Tutorial on Linear Regression

HY-539: Advanced Topics on Wireless Networks & Mobile Systems

Prof. Maria Papadopouli

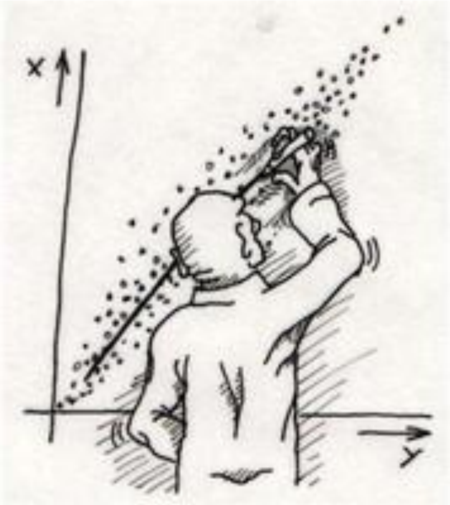
Evripidis Tzamousis

tzamusis@csd.uoc.gr

Agenda

1. Simple linear regression
2. Multiple linear regression
3. Regularization
4. Ridge regression
5. Lasso regression
6. Matlab code

Linear regression



One of the simplest and widely used statistical techniques for predictive modeling

Supposing that we have observations (i.e., targets) $y = (y_1, \dots y_n) \in \mathbb{R}^n$

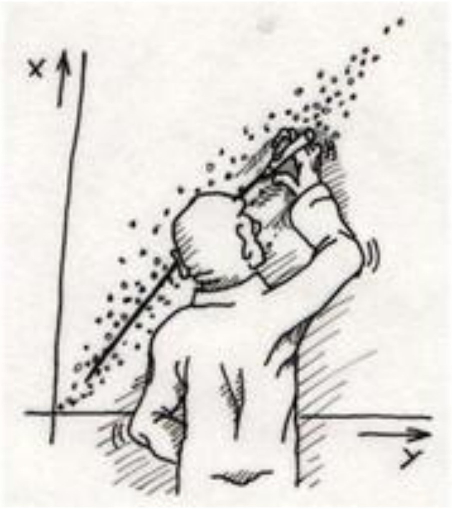
and a set of explanatory variables (i.e., predictors) $X_1, \dots X_p \in \mathbb{R}^n$

We build a linear model $y = X\beta^*$

where $\beta^* = (\beta_1^*, \dots \beta_p^*) \in \mathbb{R}^p$ are the coefficients of each predictor

y given as a weighted sum of the predictors, with the weights being the coefficients

Why using linear regression?



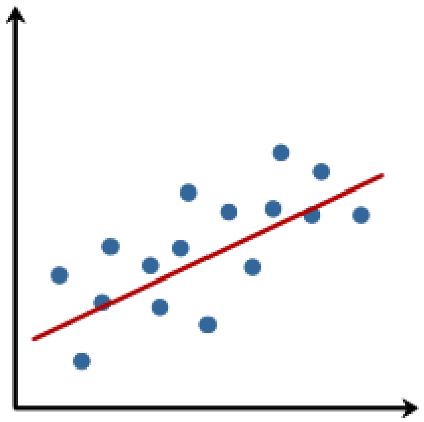
Prediction:

- Additional value of $\boldsymbol{X}$ is given without a corresponding value of $\boldsymbol{y}$
- Fitted linear model is makes a prediction of $\boldsymbol{y}$

Strength of the relationship between $\boldsymbol{y}$ and a variable $\boldsymbol{x}_i$

- Assess the impact of each predictor $\boldsymbol{x}_i$ on $\boldsymbol{y}$ through the magnitude of $\boldsymbol{\beta}_i$
- Identify subsets of $\boldsymbol{X}$ that contain redundant information about $\boldsymbol{y}$

Simple linear regression



Suppose that we have observations $y = (y_1, \ldots y_n) \in \mathbb{R}^n$

and we want to model these as a linear function of $x = (x_1, \ldots x_n) \in \mathbb{R}^n$

$$y = \beta^* x$$

To determine which is the optimal $\beta \in R^n$, we solve the least squares problem:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \beta x_i)^2 = \underset{\beta}{\operatorname{argmin}} \|y - \beta x\|_2^2$$

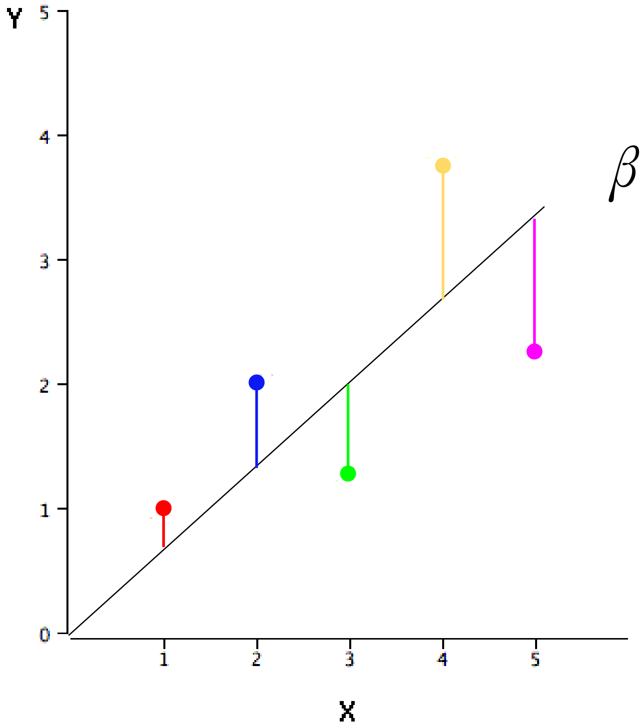
where β is the optimal β that minimizes the Sum of Squared Errors (SSE)

Example 1

Suppose that we have

- target variable $\mathbf{y} = (1, 2, 1.3, 3.75, 2.25)$
- predictor variable $\mathbf{x} = (1, 2, 3, 4, 5)$

Fit a linear model by finding the β that minimizes the Sum of Squared Errors (MSS)



$\beta = 0.7$

X	Y	Predicted Y	Squared Error
1.00	1.00	0.70	0.09
2.00	2.00	1.40	0.36
3.00	1.30	2.10	0.64
4.00	3.75	2.80	0.90
5.00	2.25	3.50	1.56

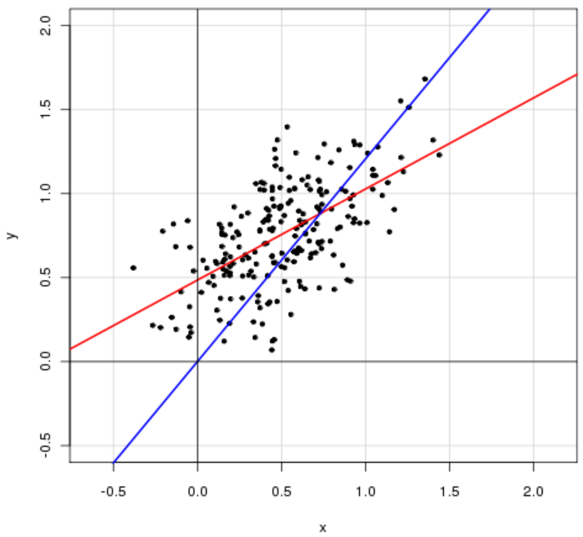
SSE = 3.55

We can add an intercept term β_0 for capturing noise not caught by predictor variable

Again we estimate $\hat{\beta}_0, \hat{\beta}_1$ using least squares

$$\hat{\beta}_0, \hat{\beta}_1 = \underset{\beta_0, \beta_1}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2 = \underset{\beta_0, \hat{\beta}_1}{\operatorname{argmin}} \|y - \beta_0 \mathbb{1} - \beta_1 x\|_2^2$$

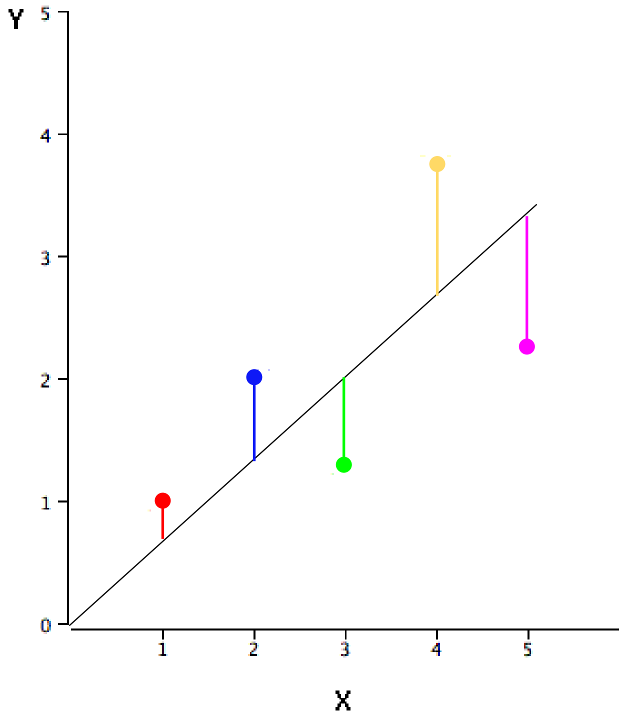
without intercept term



with intercept term

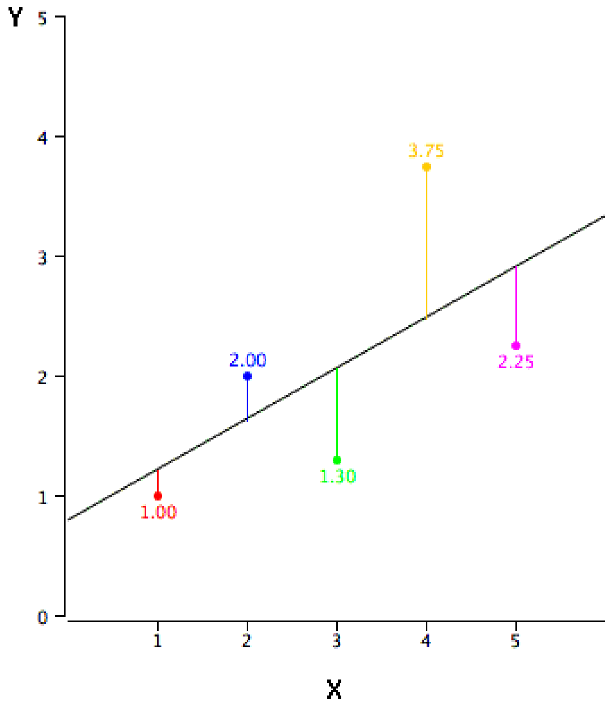
Example 2

Intercept term improves the accuracy of the model



Predicted Y	Squared Error
0.70	**0.09**
1.40	**0.36**
2.10	**0.64**
2.80	**0.90**
3.50	**1.56**

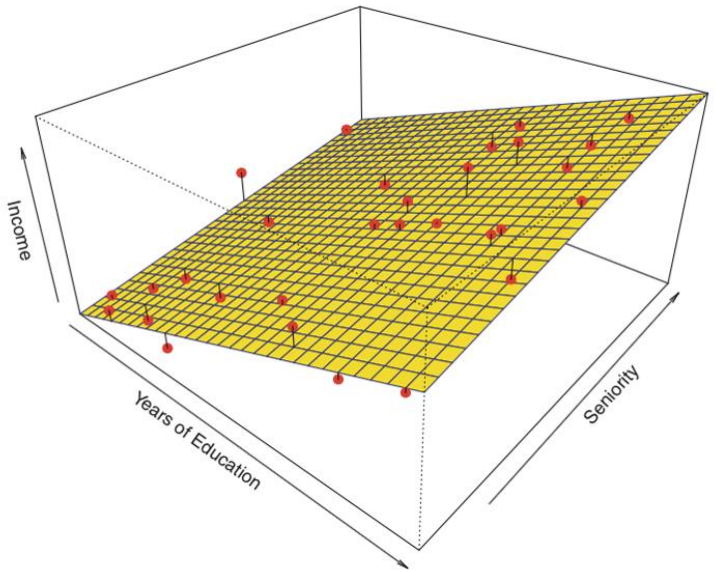
SSE = 3.55



Predicted Y	Squared Error
1.20	**0.04**
1.60	**0.16**
2.00	**0.49**
2.50	**1.56**
2.90	**0.42**

SSE = 2.67

Multiple linear regression

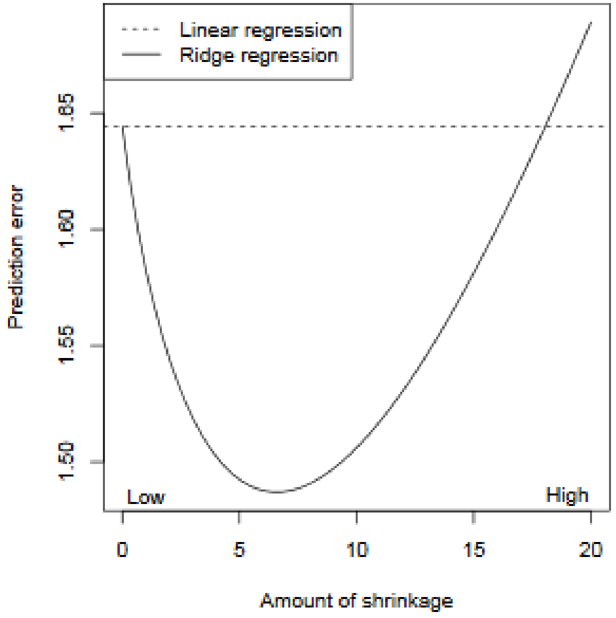


Attempts to model the relationship between two or more predictors and the target

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \|y - X\hat{\beta}\|_2^2$$

where $\hat{\beta}$ are the optimal coefficients $\beta_1, \beta_2, \ldots, \beta_p$ of the predictors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_p$ that minimize the above sum of squared errors

Regularization



Shrinks the magnitude of coefficients

Bias: error from erroneous assumptions about the training data

- High bias (underfitting) → miss relevant relations between predictors & target

Variance: error from sensitivity to small fluctuations in the training data

- High variance (overfitting) → model random noise and not the intended output

Bias – variance tradeoff: Ignore some small details, to get a more general "big picture"

Ridge regression

Given a vector with observations $y \in \mathbb{R}^n$ and a predictor matrix $X \in \mathbb{R}^{n \times p}$

the ridge regression coefficients are defined as:

$$\hat{\beta}^{\text{ridge}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$
$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}$$

Not only minimizing the squared error, but also the size of the coefficients!

Ridge regression

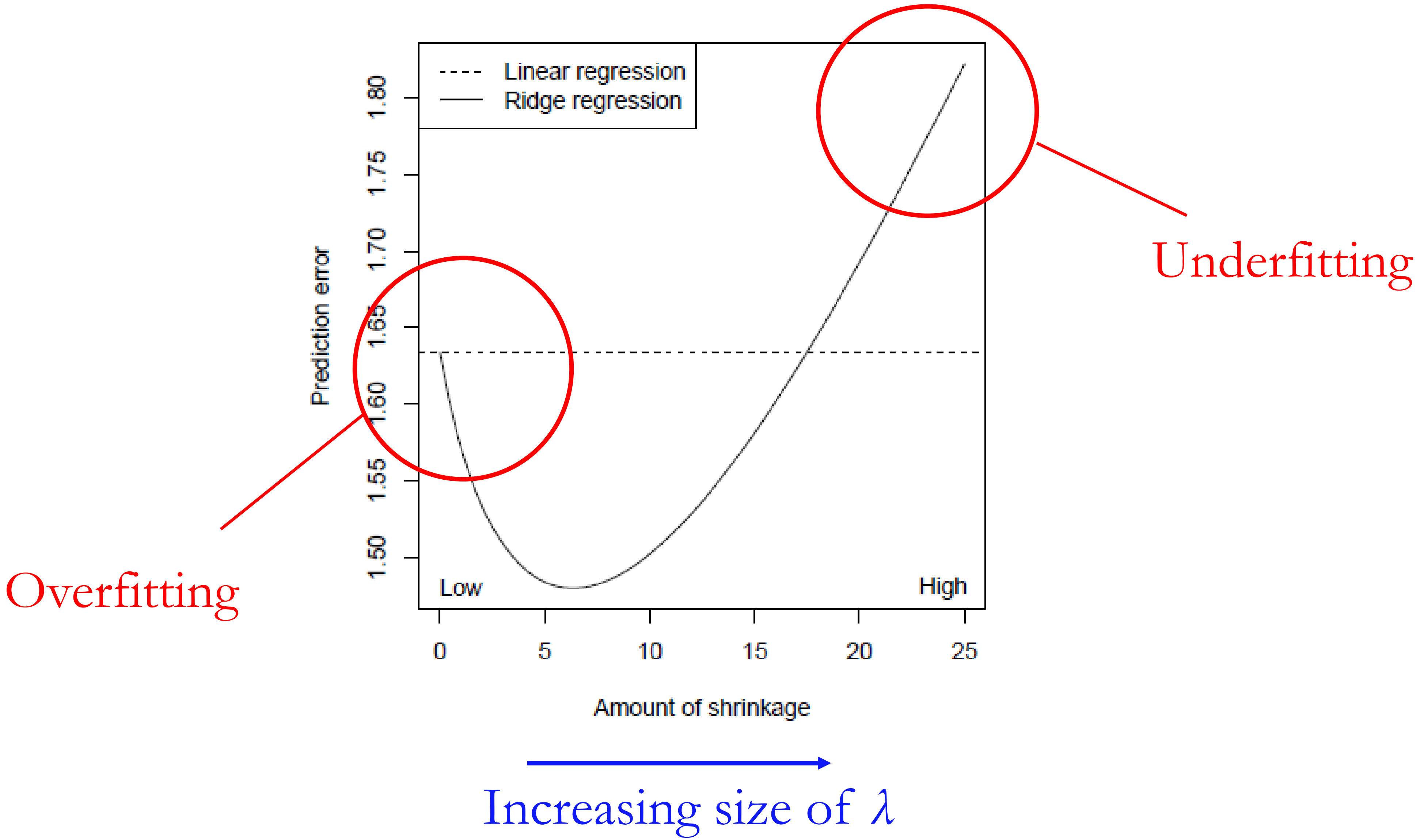
Here, $\lambda \geq 0$ is a tuning parameter for controlling the strength of the penalty

- When $\lambda = 0$, we minimize only the loss → overfitting
- When $\lambda = \infty$, we get $\hat{\beta}^{\text{ridge}} = 0$ that minimizes the penalty → underfitting

When including an intercept term, we usually leave this coefficient unpenalized

$$\hat{\beta}_0, \hat{\beta}^{\text{ridge}} = \underset{\beta_0 \in \mathbb{R},\, \beta \in \mathbb{R}^p}{\operatorname{argmin}} \; \|y - \beta_0 \mathbb{1} - X\beta\|_2^2 + \lambda \|\beta\|_2^2$$

Example 3


Linear regression
Ridge regression
Prediction error
1.50
1.55
1.60
1.65
1.70
1.75
1.80
Low
High
0
5
10
15
20
25
Amount of shrinkage
Overfitting
Underfitting
Increasing size of λ

Variable selection

Problem of selecting the most relevant predictors from a larger set of predictors

In linear model setting, this means estimating some coefficients to be exactly zero

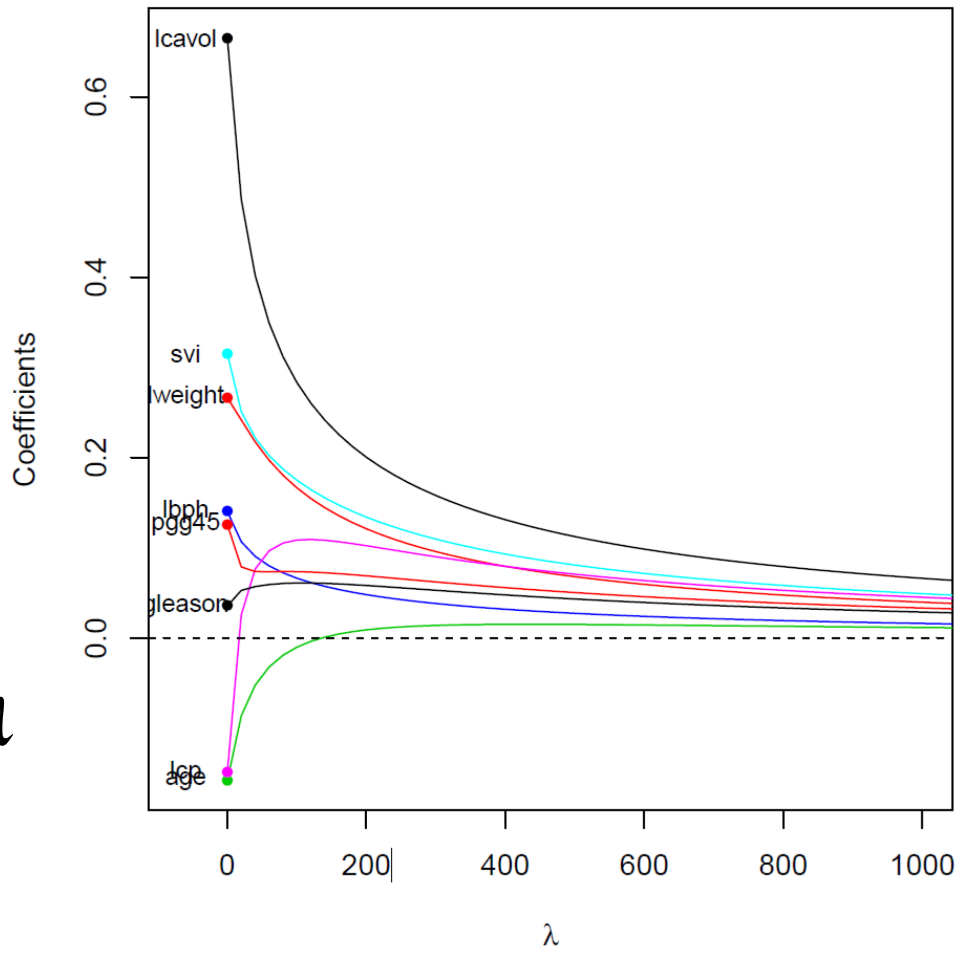
This can be very important for the purposes of model interpretation

Ridge regression cannot perform variable selection

- Does not set coefficients exactly to zero, unless $\lambda = \infty$

Example 4

Suppose that we are studying the level of prostate-specific antigen (PSA), which is often elevated in men who have prostate cancer. We look at n = 97 men with prostate cancer, and p = 8 clinical measurements. We are interested in identifying a small number of predictors, say 2 or 3, that drive PSA.



We perform ridge regression over a wide range of λ

This does not give us a clear answer...

Solution: Lasso regression

Lasso regression

The lasso coefficients are defined as:

$$\hat{\beta}^{\text{lasso}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_1}_{\text{Penalty}}$$

The only difference between lasso & ridge regression is the penalty term

- Ridge uses ℓ_2 penalty $\|\beta\|_2^2$
- Lasso uses ℓ_1 penalty $\|\beta\|_1$

Lasso regression

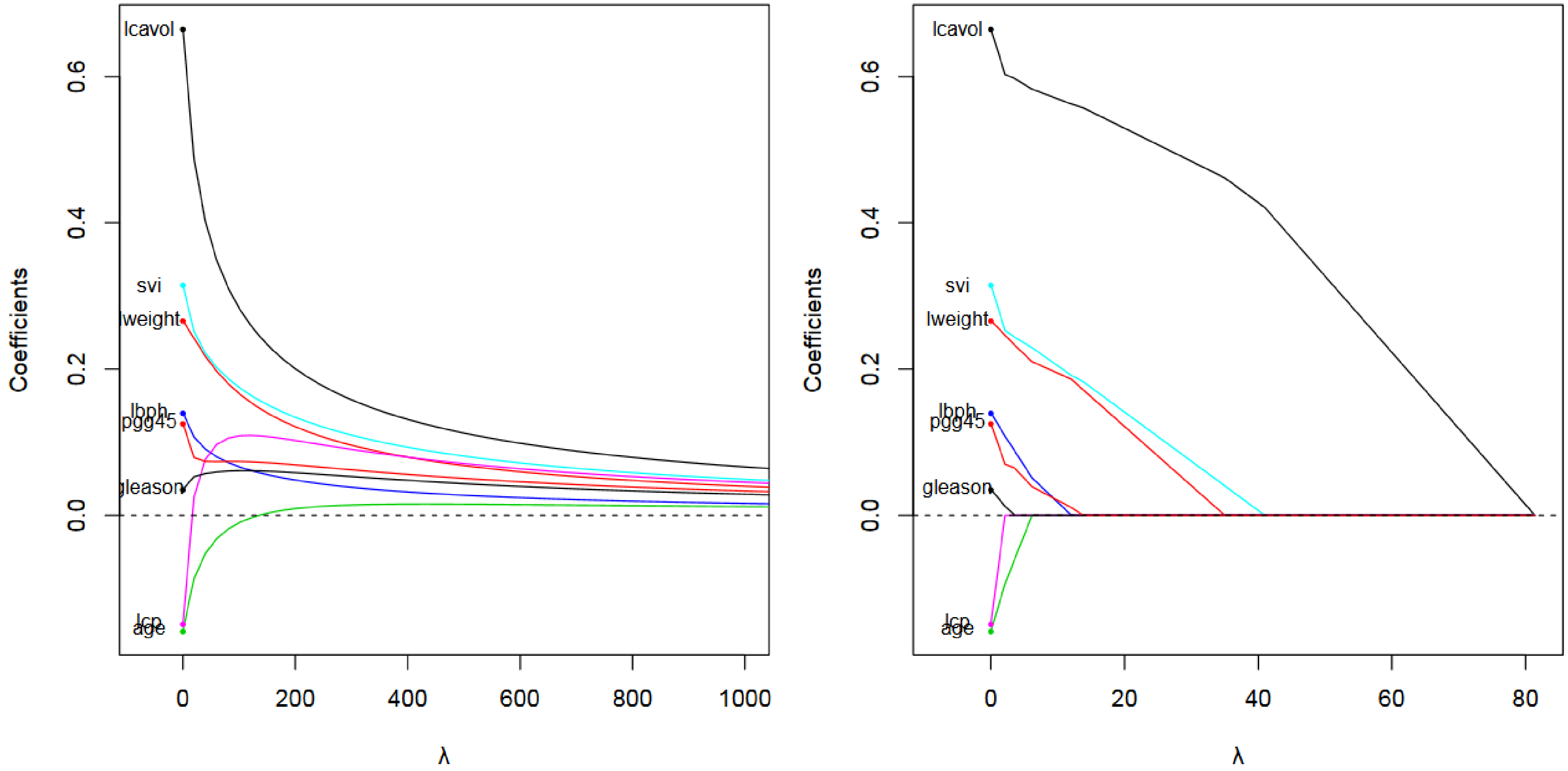
Again, $\lambda \geq 0$ is a tuning parameter for controlling the strength of the penalty

The nature of the l_1 penalty causes some coefficients to be shrunken to zero exactly

As λ increases, more coefficients are set to zero → less predictors are selected

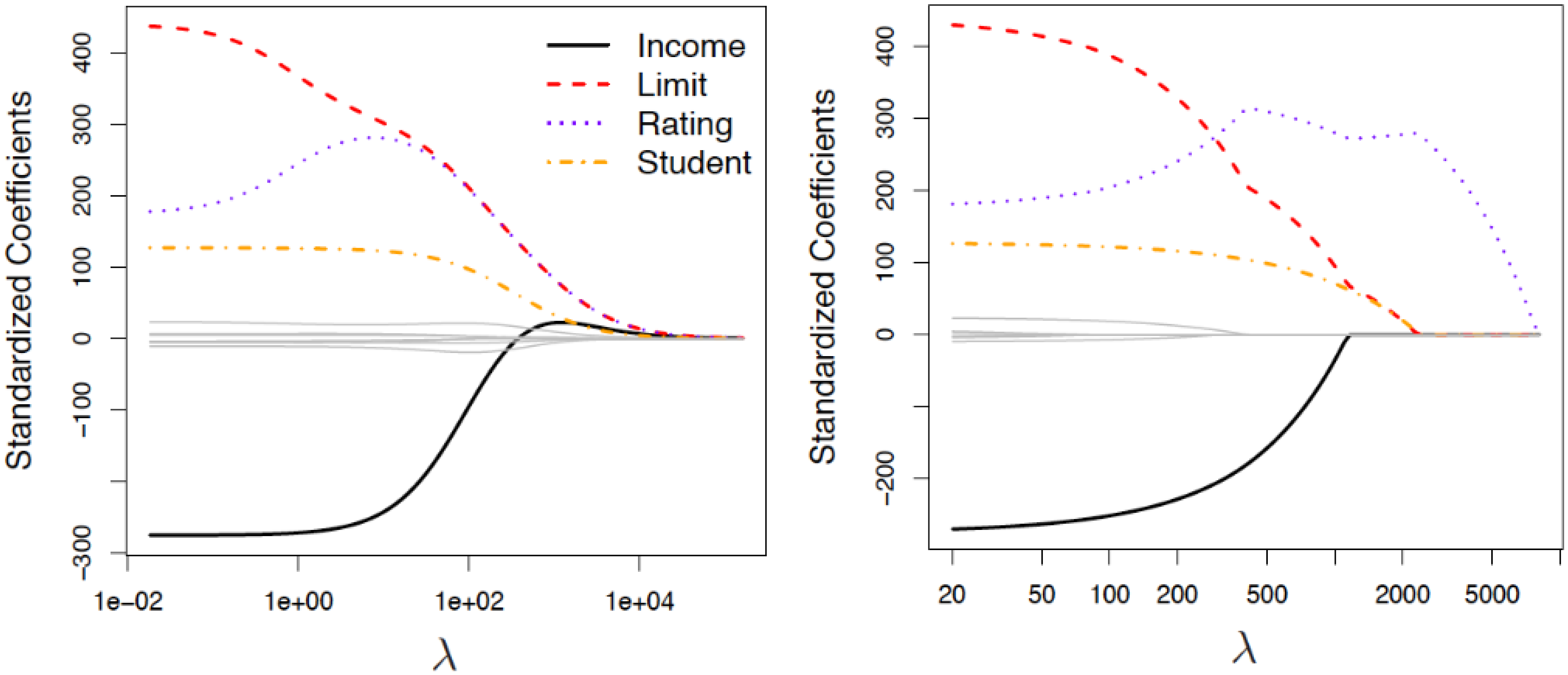
☺ **Can perform variable selection**

Example 5: Ridge vs. Lasso



lcp, age & gleason: the least important predictors → set to zero

Example 6: Ridge vs. Lasso



Constrained form of lasso & ridge

$$\hat{\beta}^{\text{ridge}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \|y - X\beta\|_2^2 \quad \text{subject to } \|\beta\|_2^2 \leq t$$

$$\hat{\beta}^{\text{lasso}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \|y - X\beta\|_2^2 \quad \text{subject to } \|\beta\|_1 \leq t$$

For any λ and corresponding solution in the penalized form, there is a value of t such that the above constrained form has this same solution. The imposed constraints constrict the coefficient vector to lie in some geometric shape centered around the origin

Type of shape (i.e., type of constraint) really matters!

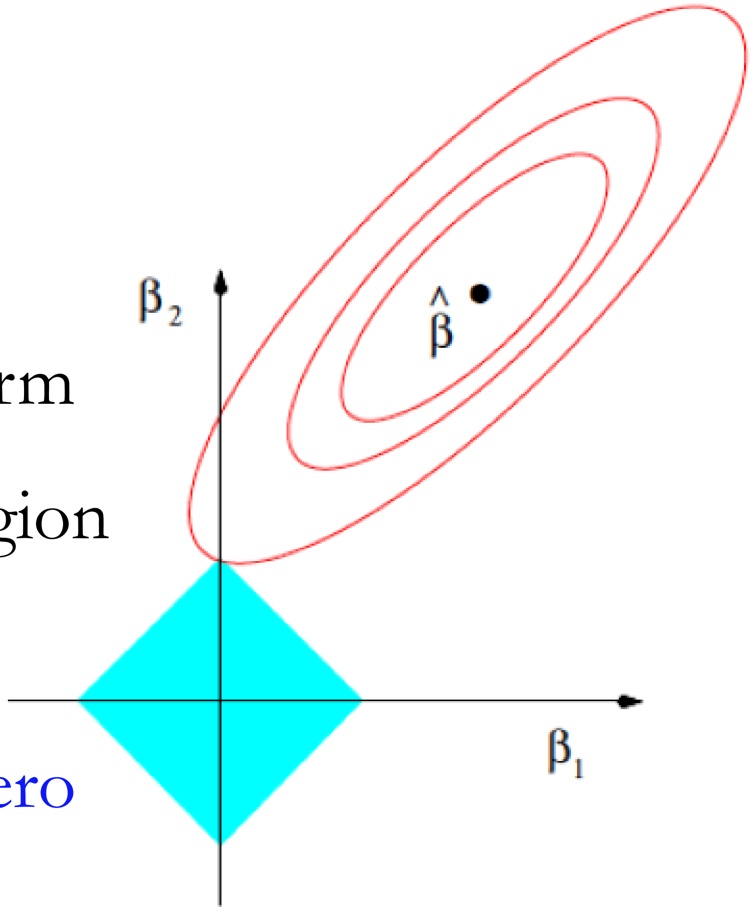
Why lasso sets coefficients to zero?

The elliptical contour plot represents sum of square error term

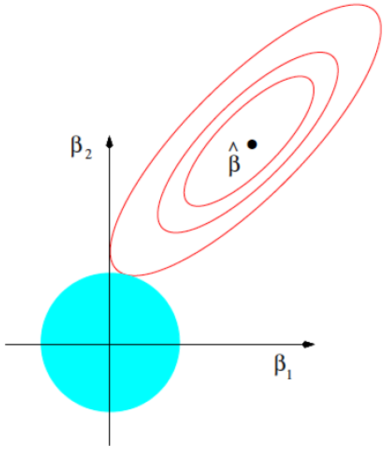
The diamond shape in the middle indicates the constraint region

Optimal point: intersection between ellipse & circle

- Corner of the diamond region, where the coefficient is zero



Instead with ridge:



Matlab code & examples

```
% Lasso regression

B = lasso(X,Y); % returns beta coefficients for a set of regularization parameters lambda
[B, I] = lasso(X,Y) % I contains information about the fitted models


% Fit a lasso model and let identify redundant coefficients
X = randn(100,5);               % 100 samples of 5 predictors
r = [0; 2; 0; -3; 0;];          % only two non-zero coefficients
Y = X*r + randn(100,1).*0.1;    % construct target using only two predictors
[B, I] = lasso(X,Y);            % fit lasso

% examining the 25th fitted model
B(:,25)         % beta coefficients
I.Lambda(25)    % lambda used
I.MSE(25)       % mean square error
```

Matlab code & examples

```
% Ridge regression

X = randn(100,5);                 % 100 samples of 5 predictors
r = [0; 2; 0; -3; 0;];            % only two non-zero coefficients
Y = X*r + randn(100,1).*0.1;  % construct target using only two predictors

model = fitrlinear(X,Y, ’Regularization’, ’ridge’, ‘Lambda’, 0.4));
predicted_Y = predict(model, X);    % predict Y, using the X data


err = mse(predicted_Y, Y);    % compute error

model.Beta     % fitted coefficients
```