Linear Methods for Regression

Hastie – Chap - 3; Part A

Introduction

- A linear regression model assumes that the regression function $E(Y\ |X)$ is linear in the inputs $X_1,\ldots,X_p$.

- They are simple and often provide an adequate and interpretable description of how the inputs affect the output. For prediction purposes they can sometimes outperform fancier nonlinear models, especially in situations with small numbers of training cases, low signal-to-noise ratio or sparse data.

Linear Regression Models and Least Squares

- Purpose: - to predict a real-valued output Y. The linear regression model has the form.

$$f(X) \ = \beta_0 + \sum_{j=1}^{p} X_j\beta_j \ . \qquad (3.1)$$

- The linear model either assumes that the regression function $E(Y\ |X)$ is linear, or that the linear model is a reasonable approximation. Here the β_{j} ’s are unknown parameters or coefficients, and the variables X_j can come from different sources:

- We have a set of training data $(x_1, y_1) \ldots (x_N, y_N)$ from which to estimate the parameters β. Each $x_i = (x_{i1}\ x_{i2} ,\ldots, x_{ip})^T$ is a vector of feature measurements for the i^{th} case. The most popular estimation method is $least\ squares$, in which we pick the coefficients $\beta = (\beta_0, \beta_1, \ldots, \beta_p)^T$ to minimize the residual sum of squares

$$RSS(\beta) = \sum_{i=1}^{N} (y_i - f(x_i))^2$$

$$= \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2. \qquad (3.2)$$

- From a statistical point of view, this criterion is reasonable if the training observations (x_i, y_i) represent independent random draws from their population. Even if the x_i's were not drawn randomly, the criterion is still valid if the y_i's are conditionally independent given the inputs x_i.

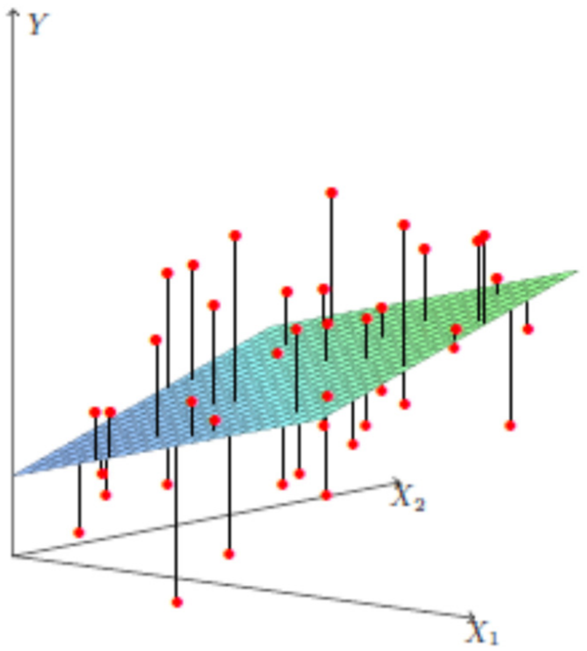


FIGURE 3.1. *Linear least squares fitting with $X \in \mathbb{R}^2$. We seek the linear function of X that minimizes the sum of squared residuals from Y.*

Figure 3.1 illustrates the geometry of least-squares fitting in the (p+1)-dimensional space occupied by the pairs (X, Y).

- Figure 3.1 illustrates the geometry of least-squares fitting in the $\mathbb{R}^{p+1}$ –dimensional space occupied by the pairs (X, Y). Note that (3.2) makes no assumptions about the validity of model (3.1); it simply finds the best linear fit to the data. Least squares fitting is intuitively satisfying no matter how the data arise; the criterion measures the average lack of fit.

- How do we minimize (3.2)?

Denote by $\mathbf{X}$ the $N \times (p+1)$ matrix with each row an input vector (with a 1 in the first position), and similarly let $\boldsymbol{y}$ be the N-vector of outputs in the training set. Then we can write the residual sum-of-squares as

$$RSS(\beta) = (\boldsymbol{y} - \mathbf{X}\beta)^T (\boldsymbol{y} - \boldsymbol{X}\beta). \tag{3.3}$$

- This is a quadratic function in the $p + 1$ parameters. Differentiating with respect to β, we obtain

$$\frac{\partial RSS}{\partial \beta} = -2\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\beta)$$

$$\frac{\partial^2 RSS}{\partial \beta \partial \beta^T} = \text{[redacted]} \qquad (3.4)$$

- Assuming (for the moment) that $\boldsymbol{X}$ has full column rank, and hence $\boldsymbol{X}^T\boldsymbol{X}$ is positive definite, we set the first derivative to zero

$$\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\beta) = 0 \qquad (3.5)$$

- To obtain the unique solution

$$\hat{\beta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}. \qquad (3.6)$$

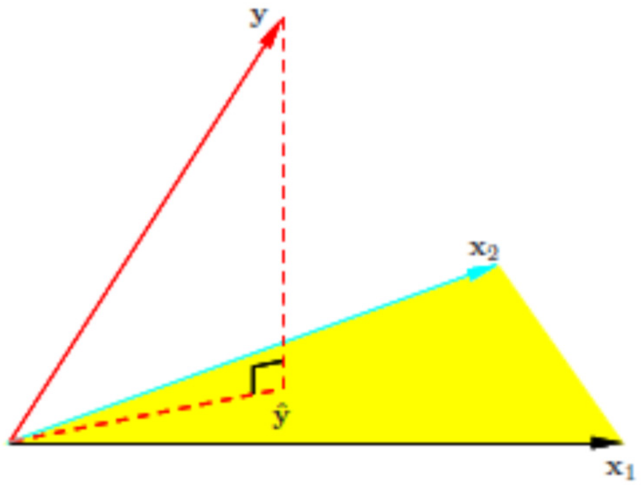


FIGURE 3.2. *The N-dimensional geometry of least squares regression with two predictors. The outcome vector y is orthogonally projected onto the hyperplane spanned by the input vectors x_1 and x_2. The projection $\hat{y}$ represents the vector of the least squares predictions*

- The predicted values at an input vector x_0 are given by $\hat{f}(x_0) = (1 : x_0)^T\hat{\beta}$;the fitted values at the training inputs are

$$\hat{y} = \boldsymbol{X}\hat{\beta} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T y, \qquad (3.7)$$

where $\hat{y}_i = \hat{f}(x_i)$ .The matrix $\mathbf{H} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$ appearing in equation (3.7) is sometimes called the “hat” matrix because it puts the hat on y.

- The hat matrix $\mathbf{H}$ computes the orthogonal projection, and hence it is also known as a projection matrix. It might happen that the columns of $\boldsymbol{X}$ are not linearly independent, so that $\mathbf{X}$ is not of full rank. for example, if two of the inputs were perfectly correlated, $(e.g., x_2 = 3\mathrm{x}_1)$.

- Then $\boldsymbol{X}^T\boldsymbol{X}$ is singular and the least squares coefficients $\hat{\beta}$ are not uniquely defined. However, the fitted values $\hat{y} \ = \boldsymbol{X}\hat{\beta}$ are still the projection of y onto the columns pace of $\boldsymbol{X}$; The non-full-rank case occurs most often when one or more qualitative inputs are coded in a redundant fashion.

- There is usually a natural way to resolve the non-unique representation, by recoding and/or dropping redundant columns in $\boldsymbol{X}$.

- Rank deficiencies can also occur in signal and image analysis, where the number of inputs p can exceed the number of training cases N. In this case, the features are typically reduced by filtering or else the fitting is controlled by regularization

- Assume that the observations y_i are uncorrelated and have constant variance σ^2, and that the x_i are fixed (non random). The variance–covariance matrix of the least squares parameter estimates is easily derived from (3.6) and is given by

$$Var\left(\hat{\beta} \right) = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \sigma^2. \qquad (3.8)$$

- Typically one estimates the variance σ^2 by.

$$\hat{\sigma}^2 = \frac{1}{N-p-1}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3.8}$$

- The $N - p - 1$ rather than N in the denominator makes $\hat{\sigma}^2$ an unbiased estimate of σ^2: $E(\hat{\sigma}^2) = \sigma^2$.

- The conditional expectation of Y is linear in $X_1,\ldots,X_p$. We also assume that the deviations of Y around its expectation are additive and Gaussian. Hence

$$Y = E(Y\,|X_1,\ldots,X_p) + \varepsilon$$

$$= \beta_0 + \textstyle\sum_{j=1}^{p} X_j\beta_j + \varepsilon\,, \qquad (3.9)$$

where the error ε is a Gaussian random variable with expectation zero and variance σ^2, written $\varepsilon \sim \mathrm{N}(0,\sigma^2)$ .Under (3.9), it is easy to show that

$$\hat{\beta} \sim N(\beta, (\boldsymbol{X}^T\boldsymbol{X})^{-1}\sigma^2). \qquad (3.10)$$

- This is a multivariate normal distribution with mean vector and variance–covariance matrix as shown.

The Gauss–Markov Theorem

- One of the most famous results in statistics asserts that the **least squares estimates of the parameters $\boldsymbol{\beta}$ have the smallest variance among all linear unbiased estimates**.

- This observation will lead us to consider biased estimates such as ridge regression later. We focus on estimation of any linear combination of the parameters $\theta = a^T\beta$; for example, predictions $f(x_0) = x_0^T\beta$ are of this form.

- The least squares estimate of $a^T\beta$ is

$$\hat{\theta} = a^T\hat{\beta} = a^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}. \qquad (3.17)$$

- Considering $\boldsymbol{X}$ to be fixed, this is a linear function $\boldsymbol{c}_0^T\boldsymbol{y}$ of the response vector $\boldsymbol{y}$. If we assume that the linear model is correct, $a^T\hat{\beta}$ is unbiased since

$$E(a^T\hat{\beta}) = E(a^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y})$$

$$= a^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{X}\beta$$

$$= a^T\beta. \qquad (3.18)$$

- The Gauss–Markov theorem states that if we have any other linear estimator $\tilde{\theta} = \boldsymbol{c}^T\boldsymbol{y}$ that is unbiased for $a^T\beta$, that is, $E(\boldsymbol{c}^T\boldsymbol{y}) = a^T\beta$, then

$$Var(a^T\,\hat{\beta}) \leq Var(\boldsymbol{c}^T\boldsymbol{y}). \qquad (3.19)$$

- Consider the mean squared error of an estimator $\bar{\theta}$ in estimating θ:

$$
\begin{aligned}
MSE(\bar{\theta}) &= E\left(\tilde{\theta} - \theta\right)^2 \\
&= Var\left(\tilde{\theta}\right) + \left[E\left(\tilde{\theta}\right) - \theta\right]^2. \qquad (3.20)
\end{aligned}
$$

- The first term is the variance, while the second term is the squared bias. The Gauss-Markov theorem implies that the least squares estimator has the smallest mean squared error of all linear estimators with no bias.

Multiple Regression from Simple Univariate Regression

- The linear model (3.1) with $p > 1$ inputs is called the multiple linear regression model.

- Suppose first that we have ***a univariate*** model with no intercept, that is,

$$Y \; = \; X\beta + \varepsilon. \qquad (3.23)$$

- The least squares estimate and residuals are

$$\hat{\beta} \; = \frac{\sum_1^N x_i y_i}{\sum_1^N x_i^2},$$

$$r_i = \; y_i - \; x_i \, \hat{\beta} \, . \qquad (3.24)$$

Recollect from Method – 1 – LSQ:

$$m = \frac{N\sum_{i=1}^{N}(X_iY_i) - \sum_{i=1}^{N}X_i\sum_{i=1}^{N}Y_i}{DEN};$$

$$C = \frac{\sum_{i=1}^{N}Y_i\sum_{i=1}^{N}X_i^2 - \sum_{i=1}^{N}X_i\sum_{i=1}^{N}(X_iY_i)}{DEN};$$

where,

$$DEN = N\sum_{i=1}^{N}X_i^2 - (\sum_{i=1}^{N}X_i)^2$$

$$DEN' = N\sum_{i=1}^{N}X^2\,; C = 0;$$

$$\mathrm{m} = \frac{N\sum_{i=1}^{N}X_iY_i}{N\sum_{i=1}^{N}X^2} =;$$

- Convenient vector notation, we let $\mathrm{y} = (y_1, \ldots, y_N)^T$, $\mathrm{x} = (x_1, \ldots , x_N)^T$ and define

$$\langle \mathrm{x}, \mathrm{y} \rangle = \textstyle\sum_{i=1}^{N} x_i y_i \,,$$

$$= \mathrm{x}^T \mathrm{y}, \qquad (3.25)$$

- *Inner product* between x and y .Then we can write

$$\hat{\beta} = \frac{\langle \mathbf{x}, \boldsymbol{y} \rangle}{\langle \mathbf{x}, \mathbf{x} \rangle} \,,$$

$$r = \mathrm{y} - \mathrm{x}\hat{\beta}. \qquad (3.26)$$

- As we will see, this simple univariate regression provides the building block for multiple linear regression.

- Suppose next that the inputs $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_p$ (the columns of the data matrix $\boldsymbol{X}$) are orthogonal; that is $\langle \mathrm{x}_\mathrm{j}, \mathrm{x}_k \rangle = 0$ for all $j \neq k$. Then it is easy to check that the multiple least squares estimates $\hat{\beta}_j$ are equal to $\langle \mathrm{x}_j, \mathrm{y} \rangle / \langle \mathrm{x}_j, \mathrm{x}_j \rangle$—the univariate estimates. In other words, when the inputs are orthogonal, they have no effect on each other's parameter estimates in the model.

- Orthogonal inputs occur most often with balanced, designed experiments (where orthogonality is enforced), but almost never with observational data.

- Hence we will have to orthogonalize them. Suppose next that we have an intercept and a single input x. Then the least squares coefficient of x has the form

$$\hat{\beta}_1 = \frac{\langle \mathrm{x} - \bar{x}\mathbf{1}, y \rangle}{\langle \mathrm{x} - \bar{x}\mathbf{1}, \mathrm{x} - \bar{x}\mathbf{1} \rangle}, \qquad (3.27)$$

- where $\bar{x} = \sum_i x_i / N$, and $\mathbf{1} = \mathrm{x}_0$, the vector of N ones.

- We can view the estimate (3.27) as the result of two applications of the simple regression(3.26). The steps are:

  1. Regress x on $\mathbf{1}$ to produce the residual $z = \mathrm{x} - \bar{x}\mathbf{1}$;

  2. Regress y on the residual $\boldsymbol{z}$ to give the coefficient $\hat{\beta}_1$.

Linear Methods for Regression

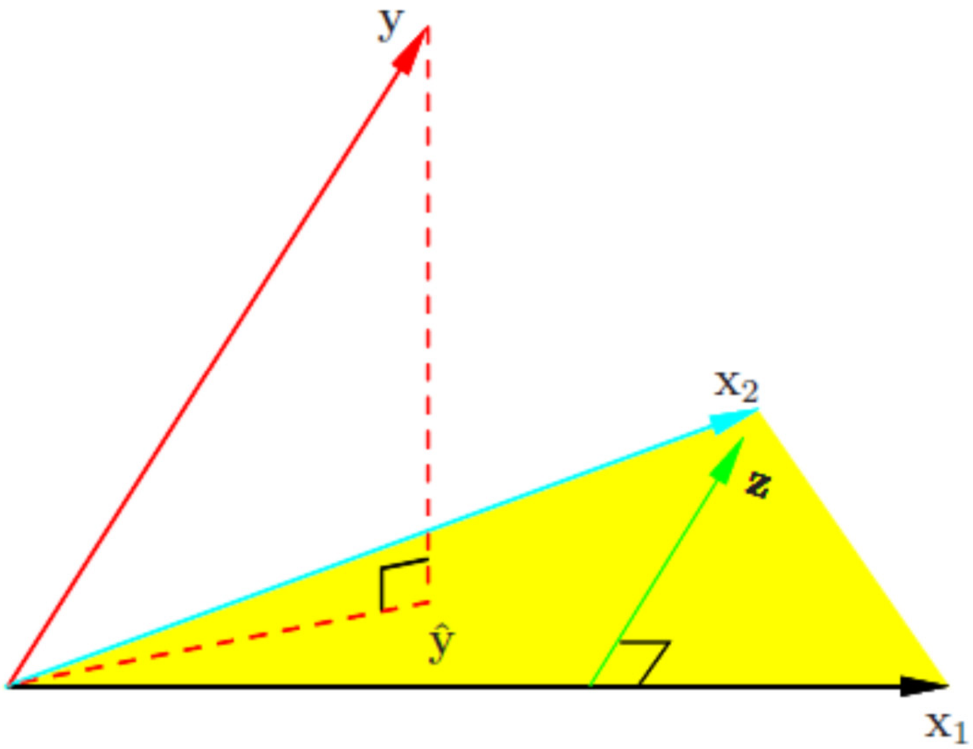


***FIGURE 3.4.** Least squares regression by orthogonalization of the inputs. The vector $\mathbf{x}_2$ is regressed on the vector $\mathbf{x}_1$, leaving the residual vector $\boldsymbol{z}$. The regression of $\boldsymbol{y}$ on $\boldsymbol{z}$ gives the multiple regression coefficient of $\mathbf{x}_2$. Adding together the projections of $\boldsymbol{y}$ on each of $\mathbf{x}_1$ and $\boldsymbol{z}$ gives the least squares fit $\hat{\boldsymbol{y}}$.*

- In this procedure, “regress b on a” means a simple univariate regression of b on a with no intercept, producing coefficient $\hat{\gamma} = \langle a, b\rangle / \langle a, a\rangle$ and residual vector $\mathbf{b} - \hat{\gamma}\mathbf{a}$. We say that $\boldsymbol{b}$ is adjusted for a, or is “orthogonalized” with respect to a.

- Step 1 orthogonalizes x with respect to $\mathrm{x}_0 = \mathbf{1}$. Step 2 is just a simple univariate regression, using the orthogonal predictors $\mathbf{1}$ and z. Figure 3.4 shows this process for two general inputs x_1 and x_2. The orthogonalization does not change the subspace spanned by x_1 and x_2, it simply produces an orthogonal basis for representing it.

- This recipe generalizes to the case of p inputs, as shown in Algorithm 3.1. Note that the inputs $\mathrm{z}_0, \ldots, \mathrm{z}_{\mathrm{j}-1}$ in step 2 are orthogonal, hence the simple regression coefficients computed there are in fact also the multiple regression coefficients.

Algorithm 3.1 *Regression by Successive Orthogonalization*.

```
1. Initialize $z_0 = \mathrm{x}_0 = \mathbf{1}$.

2. For $j = 1, 2, \ldots, p$
       Regress $\mathrm{x}_j$ on $\mathrm{z}_0, \mathrm{z}_1, \ldots, , \mathrm{z}j \,\_\, 1$ to
       produce coefficients:

           $\hat{\gamma}_{\ell j} = \frac{\langle \mathbf{z}_\ell, \mathbf{x}_\ell \rangle}{\langle \mathbf{z}_\ell, \mathbf{z}_\ell \rangle}, \ell = 0, \ldots, j - 1$

         and residual vector
         $z_j = \mathrm{x}_j - \sum_{k=0}^{j-1} \hat{\gamma}_{kj} \mathrm{z}_k$.

3. Regress $\mathrm{y}$ on the residual $\mathrm{z}_p$ to give the estimate $\hat{\beta}_p$.
```

- The result of this algorithm is

$$\hat{\beta}_p = \frac{\langle \mathrm{zp},\ \mathrm{y} \rangle}{\langle \mathrm{z_p}, \mathrm{z_p} \rangle} \qquad (3.28)$$

- Re-arranging the residual in step 2, we can see that each of the $\mathbf{x}_j$ is a linear combination of the z_k, $k \leq j$.

- Since the $\boldsymbol{z}_j$ are all orthogonal, they form a basis for the column space of $\mathbf{X}$, and hence the least squares projection onto this subspace is $\hat{\mathrm{y}}$.

- Since z_p alone involves x_p (with coefficient 1), we see that the coefficient (3.28) is indeed the multiple regression coefficient of y on x_p.

- *The multiple regression coefficient $\hat{\beta}_j$ represents the additional contribution of x_j on $\boldsymbol{y}$, after x_j has been adjusted for* $\mathrm{x}_0, \mathrm{x}_1, \ldots, \mathrm{x}_{j-1}, \mathrm{x}_{j+1} \ldots, \mathrm{x}_p$.

- Algorithm 3.1 is known as the *Gram–Schmidt* procedure for multiple regression,. We can obtain from it not just $\hat{\beta}_p$, but also the entire multiple least squares fit,

- We can represent step 2 of Algorithm 3.1 in matrix form:

$$\mathbf{X} = \mathbf{Z\Gamma}, \tag{3.30}$$

where $\mathbf{Z}$ has as columns the z_j (in order), and $\mathbf{\Gamma}$ is the upper triangular matrix with entries $\hat{\gamma}_{kj}$.

- Introducing the diagonal matrix $\mathbf{D}$ with jth diagonal entry $D_{jj} = \| \mathrm{z}_j \|$, we get

$$\mathbf{X} = \mathbf{Z}\mathbf{D}^{-1}\mathbf{D}\boldsymbol{\Gamma}$$

$$= \mathbf{Q}\mathbf{R}, \qquad (3.31)$$

the so-called QR decomposition of $\mathbf{X}$. Here $\mathbf{Q}$ is an $N \times (p+1)$ orthogonal matrix, $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, and $\mathbf{R}$ is a $(p + 1) \times (p + 1)$ upper triangular matrix.

- The $\boldsymbol{QR}$ decomposition represents a convenient orthogonal basis for the column space of $\boldsymbol{X}$. It is easy to see, for example, that the least squares solution is given by

$$\hat{\beta} =$$ [redacted] (3.32)

$$\hat{y} =$$ [redacted] (3.33)

- Equation (3.32) is easy to solve as $\boldsymbol{R}$ is upper triangular

Multiple Outputs

- Suppose we have multiple outputs $Y_1, Y_2, \ldots, Y_K$ that we wish to predict from our inputs $X_0, X_1, X_2, \ldots, X_p$. We assume a linear model for each output

$$Y_k = \beta_{0k} + \sum_{j=1}^{p} X_j \beta_{jk} + \varepsilon_k \tag{3.34}$$

$$= f_k(X) + \varepsilon_k. \tag{3.35}$$

- With N training cases we can write the model in matrix notation

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E}. \tag{3.36}$$

- Here $\mathbf{Y}$ is the $N \times K$ response matrix, with ik entry y_{ik}, $\mathbf{X}$ is the $N \times (p+1)$ input matrix, $\mathbf{B}$ is the $(p + 1) \times K$ matrix of parameters and $\mathbf{E}$ is the $N \times K$ matrix of errors.

- A straightforward generalization of the univariate loss function (3.2) is

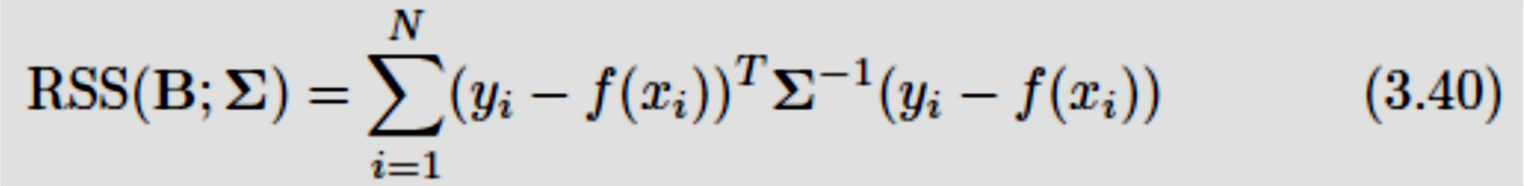
$$RSS(\mathbf{B}) = \sum_{k=1}^{K}\sum_{i=1}^{N}(y_{ik} - f_k(x_i))^2 \qquad (3.37)$$

$$= tr[(\mathbf{Y} - \mathbf{XB})^T(\mathbf{Y} - \mathbf{XB})]. \qquad (3.38)$$

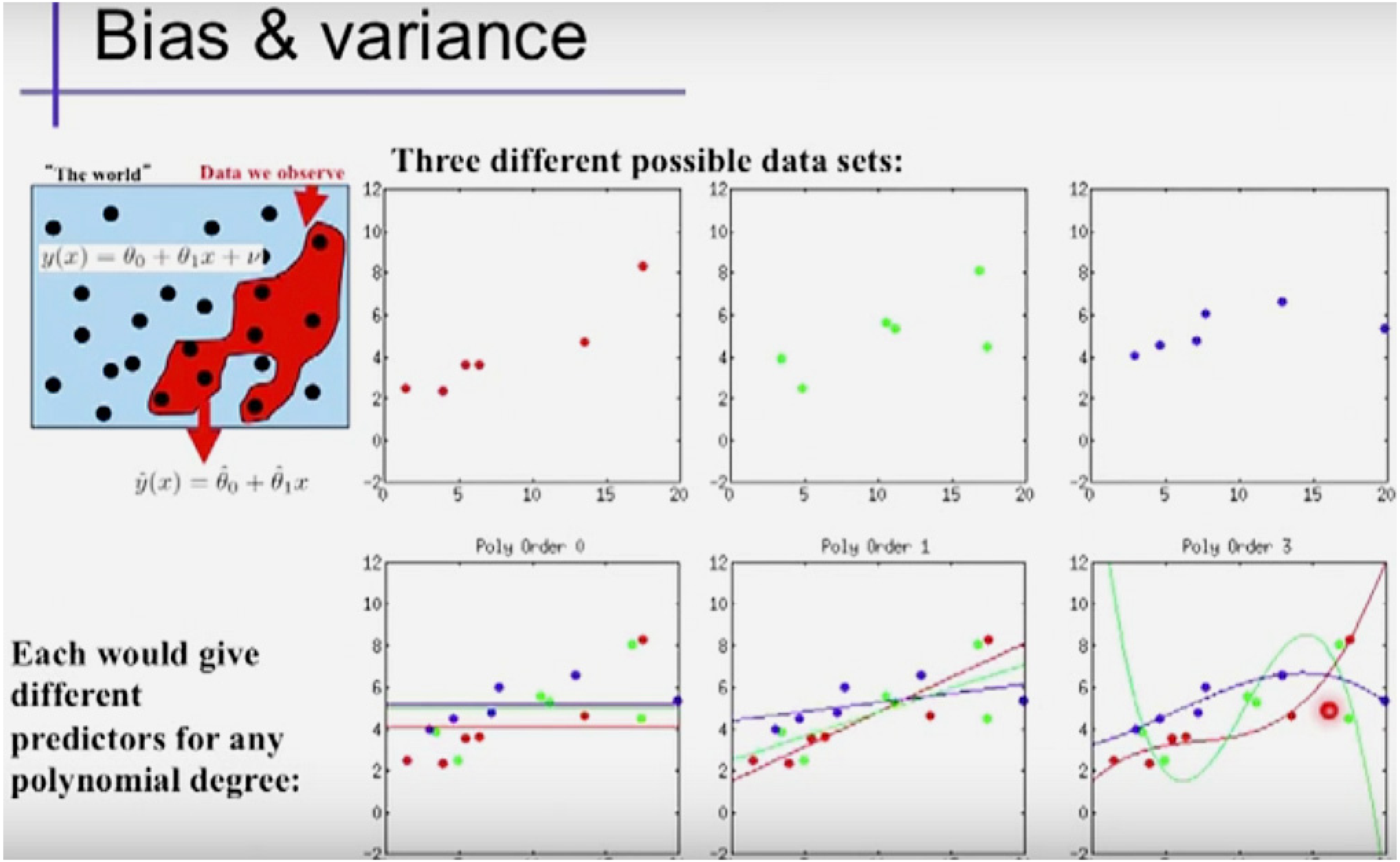
- The least squares estimates have exactly the same form as before

$$\hat{\mathbf{B}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}. \qquad (3.39)$$

If the errors $\varepsilon = (\varepsilon_1,......, \varepsilon_\kappa)$ in (3.34) are correlated; if $\mathrm{Cov}(\varepsilon) = \Sigma$, then the multivariate weighted criterion:



$$\mathrm{RSS}(\mathbf{B};\mathbf{\Sigma}) = \sum_{i=1}^{N}(y_i - f(x_i))^T\mathbf{\Sigma}^{-1}(y_i - f(x_i)) \qquad (3.40)$$



Bias & variance

Three different possible data sets:

Each would give different predictors for any polynomial degree:

Subset Selection

- There are two reasons why we are often not satisfied with the least squares estimates (3.6). $\hat{\beta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$

  ❖ The first is $prediction\ accuracy$: the least squares estimates *(not just linear)* often have low bias but large variance. Prediction accuracy can sometimes be improved by shrinking or setting some coefficients to zero. By doing so we sacrifice a little bit of bias to reduce the variance of the predicted values, and hence may improve the overall prediction accuracy.

  ❖ The second reason is $interpretation$. With a large number of predictors, we often would like to determine a smaller subset that exhibit the strongest effects. In order to get the "big picture," we are willing to sacrifice some of the small details.

- In this section, we describe a number of approaches to variable subset selection with linear regression. In later sections we discuss shrinkage and hybrid approaches for controlling variance, as well as other dimension-reduction strategies. These all fall under the general heading ***model selection*.**

- With subset selection we retain only a subset of the variables, and eliminate the rest from the model. Least squares regression is used to estimate the coefficients of the inputs that are retained. There are a number of different strategies for choosing the subset.

$\boldsymbol{Best - Subset\ Selection}$

- Best subset regression finds for each $k \in \{0, 1, 2, \ldots, p\}$ the subset of size k that gives smallest residual sum of squares (3.2). An efficient algorithm— the leaps and bounds procedure (Furnivall and Wilson, 1974)—makes this feasible for p as large as 30 or 40.

- The lower boundary represents the models that are eligible for selection by the best-subsets approach. The best-subset curve (blue lower boundary in Figure 3.5) is necessarily decreasing, so cannot be used to select the subset size k.

- There are a number of criteria that one may use; typically we choose the smallest model that minimizes an estimate of the expected prediction error. E.g. cross-validation to estimate prediction error and select k; the AIC criterion is a popular alternative.

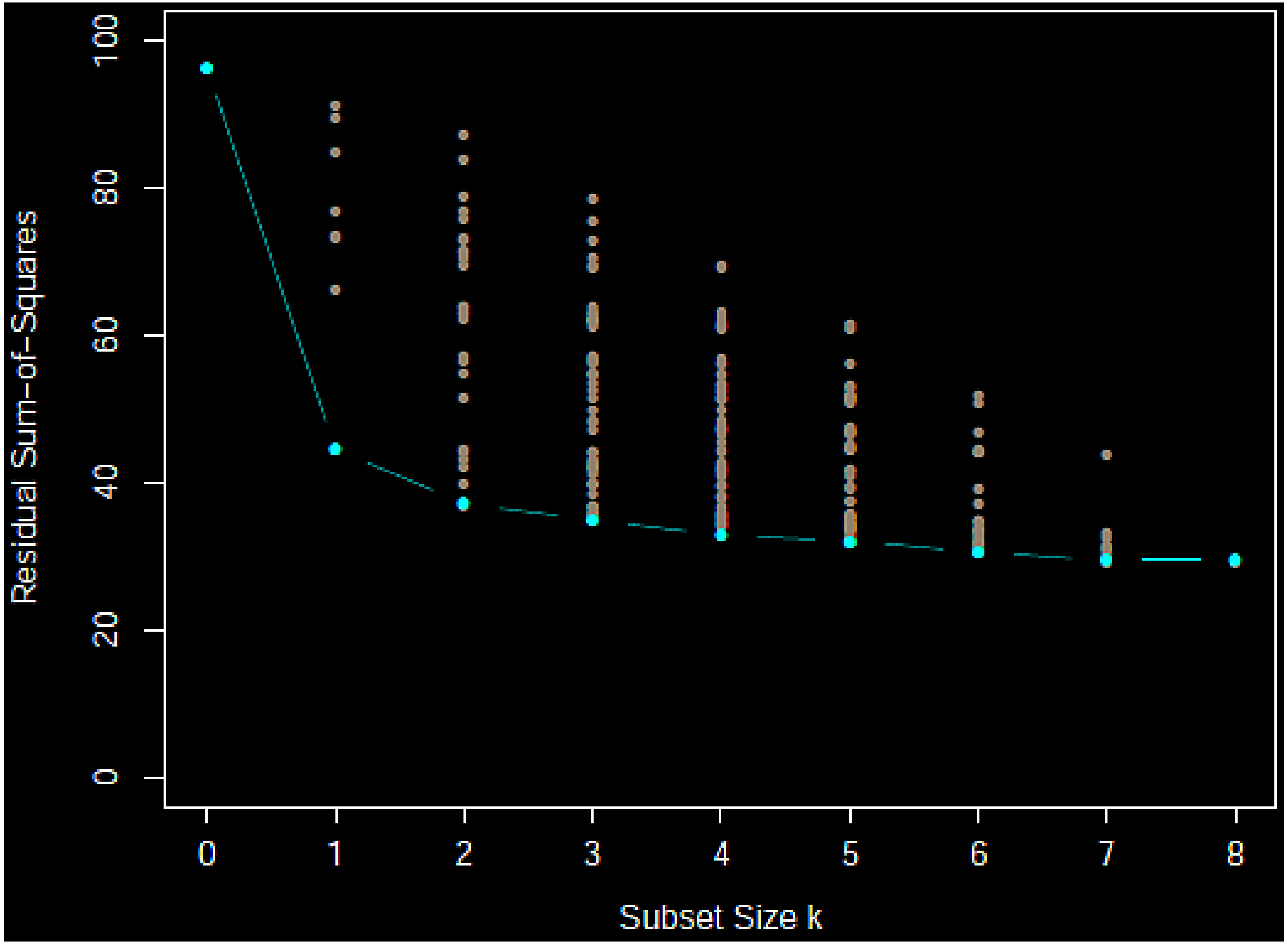


FIGURE 3.5. All possible subset models for an (the prostate cancer) example. At each subset size is shown the residual sum-of-squares for each model of that size.

Forward- and Backward-Stepwise Selection

- Rather than search through all possible subsets (which becomes infeasible for p much larger than 40), we can seek a good path through them.

- *Forward-stepwise selection* starts with the intercept, and then sequentially adds into the model the predictor that most improves the fit.

- Like best-subset regression, forward stepwise produces a sequence of models indexed by k, the subset size, which must be determined.

- Forward-stepwise selection is a $greedy\ algorithm$, producing a nested sequence of models. In this sense it might seem sub-optimal compared to best-subset selection.

- However, there are several reasons why it might be preferred:

  - ❖ $Computational$; for large p we cannot compute the best subset sequence, but we can always compute the forward stepwise sequence (even when $p \gg N$).

  - ❖ $Statistical$; a price is paid in variance for selecting the best subset of each size; forward stepwise is a more constrained search, and will have lower variance, but perhaps more bias.

- *Backward-stepwise selection* starts with the full model, and sequentially deletes the predictor that has the least impact on the fit. The candidate for dropping is the variable with the smallest Z-score. Backward selection can only be used when $N > p$, while forward stepwise can always be used.

- Figure 3.6 shows the results of a small simulation study to compare best-subset regression with the simpler alternatives forward and backward selection.

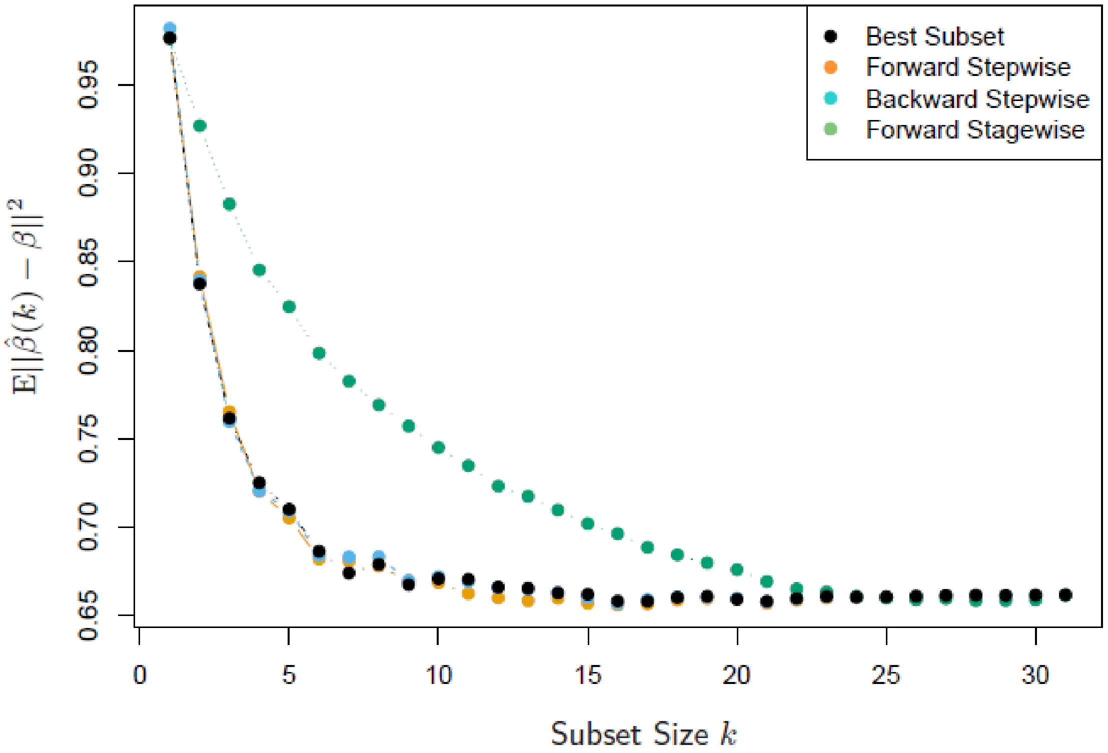


FIGURE 3.6. *Comparison of four subset-selection techniques on a simulated linear regression problem* $Y = X^T\beta + \varepsilon$. *There are* $N = 300$ *observations on* $p = 31$ *standard Gaussian variables, with pairwise correlations all equal to* 0.85. *For* 10 *of the variables, the coefficients are drawn at random from a* $N(0, 0.4)$ *distribution; the rest are zero. The noise* $\varepsilon \sim \mathrm{N}(0, 6.25)$, *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* β.

Forward-Stagewise Regression

- Forward-stagewise regression (FS) is even more constrained than forwardstepwise regression.

- It starts like forward-stepwise regression, with an intercept equal to $\bar{y}$, and centered predictors with coefficients initially all 0.

- At each step the algorithm ***identifies the variable most correlated with the current residual***. It then computes the simple linear regression coefficient of the residual on this chosen variable, and then adds it to the current coefficient for that variable.

- This is continued till none of the variables have correlation with the residuals—i.e. the least-squares fit when $N > p$.

- Unlike forward-stepwise regression, none of the other variables are adjusted when a term is added to the model.

- As a consequence, forward stagewise can take many more than p steps to reach the least squares fit, and historically has been dismissed as being inefficient.

- It turns out that this “slow fitting” can pay dividends in high-dimensional problems.

- Forward-stagewise regression is included in Figure 3.6. In this example it takes over 1000 steps to get all the correlations below 10^{-4}. For subset size k, we plotted the error for the last step for which there where k nonzero coefficients. Although it catches up with the best fit, it takes longer to do so.

Contd. - in part B;