Lecture 7

Regression in Machine Learning

Haoyu Yue / yohaoyu@washington.edu
Ph.D. Student, Interdisciplinary Urban Design and Planning
University of Washington

RE 519 Real Estate Data Analytics and Visualization
Course Website: www.yuehaoyu.com/data-analytics-visualization/
Autumn 2025

Statistical Modeling: The Two Cultures

Classical Statistics/Econometrics vs Machine Learning

	Linear Regression under Classical Statistics	Linear Regression under Machine Learning
Goals	Understanding via explanation and inference	Prediction on unseen data
Assumptions	Linearity, independence, homoscedasticity, normality of errors, no multicollinearity or omitted variable…	Fewer assumptions
Coefficient	Our interests: interpretable parameters	The weight for prediction; can be biased if interpreted
Model Size and Design	Small model and intended to be interpretable; variables chosen by domain knowledge	Larger black-box model; initial variables should include more and will have automatic variable selection (e.g., regularization)
Model Evaluation	In-sample goodness-of-fit and inferential validity. E.g., p-value, standard errors, CI, R^2, AIC, BIC	Out-of-sample predictive performance. E.g., MSE, test error

Let's almost forget the assumptions…

Polynomial Regression

Format

Recall a simple linear model, which is always a straight line.

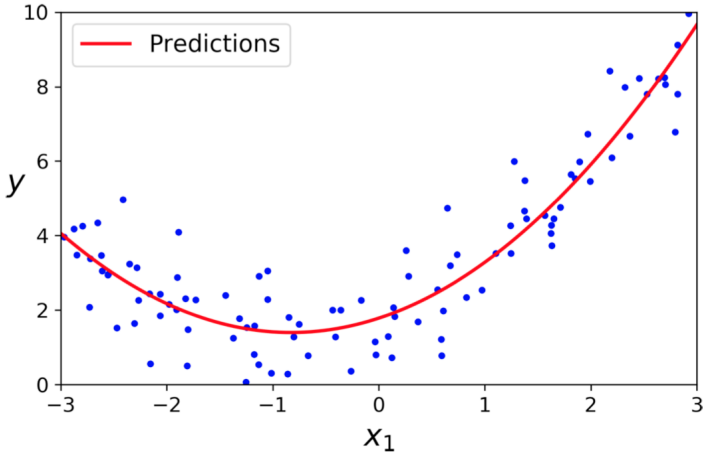
$$Y = \beta_0 + \beta_1 X + \varepsilon$$

Polynomial regression uses higher orders to fit nonlinear data. The model is still a linear model, as it is linear in the coefficients β.

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_k X^k + \epsilon$$

We will call X the features and Y as labels.

The example uses X_1^2 as one of the features and captures a better relationship. – We call this a *polynomial regression with degree 2*.



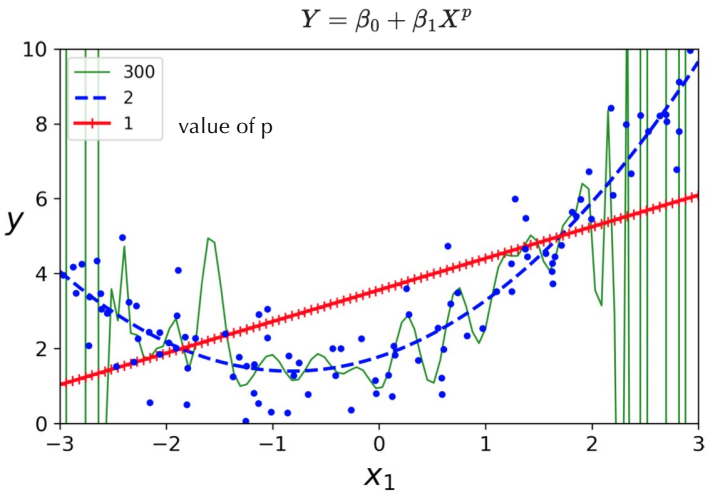
Source: Aurelien Geron. Hands-On Machine Learning with Scikit-Learn and TensorFlow.

Polynomial Regression

The Problem of Overfitting

- Which model is the best in terms of MSE?
- Which model is the best in terms of representing the true model?

Back to the purpose of machine learning, we are using a model to predict on unknown data. We care about the **generalization** ability of a model.

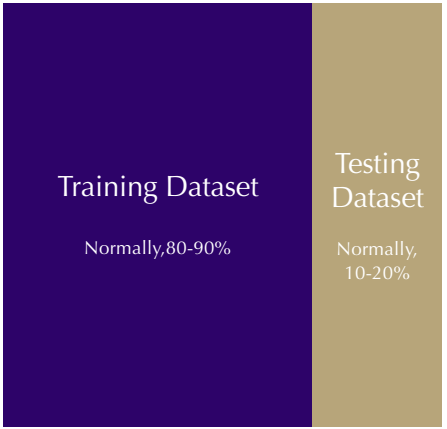


Source: Aurelien Geron. Hands-On Machine Learning with Scikit-Learn and TensorFlow.

Training and Testing

Split the Data into Training and Testing

To mimic the unseen/unknown data, we purposely split the whole dataset into training and testing data. Testing data will be used for testing and comparing models, and should **NOT be used anytime before**.



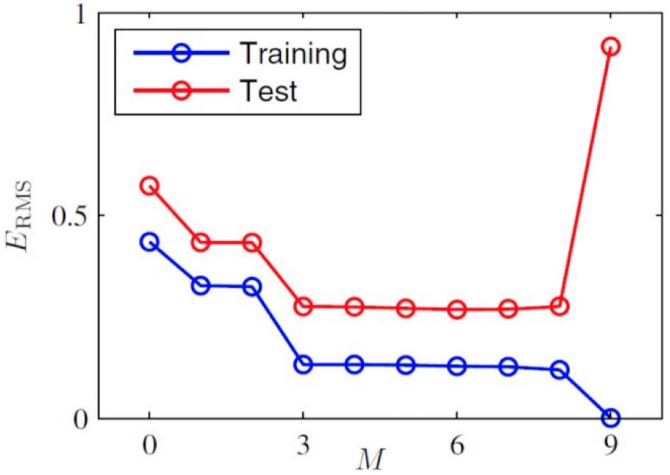
Training and Testing

Learning Curves

A direct way to compare the performance and make the model selection is by using learning curves.

We have order (model complexity) as the X axis and MSE (error function) as the Y axis, which shows the performance on both the training and testing datasets.

- **Overfitting**: good performance on the training dataset but bad performance on the testing dataset
- **Underfitting**: bad performance on both datasets



Source: Christopher M. Bishop. Pattern Recognition and Machine Learning.

Variance-Bias Tradeoff

Proof and Example (Optional)

Recall, the best predictor in terms of MSE: $E[Y|X]$

But it is too ambitious, so we use a model:

Find a model to minimize ↑ Model ↑

$$\hat{f} = \text{argmin}_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2$$

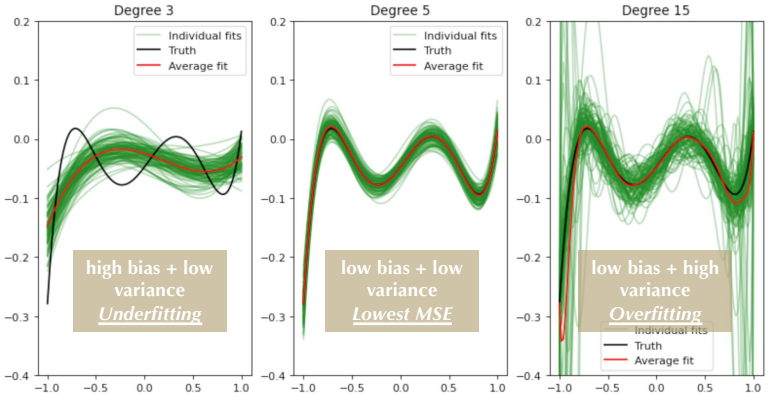
We can decompose MSE to get:

Prediction based on model ↑ $E[Y|X]$ ↑

$$\text{MSE} = \underbrace{\mathbb{E}\left[(\hat{f}(x) - \mathbb{E}[\hat{f}(x)])^2\right]}_{\text{variance}} + \underbrace{(\mathbb{E}[\hat{f}(x)] - f(x))^2}_{\text{bias}^2}$$

↓ Variance of your predictions

↓ The difference between the model's prediction and the best MSE predictor (or truth)



Source: https://courses.cs.washington.edu/courses/cse446/24sp/schedule/lecture_06/demo_tradeoff.html

Takeaway: A model cannot be either too complicated or too simple.

Regularization

In order to fix the problem of multicollinearity and overfitting, and try to find proper features (X)

Ridge Regression

In a normal linear regression, we are finding parameters to minimize MSE:

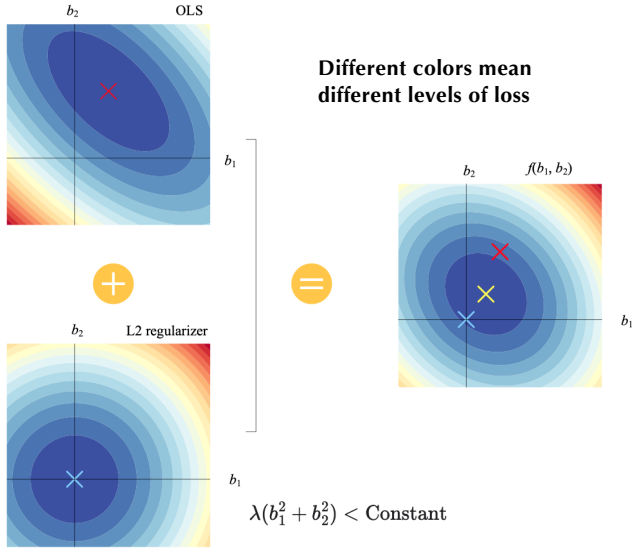
$$\hat{\beta} = \text{argmin}_\beta \sum_i (y_i - \hat{y}_i)^2$$

A model will be overfitting because it is too complex. In a ridge regression, we add some penalty terms:

$$\hat{\beta} = \text{argmin}_\beta \sum_i (y_i - \hat{y}_i)^2 + \lambda \sum_j \beta_j^2$$

penalty parameter, decided by us

We are forcing β closer to 0 by penalizing for the size of β. None of β will be changed to 0.



Source: Visualizations for Machine Learning (Iris Series)

Regularization

In order to fix the problem of multicollinearity and overfitting, and try to find proper features (X)

Lasso Regression

In a normal linear regression, we are finding parameters to minimize MSE:

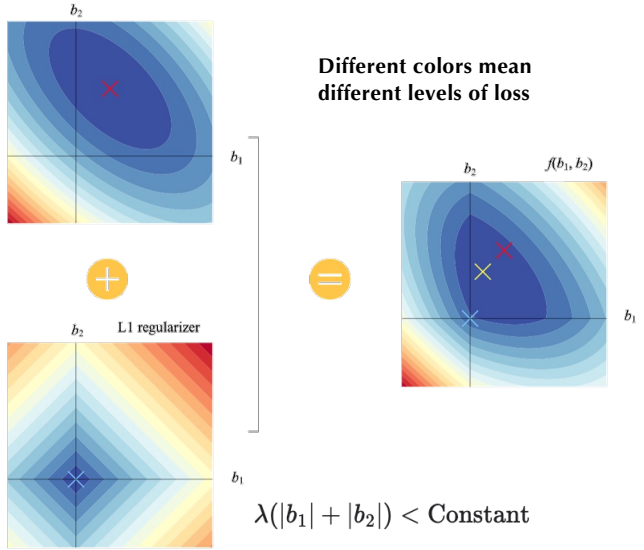
$$\hat{\beta} = \text{argmin}_\beta \sum_i (y_i - \hat{y}_i)^2$$

A model will be overfitting because it is too complex. In a lasso regression, we add some penalty terms:

$$\hat{\beta} = \text{argmin}_\beta \sum_i (y_i - \hat{y}_i)^2 + \lambda \sum_j |\beta_j|$$

penalty parameter, decided by us

We are forcing β closer to 0 by penalizing for the size of β. β can be changed to 0.



Source: Visualizations for Machine Learning (Iris Series)

Regularization

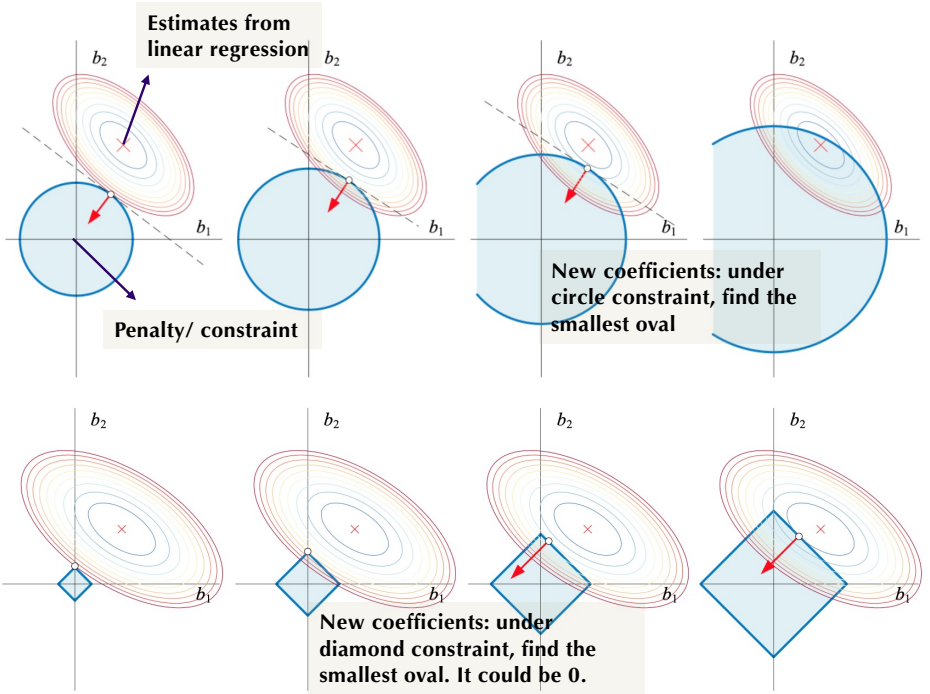
Ridge vs Lasso

We have 2 features: b_1 and b_2.

Different λ changes how strong the regularization penalty is (the size of the circle/diamond).

Ridge uses a circular constraint, so it shrinks coefficients smoothly, while Lasso uses a diamond-shaped constraint with sharp corners, making coefficients more likely to be exactly zero.

We call λ a hyperparameter. How to decide λ?



Source: Visualizations for Machine Learning (Iris Series)

Cross Validation

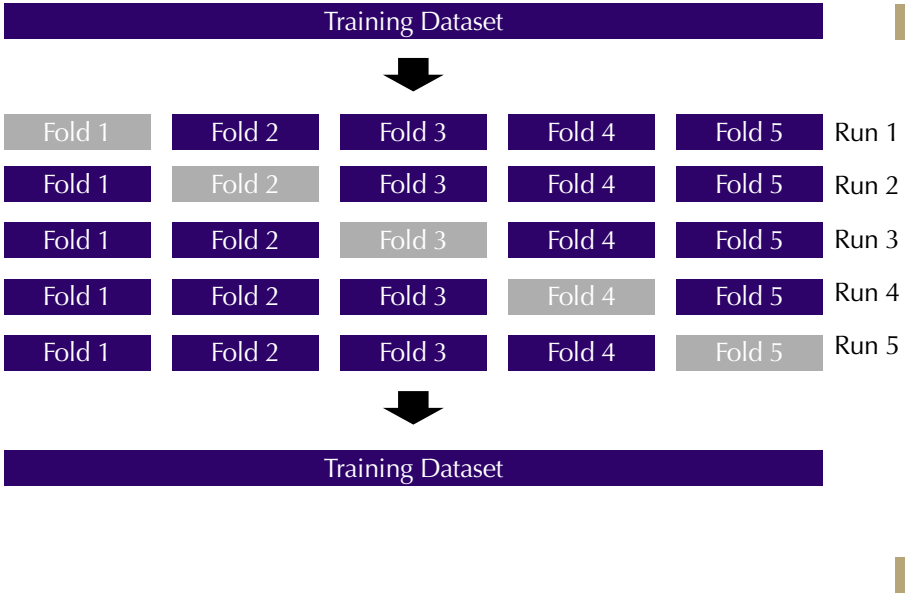
Cross Validation (K-Fold Methods)

Get a list of λ and train for 5 runs using training data; validate using validation data.

Calculate the average of the performance metric (e.g., MSE) on 5 validation datasets.

Select the best λ and train the model using all training data.

Test the model using the testing dataset.



Thank you!

Haoyu Yue / yohaoyu@washington.edu
Ph.D. Student, Interdisciplinary Urban Design and Planning
University of Washington

RE 519 Real Estate Data Analysis and Visualization
Course Website: www.yuehaoyu.com/data-analytics-visualization/
Autumn 2025

The course was developed based on previous instructors: Christian Phillips, Siman Ning, Feiyang Sun
Cover page credits: Visax