coefficients the same way).
- Ideally, an oracle selection regularization can identify truly non-important features with probability ending to 1.
- Sparsity shrinkage methods are often based on non-convex penalty, for example, the smoothly clipped absolute deviation (SCAD):

\[
p(\beta; \lambda, \gamma) = \begin{cases} 
\lambda|\beta|, & |\beta| \leq \lambda \\
\frac{2\gamma\lambda|\beta|-\beta^2-\lambda^2}{2(\gamma-1)}, & \lambda < |\beta| \leq \gamma\lambda \\
\frac{\lambda^2(\gamma+1)}{2}, & |\beta| > \gamma\lambda 
\end{cases}
\]

for \( \gamma > 2 \).

- Its derivative is

\[
\frac{\partial q_\lambda(\beta)}{\partial \beta} = \lambda \text{sign}(\beta) \left[ I(|\beta| \leq \lambda) + \frac{(\gamma\lambda-|\beta|)+I(|\beta| > \lambda)}{(\gamma-1)\lambda} I(|\beta| > \lambda) \right].
\]

Donglin Zeng, Department of Biostatistics, University of North Carolina
Minimax Concave Penalty (MCP) is defined as

\[
p(\beta; \lambda) = \begin{cases} 
\lambda|\beta| - \frac{\beta^2}{2\lambda}, & |\beta| \leq \gamma \lambda, \\
\frac{1}{2}\gamma \lambda^2, & |\beta| > \gamma \lambda 
\end{cases}
\]

Its derivative is

\[
(\lambda - |\beta|/\gamma)\text{sign}(\beta)I(|\beta| \leq \gamma \lambda).
\]

Non-convex penalty leads to computation challenge (local minimum). Some local approximation is used during iterations for parameter estimation.
Graphic comparisons of different penalities

Donglin Zeng, Department of Biostatistics, University of North Carolina
Adaptive Lasso (ALasso) method is one of sparsity shrinkage methods with a penalty function

$$\lambda \sum_{j=1}^{p} c|\beta_j|/|\tilde{\beta}_j|^\gamma, \gamma > 0,$$

where $\tilde{\beta}_j$ is a consistent initial estimator for $\beta_j$.

**Rationale:** if the true value for $\beta_j$ is zero, then the initial estimate for $\beta_j$, $\tilde{\beta}_j$ is close to zero yielding a large weight for $|\beta_j|$ in the penalty. Thus, the new solution for $\beta_j$ will be expected even closer to zero.

**Unfortunately,** ALasso requires an initial estimate so may not be applicable when $p \gg n$. 
Graphic comparisons among all penalties

**FIGURE 3.12.** Contours of constant value of $\sum_j |\beta_j|^q$ for given values of $q$.

**FIGURE 3.13.** Contours of constant value of $\sum_j |\beta_j|^q$ for $q = 1.2$ (left plot), and the elastic-net penalty $\sum_j (\alpha \beta_j^2 + (1-\alpha)|\beta_j|)$ for $\alpha = 0.2$ (right plot). Although
Sample R-codes for implementation

```r
# Sample R-code for Best Subset Selection

diab.x <- diab[,1:10]
diab.y <- diab[,11]
diab.x <- sweep(diab.x, 2, apply(diab.x, 2, mean), "=")
diab.x <- sweep(diab.x, 2, sqrt(apply(diab.x^2, 2, sum)), "/")

Best Subset Selection

the command `library(leaps)`.

The function `leaps` will go through all the possible subsets (at least the good ones), and output their $C_p$'s. For the diabetes data, the command is

diablp <- leaps(diab.x, diab.y, nbest = 10)

min(diablp$Cp)
diablp$which[diablp$Cp == min(diablp$Cp),]

The answers are that the minimum $C_p = 5.56$, and the corresponding model is

```
FALSE TRUE TRUE TRUE TRUE FALSE FALSE TRUE FALSE
```

That means the variables 2, 3, 4, 5, 6, and 9 are in the model. That includes sex, BMI, blood pressure, and three of the blood counts. To fit that model, use `lm` (for “linear model”):

diablm <- lm(diab.y ~ SEX+BMI+BP+S1+S2+S5, data = diab.x)
summary(diablm)
```

Ridge regression

```r
lambda <- .1
xpxl <- xpx + lambda * diag(11)
betal <- solve(xpxl, t(x) %*% diab.y)
```

Lasso

dlasseo <- lars(x, diab.y, type = "lasso")
print(dlasseo)

Sequence of LASSO moves:

```
BMI S5 BP S3 SEX S6 S1 S4 S2 AGE S3 S3
Var 4 10 5 8 3 11 6 9 7 2 -8 8
Step 1 2 3 4 5 6 7 8 9 10 11 12
```

Donglin Zeng, Department of Biostatistics, University of North Carolina
Tuning parameter selection

- There are often tuning parameters that need to be chosen:
  - $k$ subset size in best-subset selection;
  - regularization parameters in all shrinkage methods.
- Larger regularization parameters lead to more shrunk coefficients (more sparse model) so less variable prediction; however, the resulting model yields higher bias in prediction.
- There should also be some bias and variance trade-off in tuning parameter selection.
- Some model selection criteria such as AIC and BIC can be used; however, they are based on information criterion so do not directly serve the purpose of prediction.
Mallow’s CP criterion function for subset selection

- The criterion is based on the prediction error

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

where \( \hat{f}_k \) is the estimated function from the \( k \) best feature variables.

- Assuming \( \text{Var}(Y - f(X)) = \sigma^2 \), this prediction error is

\[ \sigma^2 + (f(x_0) - E[\hat{f}_k(x_0)])^2 + \text{Var}(\hat{f}_k(x_0)) \]

so when average over \( x_0 \) from the empirical data, it is

\[ \sigma^2 + \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - E[\hat{f}_k(X_i)])^2 + \frac{\sigma^2}{n} \text{Trace}(X_k^T (X_k^T X_k)^{-1} X_k). \]
Since the in-sample error, \( n^{-1} \sum_{i=1}^{n} (Y_i - \hat{f}_k(X_i))^2 \), has an expectation approximated by

\[
\sigma^2 + \frac{1}{n} \sum_{i=1}^{n} \left\{ f(X_i)^2 - E[\hat{f}_k(X_i)]^2 \right\} - \frac{1}{n} \sum_{i=1}^{n} \text{Var}(\hat{f}_k(X_i)),
\]

the expectation of the prediction error is equal to the expectation of the in-sample error plus

\[
2\sigma^2 n^{-1} \text{Trace}(X_k^T (X_k^T X_k)^{-1} X_k) = 2\sigma^2 k/n.
\]

Mallow’s CP selects \( k \) to minimize

\[
\frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{f}_k(X_i))^2 + 2\hat{\sigma}^2 k/n.
\]
The goal is to mimic scenarios of learning prediction rules using training samples then evaluating performance in future data.

The idea is to randomly split data into training sample and testing sample – training sample is used to train prediction rules using learning methods; – testing sample is used to evaluate prediction errors of the learned rules.

To avoid incidence of good or bad splits, this procedure repeats multiple times.

Recommendation is often leave-one-out cross-validation, 5-fold or 10-fold cross-validation.

The best tuning parameters are chosen to minimize the average of the prediction errors.
Generalized cross-validation

- CV is computationally costly, especially leave-one-out CV.
- Some approximation, called generalized cross-validation, is often used in practice:

$$n^{-1} \sum_{i=1}^{n} \left[ \frac{Y_i - \hat{f}(X_i)}{1 - \text{trace}(\Sigma)/n} \right]^2,$$

where $\Sigma Y$ is the prediction for all the subjects.