

10-315 INTRODUCTION TO MACHINE LEARNING (SCS MAJORS)

LECTURE 14: LINEAR REGRESSION

LEILA WEHBE
CARNEGIE MELLON UNIVERSITY
MACHINE LEARNING DEPARTMENT

Reading: [Elements of Statistical Learning Chapters 3.1, 3.2, 3.4](https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf)
(https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf).

LECTURE OUTCOMES

- Linear regression definition
- OLS solution and interpretation
- Ridge solution and tradeoff
- Lasso (just the penalty and what it optimizes for)

LINKS (USE THE VERSION YOU NEED)

- [Notebook \(https://github.com/lwehbe/10315_S21/blob/main/Lecture_14_Linear_Regression.ipynb\)](https://github.com/lwehbe/10315_S21/blob/main/Lecture_14_Linear_Regression.ipynb)
- [PDF slides \(https://github.com/lwehbe/10315_S21/raw/main/Lecture_14_Linear_Regression.slides.pdf\)](https://github.com/lwehbe/10315_S21/raw/main/Lecture_14_Linear_Regression.slides.pdf)

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm, multivariate_normal

# Sample data:
X = np.linspace(0,36,36)
coefs = [2,0.5]
Y = coefs[0] + coefs[1]*X + 2*norm.rvs(size = np.shape(X))
Y_seasonal = coefs[0] + coefs[1]*X + 4*np.sin(X*np.pi/6) + 0.5*norm.rvs(size = np.shape(X))
```

REGRESSION VS. CLASSIFICATION

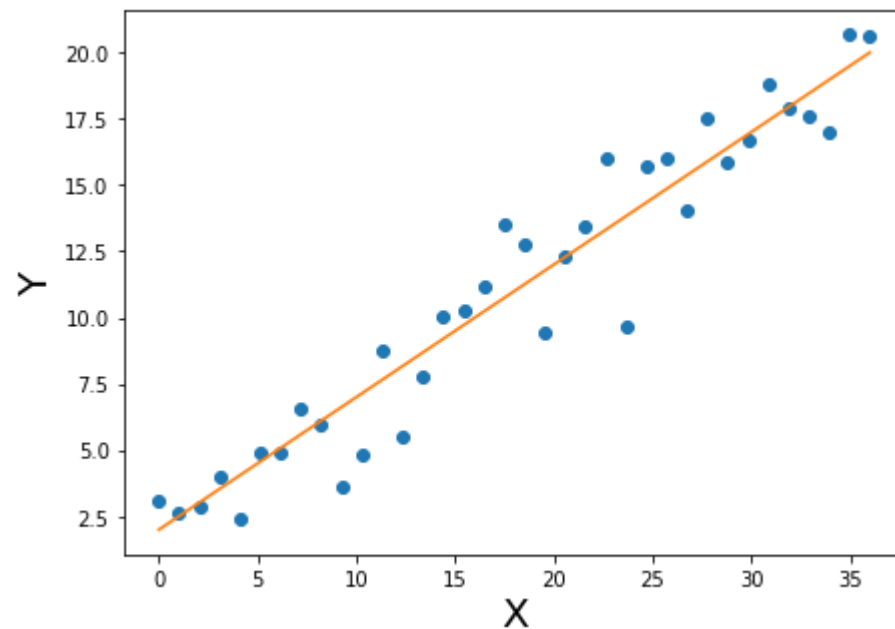
- So far, we've been interested in learning $P(Y|X)$ where Y has discrete values ('classification')
- What if Y is continuous? ('regression')
 - predict weight from gender, height, age, ...
 - predict Google stock price today from Google, Yahoo, MSFT prices yesterday
 - predict each pixel intensity in robot's next camera image, from current image and current action

LINEAR REGRESSION

- We wish to learn a linear function $f : X \rightarrow Y$ where $Y \in \mathbb{R}$ given $\{(X_1, Y_1), \dots, (X_n, Y_n)\}$ with $X_i \in \mathbb{R}^p$.
- Let's start with 1-dimensional X example (p=1):
 - We want to find the line that best "fits" the data
 - How do we define this best fit?

```
In [2]: plt.figure(figsize=(7,5))  
plt.plot(X,Y, 'o');plt.xlabel('X', fontsize=20);plt.ylabel('Y', fontsize=20);  
plt.plot(X,coefs[0]+coefs[1]*X)
```

```
Out[2]: [<matplotlib.lines.Line2D at 0x10ae7c4e0>]
```

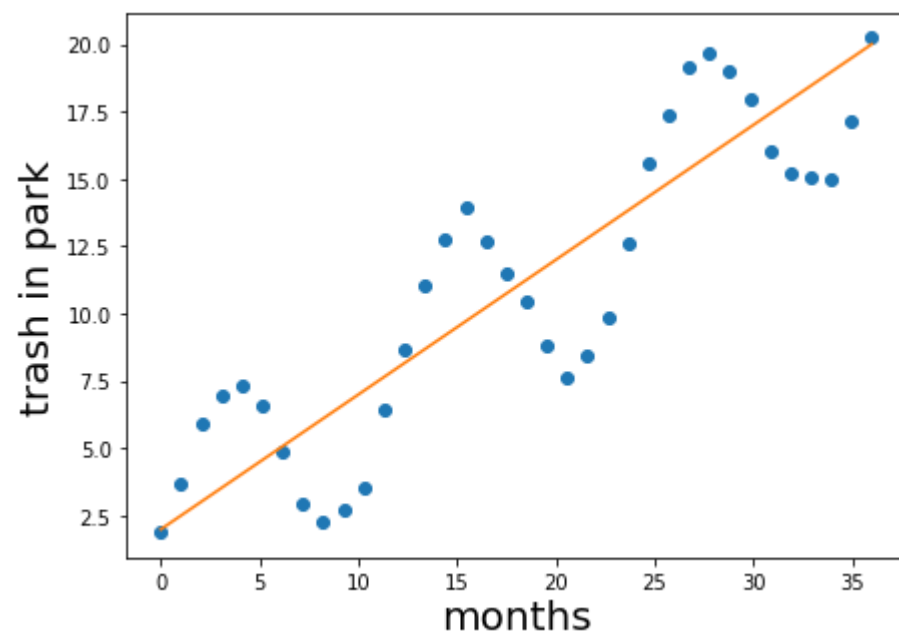


LINEAR REGRESSION

- Consider the example below where there exist a non-linear relationship that is a very good predictor of the data
 - namely there is a seasonal effect that varies with the months of the year, as well as a linear increase with time
 - linear regression is only able to capture linear relationships

```
In [3]: plt.figure(figsize=(7,5))  
plt.plot(X,Y_seasonal,'o');plt.xlabel('months', fontsize=20);plt.ylabel('trash in park',fontsize=20);  
plt.plot(X,coefs[0]+coefs[1]*X)
```

```
Out[3]: [<matplotlib.lines.Line2D at 0x145aa1f28>]
```

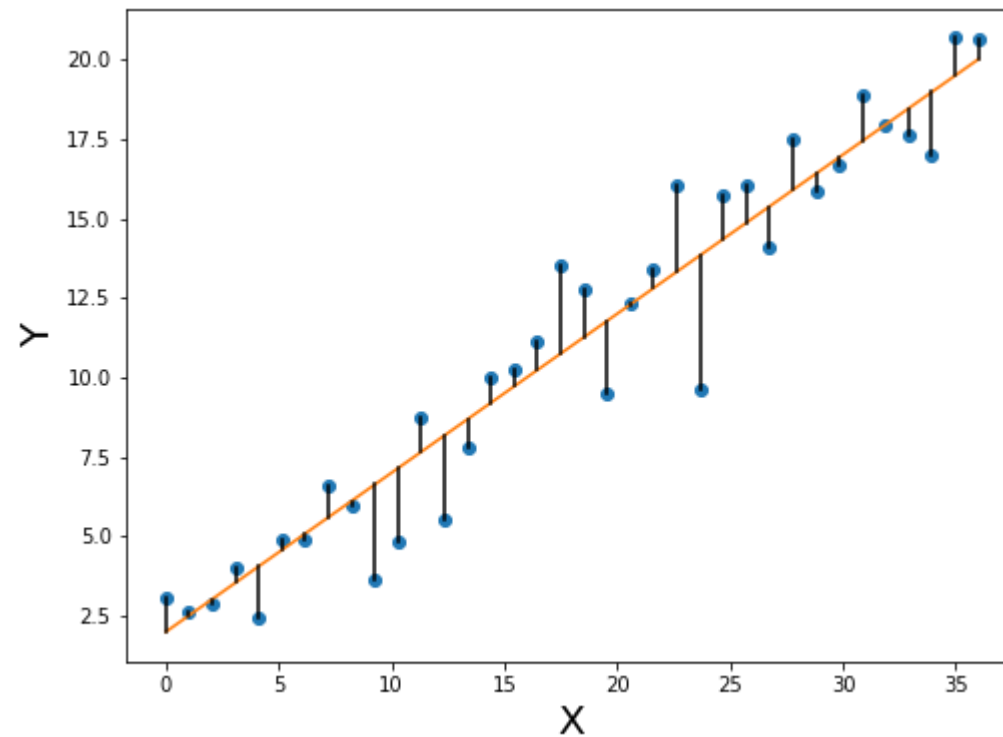


- even if the underlying relationship between X and Y is not linear, one can still use linear regression
 - the assumption is not satisfied, but we have seen previously that in some cases, a model can still perform well even if its assumptions are not specified

HOW TO DEFINE GOODNESS OF FIT?

- We define the goodness of fit based on the prediction error:
 - $\epsilon_i = y_i - \hat{y}_i = y_i - (w_0 + w_1 x_i)$
 - vertical error in the plot below

```
In [4]: plt.figure(figsize=(8,6))
plt.plot(X,Y,'o');plt.xlabel('X', fontsize=20);plt.ylabel('Y',fontsize=20);
plt.plot(X,coefs[0]+coefs[1]*X)
for i,Xi in enumerate(X):
    plt.plot( [Xi,Xi], [coefs[0]+coefs[1]*Xi, Y[i]], 'k')
```



X IN 2-D:

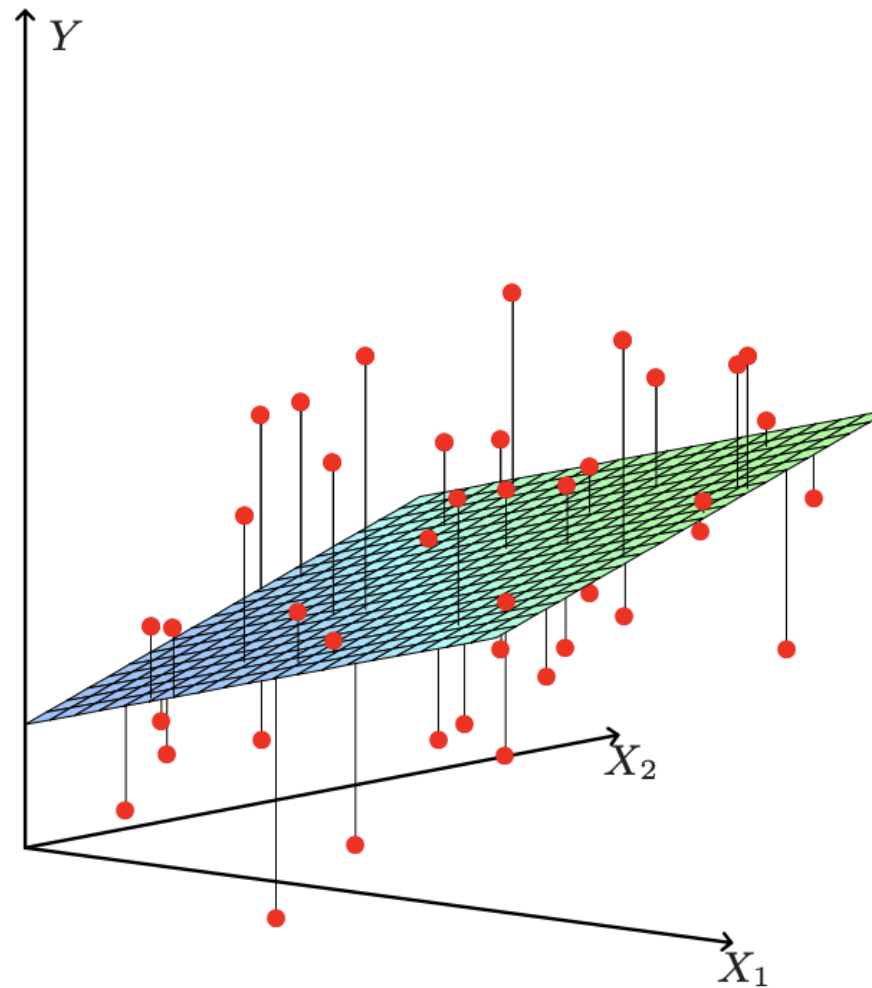


FIGURE 3.1. *Linear least squares fitting with $X \in \mathbb{R}^2$. We seek the linear function of X that minimizes the sum of squared residuals from Y .*

Source: Figure 3.1 from [ESL \(https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf\)](https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf).

APPROACH 1: MINIMIZING THE RESIDUAL SUM OF SQUARES

- Ordinary Least Squares (OLS) minimizes the Residual Sum of Squares (RSS):

$$\text{RSS}(\mathbf{w}) = \sum_{i=1}^n \left(y_i - (w_0 + \sum_j w_j x_{i,j}) \right)^2$$

- This corresponds to the sum of the square of the errors in predicting each y_i .
- If we change our notation so that now \mathbf{x}_i has an additional entry x_0 always corresponding to 1:

$$\text{RSS}(\mathbf{w}) = \sum_{i=1}^n (y_i - \mathbf{x}_i^\top \mathbf{w})^2$$

- Note that the Mean Squared Error (MSE) is also often used in regression problems and corresponds to RSS/n . It should be clear that here minimizing MSE or RSS yields the same solution

HOW TO MINIMIZE RSS?

$$\hat{\mathbf{w}}_{\text{OLS}} = \underset{\mathbf{w}}{\operatorname{argmin}} \operatorname{RSS}(\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^n (y_i - \mathbf{x}_i^\top \mathbf{w})^2$$

- Let's write RSS in matrix notation:

$$\operatorname{RSS}(\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w})$$

- Where:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \dots \\ \mathbf{x}_n^\top \end{bmatrix} = \begin{bmatrix} x_{1,0} & x_{1,1} & \dots & x_{1,p} \\ x_{2,0} & x_{2,1} & \dots & x_{2,p} \\ \dots & \dots & \dots & \dots \\ x_{n,0} & x_{n,1} & \dots & x_{n,p} \end{bmatrix} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix}$$

RSS(W) IS CONVEX IN W

$$\frac{d\text{RSS}(\mathbf{w})}{d\mathbf{w}} = \frac{d(\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w})}{d\mathbf{w}}$$

Poll: what is the size of $\frac{d\text{RSS}(\mathbf{w})}{d\mathbf{w}}$?

- RSS is a scalar
- \mathbf{w} is $(p \times 1)$

$$\frac{d\text{RSS}(\mathbf{w})}{d\mathbf{w}} = -2\mathbf{X}^\top (\mathbf{y} - \mathbf{X}\mathbf{w}) = -2(\mathbf{X}^\top \mathbf{y} - \mathbf{X}^\top \mathbf{X}\mathbf{w})$$

$$\left. \frac{d\text{RSS}(\mathbf{w})}{d\mathbf{w}} \right|_{\hat{\mathbf{w}}_{\text{OLS}}} = 0$$

- if $\mathbf{X}^\top \mathbf{X}$ is invertible*:

$$\hat{\mathbf{w}}_{\text{OLS}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

- Predict for new point \mathbf{x}_{new} : $\hat{y}_{\text{new}} = \mathbf{x}_{\text{new}}^\top \hat{\mathbf{w}}_{\text{OLS}}$

*For a review: [Zico Kolter's Linear algebra notes \(http://www.cs.cmu.edu/~zkolter/course/linalg/linalg_notes.pdf\)](http://www.cs.cmu.edu/~zkolter/course/linalg/linalg_notes.pdf).

ALTERNATIVE GEOMETRIC INTERPRETATION

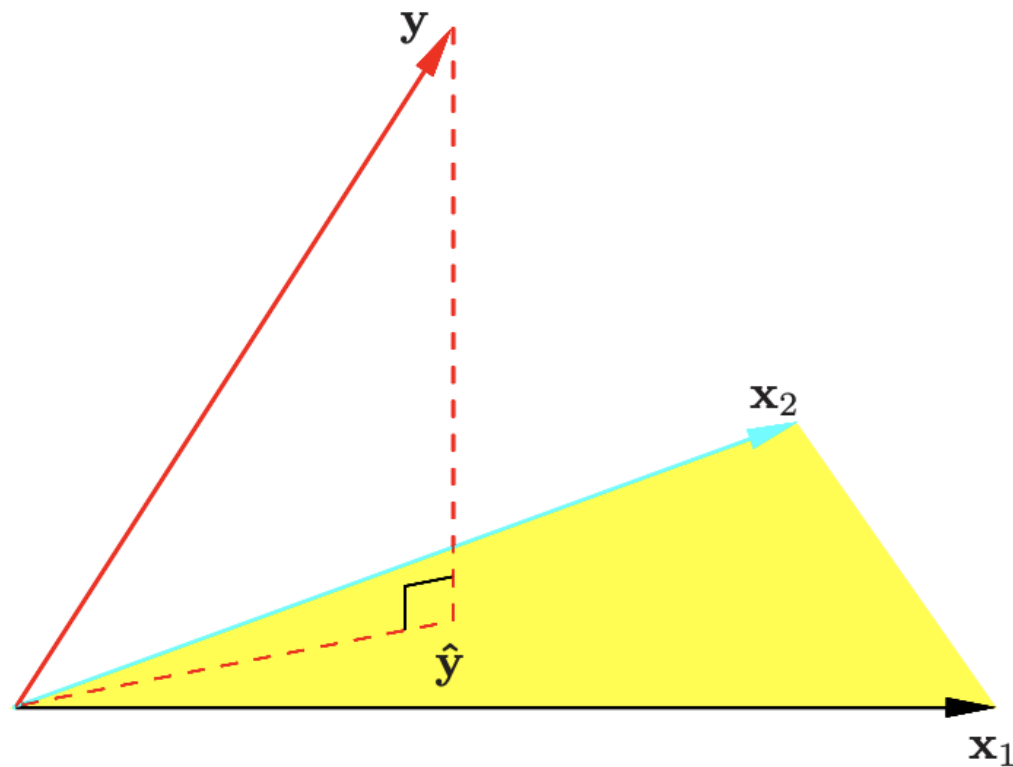


FIGURE 3.2. *The N -dimensional geometry of least squares regression with two predictors. The outcome vector \mathbf{y} is orthogonally projected onto the hyperplane spanned by the input vectors \mathbf{x}_1 and \mathbf{x}_2 . The projection $\hat{\mathbf{y}}$ represents the vector of the least squares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

Source: Figure 3.2 from [ESL \(https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf\)](https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf).

- In this representation, \mathbf{y} represents the real values for all points, and \mathbf{x}_1 and \mathbf{x}_2 are **columns** of \mathbf{X} .
- $\hat{\mathbf{y}}$ is the vectors of all predictions that lie in the space spanned by \mathbf{x}_1 and \mathbf{x}_2 .
- $\hat{\mathbf{y}}$ is the orthogonal projection of \mathbf{y} onto that space and $\mathbf{y} - \hat{\mathbf{y}}$ is the error.

ALTERNATIVE GEOMETRIC INTERPRETATION - MORE FORMALLY:

- Assume we have n tuples (\mathbf{x}_i, y_i) where $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}, \mathbf{y} \in \mathbb{R}^n$.
- \mathbf{X} is $n \times p$.
- The p columns of \mathbf{X} span a subset of \mathbb{R}^n
 - Recall from linear algebra this subset is called the column space of \mathbf{X}
- The vector of predictions for all points $\hat{\mathbf{y}}$ is the orthogonal projection of \mathbf{y} onto the linear subspace spanned by the columns of \mathbf{X} .
 - Recall this is due to our optimization procedure, in which we set:
$$\mathbf{X}^\top (\mathbf{y} - \mathbf{X}\mathbf{w}) = 0$$

WHAT HAPPENS IF $\mathbf{X}^\top \mathbf{X}$ NOT INVERTIBLE?

- Suppose \mathbf{X} is not full rank, i.e. its columns are not linearly independent
 - e.g. two of the input dimensions are perfectly correlated
 - e.g. one of the input dimensions is a perfect linear combination of the others
 - or e.g. $p > n$
- Then, $\mathbf{X}^\top \mathbf{X}$ is singular and we cannot invert it.
 - there is not a unique solution $\hat{\mathbf{w}}_{\text{OLS}}$
- Solutions to the problem: remove redundancy from \mathbf{X} , regularize (will discuss in a moment), add diagonal component (akin to specific type of regularization, to see later)...

WHAT IF $\mathbf{X}^\top \mathbf{X}$ IS INVERTIBLE BUT TOO LARGE?

- Inverting $\mathbf{X}^\top \mathbf{X}$ might still be very slow!
- Can do gradient descent:

- initialize \mathbf{w}^0

- update:

$$\mathbf{w}^{t+1} = \mathbf{w}^t - 2\eta \mathbf{X}^\top (\mathbf{X} \mathbf{w}^t - \mathbf{y})$$

- The error $\mathbf{X} \mathbf{w}^t - \mathbf{y}$ reduces as \mathbf{w}^{t+1} gets close to $\hat{\mathbf{w}}_{\text{OLS}}$
- convergence depends on learning rate (too small ==> slow, too big ==> possible oscillation and larger error if can't get close enough to $\hat{\mathbf{w}}_{\text{OLS}}$. Can use adaptive learning rate.

PROBABILISTIC INTERPRETATION: MLE

- We state the problem as:

$$Y_i = \mathbf{x}_i^\top \mathbf{w} + \epsilon_i$$

$$\epsilon_i \sim \mathcal{N}(0, \sigma)$$

$$Y_i \sim \mathcal{N}(\mathbf{x}_i^\top \mathbf{w}, \sigma)$$

- Maximizing the log-likelihood of the data simplifies to:

$$\begin{aligned}\hat{\mathbf{w}}_{\text{MLE}} &= \underset{\mathbf{w}}{\operatorname{argmax}} \log \left[\prod_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(y_i - \mathbf{x}_i^\top \mathbf{w})^2}{2\sigma^2} \right] \\ &= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_i (y_i - \mathbf{x}_i^\top \mathbf{w})^2\end{aligned}$$

This is the least squares problem! ==> same solution

APPROACH 2 - RIDGE REGRESSION, ADDING L2 REGULARIZATION

- Ridge regression minimizes the Residual Sum of Squares (RSS) with an additional penalty on the L_2 norm of \mathbf{w} :

$$\hat{\mathbf{w}}_{\text{Ridge}} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^n (y_i - \mathbf{x}_i^\top \mathbf{w})^2 + \lambda \sum_j w_j^2$$

- where $\lambda \geq 0$ is a penalty parameter
- Note: in practice, we often don't penalize the intercept term. Instead we first estimate the intercept as \bar{y} and remove it from \mathbf{y} , and run ridge regression with no intercept. Then we set the intercept as \bar{y} .

- In matrix notation:
$$\hat{\mathbf{w}}_{\text{Ridge}} = \underset{\mathbf{w}}{\operatorname{argmin}} (\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda \mathbf{w}^\top \mathbf{w}$$

- Solving

$$\begin{aligned} \frac{d\text{RSS}(\mathbf{w}) + \lambda \mathbf{w}^\top \mathbf{w}}{d\mathbf{w}} &= -2\mathbf{X}^\top (\mathbf{y} - \mathbf{X}\mathbf{w}) + 2\lambda \mathbf{w} = 0 \\ (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_p) \mathbf{w} &= \mathbf{X}^\top \mathbf{y} \end{aligned}$$

- We get:

$$\hat{\mathbf{w}}_{\text{Ridge}} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_p)^{-1} \mathbf{X}^\top \mathbf{y}$$

PROBABILISTIC INTERPRETATION

- We state the problem as:

$$Y_i = \mathbf{x}_i^\top \mathbf{W} + \epsilon_i$$

$$\epsilon_i \sim \mathcal{N}(0, \sigma)$$

$$Y_i \mid \mathbf{W} \sim \mathcal{N}(\mathbf{x}_i^\top \mathbf{W}, \sigma)$$

$$W_j \sim \mathcal{N}(0, \gamma)$$

- Maximizing the log-posterior probability of \mathbf{W} :

$$\hat{\mathbf{w}}_{\text{MAP}} = \underset{\mathbf{w}}{\operatorname{argmax}} \log P(\mathbf{W})P(Y \mid \mathbf{W})$$

$$= \underset{\mathbf{w}}{\operatorname{argmax}} \log \left(\left[\prod_j \frac{1}{\sqrt{2\pi\gamma^2}} \exp \frac{-w_j^2}{2\gamma^2} \right] \left[\prod_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(y_i - \mathbf{x}_i^\top \mathbf{w})^2}{2\sigma^2} \right] \right)$$

- Exercise: show that this results in the same problem as ridge regression
 - Ridge regression is equivalent to enforcing a zero mean gaussian prior on the individual weights.

WHAT IS THE EFFECT OF λ ? WHICH λ TO CHOOSE?

$$\begin{aligned}\hat{\mathbf{w}}_{\text{Ridge}} &= \underset{\mathbf{w}}{\operatorname{argmin}} (\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda \mathbf{w}^\top \mathbf{w} \\ &= (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_p)^{-1} \mathbf{X}^\top \mathbf{y}\end{aligned}$$

- think of λ as a shrinkage parameter varying how much the weights are allowed to be close to the OLS solution.
 - when $\lambda \rightarrow 0$, $\hat{\mathbf{w}}_{\text{Ridge}} \rightarrow \hat{\mathbf{w}}_{\text{OLS}}$
 - when $\lambda \rightarrow \infty$, $\hat{\mathbf{w}}_{\text{Ridge}} \rightarrow \mathbf{0}_p$ (vector of 0s)

Let's look at a specific problem with two input features x_1 and x_2 .

```
In [5]: w1x = np.linspace(-3,3,100)
w2x = np.linspace(-3,3,100)
W1,W2 = np.meshgrid(w1x, w2x)

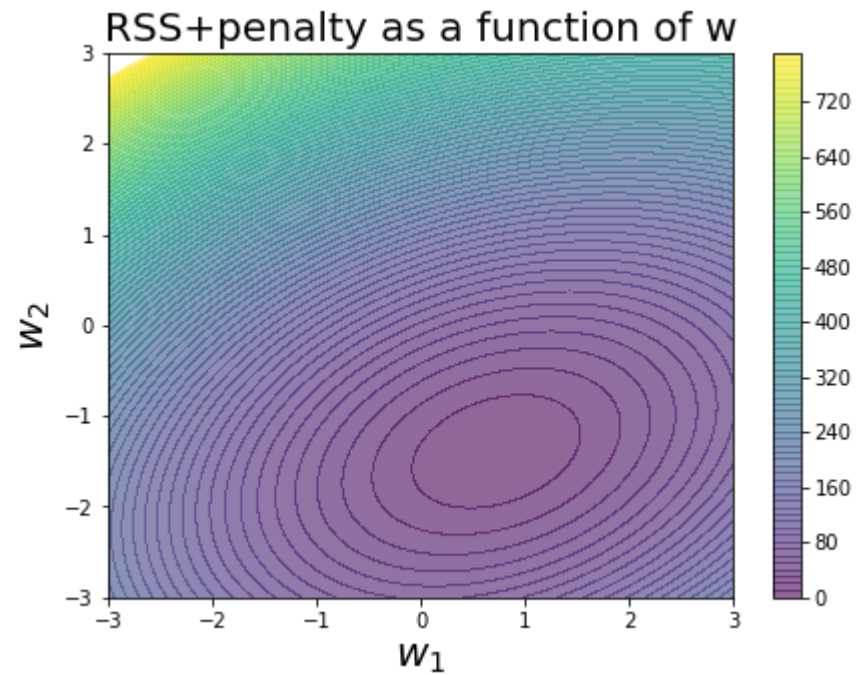
X = multivariate_normal.rvs(mean=np.array([0,0]),cov=1,size=20)
real_w = np.array([[0.8],[-1.5]])
Y = X.dot(real_w) + 0.4*norm.rvs(size=(20,1))

def rss(w1,w2):
    w = np.array([[w1],[w2]])
    loss = np.sum( (Y - X.dot(w)) **2)
    return loss
```

```
In [6]: plt.figure(figsize=(7,5))

lambda = 0.1
L_w = np.vectorize(rss)(*np.meshgrid(w1x, w2x)) +lambda*(W1**2+W2**2)

cs = plt.contourf(W1, W2, L_w,levels=np.arange(0,800,10),alpha=0.6);
plt.colorbar()
plt.xlabel(r'$w_1$',fontsize=20)
plt.ylabel(r'$w_2$',fontsize=20)
plt.title('RSS+penalty as a function of w',fontsize=20);
```



ALTERNATIVE FORMULATION OF OPTIMIZATION PROBLEM

$$\hat{\mathbf{w}}_{\text{Ridge}} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_i (y_i - \mathbf{x}_i^\top \mathbf{w})^2 + \lambda \sum_j w_j^2$$

- Can also be written as

$$\begin{aligned} \hat{\mathbf{w}}_{\text{Ridge}} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_i (y_i - \mathbf{x}_i^\top \mathbf{w})^2 \\ \text{subject to } \sum_j w_j^2 \leq t \end{aligned}$$

- where, for each problem, there is a one-to-one correspondence between specific values of λ and t . We can use this formulation to better understand the effect of the constraint on the value of the parameters that is chosen

```

In [7]: plt.figure(figsize=(5,5))

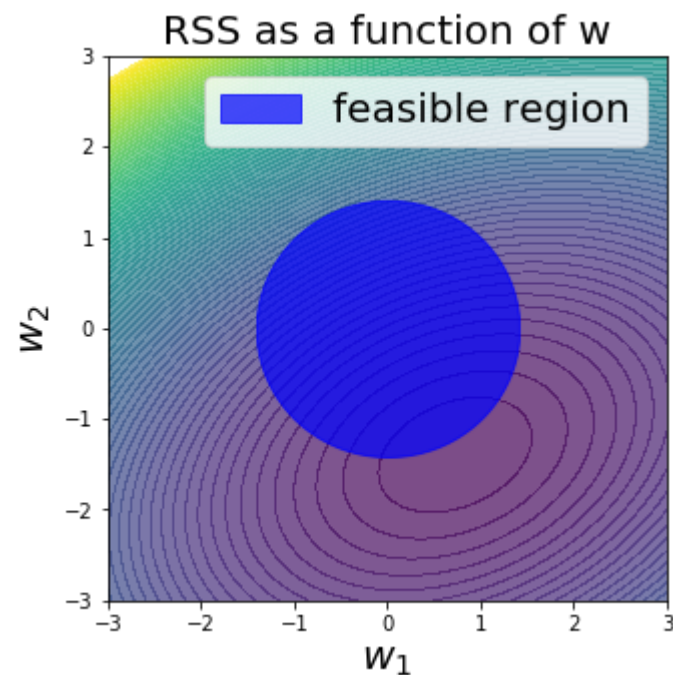
L_w = np.vectorize(rss)(*np.meshgrid(w1x, w2x))

cs = plt.contourf(W1, W2, L_w, levels=np.arange(0,800,10), alpha=0.7, aspect='equal');
plt.xlabel(r'$w_1$', fontsize=20)
plt.ylabel(r'$w_2$', fontsize=20)
plt.title('RSS as a function of w', fontsize=20);

t = 2
w1x_plot = np.linspace(-3,3,1000)
w1x_plot = w1x_plot[w1x_plot**2<=t]
w2_plot = np.nan_to_num(np.sqrt(t - w1x_plot**2))
plt.fill_between(w1x_plot, w2_plot, -w2_plot, color='b', alpha=0.7, label='feasible region');
plt.legend(fontsize=20);

```

/Users/lwehbe/env/py3/lib/python3.7/site-packages/matplotlib/contour.py:1000: UserWarning: The following kwargs were not used by contour: 'aspect'
s)



```

In [8]: plt.figure(figsize=(6,5))
        from numpy.linalg import inv

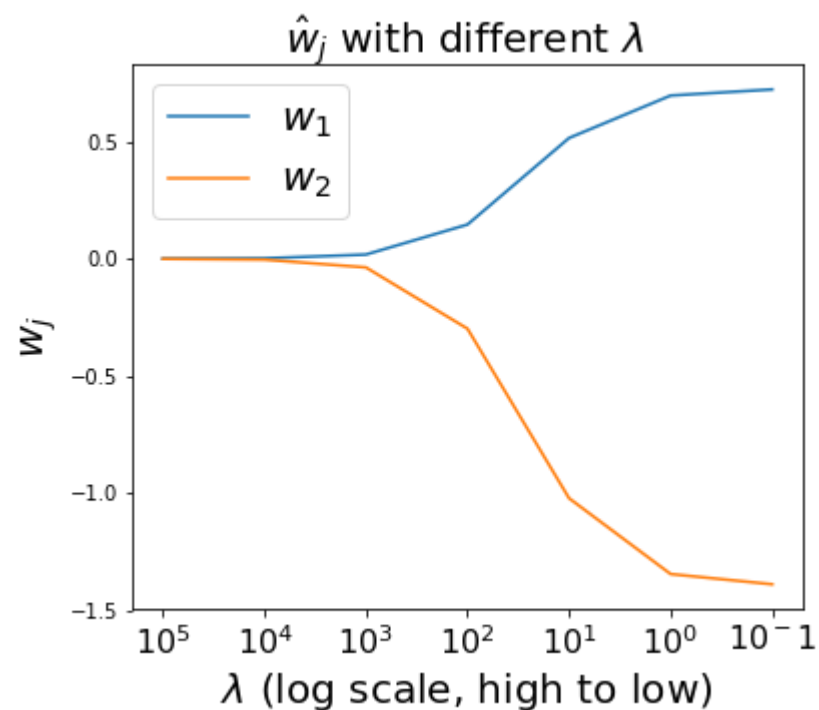
        def ridge(X,Y,lmbda):
            p = X.shape[1]
            return inv(X.T.dot(X) + lmbda*np.eye(p)).dot(X.T.dot(Y))

        lmbdas = np.array([0.1,1,10,100,1000,10000,100000])
        w_lambda = np.hstack([ridge(X,Y,L) for L in lmbdas])
        plt.plot(np.arange(len(lmbdas)),w_lambda[0][::-1],label=r'$w_1$')
        plt.plot(np.arange(len(lmbdas)),w_lambda[1][::-1],label=r'$w_2$')

        xlabels = [r'$10^{\{ \}}$'.format(int(np.log10(L))) for L in lmbdas]
        plt.xticks(np.arange(len(lmbdas)), xlabels[::-1],fontSize=16 )

        plt.xlabel(r'$\lambda$ (log scale, high to low)',fontSize=20)
        plt.ylabel(r'$w_j$',fontSize=20)
        plt.title('$\hat{w}_j$ with different $\lambda$',fontSize=20)
        plt.legend(fontsize=20);

```



EFFECT OF λ

- when $\lambda \rightarrow 0$: $\hat{\mathbf{w}}_{\text{Ridge}} \rightarrow \hat{\mathbf{w}}_{\text{OLS}}$
- when $\lambda \rightarrow \infty$: $\hat{\mathbf{w}}_{\text{Ridge}} \rightarrow \mathbf{0}_p$ (vector of 0s)

BIAS-VARIANCE TRADE-OFF

- Given a joint distribution $P(X, Y)$, let $\mathbf{w}^* \in \mathbb{R}^p$ the parameters of the best linear approximation of Y given X .
 - We attempt to estimate \mathbf{w}^* using a finite sample from $P(X, Y)$.
- How good is our estimate $\hat{\mathbf{w}}$?
 - bias: if we could repeat the experiment multiple times (and thus calculate $\hat{\mathbf{w}}$ multiple times), would the average $\hat{\mathbf{w}}$ be close to \mathbf{w}^* ?
 - variance: if we could repeat the experiment multiple times, how much would the $\hat{\mathbf{w}}$ s agree? would they be very different?

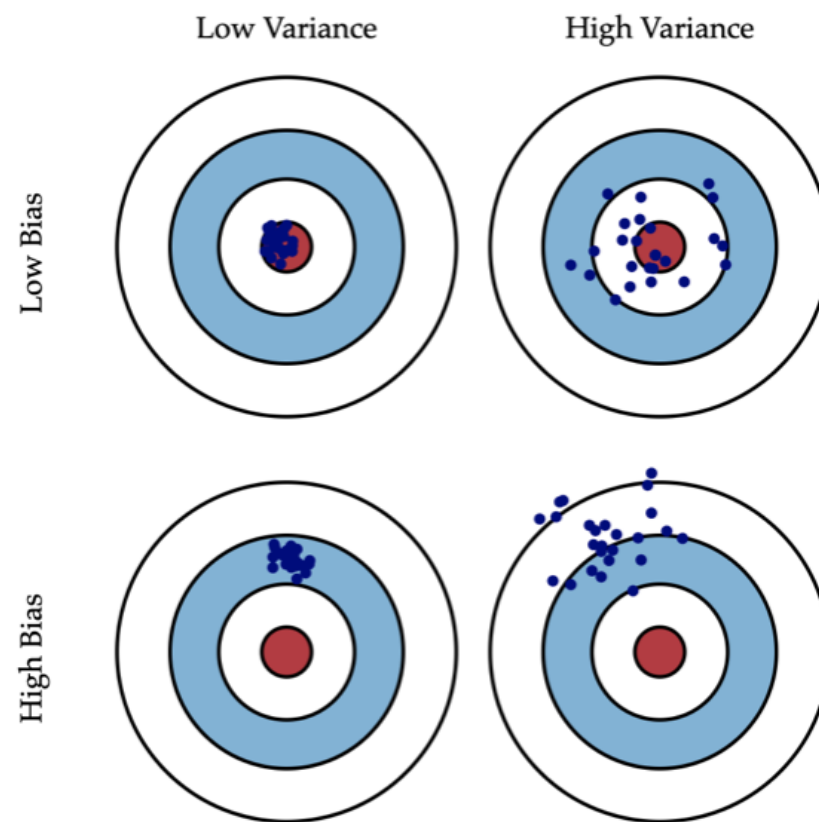


Fig. 1 Graphical illustration of bias and variance.

Source: [Understanding the Bias Variance Tradeoff by Scott Fortmann-Roe \(http://scott.fortmann-roe.com/docs/BiasVariance.html\)](http://scott.fortmann-roe.com/docs/BiasVariance.html),

EFFECT OF λ - TRADEOFF BETWEEN BIAS AND VARIANCE

- when $\lambda \rightarrow 0$: high variance, bias $\rightarrow 0$ (OLS solution is unbiased)
- when $\lambda \rightarrow \infty$: high bias, variance $\rightarrow 0$ (since converging to the zero vector)

APPROACH 3 - LASSO, ADDING L1 REGULARIZATION

- The minimizes the Residual Sum of Squares (RSS) with an additional penalty on the L_1 norm of \mathbf{w} :

$$\hat{\mathbf{w}}_{\text{Lasso}} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^n (y_i - \mathbf{x}_i^\top \mathbf{w})^2 + \lambda \sum_j |w_j|$$

- where $\lambda \geq 0$ is a penalty parameter
- Alternative formulation of optimization problem

$$\begin{aligned} \hat{\mathbf{w}}_{\text{Lasso}} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_i (y_i - \mathbf{x}_i^\top \mathbf{w})^2 \\ \text{subject to } \sum_j |w_j| \leq t \end{aligned}$$

- where, for each problem, there is a one-to-one correspondance between specific values of λ and t .
- The Lasso is also equivalent to imposing a Laplace prior on the parameters $w_j \sim \exp \frac{-|w_j|}{b}$.

LASSO OPTIMIZATION PROBLEM

- The Lasso optimization problem does not have a closed form solution, quadratic optimization problem.
 - More in 10-725
- The Lasso problem encourages sparsity! With high penalty (high λ or low t), few parameters will be non-zero
 - Think of it as taking a bet that only a few parameters are important

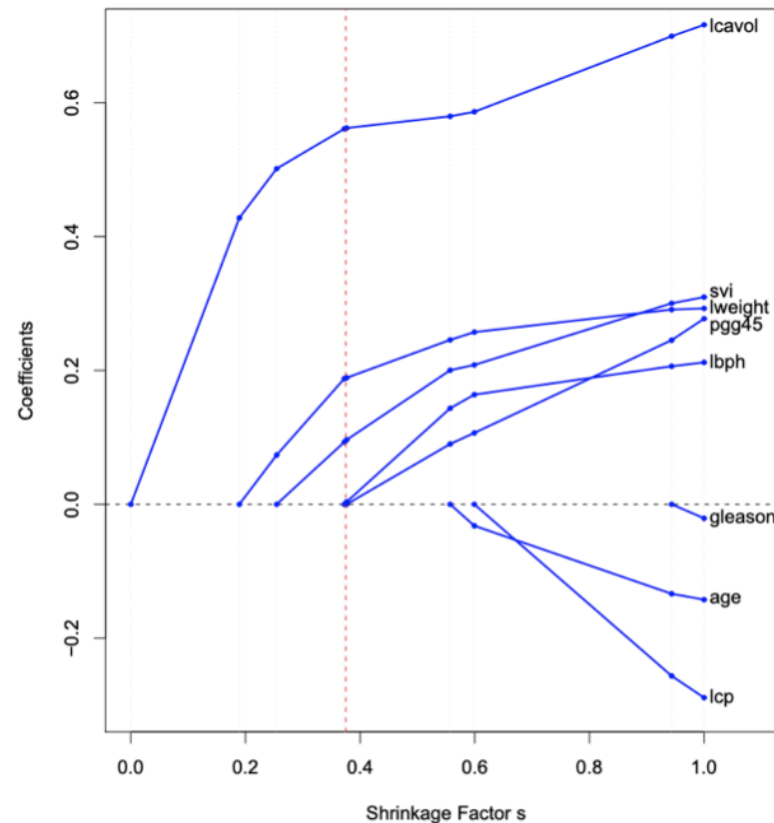


FIGURE 3.10. Profiles of lasso coefficients, as the tuning parameter t is varied. Coefficients are plotted versus $s = t / \sum_1^p |\hat{\beta}_j|$. A vertical line is drawn at $s = 0.36$, the value chosen by cross-validation. Compare Figure 3.8 on page 65; the lasso profiles hit zero, while those for 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.

Source: Figure 3.10 from [ESL](#)

(https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf).

COMPARE TO RIDGE SOLUTION

- High penalty causes weights to become smaller, but without being exactly 0.

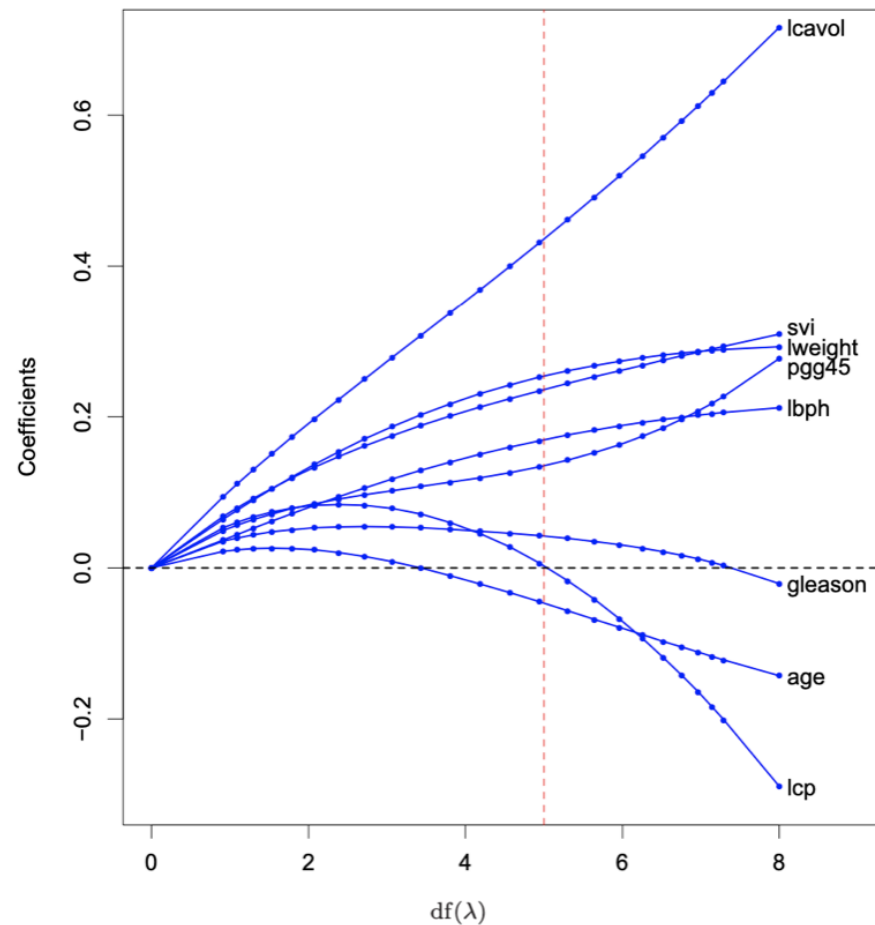


FIGURE 3.8. Profiles of ridge coefficients for the prostate cancer example, as the tuning parameter λ is varied. Coefficients are plotted versus $df(\lambda)$, the effective degrees of freedom. A vertical line is drawn at $df = 5.0$, the value chosen by cross-validation.

Source: Figure 3.8 from [ESL \(https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf\)](https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf).

LASSO VS. RIDGE SOLUTION

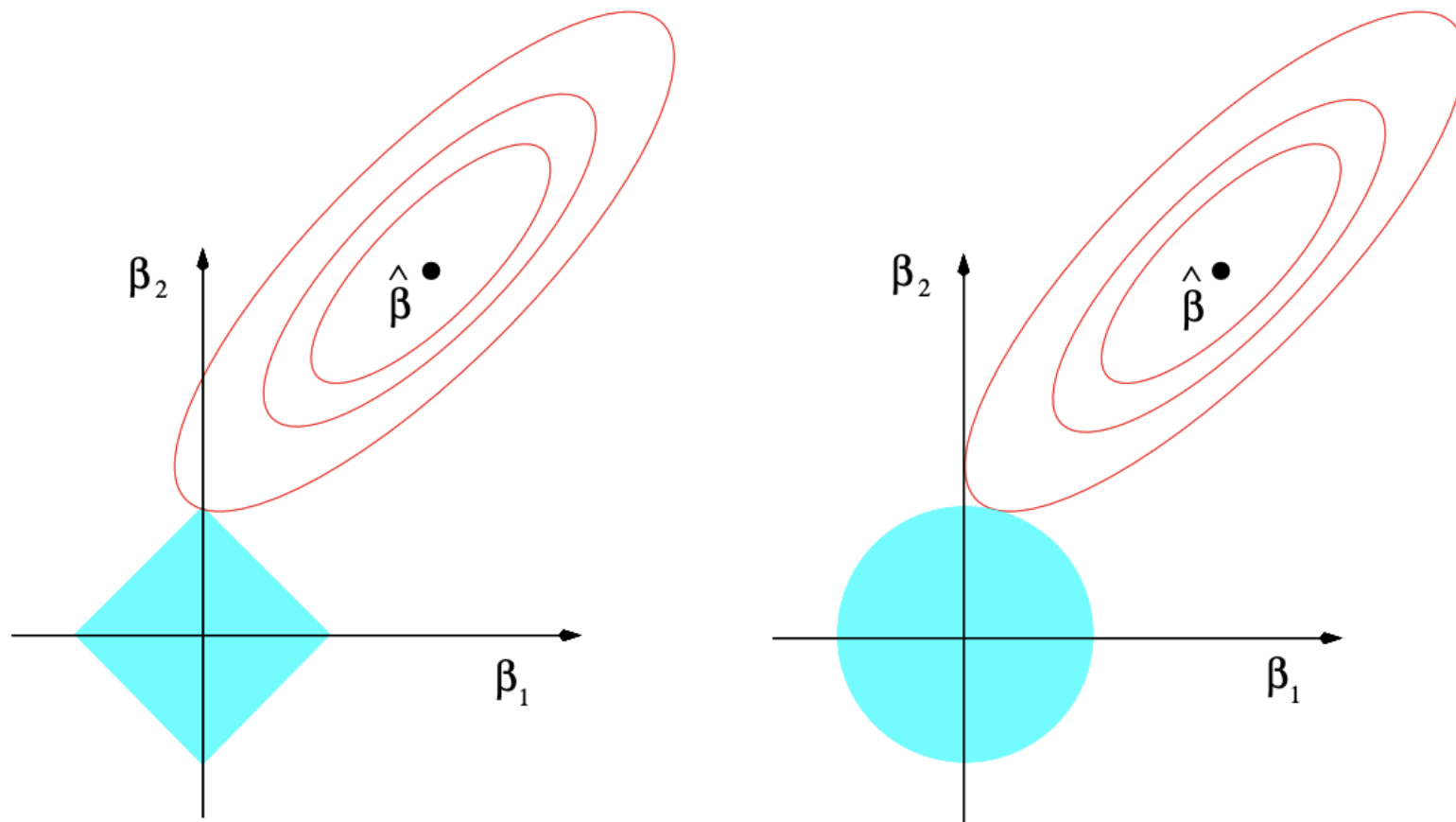


FIGURE 3.11. Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions $|\beta_1| + |\beta_2| \leq t$ and $\beta_1^2 + \beta_2^2 \leq t^2$, respectively, while the red ellipses are the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

Source: Figure 3.11 from [ESL \(https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf\)](https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf).

- In high dimensions, in Lasso, more likely to encounter edges or peaks.

HOW TO PICK λ ?

- Divide training set into train and validation:
 - train with different λ settings
 - pick the λ with smallest **validation** error (not test error!)
- K-fold cross-validation:
 - Divide your training set into K folds, for each fold i:
 - train with different λ settings on the other K-1 folds
 - compute error on fold i for each λ
 - average error across fold and pick λ with smallest cross-validation error
- Other types of cross-validation (leave-one-out cross-validation etc...)

WHAT YOU SHOULD KNOW

- Linear regression definition
- OLS solution and interpretation
- Ridge solution and tradeoff
- Lasso (just the penalty and what it optimizes for)

There is a lot more to learn about regression!

- Class in statistics department (e.g. 36-707)
- questions to think about:
 - what happens when Y is multidimensional? How to adapt the solution?
 - see section 3.4.1 for an interpretation of the effect of Ridge on different dimensions in the X (there is more shrinkage applied to the directions of variance corresponding to the small eigenvalues).
 - how can we use the ridge regression solution to formulate kernel regression?