Linear Methods for Regression

Statistical Data Mining I

Rachael Hageman Blair

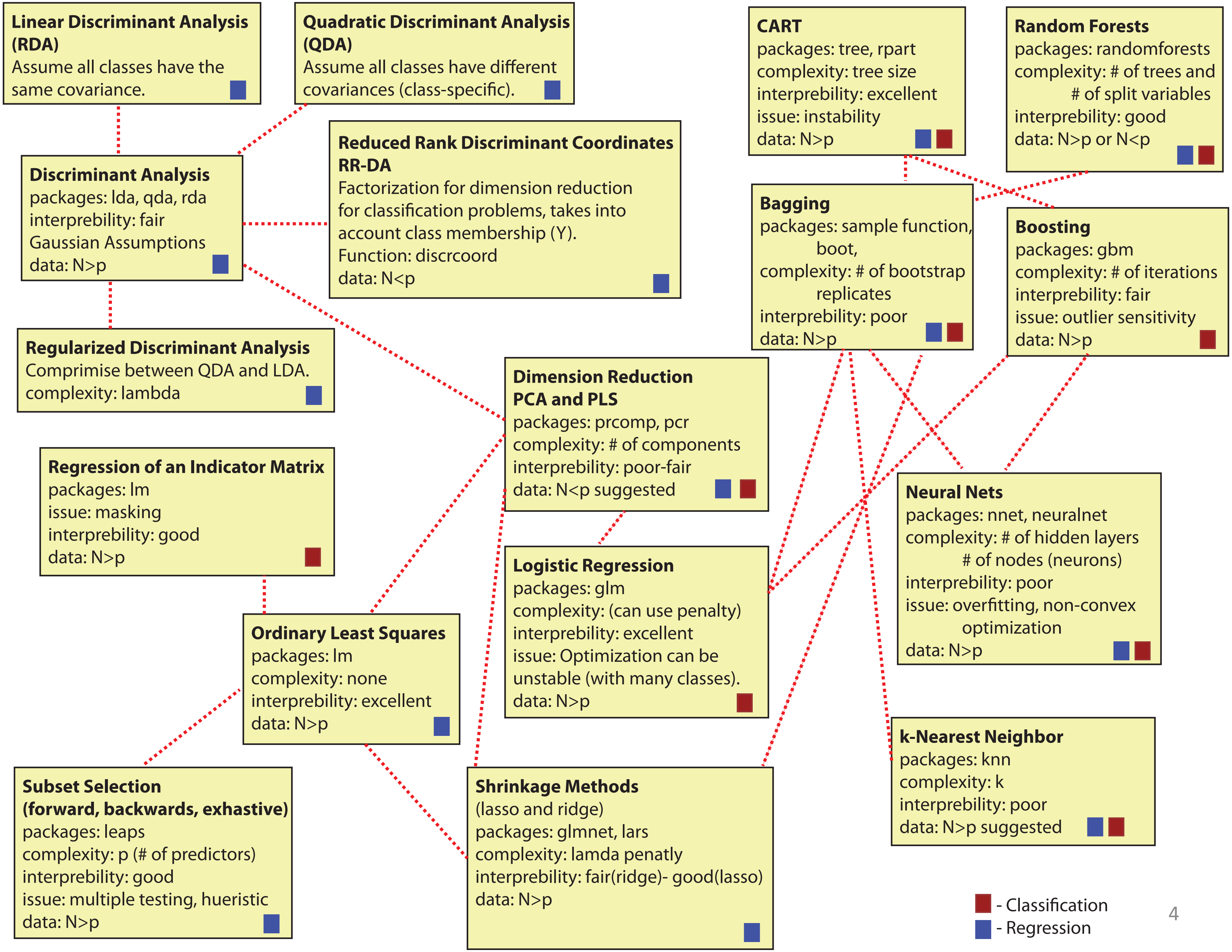
Recall: Functional Approximation

There are two reasons to estimate f(x):

(1) Prediction – minimize “reducible error”.

(2) Inference

$\hat{f}(x)$



Linear Discriminant Analysis (RDA)
Assume all classes have the same covariance. 🟦

Quadratic Discriminant Analysis (QDA)
Assume all classes have different covariances (class-specific). 🟦

Discriminant Analysis
packages: lda, qda, rda
interprebility: fair
Gaussian Assumptions
data: N>p 🟦

Reduced Rank Discriminant Coordinates RR-DA
Factorization for dimension reduction for classification problems, takes into account class membership (Y).
Function: discrcoord
data: N<p 🟦

Regularized Discriminant Analysis
Comprimise between QDA and LDA.
complexity: lambda 🟦

Regression of an Indicator Matrix
packages: lm
issue: masking
interprebility: good
data: N>p 🟥

Ordinary Least Squares
packages: lm
complexity: none
interprebility: excellent
data: N>p 🟦

Subset Selection (forward, backwards, exhastive)
packages: leaps
complexity: p (# of predictors)
interprebility: good
issue: multiple testing, hueristic
data: N>p 🟦

Dimension Reduction PCA and PLS
packages: prcomp, pcr
complexity: # of components
interprebility: poor-fair
data: N<p suggested 🟦 🟥

Logistic Regression
packages: glm
complexity: (can use penalty)
interprebility: excellent
issue: Optimization can be unstable (with many classes).
data: N>p 🟥

Shrinkage Methods
(lasso and ridge)
packages: glmnet, lars
complexity: lamda penatly
interprebility: fair(ridge)- good(lasso)
data: N>p 🟦

CART
packages: tree, rpart
complexity: tree size
interprebility: excellent
issue: instability
data: N>p 🟦 🟥

Random Forests
packages: randomforests
complexity: # of trees and
# of split variables
interprebility: good
data: N>p or N<p 🟦 🟥

Bagging
packages: sample function,
boot,
complexity: # of bootstrap
replicates
interprebility: poor
data: N>p 🟦 🟥

Boosting
packages: gbm
complexity: # of iterations
interprebility: fair
issue: outlier sensitivity
data: N>p 🟥

Neural Nets
packages: nnet, neuralnet
complexity: # of hidden layers
# of nodes (neurons)
interprebility: poor
issue: overfitting, non-convex
optimization
data: N>p 🟦 🟥

k-Nearest Neighbor
packages: knn
complexity: k
interprebility: poor
data: N>p suggested 🟦 🟥

🟥 - Classification
🟦 - Regression

Model Selection and Bias-Variance tradeoff

Back to nearest neighbors: The expected prediction error (EPE) at x_0:

$$
\begin{aligned}
EPE_k(x_0) &= E\left[(Y - \hat{f}_k(x_0))^2 \mid X = x_0\right] \\
&= \sigma^2 + \left[Bias^2(\hat{f}_k(x_0)) + Var_T(\hat{f}_k(x_0))\right] \\
&= \sigma^2 + \boxed{\left[f(x_0) - \frac{1}{k}\sum_{l=1}^{k} f(x_l)\right]^2 + \frac{\sigma^2}{k}.}
\end{aligned}
$$

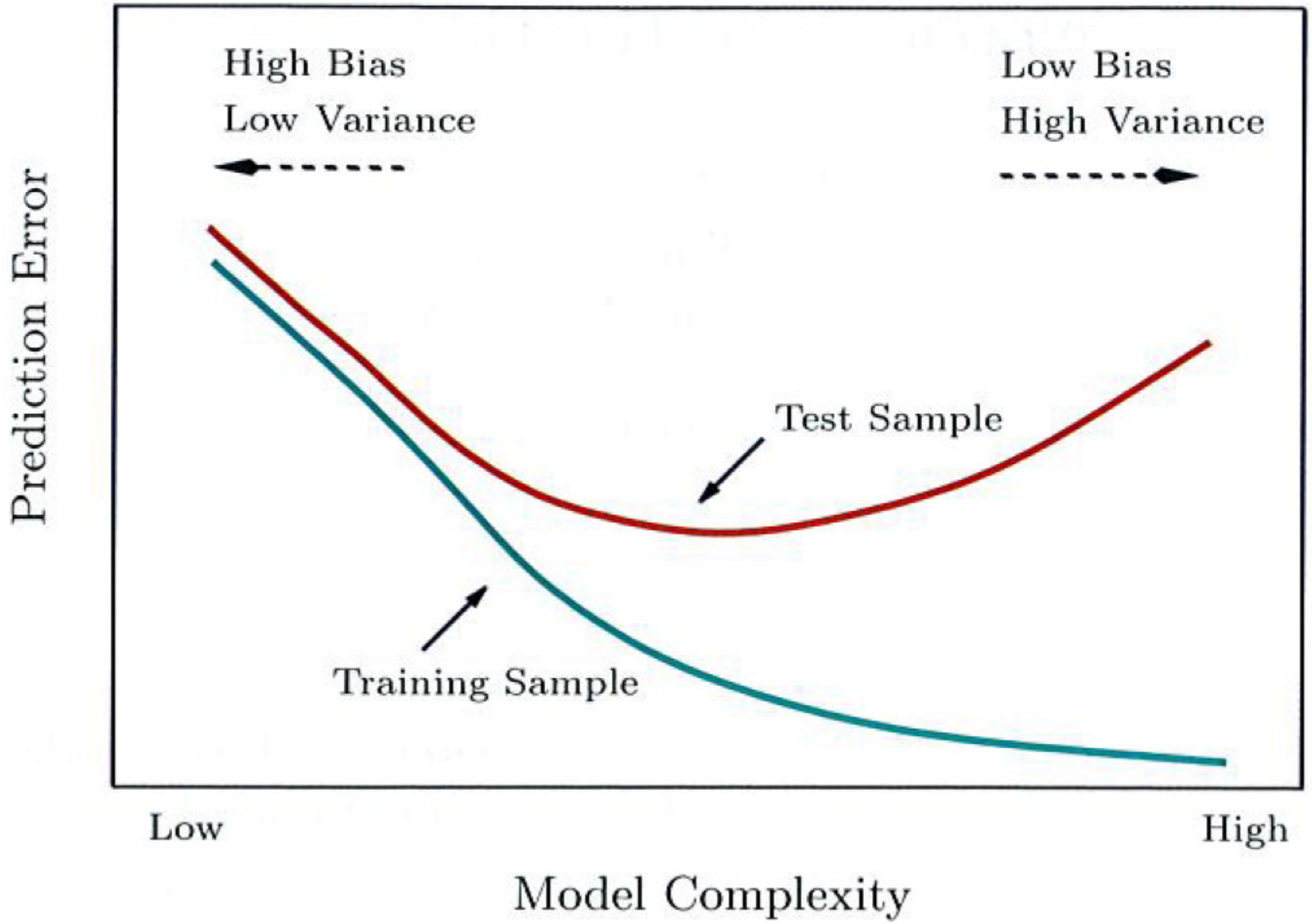
MSE

Irreducible error – beyond out control.

Squared difference between the true mean and the estimated. Likely to Increase with k.

The variance of an average. As k Increases this decreases.

Model Selection and Bias-Variance tradeoff



Introduction to Regression

- The linear regression model:

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

- Input vector: $X^T = (X_1, X_2, \ldots, X_p)$.
- Output vector (real-valued): Y.
- Predict Y from X by $f(X)$ such that the expected loss function: $E(L(Y, f(X)))$ is minimized.
- Square Loss: $L(Y, f(X)) = (Y - f(X))^2$.
- The optimal predictor:

$$f^*(X) = \arg\min_{f(X)} E(Y - f(X))^2$$

$$= E(Y \mid X)$$ aka The Regression Function

Introduction to Regression

An Example: The number of active physicians in a Standard Metropolitan Statistical Area (SMSA), denoted by Y, is expected to be related to the total population (X_1 , measured in thousands), (X_2, measured in square miles), and total personal income ($X_{3,}$ measured in millions of dollars).   Data are collected for 141 SMSAs, and shown in the table:

i:	1	2	3	…	139	140	141
X_1	9387	7031	7017	…	232	232	231
X_2	1348	4069	3719	…	1011	813	654
X_3	72100	52737	54542	…	1337	1589	1148
Y	25627	15389	13326	…	264	371	140

Goal: Predict Y from X_1, X_2, and X_3.

Linear Methods and Least Squares

- Assumption: the regression function is linear:

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

- What if the model is not true?
  - It is a good approximation.
  - Because of lack of training data/ or smarter algorithms, it is the most we can extract robustly from the data.

- Comments on X_j:
  - Quantitative inputs
  - Quantitative inputs: dummy coding of “x-level factors”.
  - Transformations of quantitative inputs, e.g., log, square root.
  - Basis expansions: $X_2 = X_1^2, X_3 = X_1^3$.
  - Interactions: $X_1 \cdot X_2$.

Linear Methods and Least Squares

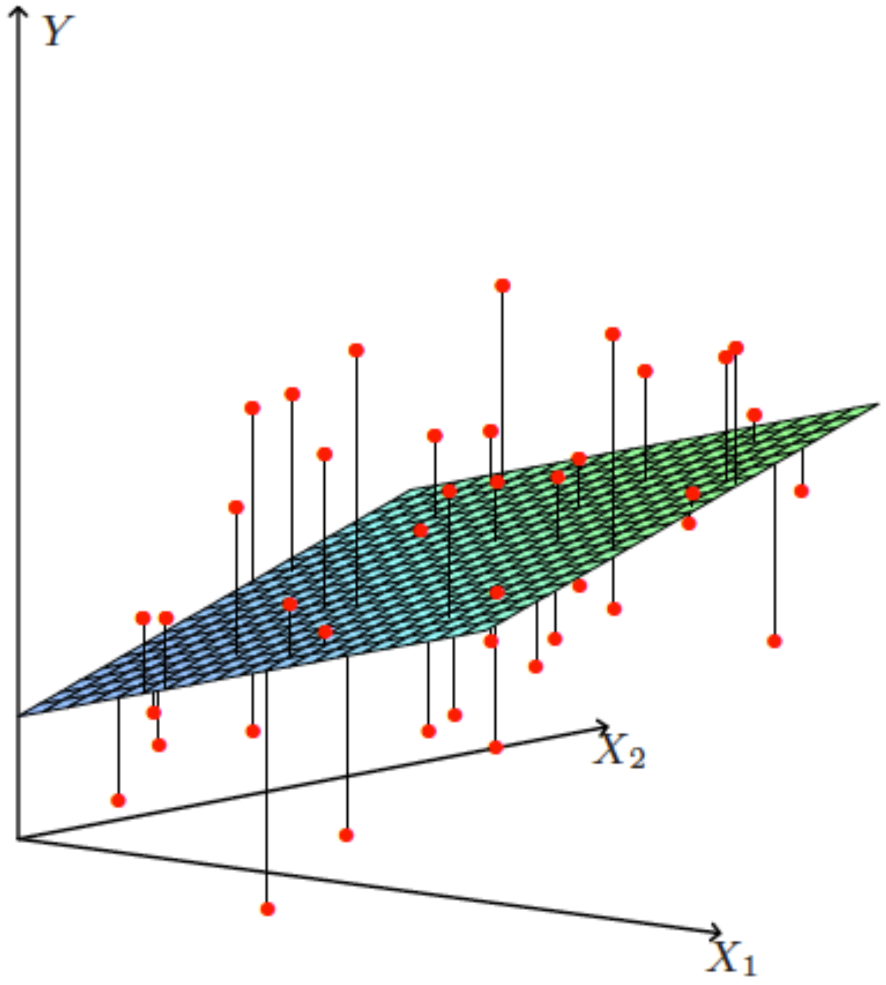
- **Least Squares Estimation:** the problem of finding the regression function $E(Y \mid X)$ comes down to estimating the regression parameters β, such that the residual sum of squares is minimized:

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

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

- **Training Data:** $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N).$

Linear Methods and Least Squares



Linear Methods and Least Squares

In matrix-vector form:

Input matrix:

$$X = \begin{pmatrix} 1 & x_{1,1} & \cdots & x_{1,p} \\ 1 & x_{2,1} & \cdots & x_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N,1} & \cdots & x_{N,p} \end{pmatrix} \in R^{N\times(p+1)}.$$

Output vector:

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \in R^{N\times 1}.$$

Linear Methods and Least Squares

The estimated regression parameters: $\hat{\beta}$.

The fitted values at the training points: $\hat{y}_i = \hat{\beta}_0 + \sum_{j=1}^{p} x_{ij}\hat{\beta}_j$.

The least squares estimate. $X\hat{\beta} = y$

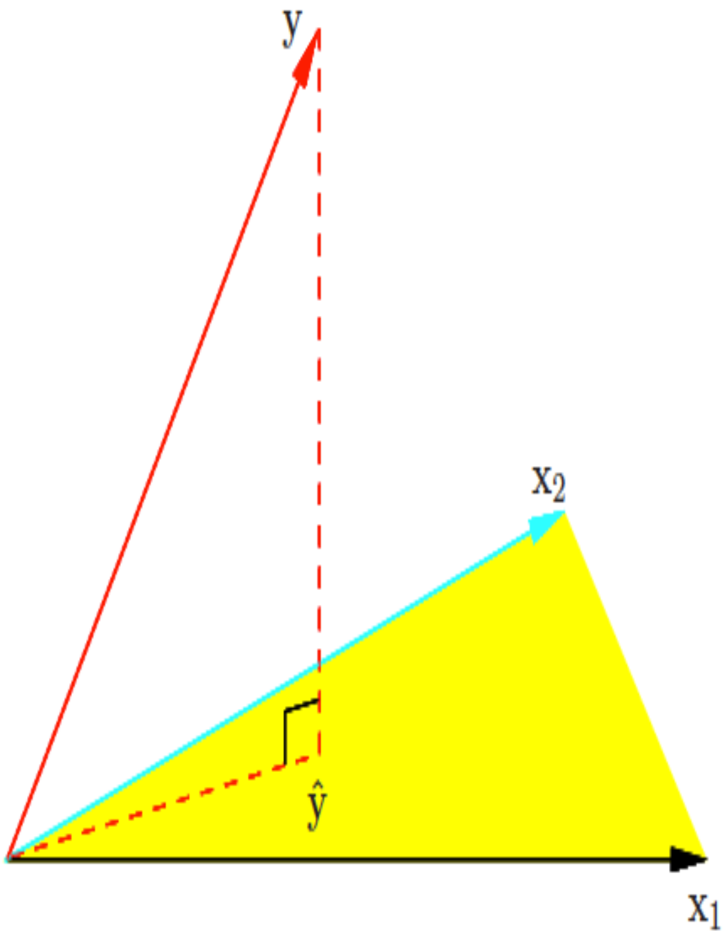
$$X^T X\hat{\beta} = X^T y$$

$$\hat{\beta} = \underbrace{(X^T X)^{-1}}_{\text{The Moore-Penrose psuedoinverse}} X^T y$$

The fitted values: $\hat{y} = X\hat{\beta} = \underbrace{X(X^T X)}_{\text{The Hat Matrix}}{}^{-1} X^T$

Geometric Interpretation

- The input vectors span a N-dimensional subspace $\mathbf{R}^N$.
- The output vector y is orthogonally projected onto the hyper-plane spanned by the input vectors.
- The residual: $y - \hat{y}$ is orthogonal to the subspace spanned by X.
- The fitted value $\hat{y}$ lies in the subspace spanned by X.
- The geometric interpretation is useful for understanding coefficient shrinkage and subset selection.



Least Squares Properties

Assumptions: The linear model is true, the observations, y_i , are uncorrelated and have a constant variance σ^2 .

The **variance-covariance matrix:**

$$Var(\hat{\beta}) = (X^T X)^{-1}\sigma^2$$

Where the variance σ^2 is estimated by:

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

If we further assume: $Y = E(Y \mid X_1,\ldots,X_p) + \varepsilon$, where $\varepsilon \sim N(0,\sigma^2)$, and is independent of X . The input, X , is regarded as fixed, and Y is random due to ε. Then we have the following properties:

- Estimated coefficients are from a multivariate normal: $\hat{\beta} \sim N(\beta,(X^T X)^{-1}\sigma^2)$.
- Confidence intervals can be computed and significance tests can be done.

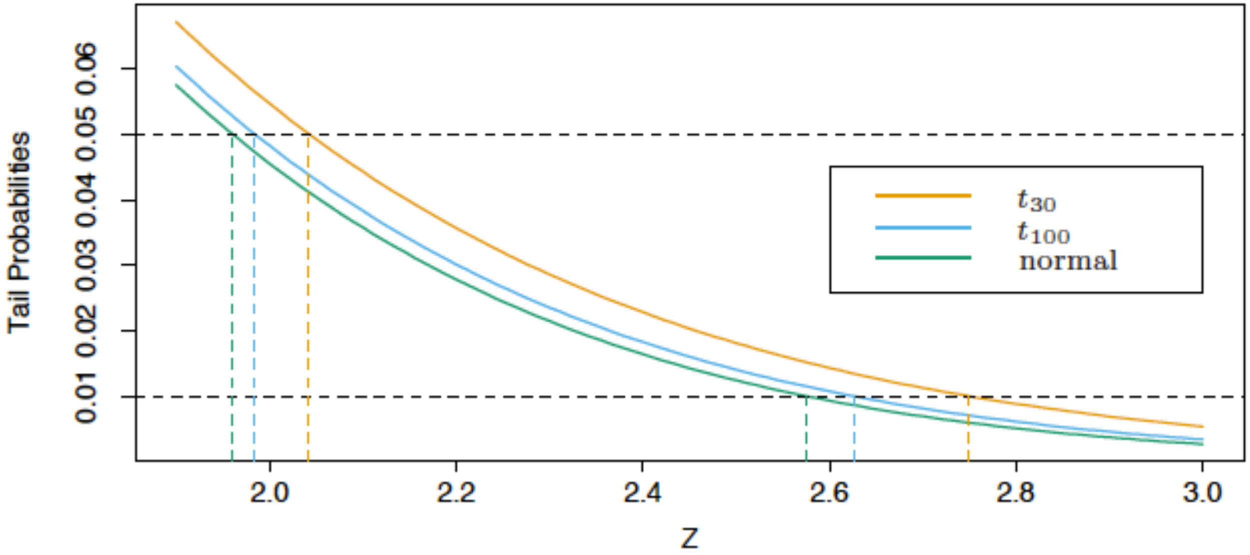
Testing coefficient significance

Coefficient significance: To test the hypothesis that $\beta_j = 0$ we us the z-score:

$$z_j = \frac{\hat{\beta}_j}{\hat{\sigma}\sqrt{v_j}},$$

where v_j is the *j*th diagonal element of $(X^T X)^{-1}$.

Under the null hypothesis, z_j, is distributed as t_{N-p-1}.



Testing group coefficient significance

Simultaneous Testing of Coefficient significance: To test the hypothesis that a group of particular coefficients is significant, we use the F statistic:

$$F = \frac{(RSS_0 - RSS_1)/(p_1 - p_0)}{RSS_1/(N - p_1 - 1)},$$

RSS_1 is for the least squares fit for the large model with $p_1 + 1$ parameters.

RSS_0 is for the least squares fit for the smaller model with $p_0 + 1$ parameters.

Least Squares - An Example:

Prostate Cancer:

Input variables: log cancer volume (lcavol), log prostate weight (lweight), age, log of the amount of benine tumor (lbph), seminal vesicle invasion (svi), log of capsular penetration (lcp), Gleason score (gleason), and percent of gleason scores 4 or 5 (pgg45).

Output: prostate-specific antigen (lpsa).

Least Squares - An Example:

Prostate Cancer:

Input variables: log cancer colume (lcavol), log prostate weight (lweight), age, log of the amount of bening tumor (lbph), seminal vesicle invasion (svi), log of capsular penetration (lcp), Gleason score (gleason), and percent of gleason scores 4 or 5 (pgg45).

Output: prostate-specific antigen (lpsa).

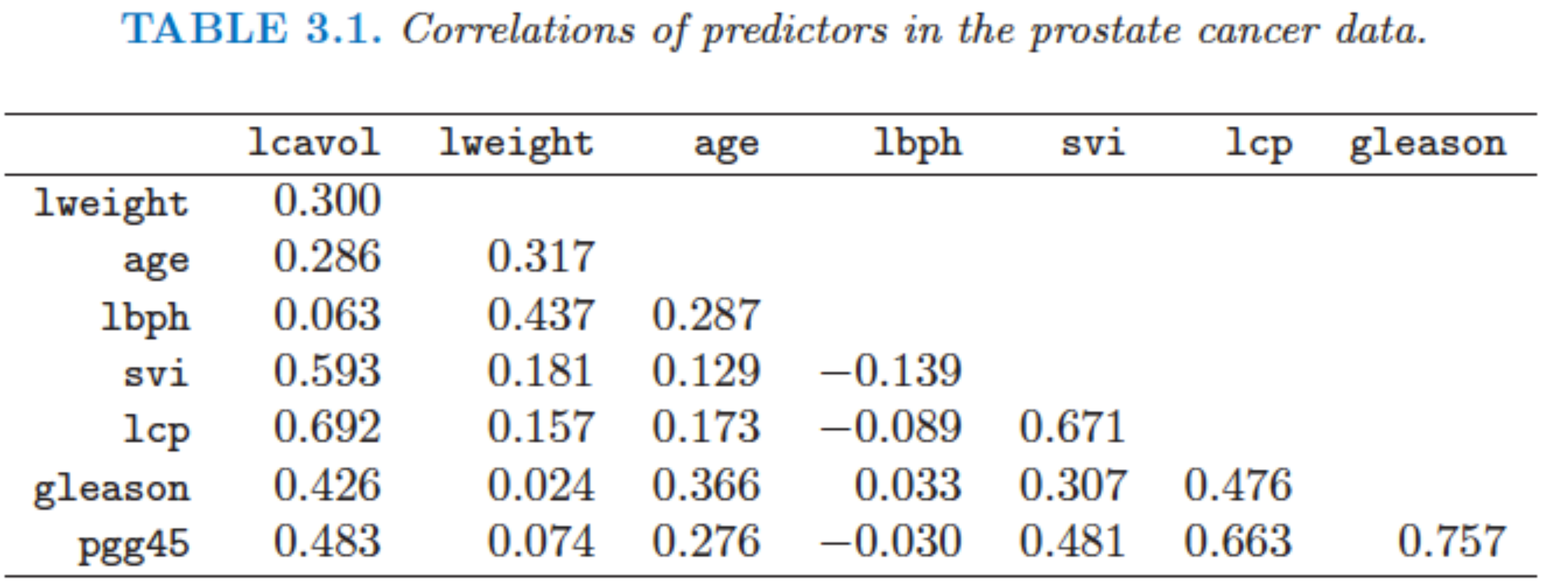


TABLE 3.1. *Correlations of predictors in the prostate cancer data.*

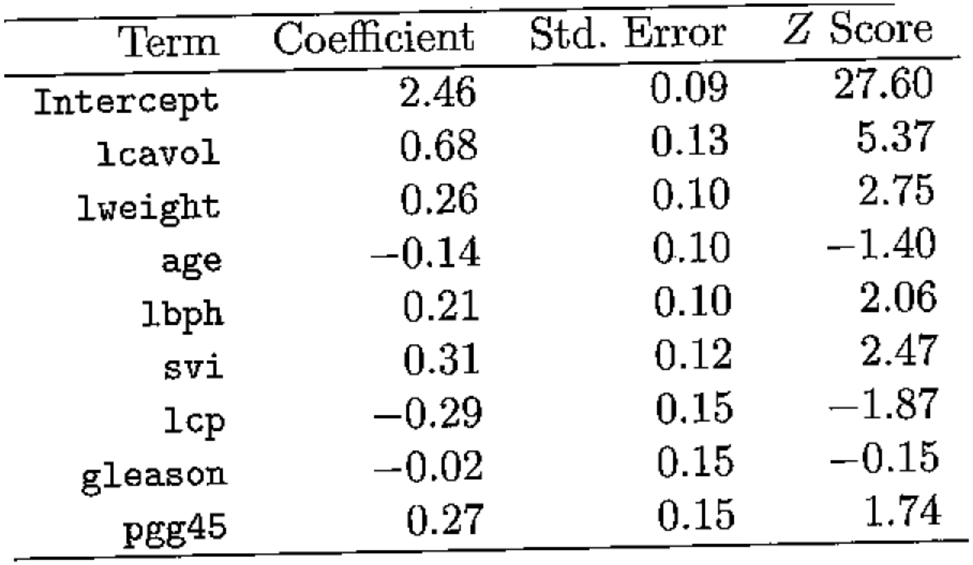
	lcavol	lweight	age	lbph	svi	lcp	gleason
lweight	0.300						
age	0.286	0.317					
lbph	0.063	0.437	0.287				
svi	0.593	0.181	0.129	−0.139			
lcp	0.692	0.157	0.173	−0.089	0.671		
gleason	0.426	0.024	0.366	0.033	0.307	0.476	
pgg45	0.483	0.074	0.276	−0.030	0.481	0.663	0.757

Least Squares - An Example:

Prostate Cancer:

Input variables: log cancer colume (lcavol), log prostate weight (lweight), age, log of the amount of bening tumor (lbph), seminal vesicle invasion (svi), log of capsular penetration (lcp), Gleason score (gleason), and percent of gleason scores 4 or 5 (pgg45).

Output: prostate-specific antigen (lpsa).



Term	Coefficient	Std. Error	Z Score
Intercept	2.46	0.09	27.60
lcavol	0.68	0.13	5.37
lweight	0.26	0.10	2.75
age	−0.14	0.10	−1.40
lbph	0.21	0.10	2.06
svi	0.31	0.12	2.47
lcp	−0.29	0.15	−1.87
gleason	−0.02	0.15	−0.15
pgg45	0.27	0.15	1.74

Can we eliminate some of these parameters?

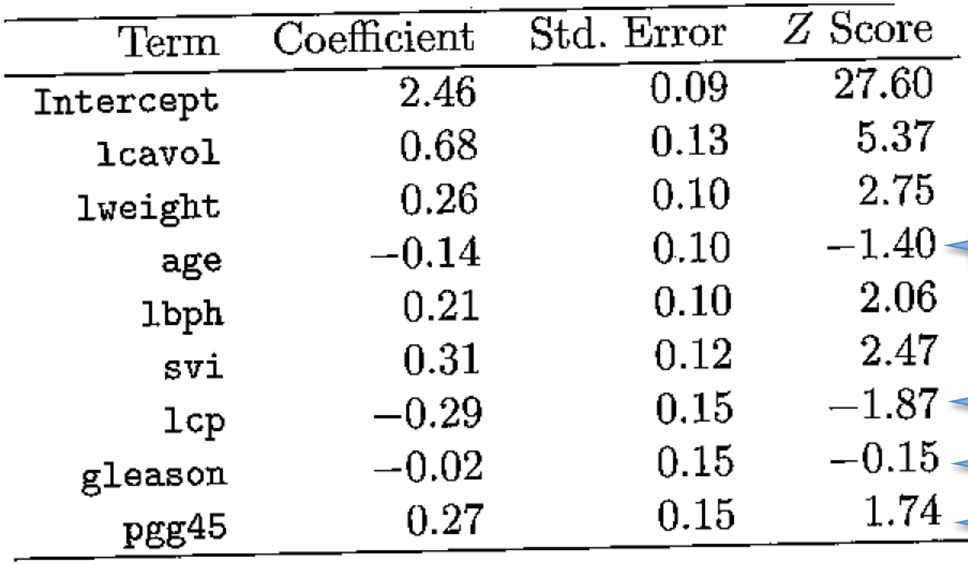
$$z_j = \frac{\hat{\beta}_j}{\hat{\sigma}\sqrt{v_j}}$$

Least Squares - An Example:

Prostate Cancer:

Input variables: log cancer colume (lcavol), log prostate weight (lweight), age, log of the amount of bening tumor (lbph), seminal vesicle invasion (svi), log of capsular penetration (lcp), Gleason score (gleason), and percent of gleason scores 4 or 5 (pgg45).

Output: prostate-specific antigen (lpsa).



Term	Coefficient	Std. Error	Z Score
Intercept	2.46	0.09	27.60
lcavol	0.68	0.13	5.37
lweight	0.26	0.10	2.75
age	−0.14	0.10	−1.40
lbph	0.21	0.10	2.06
svi	0.31	0.12	2.47
lcp	−0.29	0.15	−1.87
gleason	−0.02	0.15	−0.15
pgg45	0.27	0.15	1.74

Can we
eliminate a set of parameters
from the model?

$$F = \frac{(RSS_0 - RSS_1)/(p_1 - p_0)}{RSS_1/(N - p_1 - 1)}$$

The Gauss-Markov Theorem

- Assume that the linear model is true.
- For any linear combination of the parameters, $\beta_0 \dots \beta_p$, denoted by $\theta = a^T \beta$ (for example $\theta = f(x_0) = x_0^T \beta$). $\theta = a^T \hat{\beta}$ is an unbiased estimator.

- The least squares estimate of θ is:

$$\begin{aligned}\hat{\theta} &= a^T \beta \\ &= a^T (X^T X)^{-1} X^T y.\end{aligned}$$

which is linear in y .

The Gauss-Markov Theorem

- Suppose that $c^T y$ is another unbiased linear estimate of θ, i.e., $E(c^T y) = 0$.

- The Gauss-Markov theorem states that the least squares estimate yields the minimum variance among all linear unbiased estimates:

$$Var(a^T y) \le Var(c^T y).$$

The Gauss-Markov Theorem

- The big picture:

$$
\begin{aligned}
MSE(\hat{\theta}) &= E(\hat{\theta}-\theta)^2 \\
&= \underbrace{Var(\hat{\theta})}_{\text{variance}} + \underbrace{\left[E(\hat{\theta})-\theta\right]^2}_{\text{squared bias}}
\end{aligned}
$$

- The Gauss-Markov theorem implies that the OLS estimator has the smallest mean squared error of ALL linear estimators with no bias.
- However, there may exist a biased estimator with smaller mean squared error. Such an estimator would trade a little bias for a larger reduction in variance. e.g., shrinkage methods, subset selection.

Gram-Schmitt Orthogonalization

Some more linear algebra…..

- Inner product: $\langle x, y \rangle = \sum_{i=1}^{N} x_i y_i = x^T y$

- Two vectors are orthogonal if and only if their inner product is zero.

- Suppose we have a **univariate model** (p=1) with no intercept:

$$\hat{\beta} = \frac{\sum_{i=1}^{N} x_i y_i}{\sum_{i=1}^{N} x_i^2} = \frac{\langle x, y \rangle}{\langle x, x \rangle}, \qquad \text{and} \qquad r_i = y_i - x_i \hat{\beta}.$$

- If multiple **orthogonal** inputs, then the the LS estimates are:

$$\boxed{\hat{\beta}_j = \frac{\langle x_j, y \rangle}{\langle x_j, x_j \rangle}.}$$

When inputs are orthogonal… the Predictors have no effect on other Coefficient estimates… only their own.

Gram-Schmitt Orthogonalization

In reality …..

- Orthogonality can be achieved by careful experimental design, but is rare in observational data.
- We can do some tricks to achieve it (assume univariate model):
  1. Regress x on 1 to produce the residual $z = x - \bar{x}1$.
  2. Regress y on z to give the coefficients $\hat{\beta}_1$.

$$\hat{\beta}_1 = \frac{\langle x - \bar{x}1, y \rangle}{\langle x - \bar{x}1, x - \bar{x}1 \rangle}.$$

Note: Orthogonalization **does not change the subspace** spanned by the input vectors, it only finds an orthogonal basis for representing it.

Gram-Schmitt Orthogonalization

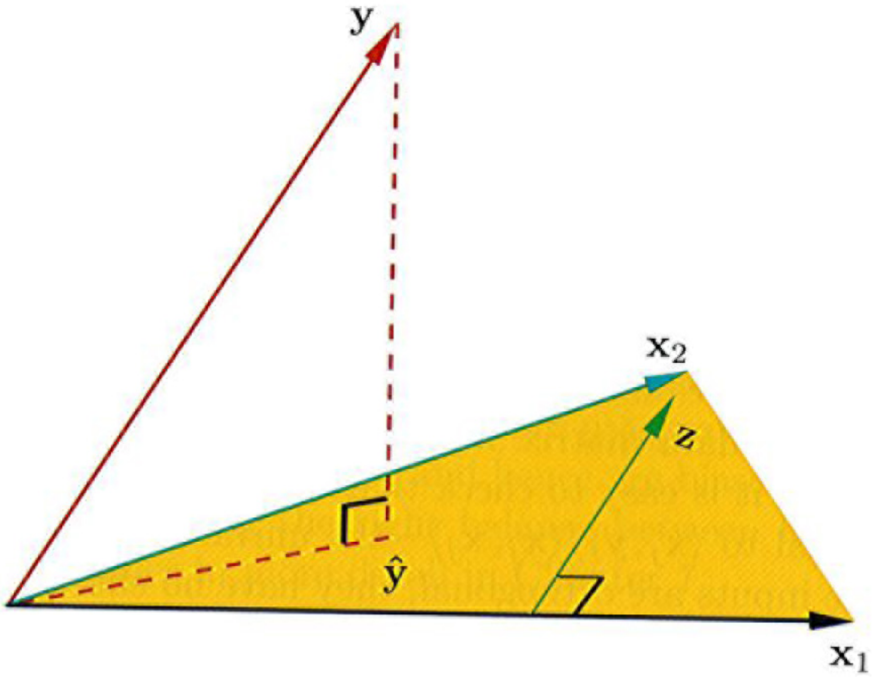
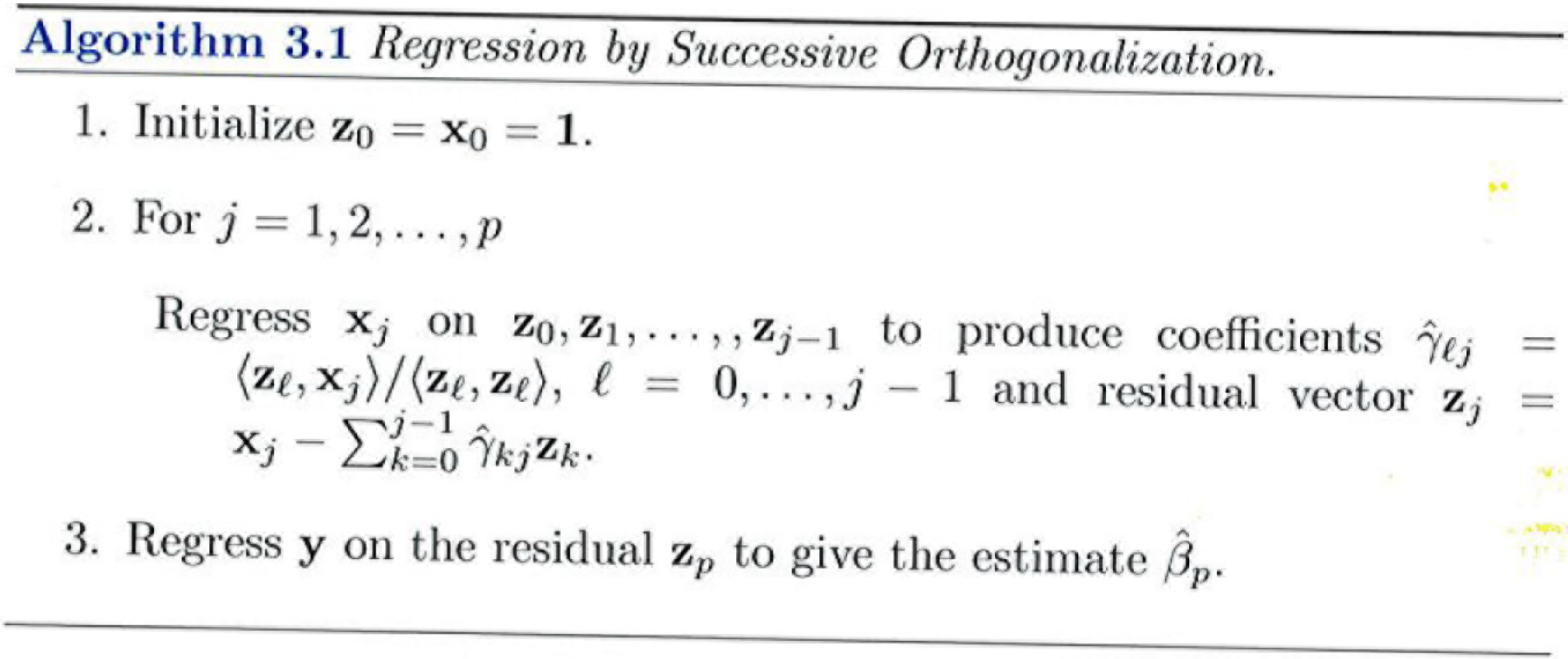


FIGURE 3.4. *Least squares regression by orthogonalization of the inputs. The vector* $\mathbf{x}_2$ *is regressed on the vector* $\mathbf{x}_1$*, leaving the residual vector* $\mathbf{z}$*. The regression of* $\mathbf{y}$ *on* $\mathbf{z}$ *gives the multiple regression coefficient of* $\mathbf{x}_2$*.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

Gram-Schmitt Orthogonalization

The algorithm:

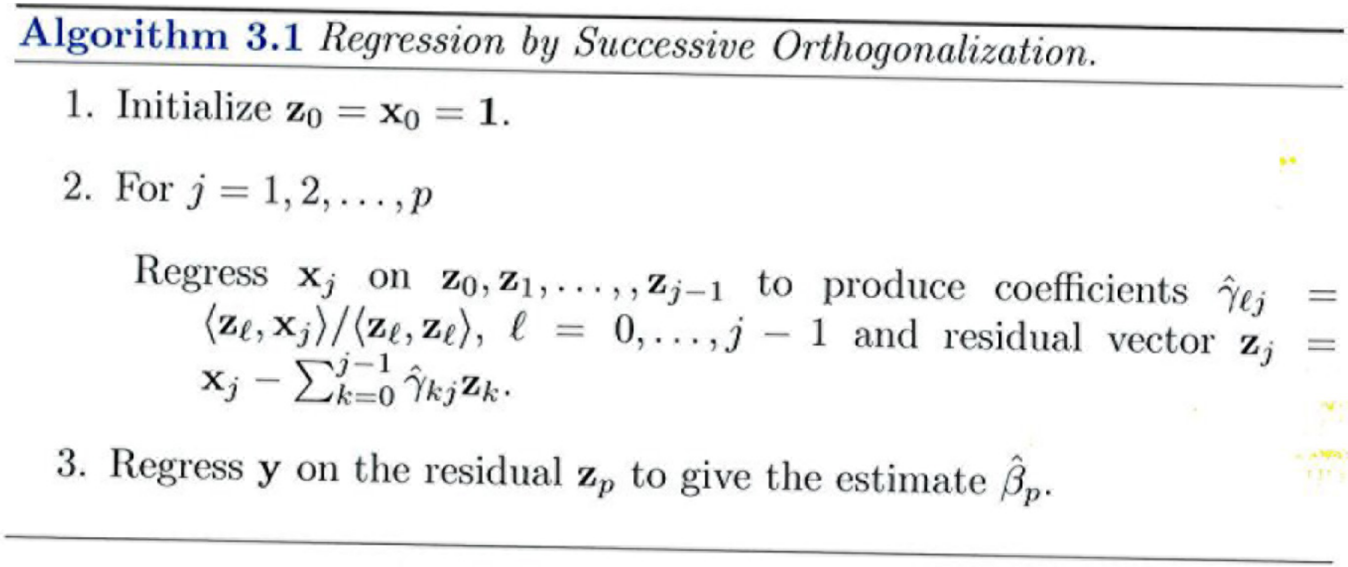


Algorithm 3.1 *Regression by Successive Orthogonalization.*

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

2. For $j = 1, 2, \ldots, p$

     Regress $\mathbf{x}_j$ on $\mathbf{z}_0, \mathbf{z}_1, \ldots, , \mathbf{z}_{j-1}$ to produce coefficients $\hat{\gamma}_{\ell j} = \langle \mathbf{z}_\ell, \mathbf{x}_j \rangle / \langle \mathbf{z}_\ell, \mathbf{z}_\ell \rangle$, $\ell = 0, \ldots, j-1$ and residual vector $\mathbf{z}_j = \mathbf{x}_j - \sum_{k=0}^{j-1} \hat{\gamma}_{kj} \mathbf{z}_k$.

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

Gram-Schmitt Orthogonalization

The algorithm:



Algorithm 3.1 *Regression by Successive Orthogonalization.*

1. Initialize $\mathbf{z}_0 = \mathbf{x}_0 = \mathbf{1}$.
2. For $j = 1, 2, \ldots, p$

   Regress $\mathbf{x}_j$ on $\mathbf{z}_0, \mathbf{z}_1, \ldots, , \mathbf{z}_{j-1}$ to produce coefficients $\hat{\gamma}_{\ell j} = \langle \mathbf{z}_\ell, \mathbf{x}_j \rangle / \langle \mathbf{z}_\ell, \mathbf{z}_\ell \rangle$, $\ell = 0, \ldots, j-1$ and residual vector $\mathbf{z}_j = \mathbf{x}_j - \sum_{k=0}^{j-1} \hat{\gamma}_{kj} \mathbf{z}_k$.
3. Regress $\mathbf{y}$ on the residual $\mathbf{z}_p$ to give the estimate $\hat{\beta}_p$.

Note: If the input vectors are highly correlated this is numerically unstable. Why? Because the residuals z will be close to zero! Can be stabilized in **modified Gram-Schmidt**.

Gram-Schmitt Orthogonalization

- In matrix form we can write step 2 as:

$$X = Z\Gamma,$$

where Z has the columns of z, and Γ is an upper triangular matrix.

- We define D as the diagonal matrix with entry $D_{jj} = \|z_j\|$, we get the QR factorization:

$$\begin{aligned} X &= ZD^{-1}D\Gamma \\ &= QR. \end{aligned}$$

$Q \in \mathbf{R}^{N\times(p+1)}$
Is an Orthogonal matrix

$R \in \mathbf{R}^{(p+1)\times(p+1)}$
Upper triangular

- The least squares solution: $\hat{\beta} = R^{-1}Q^T y$

$$\hat{y} = QQ^T y.$$