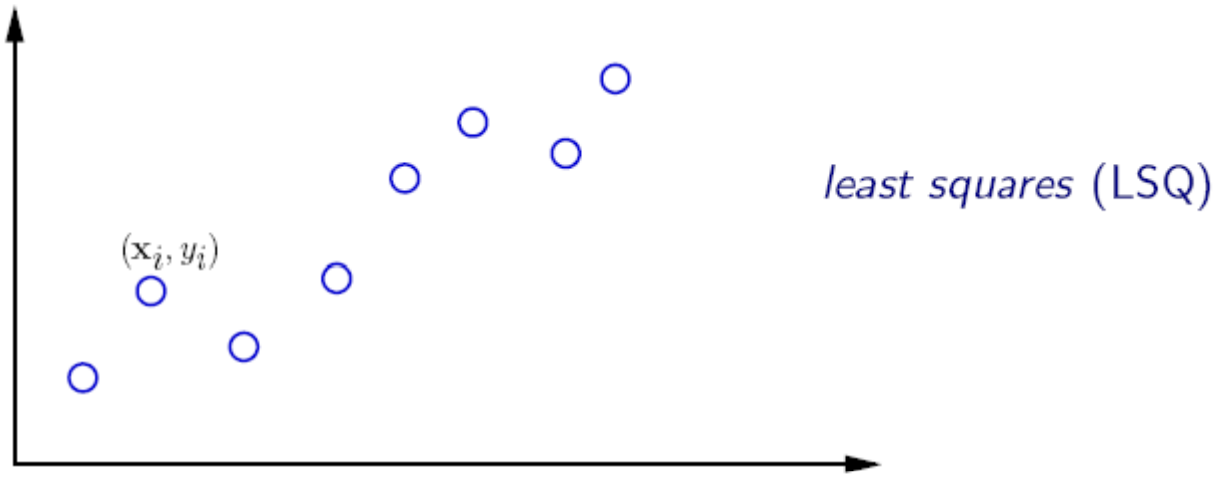
Regression

Linear Regression

Linear fitting to data

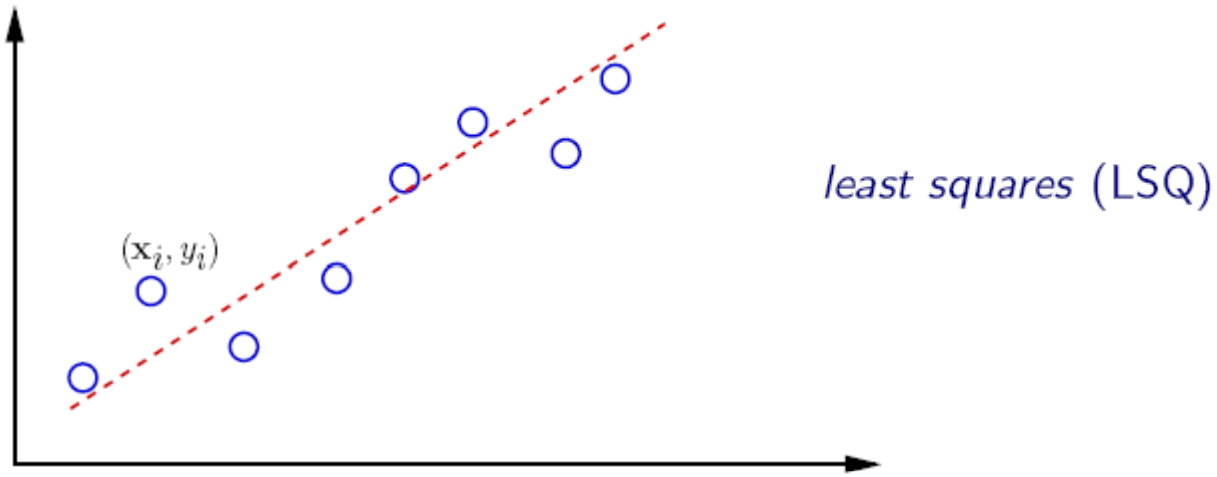
- We want to fit a linear function to an observed set of points $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ with associated labels $Y = [y_1, \ldots, y_N]$.
  - Once we fit the function, we can use it to *predict* the y for new $\mathbf{x}$.
- Find the function that minimizes sum (or average) of square distances between actual ys in the training set and predicted ones.



slide thanks to Greg Shakhnarovich (CS195-5, Brown Univ., 2006)

Linear fitting to data

- We want to fit a linear function to an observed set of points $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ with associated labels $Y = [y_1, \ldots, y_N]$.
  - Once we fit the function, we can use it to *predict* the y for new $\mathbf{x}$.
- Find the function that minimizes sum (or average) of square distances between actual ys in the training set and predicted ones.

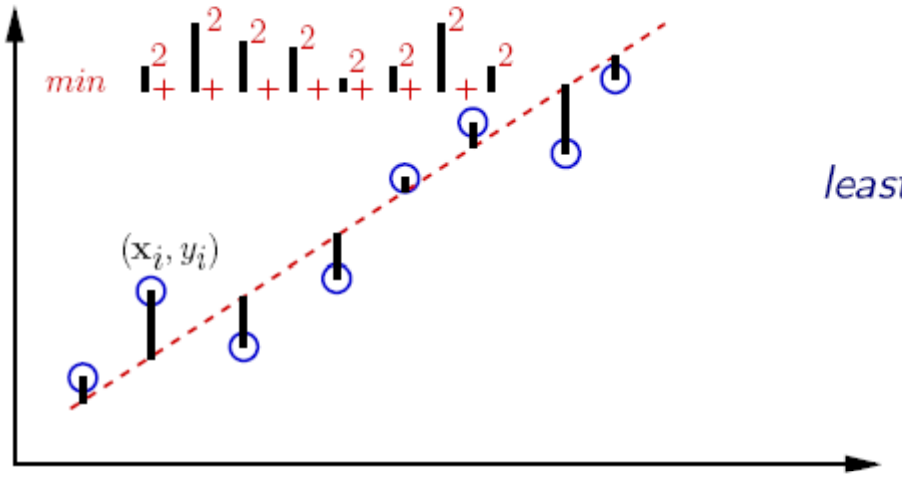


least squares (LSQ)

slide thanks to Greg Shakhnarovich (CS195-5, Brown Univ., 2006)

Linear fitting to data

- We want to fit a linear function to an observed set of points $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ with associated labels $Y = [y_1, \ldots, y_N]$.
  - Once we fit the function, we can use it to *predict* the y for new $\mathbf{x}$.
- Find the function that minimizes sum (or average) of square distances between actual ys in the training set and predicted ones.

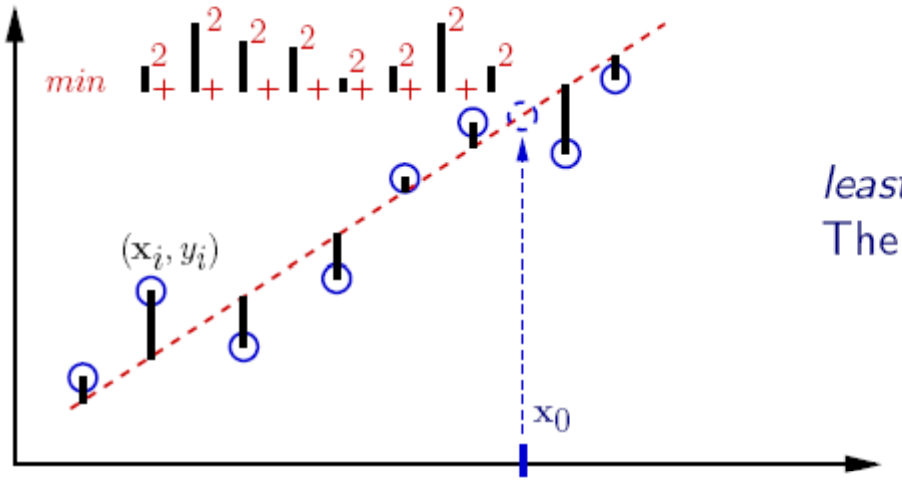


least squares (LSQ)

slide thanks to Greg Shakhnarovich (CS195-5, Brown Univ., 2006)

Linear fitting to data

- We want to fit a linear function to an observed set of points $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ with associated labels $Y = [y_1, \ldots, y_N]$.
  - Once we fit the function, we can use it to *predict* the y for new $\mathbf{x}$.
- Find the function that minimizes sum (or average) of square distances between actual ys in the training set and predicted ones.

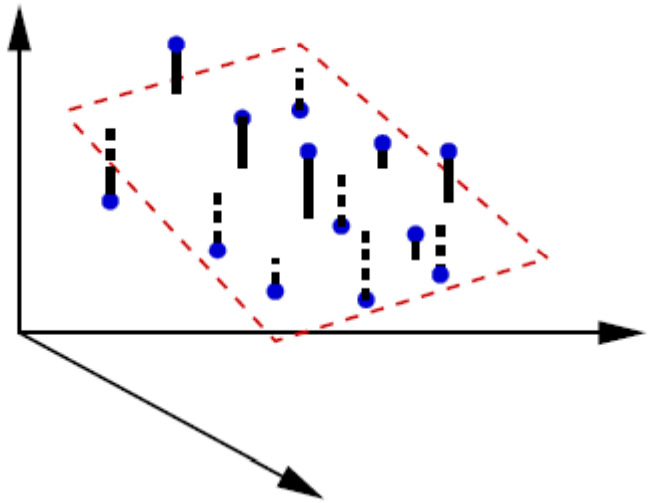


least squares (LSQ)
The fitted line is used as a predictor

slide thanks to Greg Shakhnarovich (CS195-5, Brown Univ., 2006)

Linear functions

- General form: $f(\mathbf{x}; \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_d x_d$
- 1D case ($\mathcal{X} = \mathbb{R}$): a line
- $\mathcal{X} = \mathbb{R}^2$: a plane



- *Hyperplane* in general, d-D case.

slide thanks to Greg Shakhnarovich (CS195-5, Brown Univ., 2006)

Loss function

- Suppose target labels come from set Y
  - Binary classification: $Y = \{ 0, 1 \}$
  - Regression: $Y = \Re$ (real numbers)
- A loss function maps decisions to costs:
  - $L(y, \hat{y})$ defines the penalty for predicting $\hat{y}$ when the true value is y.
- Standard choice for classification:
  0/1 loss (same as misclassification error)

$$L_{0/1}(y, \hat{y}) = \begin{cases} 0 & \text{if } y = \hat{y} \\ 1 & \text{otherwise} \end{cases}$$

- Standard choice for regression:
  squared loss

$$L(y, \hat{y}) = (\hat{y} - y)^2$$

Least squares linear fit to data

- Most popular estimation method is least squares:
  - Determine linear coefficients α, β that minimize sum of squared loss (SSL).
  - Use standard (multivariate) differential calculus:
    - differentiate SSL with respect to α, β
    - find zeros of each partial differential equation
    - solve for α, β
- One dimension:

$$\text{SSL} = \sum_{j=1}^{N} (y_j - (\alpha + \beta \cdot x_j))^2 \qquad N = \text{number of samples}$$

$$\beta = \frac{\text{cov}[x, y]}{\text{var}[x]} \qquad \alpha = \bar{y} - \beta \cdot \bar{x} \qquad x,y = \text{means of training } x, y$$

$$\hat{y}_t = \alpha + \beta \cdot x_t \qquad \text{for test sample } x_t$$

Least squares linear fit to data

- Multiple dimensions
  - To simplify notation and derivation, change α to β_0, and add a new feature $x_0 = 1$ to feature vector $\mathbf{x}$:

$$\hat{y} = \beta_0 \cdot 1 + \sum_{i=1}^{d} \beta_i \cdot x_i = \boldsymbol{\beta} \cdot \mathbf{x}$$

  - Calculate SSL and determine $\boldsymbol{\beta}$:

$$\text{SSL} = \sum_{j=1}^{N} (y_j - \sum_{i=0}^{d} \beta_i \cdot x_i)^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}} \cdot (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

$$\mathbf{y} = \text{vector of all training responses } y_j$$

$$\mathbf{X} = \text{matrix of all training samples } \mathbf{x}_j$$

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

$$\hat{y}_t = \boldsymbol{\beta} \cdot \mathbf{x}_t \qquad \text{for test sample } \mathbf{x}_t$$

Least squares linear fit to data

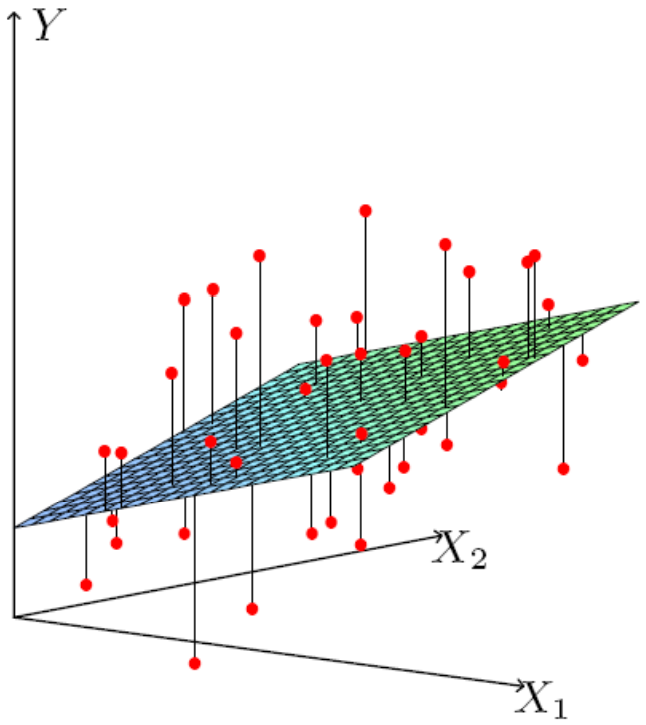


FIGURE 3.1. *Linear least squares fitting with* $X \in \mathbb{R}^2$. *We seek the linear function of* X *that minimizes the sum of squared residuals from* Y.

Least squares linear fit to data

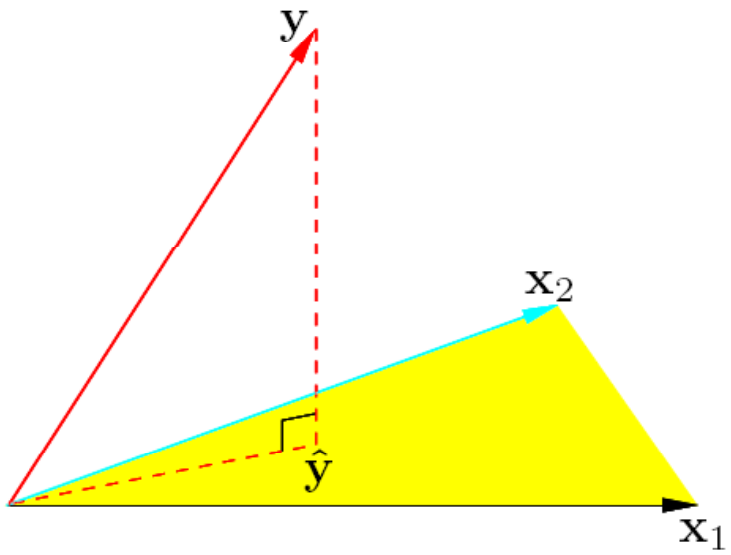
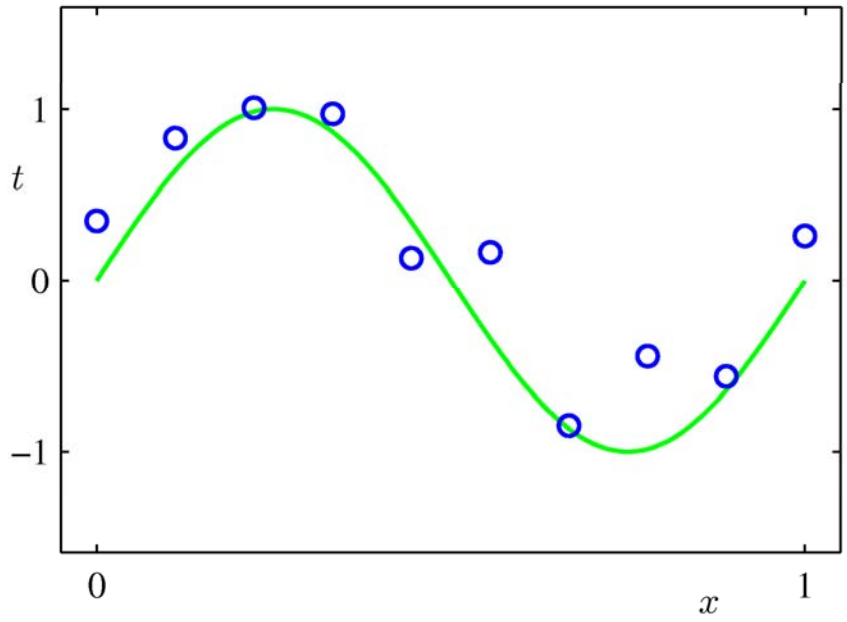


FIGURE 3.2. *The N-dimensional geometry of least squares regression with two predictors. The outcome vector $\mathbf{y}$ is orthogonally projected onto the hyperplane spanned by the input vectors $\mathbf{x}_1$ and $\mathbf{x}_2$. The projection $\hat{\mathbf{y}}$ represents the vector of the least squares predictions*

Extending application of linear regression

- The inputs **X** for linear regression can be:
  - Original quantitative inputs
  - Transformation of quantitative inputs, e.g. log, exp, square root, square, etc.
  - Polynomial transformation
    - example: $y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot x^2 + \beta_3 \cdot x^3$
  - Basis expansions
  - Dummy coding of categorical inputs
  - Interactions between variables
    - example: $x_3 = x_1 \cdot x_2$
- This allows use of linear regression techniques to fit much more complicated non-linear datasets.

Example of fitting polynomial curve with linear model



$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Prostate cancer dataset

- 97 samples, partitioned into:
  - 67 training samples
  - 30 test samples
- Eight predictors (features):
  - 6 continuous (4 log transforms)
  - 1 binary
  - 1 ordinal
- Continuous outcome variable:
  - `lpsa`: log(prostate specific antigen level)

Correlations of predictors in prostate cancer dataset

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

Fit of linear model to prostate cancer dataset

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.*

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

Regularization

- Complex models (lots of parameters) often prone to overfitting.
- Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters.
- Two common types of regularization in linear regression:
  - L_2 regularization (a.k.a. ridge regression). Find β which minimizes:

$$\sum_{j=1}^{N}(y_j - \sum_{i=0}^{d}\beta_i \cdot x_i)^2 + \lambda \sum_{i=1}^{d}\beta_i^2$$

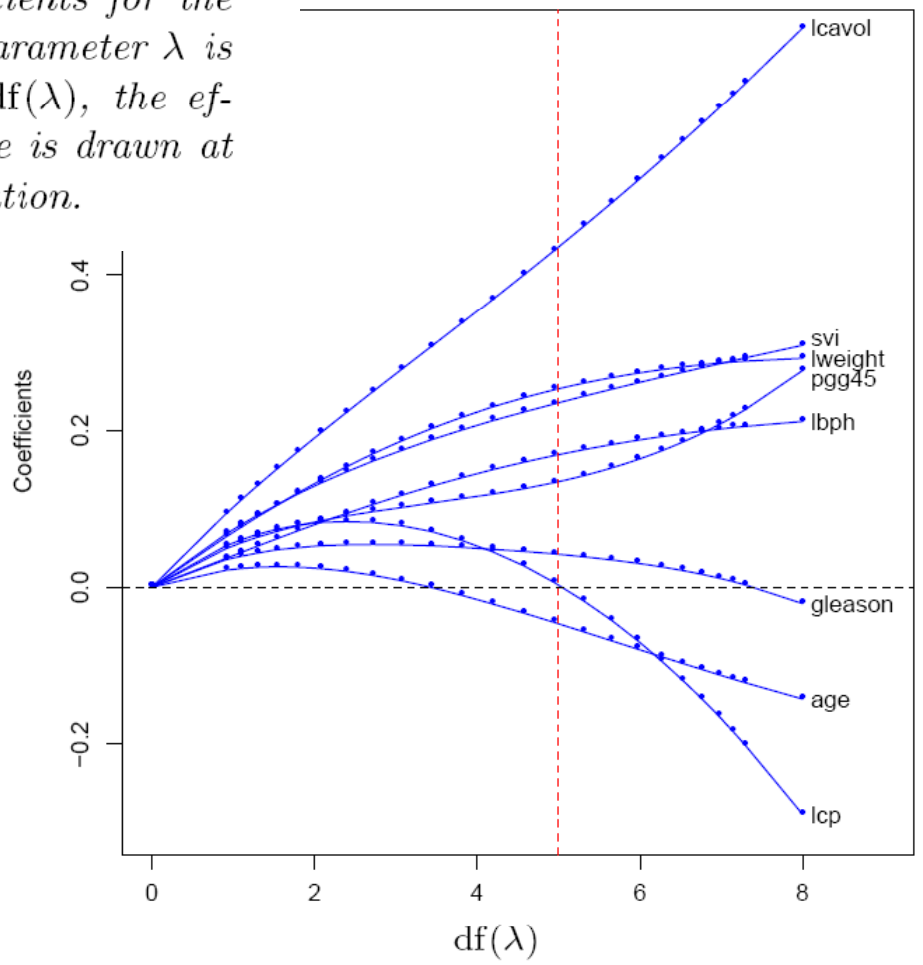
    - λ is the regularization parameter: bigger λ imposes more constraint

  - L_1 regularization (a.k.a. lasso). Find β which minimizes:

$$\sum_{j=1}^{N}(y_j - \sum_{i=0}^{d}\beta_i \cdot x_i)^2 + \lambda \sum_{i=1}^{d}|\beta_i|$$

Example of L_2 regularization

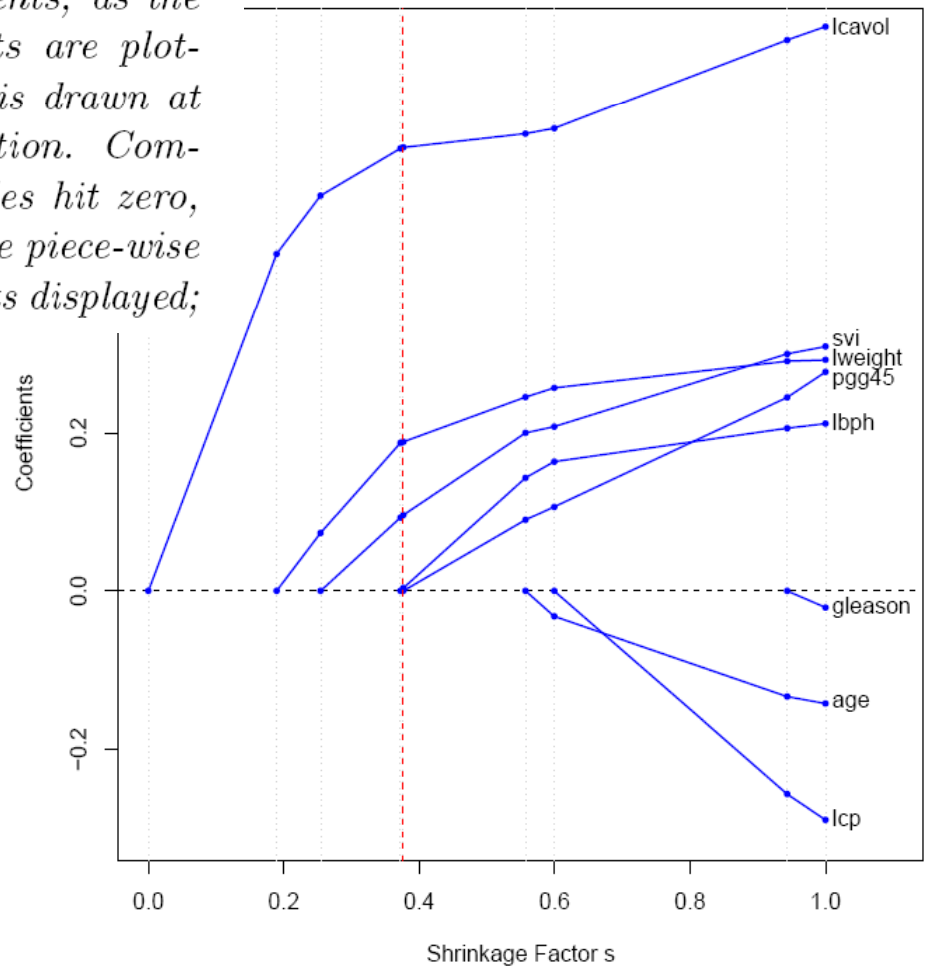
FIGURE 3.8. *Profiles of ridge coefficients for the prostate cancer example, as the tuning parameter λ is varied. Coefficients are plotted versus $\mathrm{df}(\lambda)$, the effective degrees of freedom. A vertical line is drawn at $\mathrm{df} = 5.0$, the value chosen by cross-validation.*



L2 regularization shrinks coefficients towards (but not to) zero, and towards each other.

Example of L_1 regularization

FIGURE 3.10. *Profiles of lasso coefficients, as the tuning parameter t is varied. Coefficients are plotted versus $s = t/\sum_1^p |\hat{\beta}_j|$. A vertical line is drawn at $s = 0.36$, the value chosen by cross-validation. Compare Figure 3.8 on page 9; 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;*



L1 regularization shrinks coefficients to zero at different rates; different values of λ give models with different subsets of features.

Example of subset selection

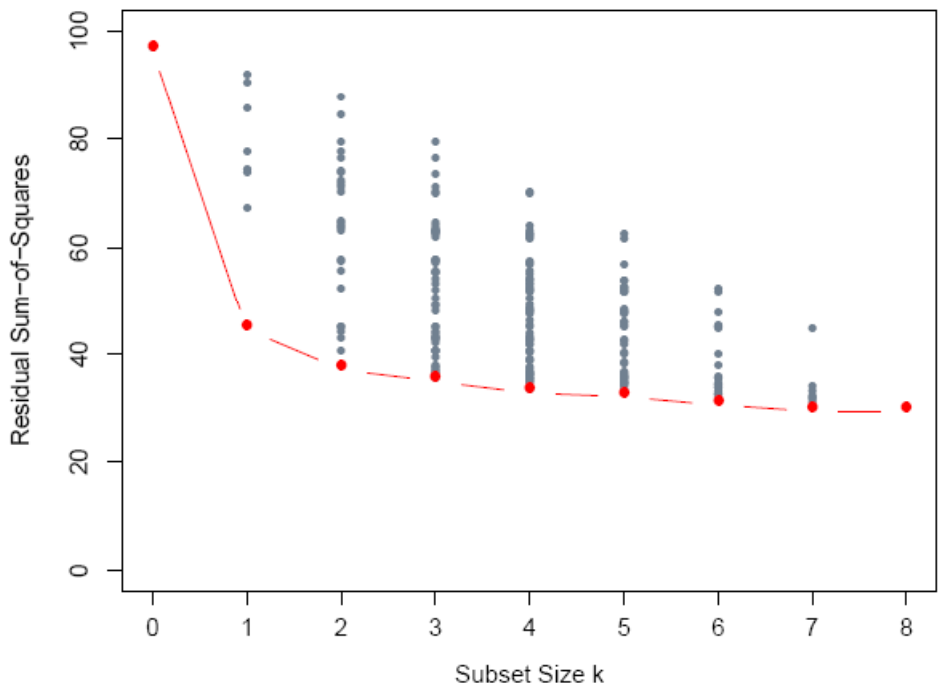


FIGURE 3.5. *All possible subset models for the prostate cancer example. At each subset size is shown the residual sum-of-squares for each model of that size.*

Comparison of various selection and shrinkage methods

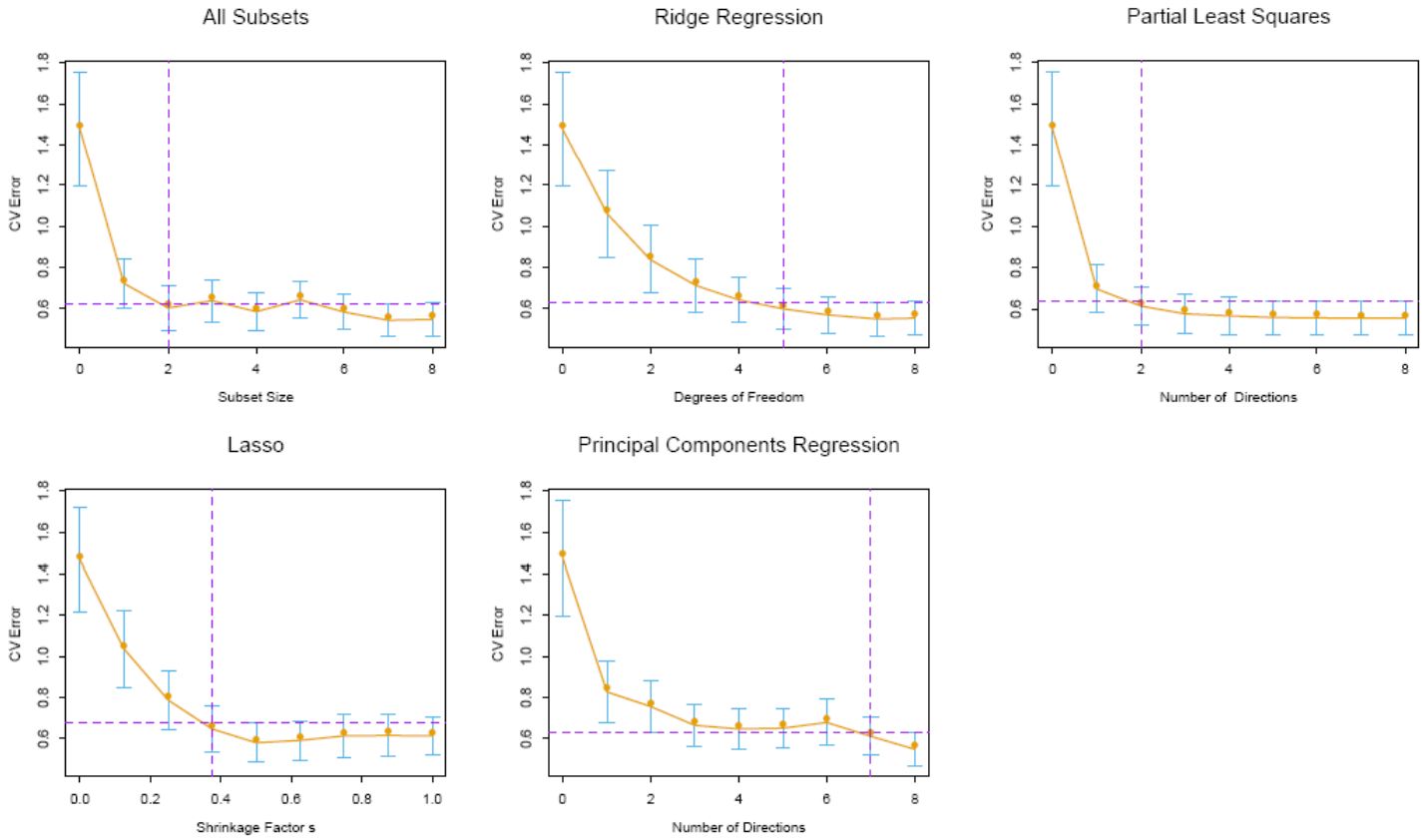


FIGURE 3.7. *Estimated prediction error curves and their standard errors for the various selection and shrinkage methods. Each curve is plotted as a func-*

L_1 regularization gives sparse models, L_2 does not

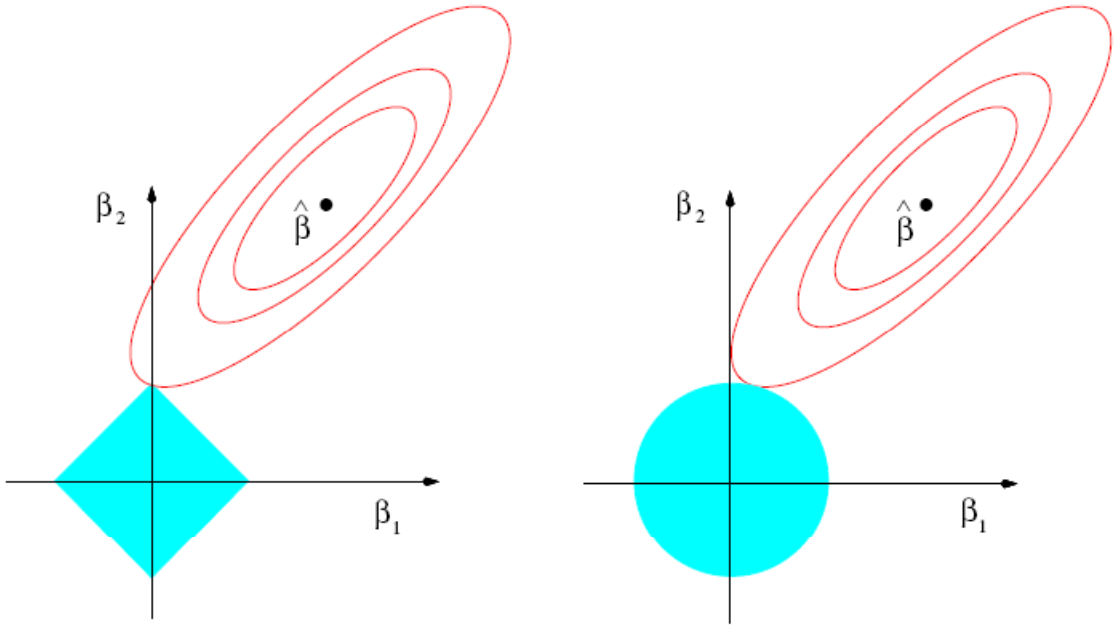


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$, *respectively, while the red ellipses are the contours of the least squares error function.*

Other types of regression

- In addition to linear regression, there are:
  - many types of non-linear regression
    - decision trees
    - nearest neighbor
    - neural networks
    - support vector machines
  - locally linear regression
  - etc.

MATLAB interlude

matlab_demo_07.m

Part B