- X=1 if female

	Coefficient	Std. Error	t-statistic	p-value
Intercept	509.80	33.13	15.389	< 0.0001
gender[Female]	19.73	46.05	0.429	0.6690

- X=1 if male

- X=1 if female

$y_i = \begin{cases} B_0 + B_1 + \epsilon & \text{female} \\ B_0 & \text{male} \end{cases}$

	Coefficient	Std. Error	t-statistic	p-value
Intercept	509.80	33.13	15.389	< 0.0001
gender[Female]	19.73	46.05	0.429	0.6690

Avg Balance for female = $B_0 + B_1 = 509.8 + 19.73$

" " " male = $B_0 = 509.8$

- X=1 if male

$y_i = B_0' + B_1' X + \epsilon \; : \begin{cases} B_0' + B_1' + \epsilon & \text{male} \\ B_0' & \text{female} \end{cases}$

female $\Rightarrow$ $B_0' = 509.8 + 19.73$, males: $B_0' + B_1' = 509.8$

$B_1' = -19.73$

University of Pittsburgh

ECE 2195: Special Topics – Computers Machine Learning

Shrinkage Methods – Regularization

Mai Abdelhakim, PhD

ECE Department

Swanson School of Engineering

University of Pittsburgh

maia@pitt.edu

Objectives of this Unit

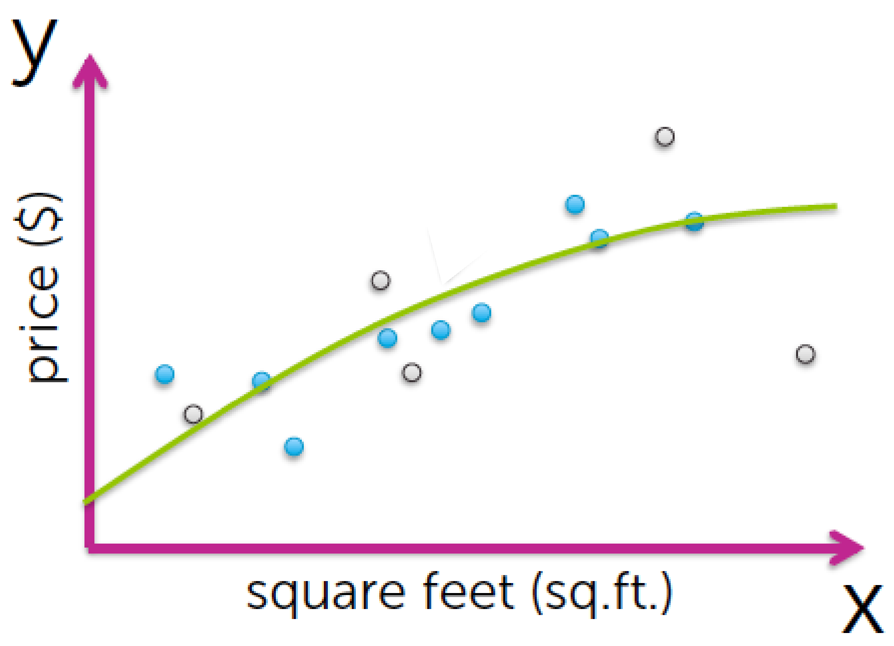
- Shrinkage methods:
  - Ridge regression
  - Lasso regression

Impact of Number of Features

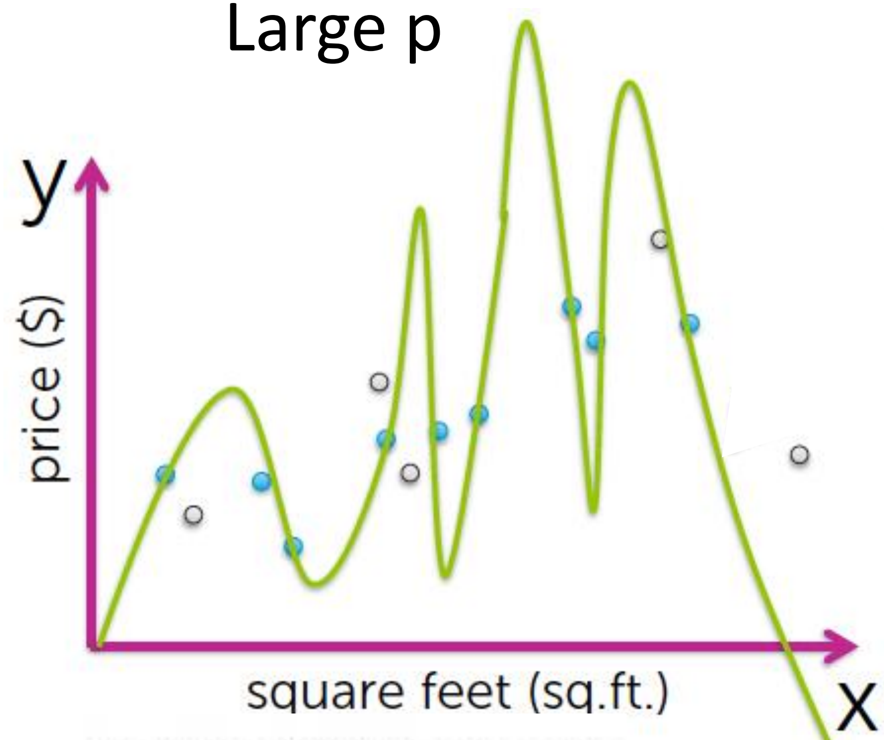
- We can define a polynomial regression function with p features as follows:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_1^3 + \ldots + \beta_p X_1^p$$

Small p



Large p



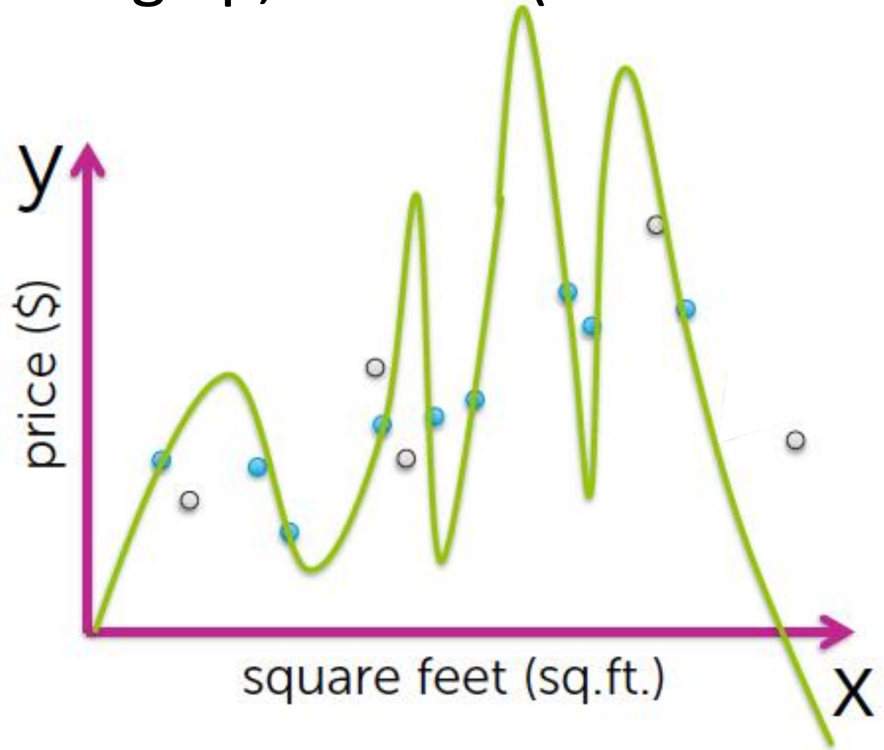
Emily Fox & Carlos Guestrin

Impact of the Number of Observations

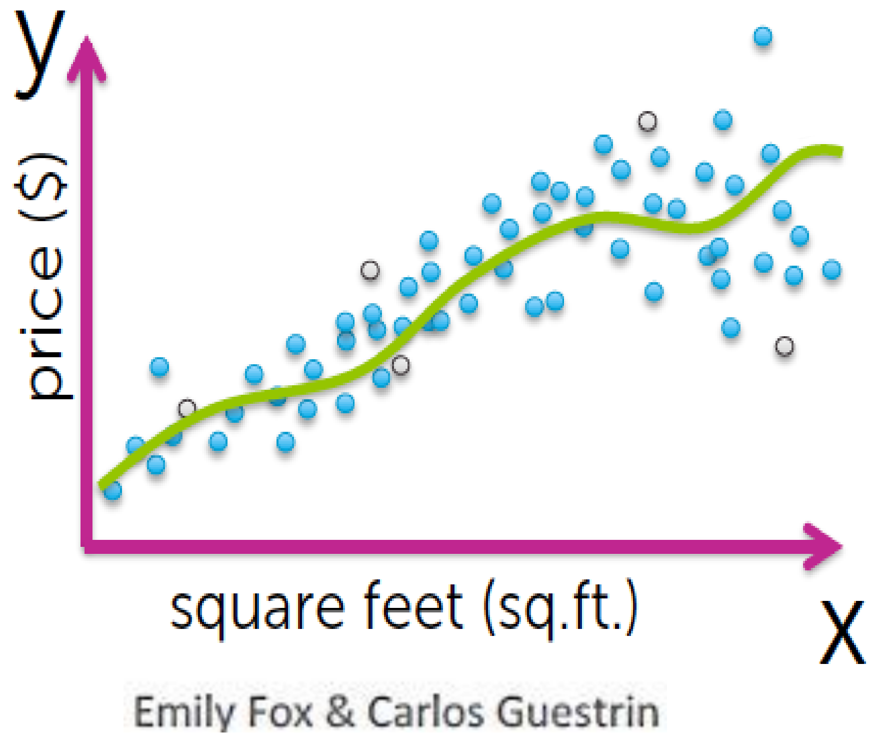
- Needs a lot of observations to avoid overfitting

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_1^3 + \ldots + \beta_p X_1^p$$

Large p, small n (n is # observations)



Large p, large n



Need of sufficient training observations

- Same phenomena applies when there are many features in a linear regression model without using polynomial terms
  - We need number of features $p << n$
    - Data to reflect all possible combinations between the features and the response

- Accuracy: if number of features (p) is large without large number of observations, accuracy will degrade (large variance)

Feature Selection

- Recall the concept of feature selection methods:
  - Best subset: search over all possible combinations of features
  - Forward selection
  - Backward selection
  - Mixed selection

Can we include large number of features without overfitting?

- Can we do better with linear regression?
  - **Can we include large number of features, without overfitting?**

- Can we replace the ordinary least square fitting by another fitting that solve this problem?
  - We can fit a single model and include all features - but use a technique that shrinks some coefficient estimates towards zero. (why zero?)
    - This is the main idea behind **Ridge and Lasso regression**

Ridge Regression

- Ordinary Least Squares (OLS) estimates the coefficients by minimizing

$$\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

- Ridge Regression, also called L_2 *regularization (as it uses the L2 norm),*
  - Modifies the objective function (that needs to be minimized) to

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \boxed{\sum_{j=1}^{p} \beta_j^2}$$

λ is a tuning parameter

$$= \mathrm{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2$$

Shrinkage penalty

➔ ***L_2 norm of coefficients (excluding $\boldsymbol{\beta_0}$)***

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

- The first term: Ridge regression tries to find coefficient estimate that **minimizes** the **RSS** (same as least squares)
  - To better fit to the training data

- The second term is called **shrinkage penalty**: has the effect of **shrinking** coefficients towards **zero**
  - To avoid overfitting and reducing the variance of the fitted model

- **λ is a tuning parameter** $(\lambda \geq 0)$ controls the **relative impact of these two terms**

Finding Coefficients of Ridge Regression

The optimal solution can be obtained by:

- Close-form solution: $\frac{\partial \mathrm{J}(\beta)}{\partial \beta}$ =0 ➔ $\hat{\beta}$ =$\left(X^T X + \lambda\, I_m\right)^{-1} X^T\, y$
  - I_m is the (p+1)x (p+1) identity matrix with first row all zeros, and rest of rows have ones on diagonal elements
    - For example, if p=2, then I_m= $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- Gradient descent, same iterative procedure as described before

Impact of the tuning parameter on regularization

- The objective function to minimize is: **J($\boldsymbol{\beta}$)= RSS($\boldsymbol{\beta}$) +** $\boldsymbol{\lambda \sum_{j=1}^{p} \beta_j^2}$

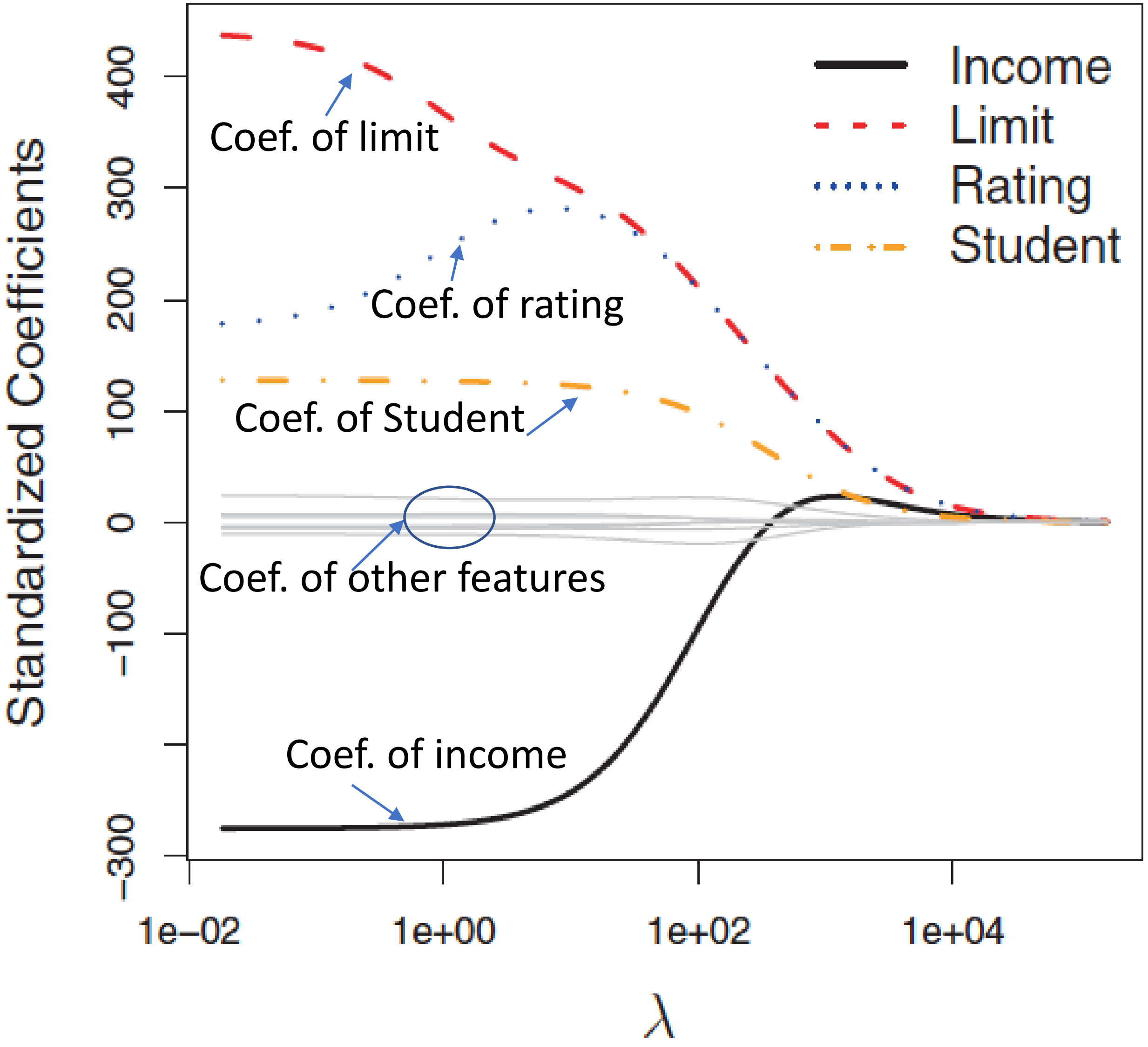
- **If λ=0 ➔** J(β)= RSS(β) , same least squares solution as before
  - May result in overfitting

- **If λ is very large (λ=∞) ➔** minimizing J(β) will result in setting all coefficients to zero (low magnitude)
  - This results in underfitting

Example: Credit dataset

- Credit data set (10 features): Records <u>balance</u> (average credit card debt for a number of individuals), <u>*age*</u> ,<u>number of *cards*</u>, years of <u>*education*</u>, <u>income</u>, credit <u>*limit*</u>, <u>*student*</u> status, and credit <u>*rating*</u>, other features
- Using ridge regression with different values of λ
  - Figure shows the change of coefficient with λ
    - λ close to zero ➔ least square estimates
    - λ large ➔ coefficient shrinks to zero

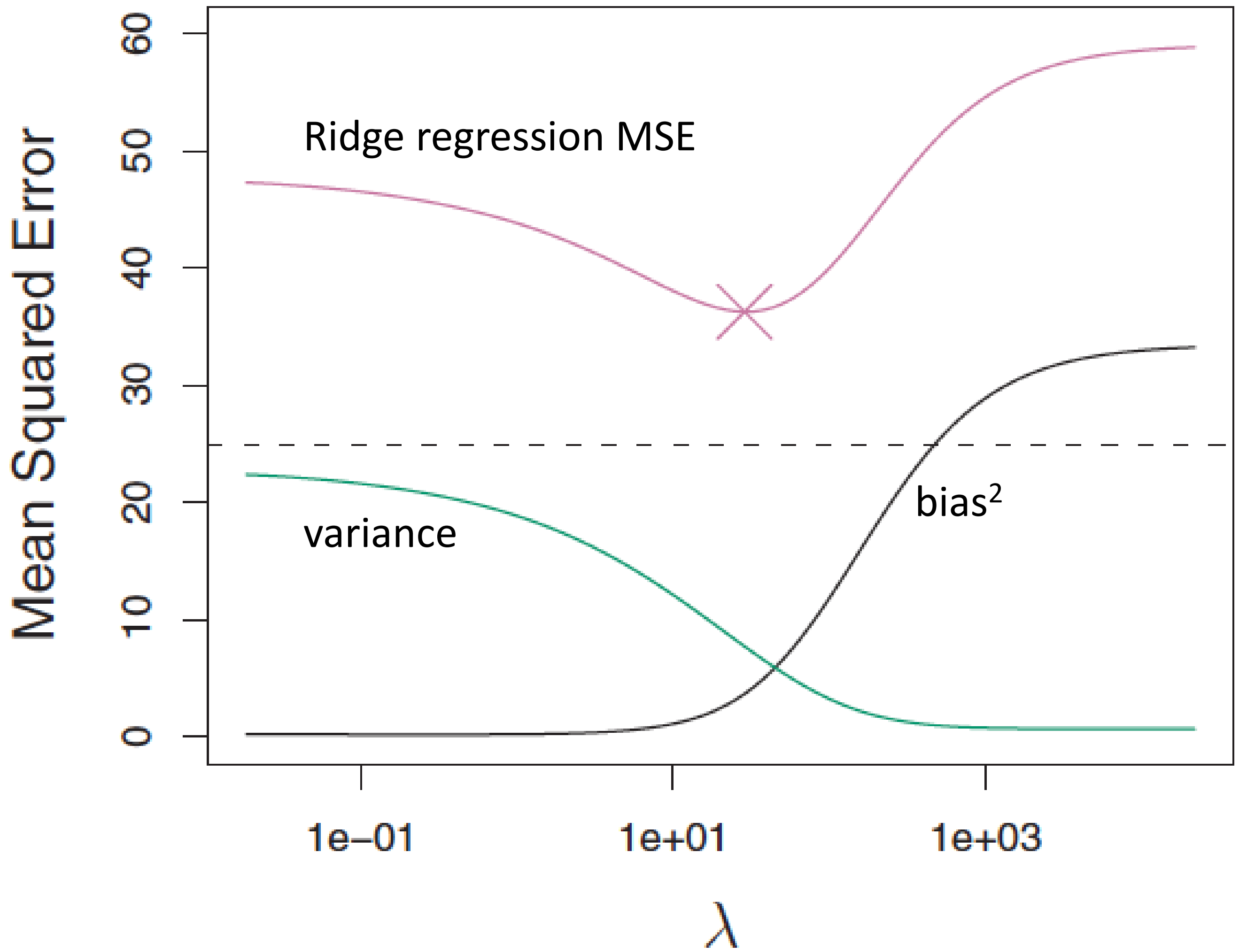
Standardized coefficient are the coefficient estimate when **features** are **scaled to have unit variance**



Select the tuning parameter to avoid overfitting

Figure shows simulated data with **n=50 training** examples and **p=45 features**

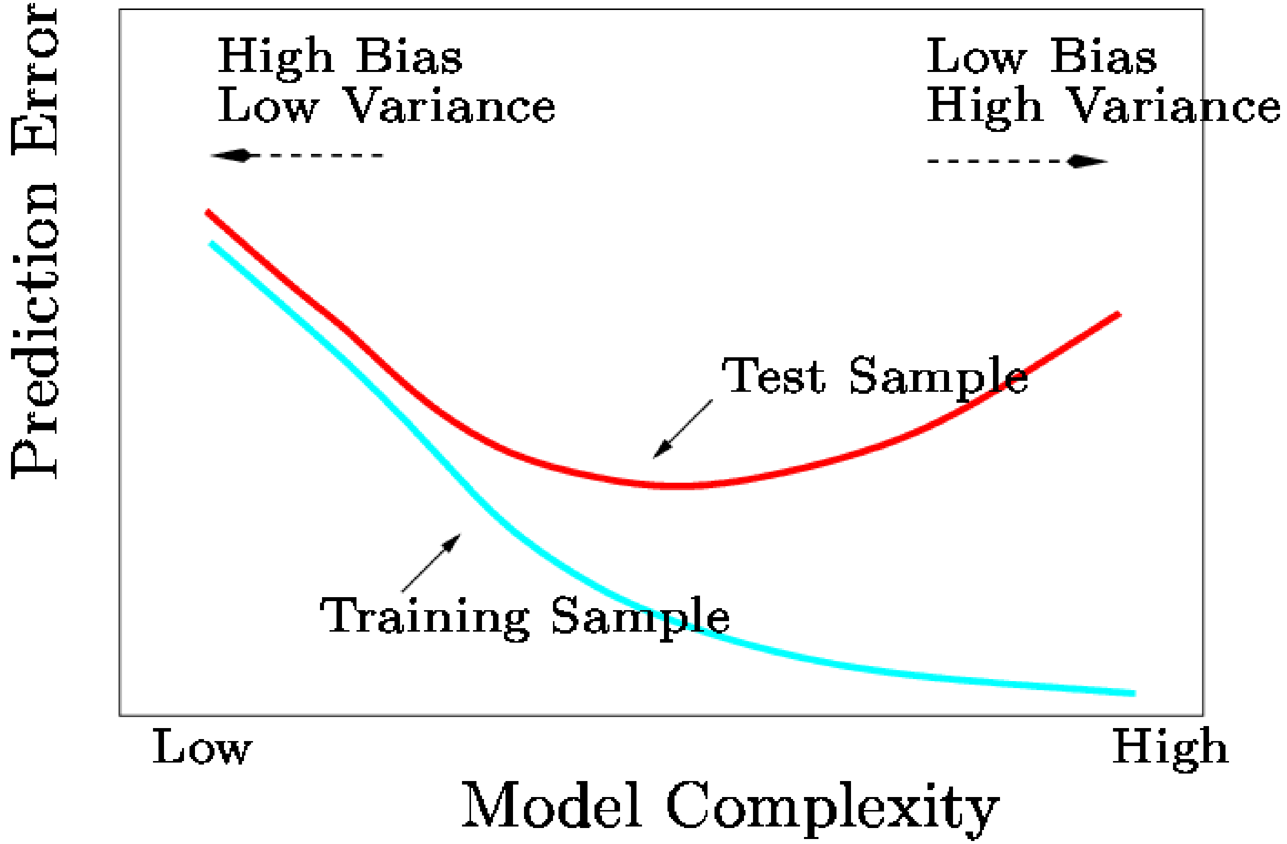
The shrinkage parameter is selected to achieve good bias-variance tradeoff



Ridge regression works in situations where OLS has high variance ($p \approx n$ or $p > n$)

Bias-Variance Tradeoff

- ***λ* increases** => flexibility of the model decreases (**less complex**)
  - At extreme case with very large λ : no features will be included (simple/trivial model)
- Ridge regression works in situations where OLS has high variance (p≈n or p > n)



Ridge Regression

- Advantages:
  - **Reduce variance**, avoid overfitting when p is large
  - Fit **single model**
- Disadvantages:
  - All coefficients shrink towards zero, but non of them will be set exactly to zero (if $\lambda \neq \infty$)
    - Will **not exclude any feature**
      - Credit card data: Ridge will always include all 10 features instead of selecting the most relevant ones
    - **Challenge in the model interpretation**

Note : L1 and L2 Norms (Linear Algebra)

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$

Lasso Regression

- Tries to overcome disadvantages of Ridge regression
- Modifies the objective function to use the L_1 norm (instead of the L_2 norm in Ridge)

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda \boxed{\sum_{j=1}^{p}|\beta_j|}$$

$$= \mathrm{RSS} + \lambda \sum_{j=1}^{p}|\beta_j|$$

➔ L_1 ***norm of coefficients (excluding*** $\boldsymbol{\beta_0}$)

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

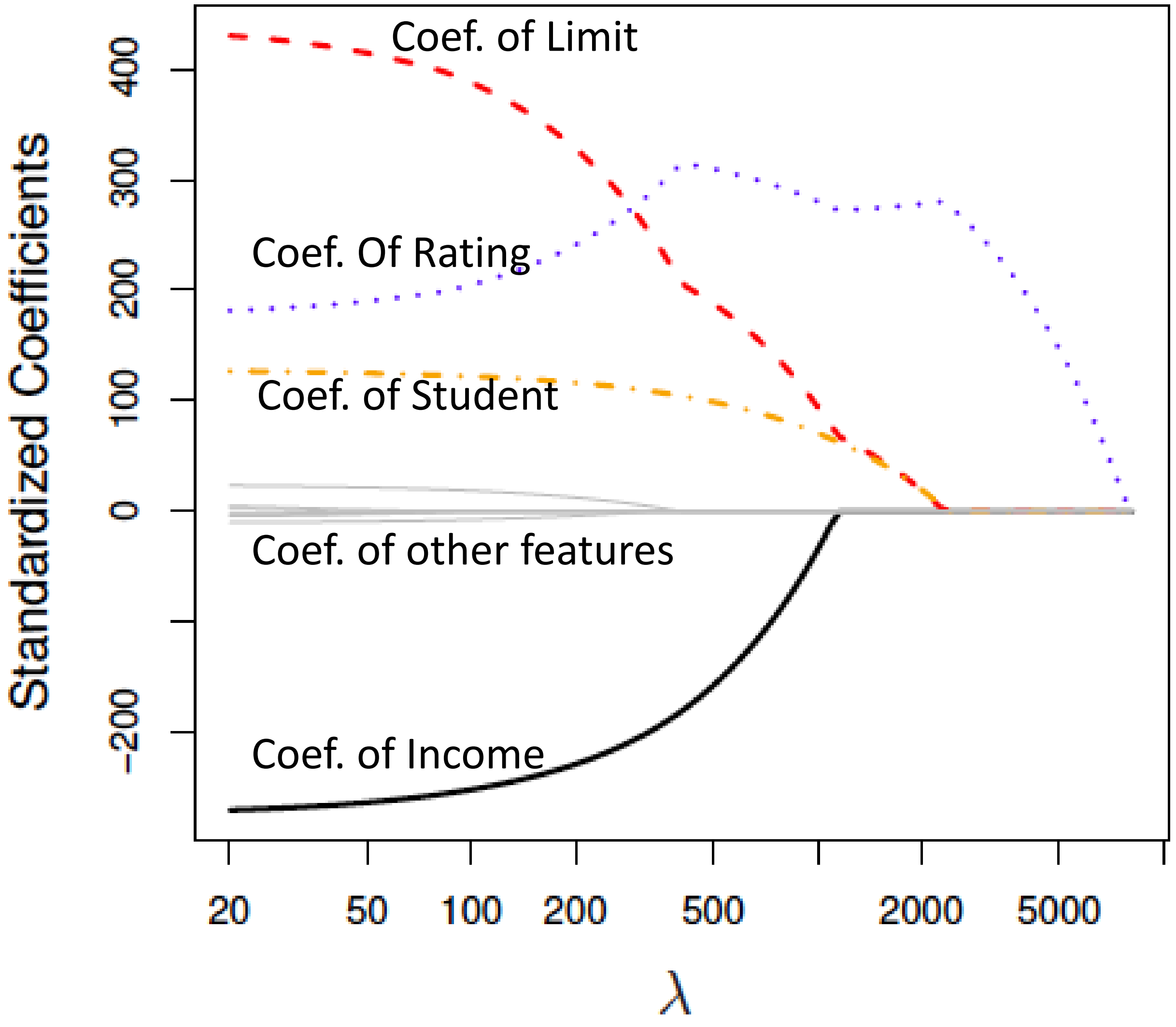
- When the **tuning parameter** (λ), **<u>some</u> coefficients will be forced to be zero**
  - Equivalent to feature selection
  - Easy to interpret

- Called **sparse model**, as it contains subset of features

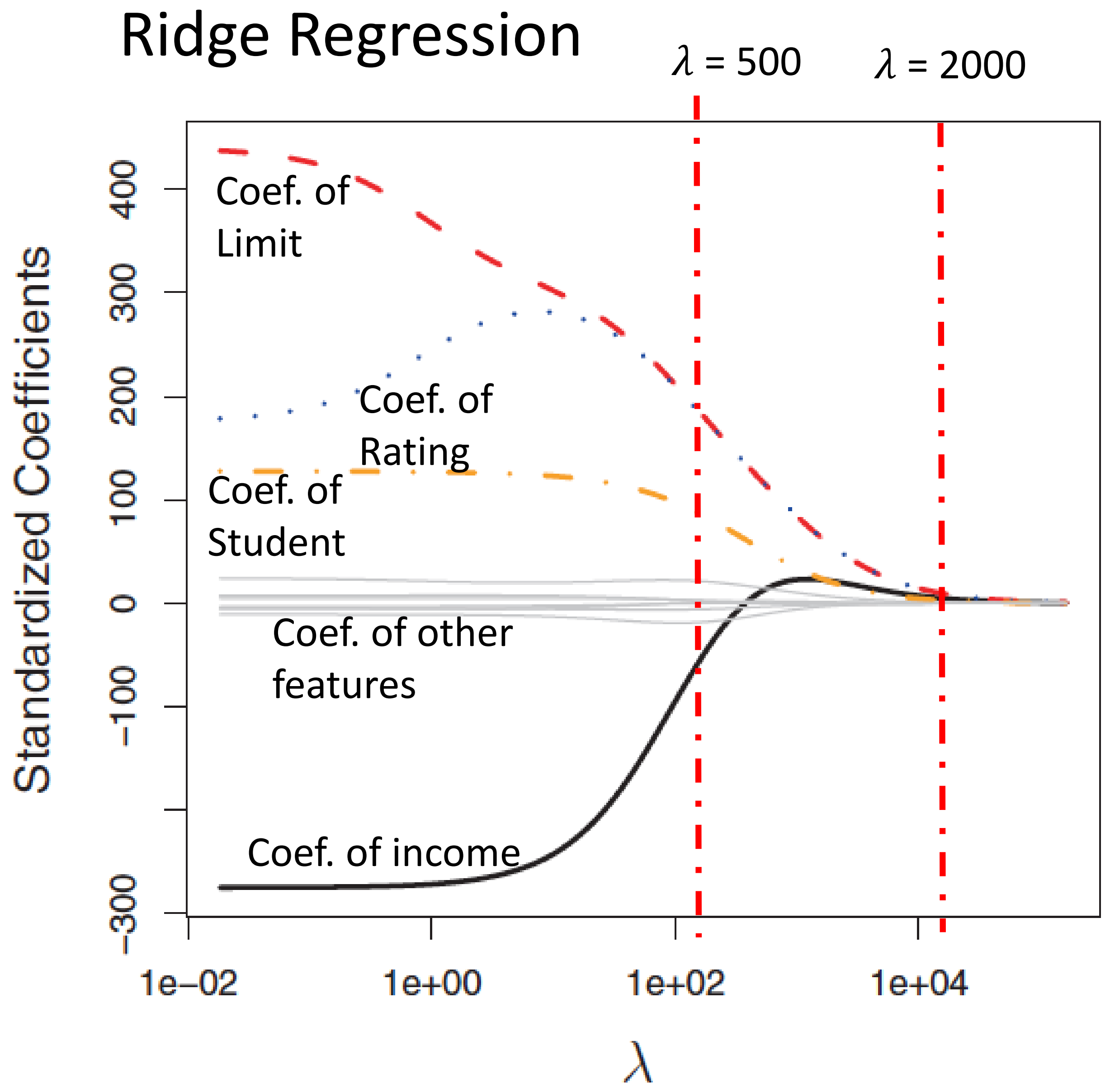
Assume one feature and find B1 with Lasso

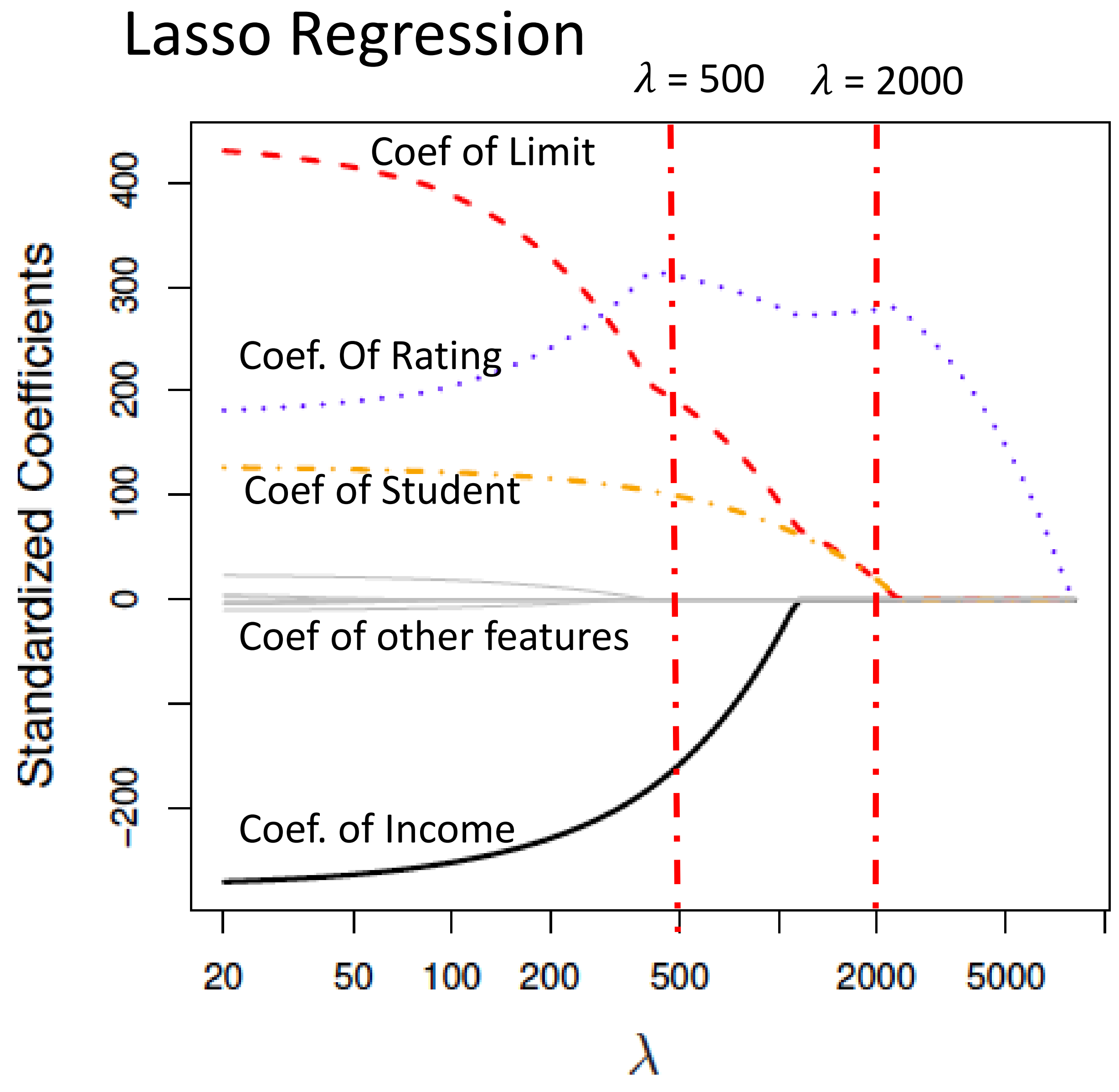
Example: Credit Dataset with Lasso Regression

- Features: Limit, Income, Rating Student, other features
- Apply the Lasso to the credit data set
  - λ close to zero ➔ least square estimates
  - λ large ➔ coefficient shrinks to zero
- For a given λ, subset of features can be selected, and other coefficients are set to zero



Example: Compare Ridge and Lasso





Ridge vs Lasso

- **Lasso** performs better when **small number of features are in fact related to the response** (have substantial coefficients)

- **Ridge** performs better when the **response is a function of all features**
  - All contribute to response with a small amount

- But the number of features that are related to response is typically unknown

- Cross-validation can be used to find which approach works better on a particular data set

Ridge Regression in Python

- Default value for tuning parameter (called alpha in python) is $\lambda = 1$

  from **sklearn.linear_model** import **Ridge**

  # train and fit the ridge regression model with training data
  RidgeModel=Ridge( ).fit(X_train, Y_train) # this uses default **alpha (=lambda) of 1**

  #find the R^2 metric with the .score
  RidgeModel.score(X_test,Y_test)

- To specify a value of λ (referred to as alpha in python): for example set $\lambda = 10$
  - RidgeModel10=**Ridge(alpha=10)**.fit(X_train, Y_train)

Lasso Regression in Python

- Default value for tuning parameter (called alpha in python) is λ = 1

  from **sklearn.linear_model** import **Lasso**

  lassoModel=**Lasso( ).**fit(X_train, Y_train)

- **Update the tuning parameter** to 0.01

  LassoModel001=Lasso(alpha=0.01). fit(X_train, Y_train)

- Use the .score method to get the performance

- You can find number of coefficients that are equal to zero using: numpy.sum(LassoModel001.coef_==0)

Note

- Ridge problem formulation is analytically similar to maximizing likelihood function when coefficients have prior probability that is Gaussian (B $\sim N\left(0, c^2 I\right)$, error $e_i \sim N\left(0, \sigma^2\right)$)

Note

- Lasso problem formulation is analytically similar to maximizing likelihood function P(Y|x) when coefficients have prior probability that is Laplace $(\mathrm{P(B)} \sim e^{-|B|/c}$, error $e_i \sim N\left(0, \sigma^2\right))$