NYC DATA SCIENCE
ACADEMY

Python Machine Learning

Penalized Linear Regression

NYC Data Science Academy

Outline

- **Biases and Model Variance - First Visit**
- **The Concept of Penalized/Regularized Linear Regression**
  - Ridge Linear Regression
  - Lasso Linear Regression

Biases and Model Variances

- The predictive goal of a supervised machine learning task is to find a model which best describes the underlying population of the data
- The model bases its understanding on the given population by investigating a finite **subset** of data, along with its target
- The algorithm tries to find patterns in the given dataset and expects to generalize it to the unseen data from the same population
- Statisticians have identified two major sources of performance errors in such a task
    - Biases
    - Model variances

The Meaning of Biases/Variance

- The biases in a regression task refers to the systematic error of overestimating/underestimating the target
- For example in a housing price prediction model, the model might on average overestimate the house prices
- The model variance, on the other hand, reflects the sensitivity of the model coefficients on the fed-in data points
- In the context of multiple linear regression, the model variance refers to the slopes and intercept $\widehat{SE}^2$

- Both of biases and model variances contribute to the final prediction error

Penalized Linear Regression

- Multiple linear regression is known to be an optimal **unbiased** linear estimator
- This does not mean it is impossible to improve its performance--for unseen new data points
- It turns out that there are techniques which produces slightly biased linear models with much less model variances --- at the end producing models potentially with a higher predictive accuracy
- This is achieved by penalized (regularized) linear regression
- In the literature, penalization is sometimes called regularization, penalized linear model is also called regularized linear model

Why Penalizing the Linear Coefficients?

- Recall that the root cause of the high model variance is due to the appearance of **multicollinearity** among the data features
- Intuitively the appearance of such hidden linear relationship makes our model over-confident about what it learns in the data --- which can **NOT** be generalized to unseen new test sets
- This suggests that as long as **multicollinearity** occurs, the linear coefficients estimated by the normal equation can be larger (in magnitude) than the un-observed true model's coefficients

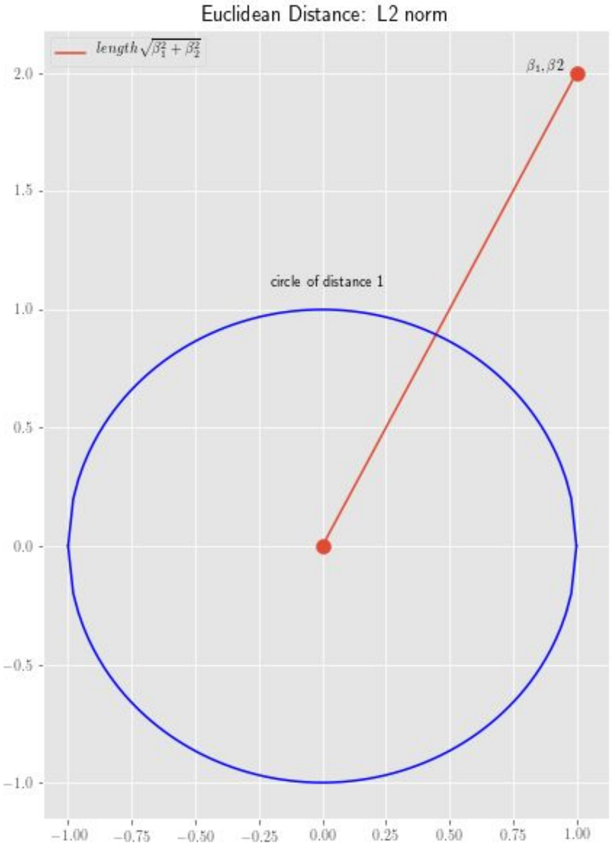
The Ways to Penalize the Coefficients

- We are aware that the slopes coefficients of features with different order of magnitudes (reflected by their stds) are quite different in their magnitudes. To make the slope comparison possible, we often standardize the input features, such that the new features all have unit standard deviation
- With this understood, the way to measure the total size of the slope vector is called the Euclidean distance measure in mathematics

The Euclidean Distance

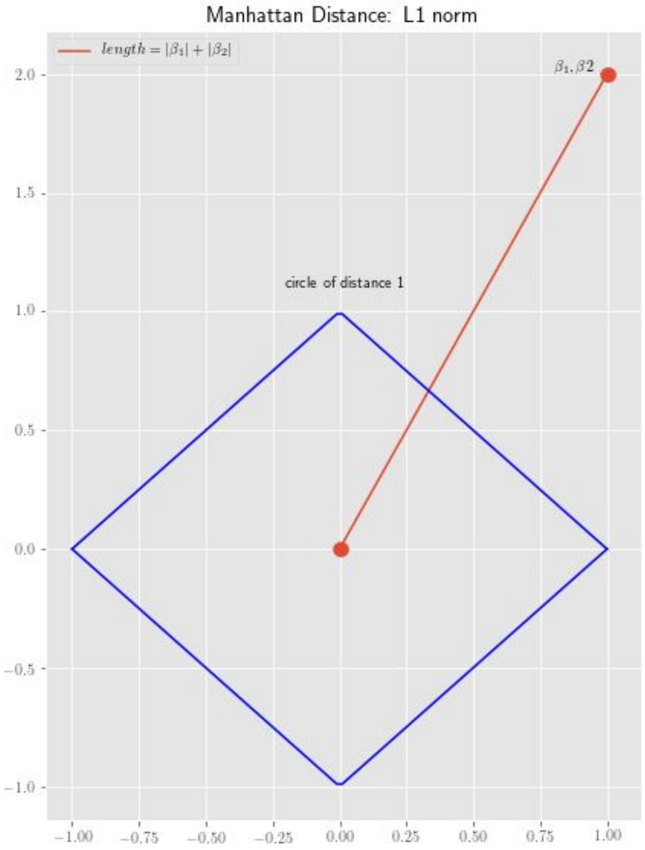
- The **Euclidean distance** is also called **Euclidean norm** (or **L2 norm**)
- The concentric circle is from those points which have a constant distance to the origin
- The L2 norm is often denoted

  by $||\beta||_{L^2}$symbolically



The L1 (Manhattan) Distance

- The **Manhattan distance** is also called **L1 distance/norm**
- The curve of constant distance to origin is the diamond shape shown in the right-hand plot
- **L1 norm** is denoted by $||\beta||_{L^1} = \sum_i |\beta_i|$



The Penalized RSS

- Recall that the **RSS** function is of the form

$$RSS(\beta_0, \cdots, \beta_p) \equiv \sum_{i \leq N} (y_i - \beta_0 - \sum_{j \leq p} \beta_j x_{ij})^2$$

- This is a **concave** quadratic function of multiple variables
- The idea of **penalization** is to add a penalization term to **RSS**
- For **Ridge** regression, we have

$$RSS(\beta_0, \cdots, \beta_p) + \lambda \cdot ||\beta||_{L^2}^2 = \sum_{i \leq N} (y_i - \beta_0 - \sum_{j \leq p} \beta_j x_{ij})^2 + \lambda \sum_{1 \leq j \leq p} \beta_j^2$$

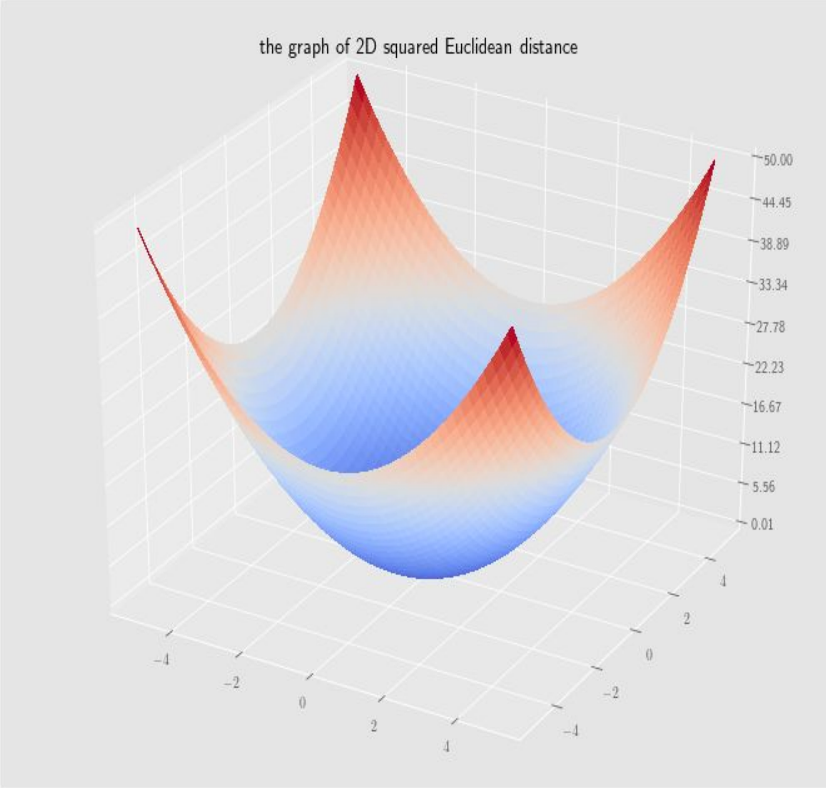
- For **Lasso** regression, we have

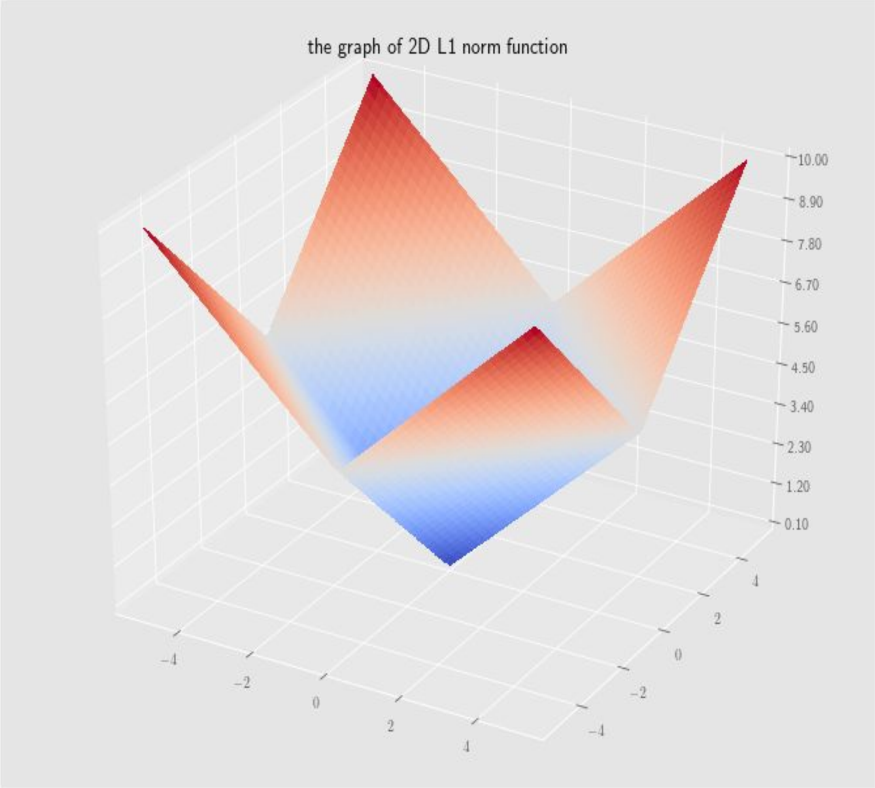
$$RSS(\beta_0, \cdots, \beta_p) + \lambda \cdot ||\beta||_{L^1} = \sum_{i \leq N} (y_i - \beta_0 - \sum_{j \leq p} \beta_j x_{ij})^2 + \lambda \sum_{1 \leq j \leq p} |\beta|_j$$

- There is a hybrid of Ridge and Lasso called Elastic-Net. We will discuss it at the end of the slides and in the lecture code

What is the Penalization Term Doing?

- To understand the effect of penalization, let us visualize the penalization terms (squared L2 norm and L1 norm, respectively) graphically





The Concavity of the Penalization Terms

- The hyperparameter lambda is a fixed positive constant
- When we take the horizontal contours (level curves) of both surfaces, we recover the concentric circles/concentric diamond-shapes, respectively
- The key property of these surfaces is that they are **concave up**, with a unique minimum at origin
- By adding them to the original **RSS**, the penalized **RSS** is still concave up
- The constant lambda determines the influence of the penalization. When lambda is sufficiently large, the penalization terms tend to ‘penalize’ large (in magnitude) regression slopes
- If we find the best regression coefficients by minimizing the penalized **RSS**, our new solution tends to be smaller than the solution from normal equation

The Limiting Behavior of Lambda

- When lambda approaches zero, the penalization term disappears, we recover the original **RSS**, defined in the multiple linear regression lecture
- On the other hand, when lambda approaches positive infinity, the effect of the penalization term takes over the original **RSS**, thus the location of minimum penalized **RSS** approach the origin in the space of slopes
- This is nothing but the **ANOVA** null model with only an intercept
- A penalized model is a compromise between the original data driven **MLR** and the **ANOVA** null model
- No matter whether we use **Ridge** or **Lasso** penalization, the tuning of the hyperparameter lambda establishes a one parameter family of models relating the **MLR** model and the null model
- ANOVA (analysis of variance) null model refers to the simple model defined by the sample means with NO slopes

The Benefit of Penalized Linear Models

- The original **MLR** model is unbiased, but it often has very high model variance induced by multicollinearity of the features
- The lambda-penalized linear model are biased, but with a suitably chosen lambda, the penalized linear model could have a much reduced variance, which translates to a lower overall prediction error
- This phenomenon is known to be ‘**bias variance trade-off**’, which plays an important role in building more sophisticated machine learning models

Ridge vs Lasso

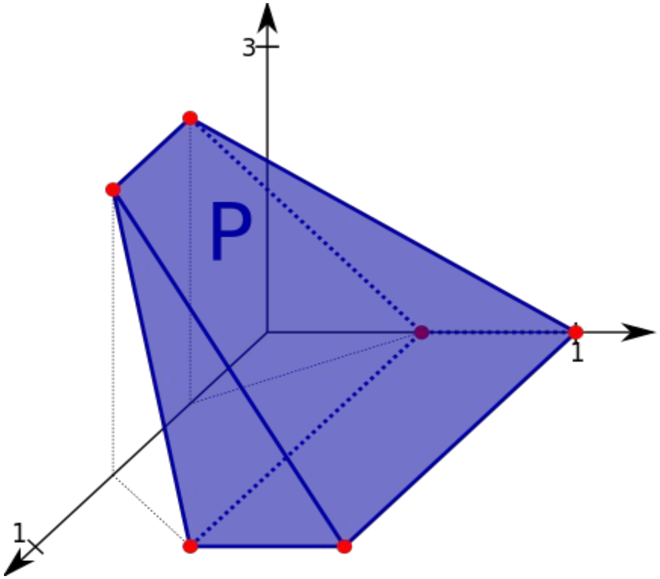
- Both **Ridge** and **Lasso** penalties can be used to regularize the linear models, but they have different behaviors
- **Ridge** penalty is a quadratic function of the slope coefficients, which makes the overall penalized **RSS** quadratic. Thus **Ridge** regression slope coefficients can be expressed explicitly. In fact it follows the regularized normal equation:

$$\vec{\beta} = (X^T X + \lambda I)^{-1} X^T y$$

- On the other hand, the coefficients of Lasso penalty is NOT a smooth function. As a consequence there is no closed-form solution to Lasso model's slope coefficients
- Instead one needs to adapt an iterative procedure to find the solution
- Because the penalty function is piecewise linear, the classical technique of **linear-programming** is often used to find the solution numerically

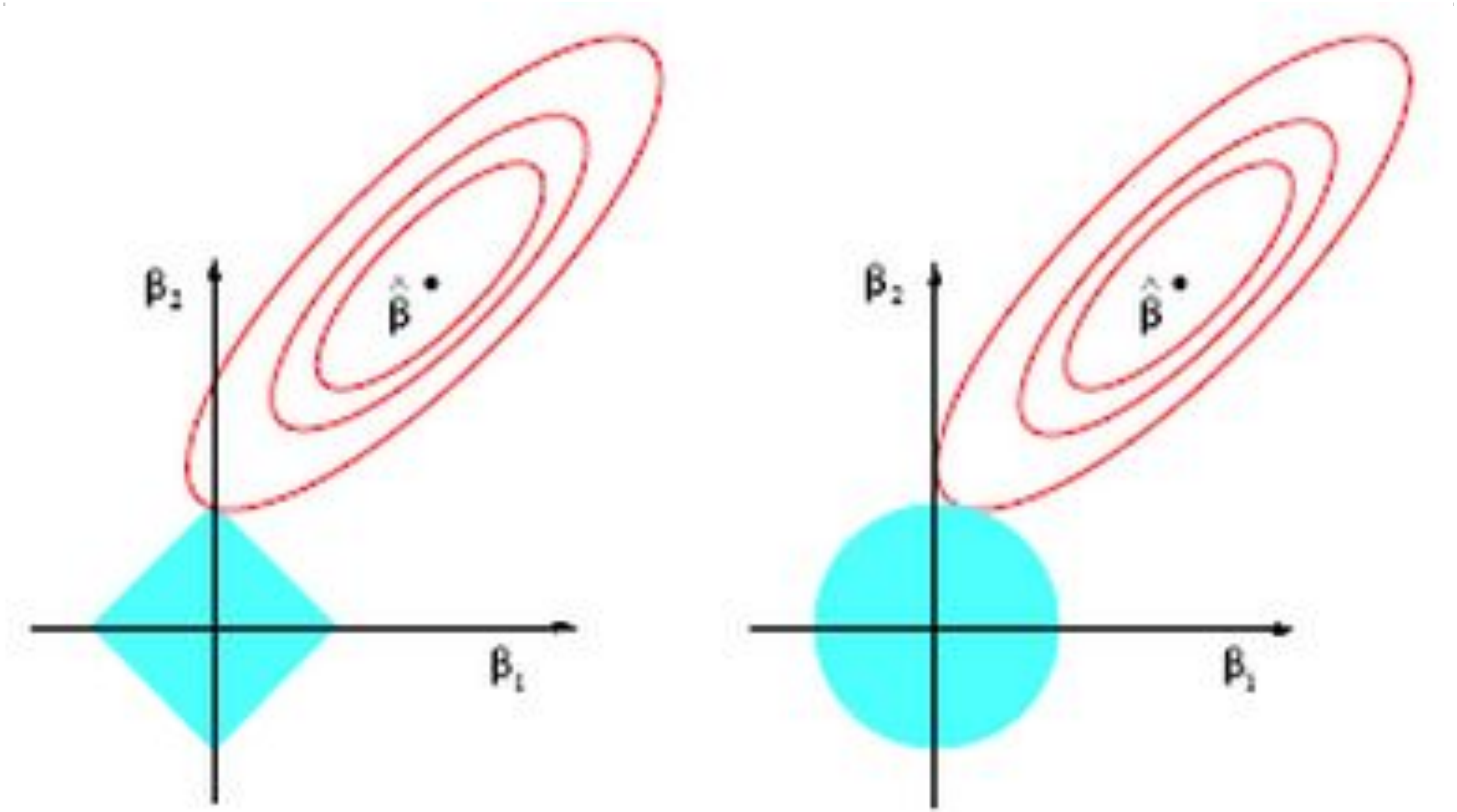
Lasso and Linear Programming

- Instead one needs to adapt an iterative procedure to find the solution
- Because the penalty function is piecewise linear, the classical technique of **linear-programming** is often used to find the solution numerically
- **linear programming** is a classical constrained optimization technique to maximize or minimize a multivariate linear function within a bounded or unbounded convex polytope



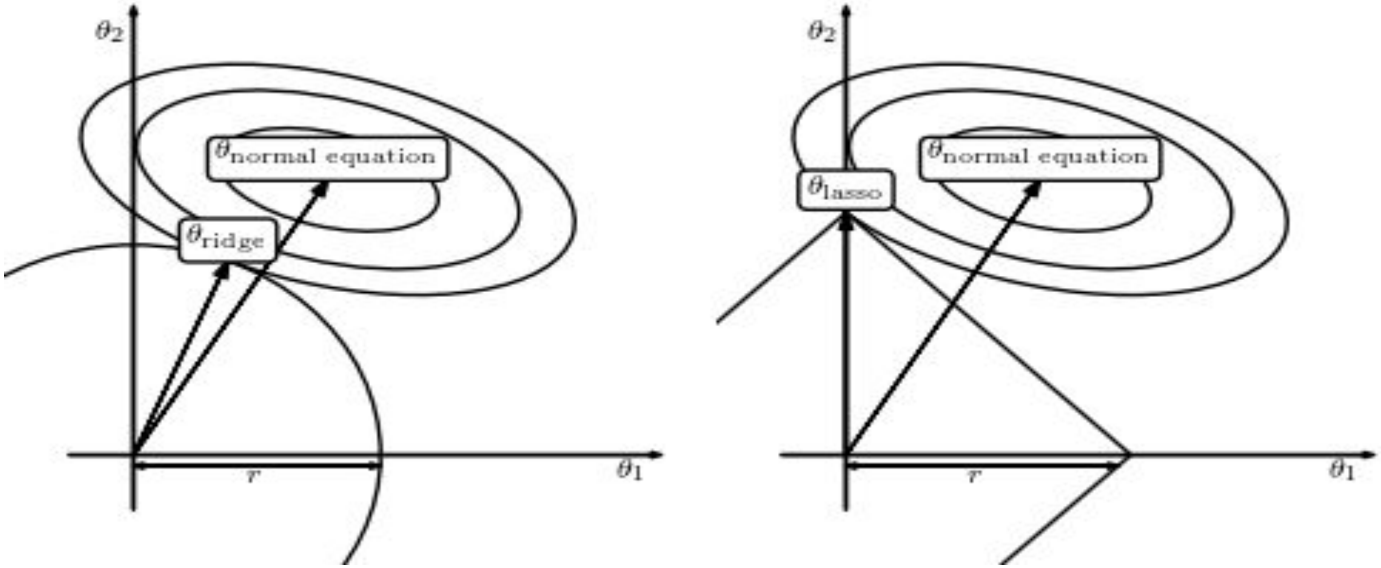
- The right hand plot is taken from wikipedia

Lasso (Left) vs Ridge (Right)



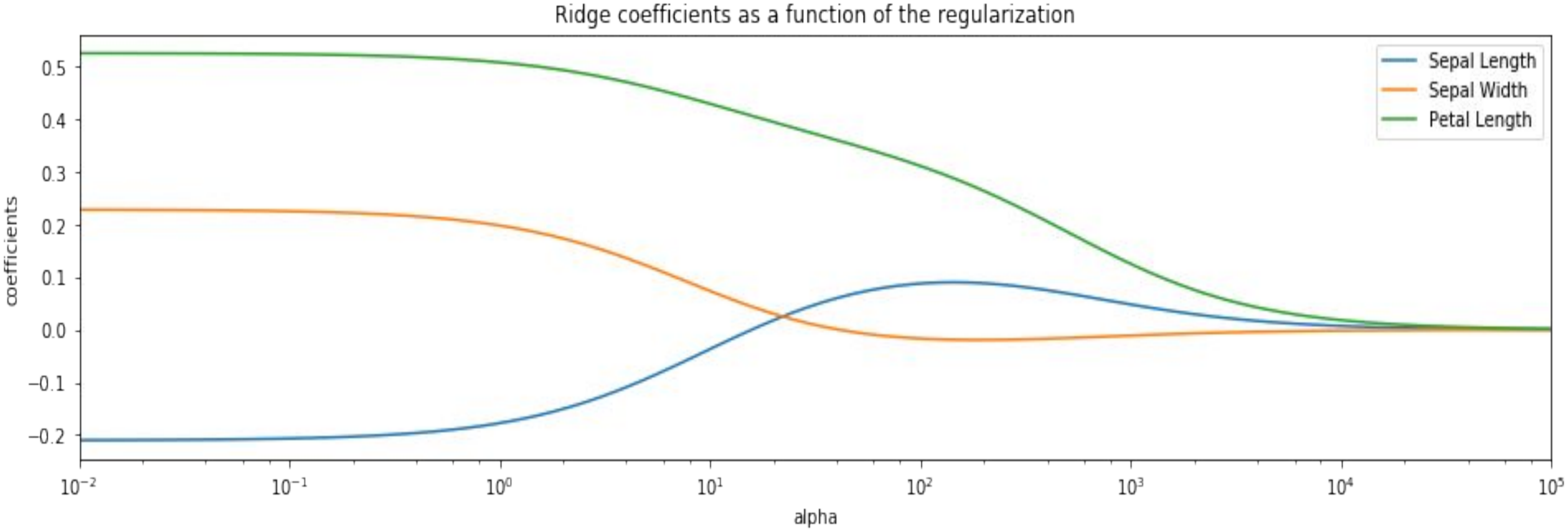
Ridge vs Lasso

- The concentric ellipses are the level curves of **MLR RSS**
- The circle and the square (interiors included) are the **Ridge** and **Lasso** constraints. The points where the boundaries of the constraints first touch the concentric ellipses (level curves of **RSS**) are the solutions of the penalized regressions



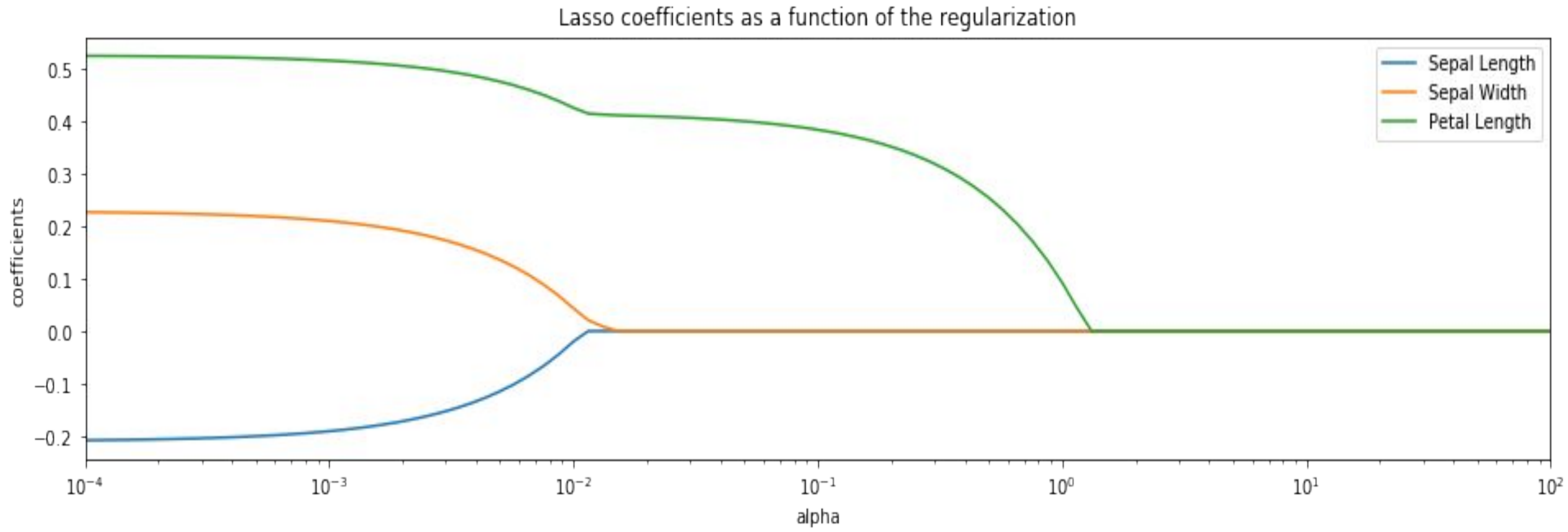
Ridge Coefficient Plot Illustration

- Suppose that we run a **Ridge/Lasso** regression of iris petal width on sepal length, sepal width, petal length. We observe how do these slope coefficients depending on lambda
- **Ridge** coefficients approach zero gradually but they remain nonzero for any finite lambda



Lasso Coefficient Plot Illustration

- The similar plot for Lasso regression
- The coefficients drop to exactly zero one by one. They all become zero at a large finite lambda



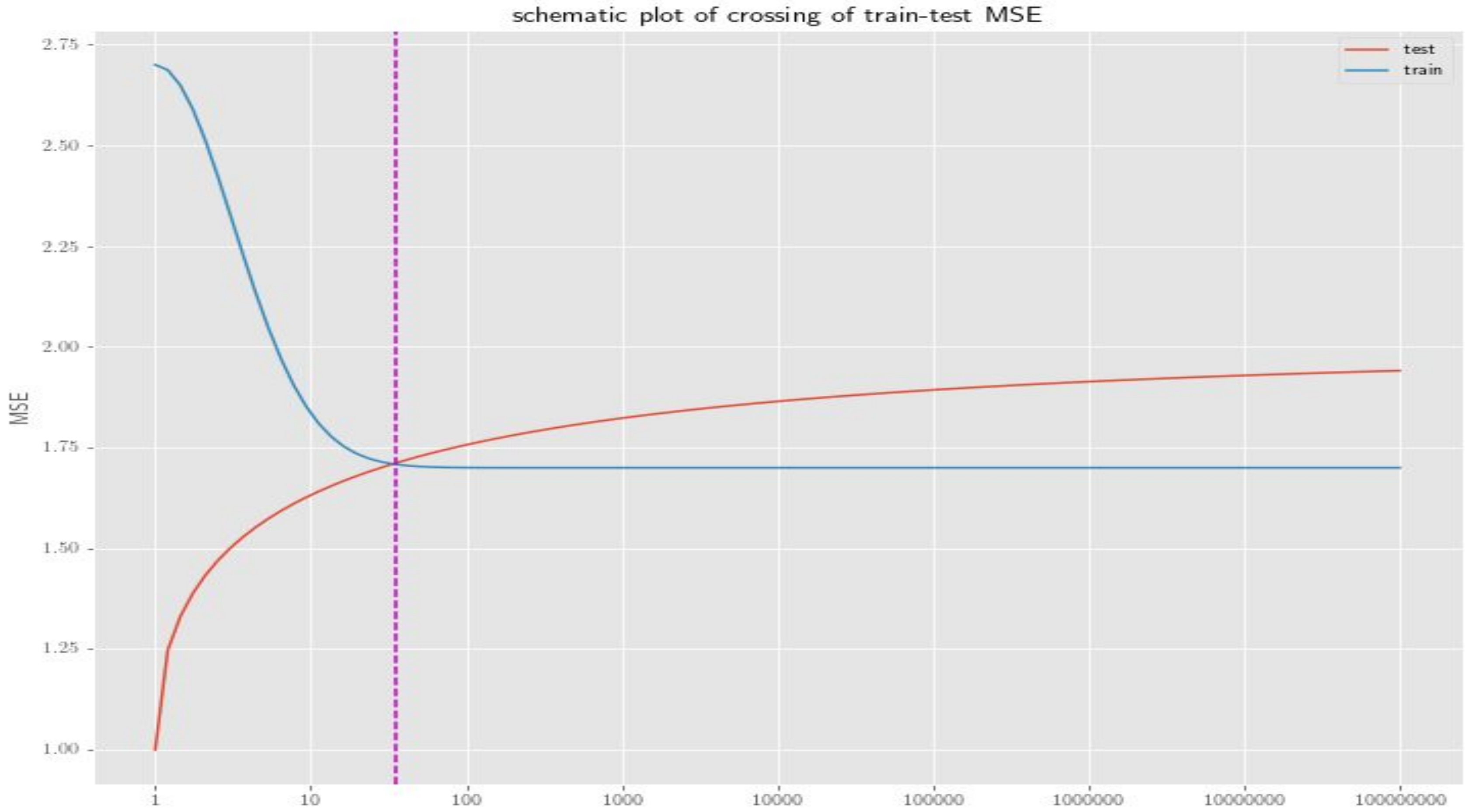
The Search of an Optimal Lambda

- The hyperparameter lambda induces a one parameter family of linear models which are biased
- Which lambda should we use?
- Intuitively a good lambda is the one whose associated model offers the least overfitting (the performance drop in handling new samples from the same distribution). This is exactly the purpose of introducing penalized (regularized) regression
- When there is no unseen new data available, we can set aside a portion of the data for testing purpose

The Crossing of Train and Test MSE

- We train the **Ridge/Lasso** model based on the remaining samples, which is usually called a training set
- When the original **MLR** model is overfitting, we will see train-set **MSE** significantly lower than test-set **MSE**
- On the other hand, when lambda approaches infinity, the null model underfits the data, which means train-set **MSE** can be lower than test-set **MSE**
- When we tune the hyperparameter lambda from 0 to a large number (a proxy of infinity), we are looking for a particular lambda such that train-set **MSE** is approximately equal to test-set **MSE**

Schematic Plot of MSE Crossing



Hyperparameter lambda

Penalized Regression and Feature Importance

- Because the features have been standardized before fitting a penalized linear model, their slope coefficients can be compared for their relative strength
- Thus one may use the magnitudes of these slopes as scores to rank the relative importance of features
- As we stretch lambda larger and larger, the relative importance of features may change
- For **Lasso** at a particular lambda, some of the feature slopes have dropped to zero identically. This suggests that these features have no effect on the penalized model. This enables us to use **Lasso** to perform **feature selection**

Pros and Cons of Penalized Regression

- By introducing an additional hyperparameter **lambda**, a fine-tuned penalized regression trades off a slight increase of biases for a significant drop of its model variance, which results in an improved accuracy

- On the other hand, the penalization procedure shrinks the slope coefficients to battle multicollinearity. This makes the penalized model less interpretable compared to **MLR**
- For a regular **MLR**, each individual slope represents the change of the target value per unit change of the feature, fixing all the other features. But this type of interpretation does not work for penalized regression

Applications of Penalization Technique

- The idea of penalization is rather useful in machine learning
- The same idea has been used in logistic regression/GLM, spline regression, in the design of recommendation systems, tree boosting (e.g. xgboost), neural network, ….., etc
- Fitting a support vector machine/regression model can be naturally formulated as minimizing a penalized nonlinear loss function
- Gradient boosting (discussed in the latter lecture) can be interpreted as an **implicit regularization** scheme of growing an ensemble of trees

A Remark on Elastic-Net

- As has been mentioned earlier, elastic-net is a hybrid of **Ridge** and **Lasso** regression, which has partial characteristic from both approaches
- **Elastic-net** penalization is a convex combination of **Ridge** and **Lasso**
- Its regularization term has the following form:

$$\lambda \cdot (1 - \rho) \cdot ||\vec{\beta}||_{L^2}^2 + \lambda \cdot \rho \cdot ||\vec{\beta}||_{L^1}$$

- As before **lambda** represents the regularization strength
- The new parameter **rho** controls the different mixtures of **Ridge** and **Lasso**
- **rho** = 1 degenerates to **Lasso**
- **rho** = 0 degenerates to **Ridge**
- The hyperparameters **rho** and **lambda** constitute a 2d family to tune