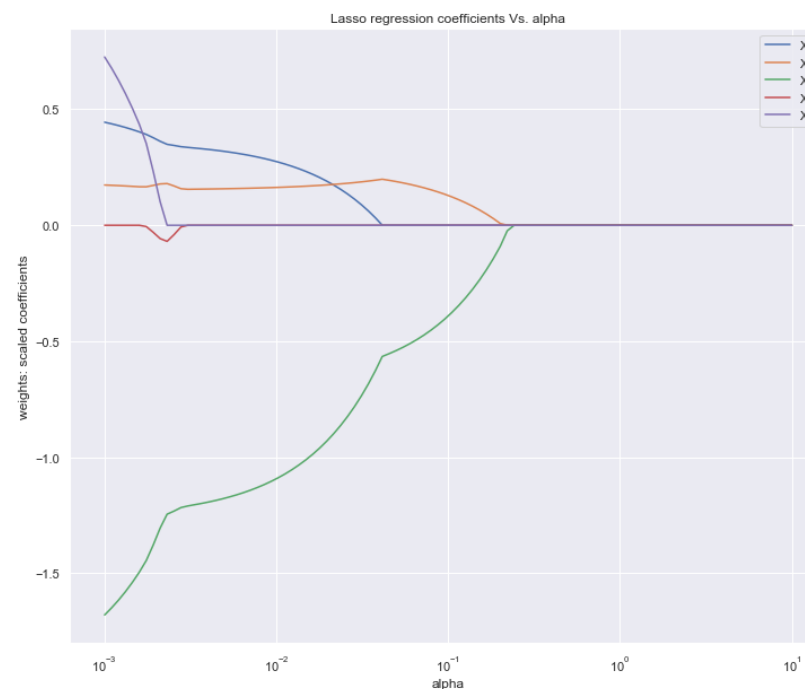
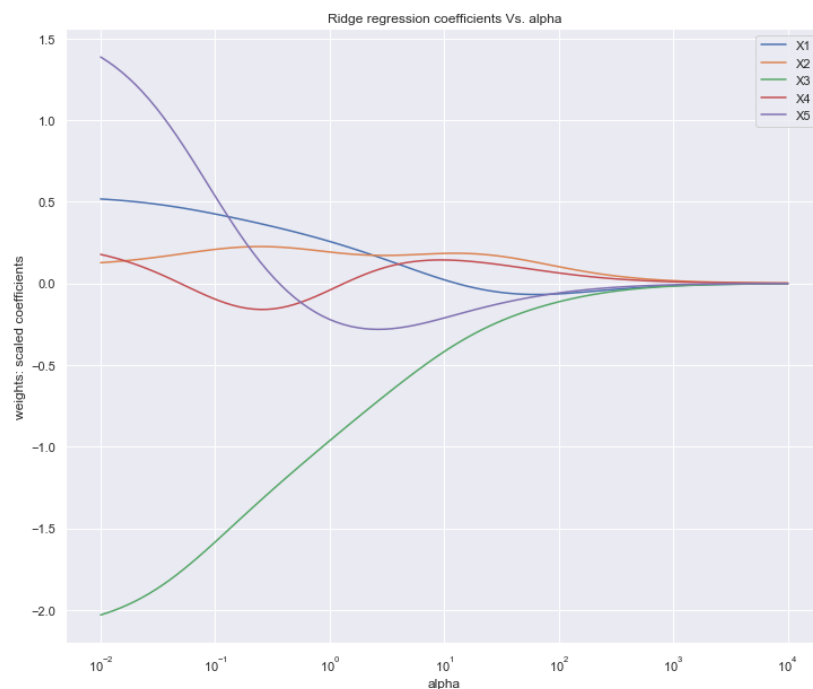




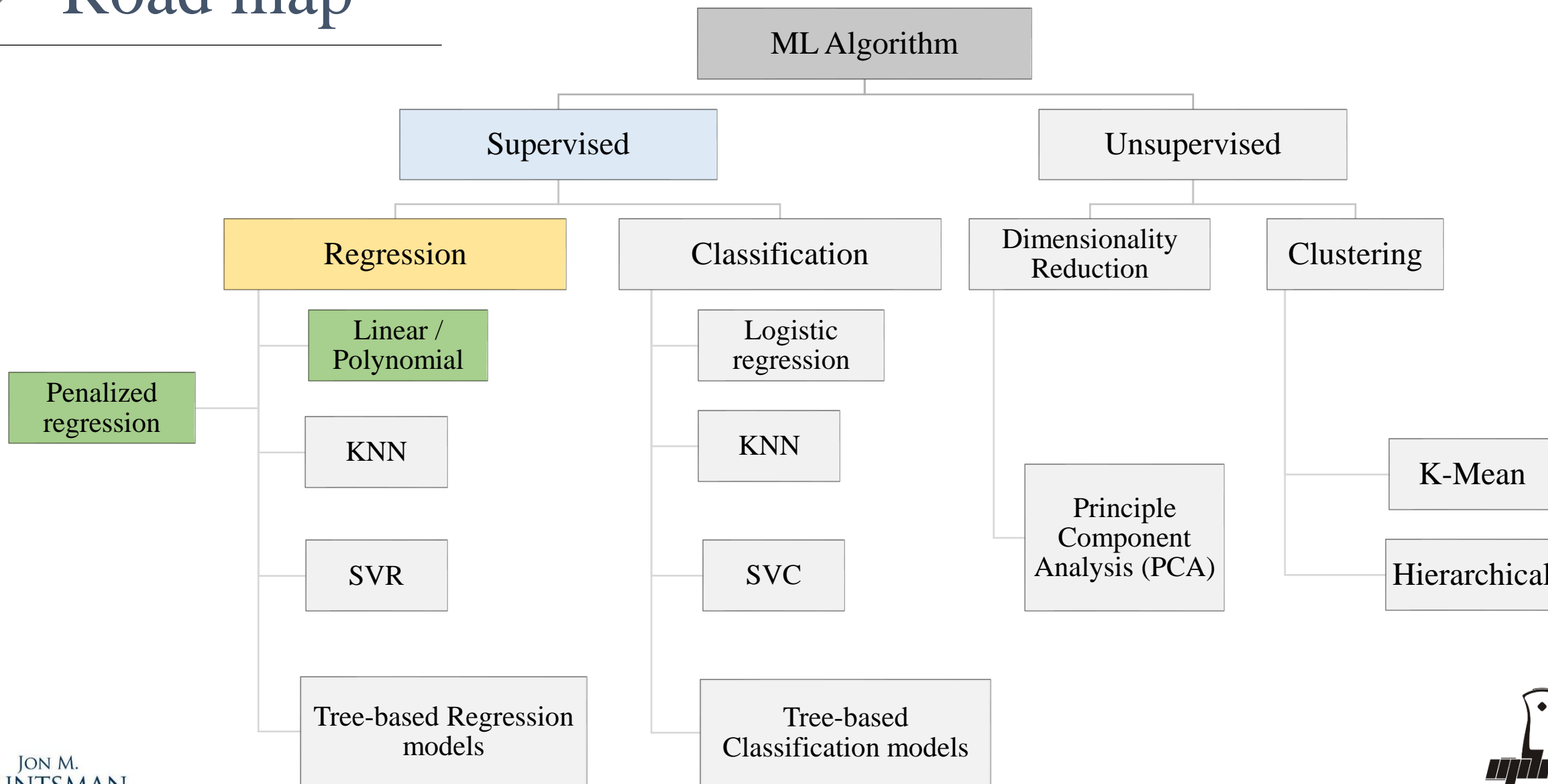
Module 6 - Penalized Regressions

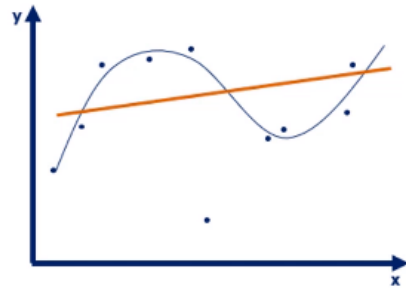
Ridge, LASSO, Elastic Net





Road map

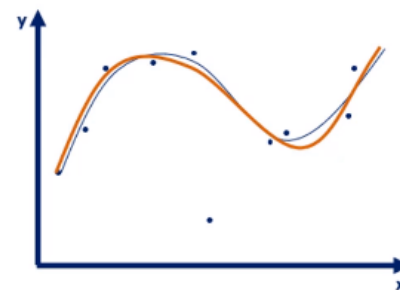




underfit

- Does not fit the training set well

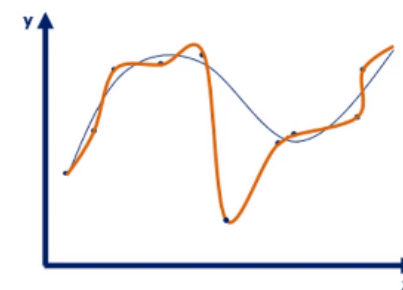
high bias



just right

- Fits training set pretty well

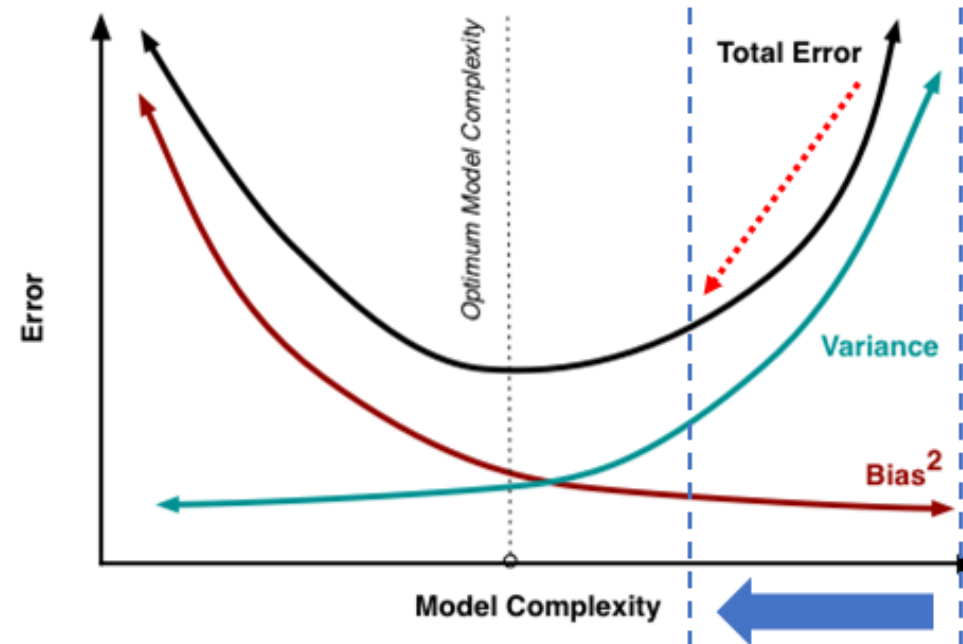
generalization



overfit

- Fits the training set extremely well

high variance



Regularization / Penalized regression

➔ Norms

- In mathematics, the **norm** of a vector is its **length**.
- In regression analysis, to fit our linear model, we need a measure of **mismatch**!
- Our vector is error at each training data. **We want to measure the length of error!**

- **L1** norm: Least absolute errors

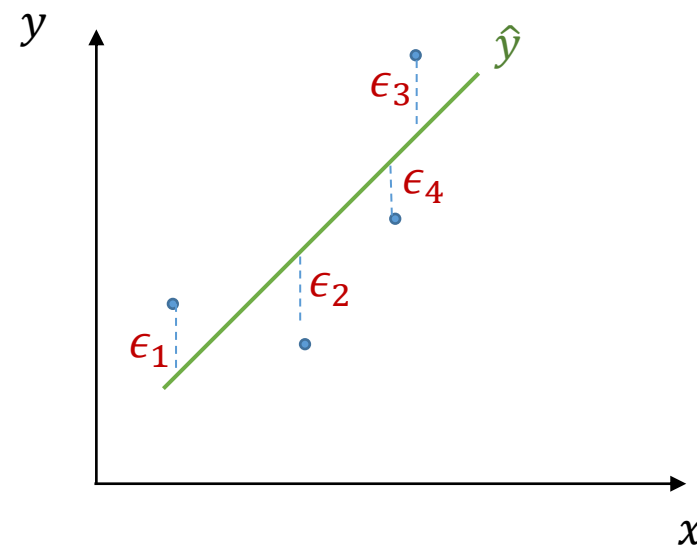
Manhattan norm

$$L^1 = \sum_i |\epsilon_i|$$

- **L2** norm: Least squares

Euclidean norm

$$L^2 = \sum_i (\epsilon_i)^2$$

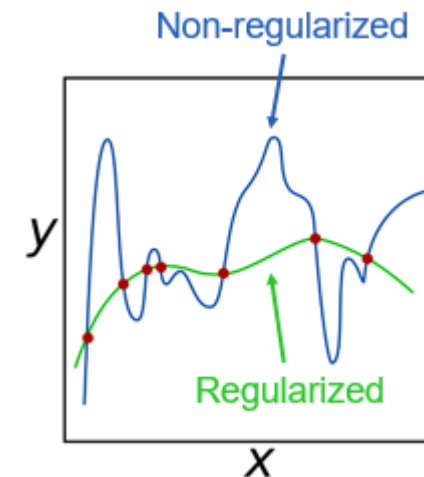




Regularization

- ❑ In machine learning there are often **many features** (usually **correlated** with each other). This can lead to **overfitting** and models that are **unnecessarily complex**.
- ❑ **Regularization** force the learning algorithm to build a **less complex model**. In practice, that often leads to **slightly** higher bias but **significantly** reduces the variance.
- ✓ The two most widely used types of regularization are called **L1** and **L2** regularization. The idea is quite simple. To create a regularized model, we modify the loss function by adding a penalizing term whose value is higher when the model is more complex.

$$\text{Min}_{w,b} (\text{MSE} + \text{penalty}) = \text{Min} \left[\underbrace{\frac{1}{N} \sum_{i=1}^N \left(y_i - f_{w,b}(X_i) \right)^2}_{\text{(fit data)}} + \underbrace{\text{penalty}(w)}_{\text{(regularize)}} \right]$$



→ Penalized regression

$$\text{Min}_{w,b} (\text{MSE} + \text{penalty}) = \text{Min} \left[\frac{1}{N} \sum_{i=1}^N (y_i - f_{w,b}(X_i))^2 + \text{penalty}(w) \right]$$

- **Penalized regression** is useful for reducing a large number of features to a manageable set and for making good predictions especially where features are correlated (i.e., when classical linear regression breaks down).
- **Penalized regression** can be used to avoid **overfitting**.
- To use the penalized regression, we need to first **standardize the features**. This will allow us to compare the magnitudes of regression coefficients for the feature variables.

- 1) Ridge regression
- 2) LASSO regression
- 3) Elastic Net regression

The only
difference is
in the penalty
term

Part I

Ridge Regression

➔ 1) Ridge regression

$$\begin{aligned} \text{Min}_{w,b} (\text{MSE} + \text{penalty}) &= \text{Min} \left[\frac{1}{N} \sum_{i=1}^N \left(y_i - f_{w,b}(X_i) \right)^2 + \text{penalty}(w) \right] \\ &= \text{Min} \left[\frac{1}{N} \sum_{i=1}^N \left(y_i - f_{w,b}(X_i) \right)^2 + \lambda \sum_{j=1}^D w_j^2 \right] \end{aligned}$$

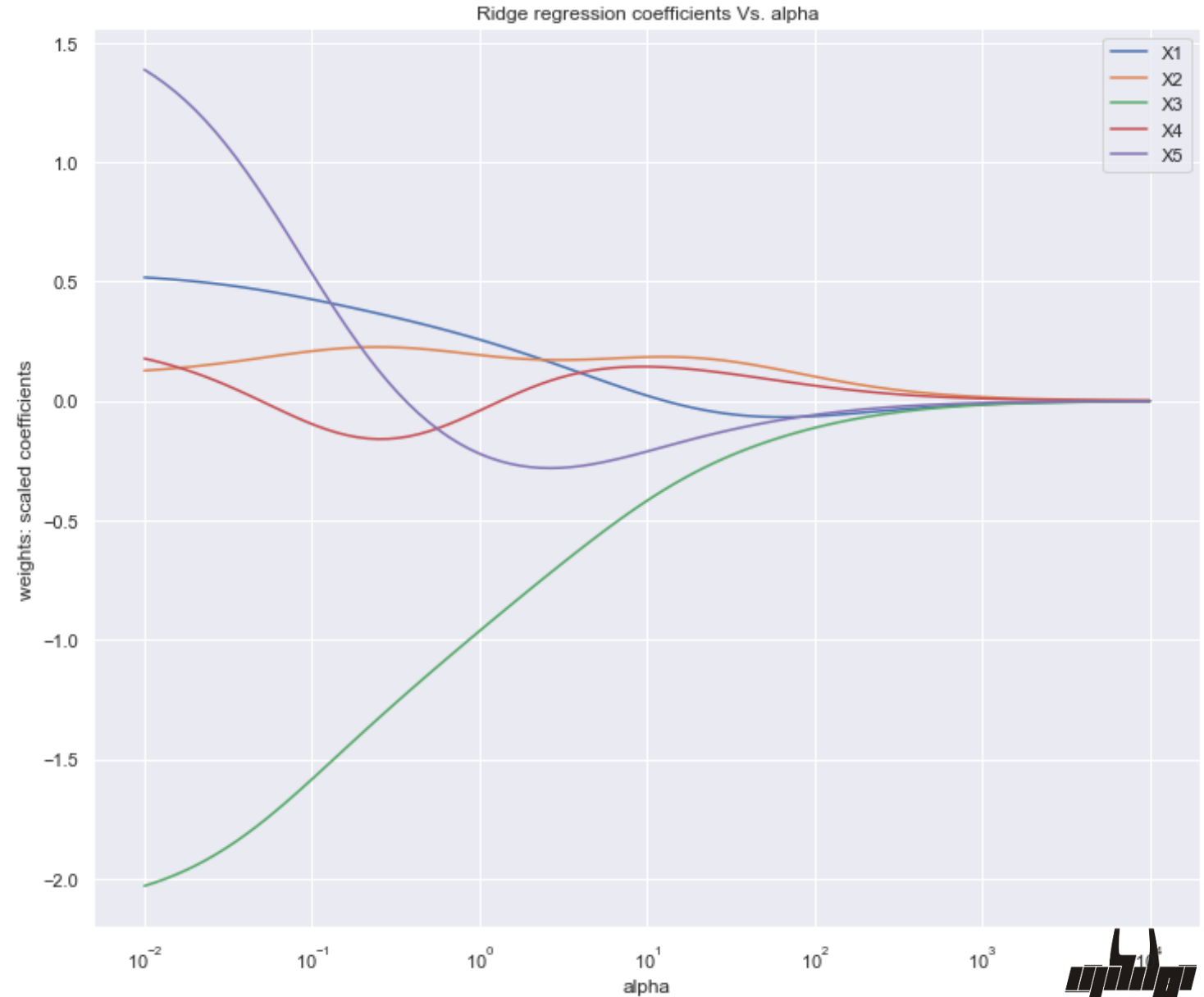
- Ridge regression uses **L2** norm.
- The shrinkage penalty has the effect of shrinking the estimates of w_j towards zero.
- The tuning parameter λ serves to control the relative impact of the penalty term on the regression coefficient estimates.
- Selecting a good value for λ is critical; cross-validation is used for this.
- It is best to apply ridge regression after variable **standardization**.

The true model is:

$$y = f(x) = x + 2x^2 - 3x^3 + \epsilon$$

Imposed functional form:

$$\hat{y} = b + w_1x + w_2x^2 + w_3x^3 + w_4x^4 + w_5x^5$$



Part II

LASSO Regression

→ 2) LASSO regression

$$\begin{aligned} \text{Min}_{w,b} (\text{MSE} + \text{penalty}) &= \text{Min} \left[\frac{1}{N} \sum_{i=1}^N \left(y_i - f_{w,b}(X_i) \right)^2 + \text{penalty}(w) \right] \\ &= \text{Min} \left[\frac{1}{N} \sum_{i=1}^N \left(y_i - f_{w,b}(X_i) \right)^2 + \lambda \sum_{j=1}^D |w_j| \right] \end{aligned}$$

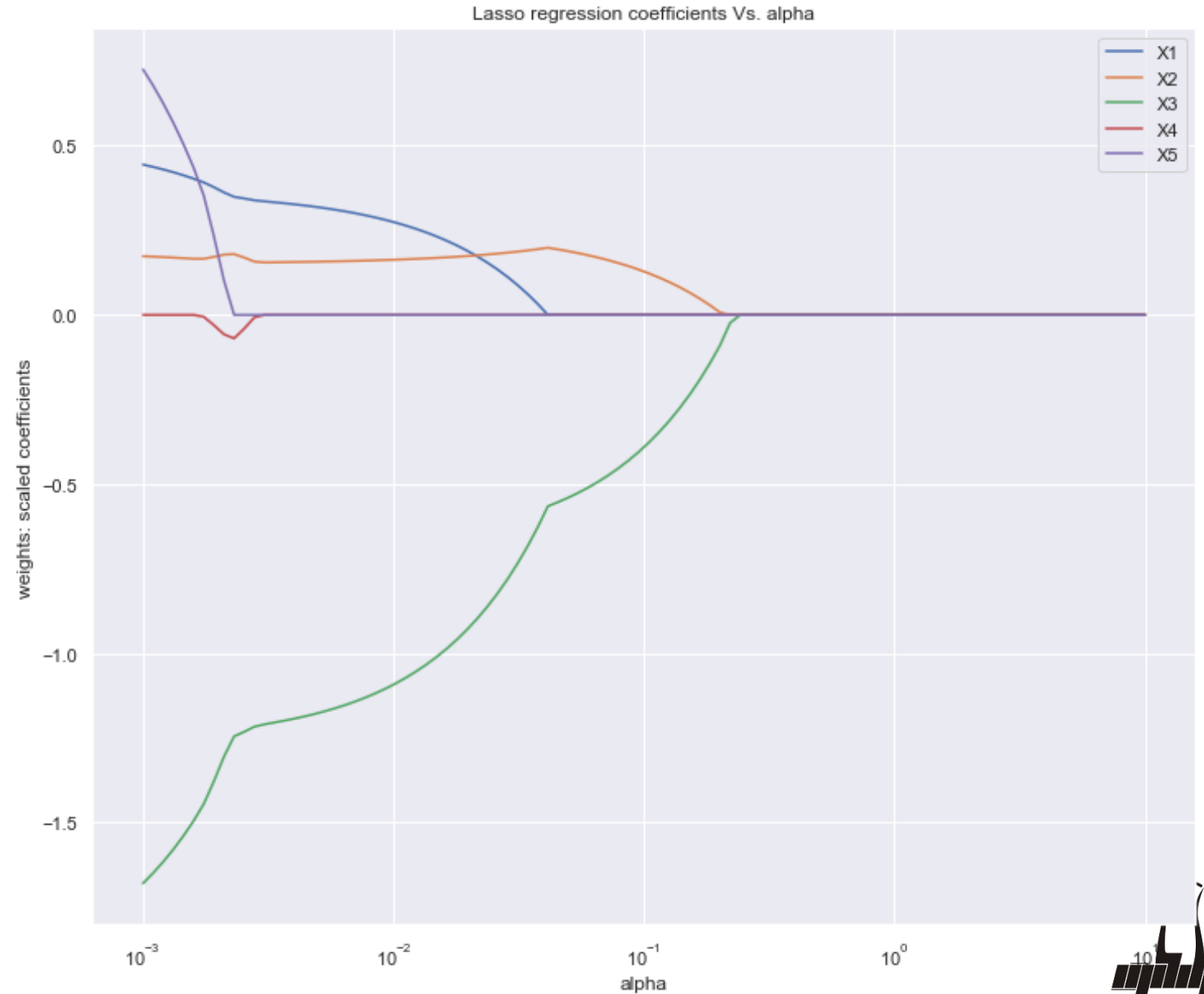
- LASSO stands for “Least Absolute Shrinkage and Selection Operator”
- LASSO regression uses **L1** norm.
- LASSO eliminates the least important features from the model, it automatically performs a type of **feature selection**.
- Selecting a good value for λ is critical; cross-validation is used for this.
- It is best to apply LASSO regression after variable **standardization**.

The true model is:

$$y = f(x) = x + 2x^2 - 3x^3 + \epsilon$$

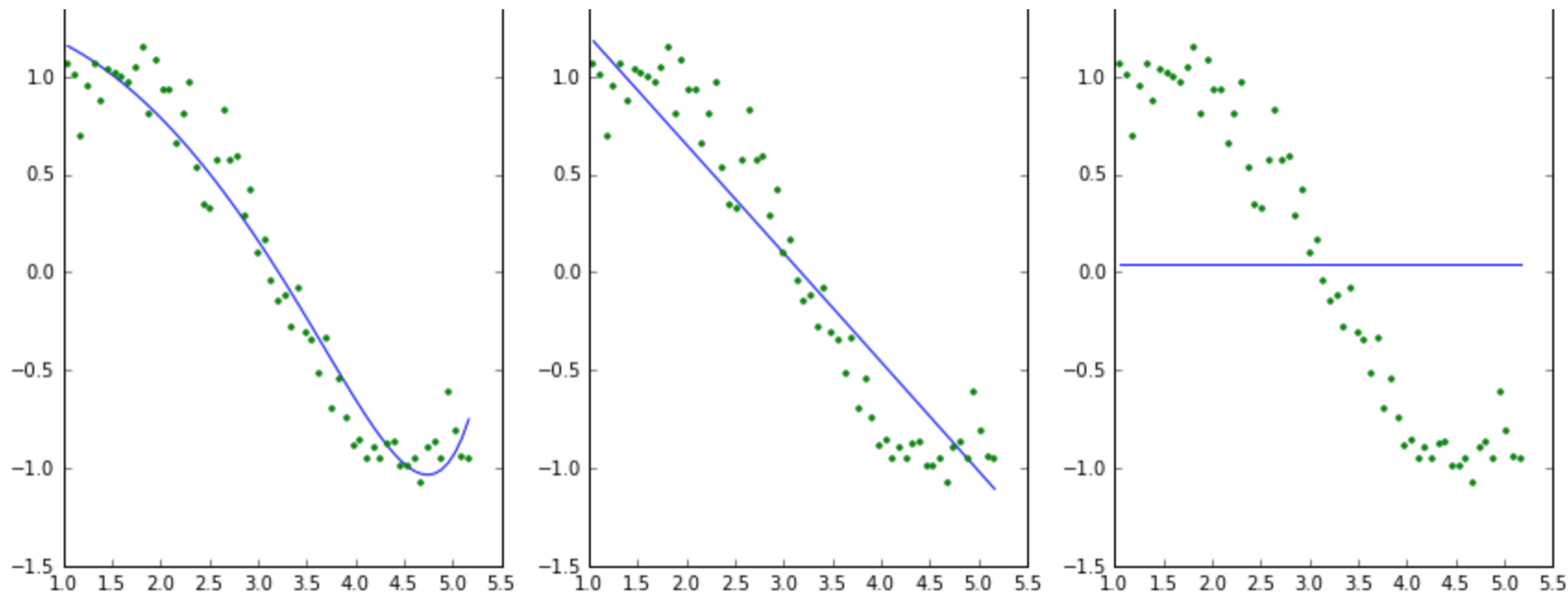
Imposed functional form:

$$\hat{y} = w_1x + w_2x^2 + w_3x^3 + w_4x^4 + w_5x^5$$

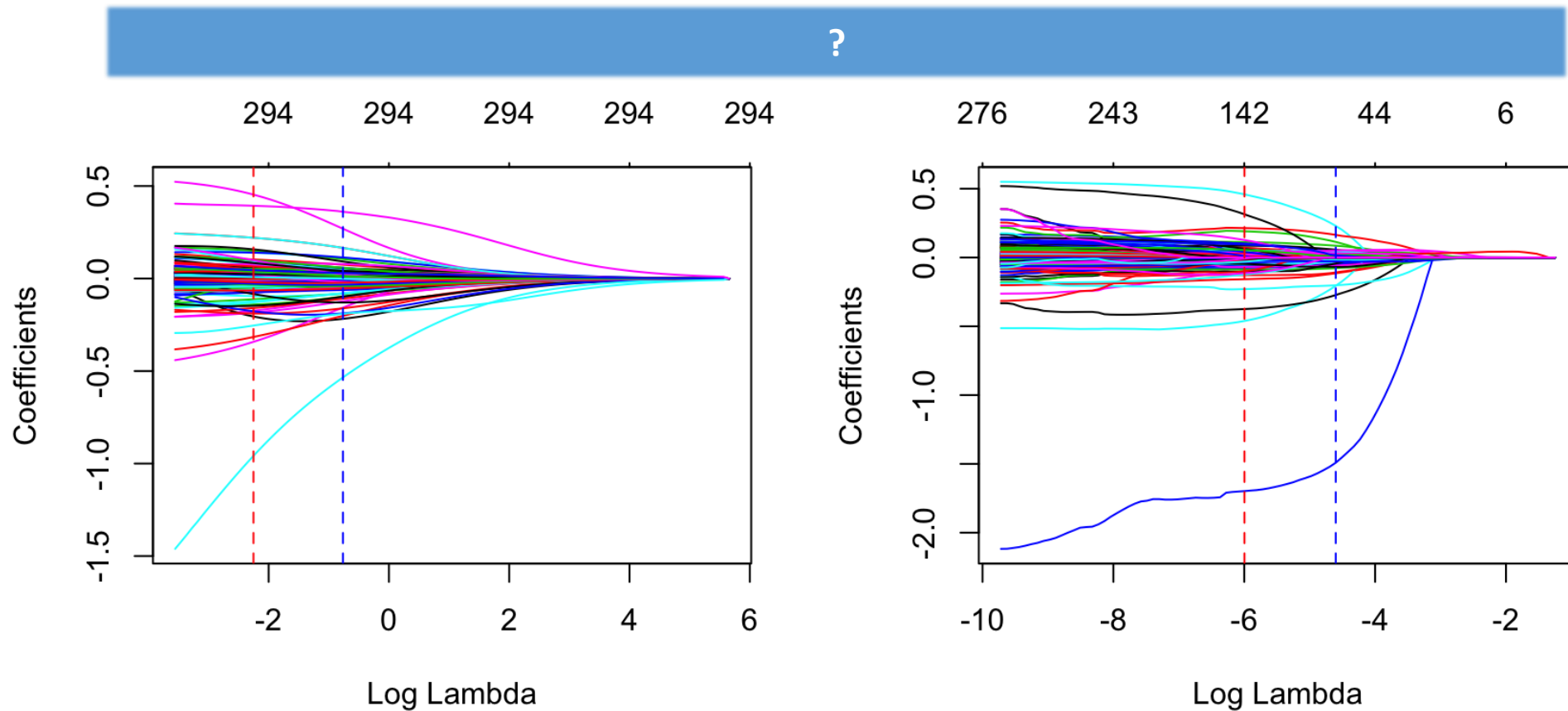


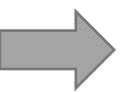
➔ Ridge and LASSO vs Lambda

As λ increases, the model becomes simpler



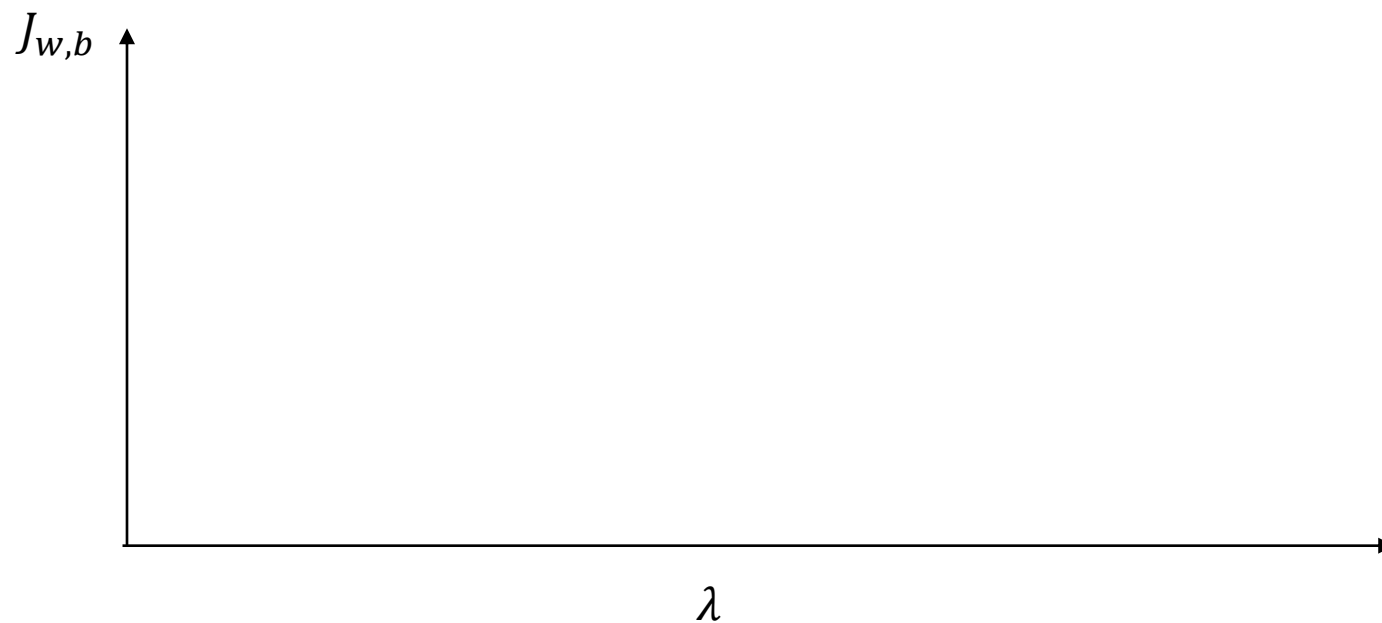
Question: Ridge vs LASSO?





Question: Cost function vs Lambda

$$J_{w,b} (\text{MSE} + \text{penalty}) = \frac{1}{N} \sum_{i=1}^N \left(y_i - f_{w,b}(X_i) \right)^2 + \lambda \sum_{j=1}^D |w_j|$$



Part III

Elastic Net Regression

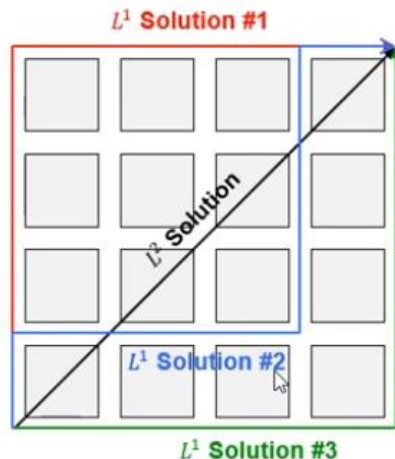
➔ 3) Elastic Net Regression

$$\begin{aligned} \text{Min}_{w,b} (\text{MSE} + \text{penalty}) &= \text{Min} \left[\frac{1}{N} \sum_{i=1}^N \left(y_i - f_{w,b}(X_i) \right)^2 + \text{penalty}(w) \right] \\ &= \text{Min} \left[\frac{1}{N} \sum_{i=1}^N \left(y_i - f_{w,b}(X_i) \right)^2 + \lambda_1 \sum_{j=1}^D |w_j| + \lambda_2 \sum_{j=1}^D w_j^2 \right] \end{aligned}$$

- In LASSO some weights are reduced to zero, but others may be quite large. In Ridge, weights are small in magnitude, but they are not reduced to zero.
- In Elastic Net, we may be able to get the **best of both worlds** by making some weights zero while reducing the magnitude of the others.

→ Ridge vs LASSO vs Elastic Net

Property	Ridge	LASSO	Elastic Net
Can shrink the coefficient estimate toward zero?			
Can include all the features in the model even with large λ ?			
Can force some of the coefficient estimates to be exactly = 0? Hence, can be used for <u>feature selection</u> ? Or <u>sparse output</u> ? More explainable?			
Is robust : resistant to <u>outliers</u> ?			
No Analytical solution i.e., requires gradient descent?			
Always unique solution?			





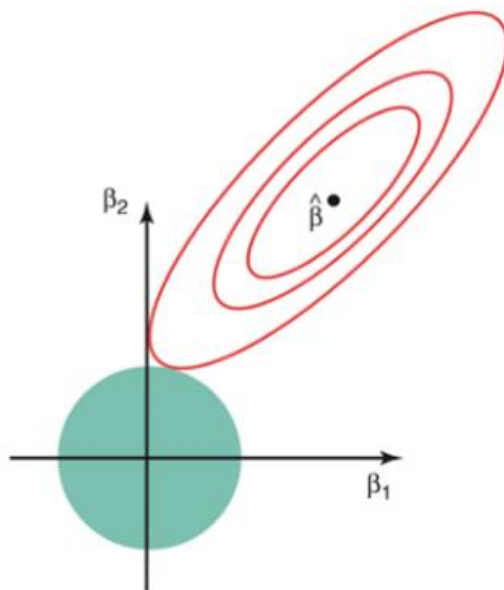
Appendix Behind the Scenes!



➔ Ridge regression, behind the scene!

Why the ridge regression shrinks the estimates of coefficients towards zero and NOT exactly zero? (why ridge regression cannot be used for feature selection?)

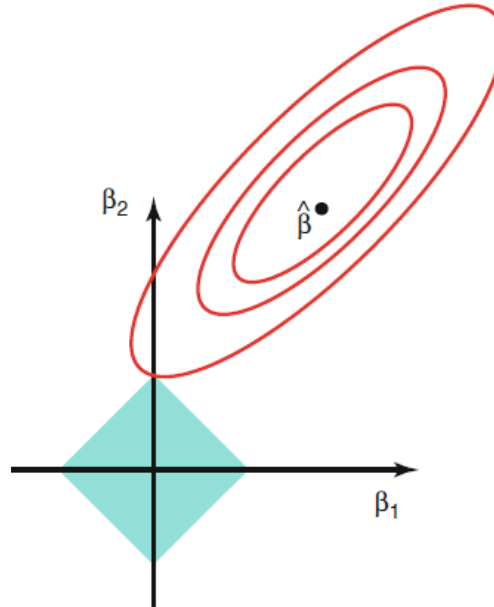
$$\underset{\beta}{\text{minimize}} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 \quad \text{subject to} \quad \sum_{j=1}^p \beta_j^2 \leq s,$$



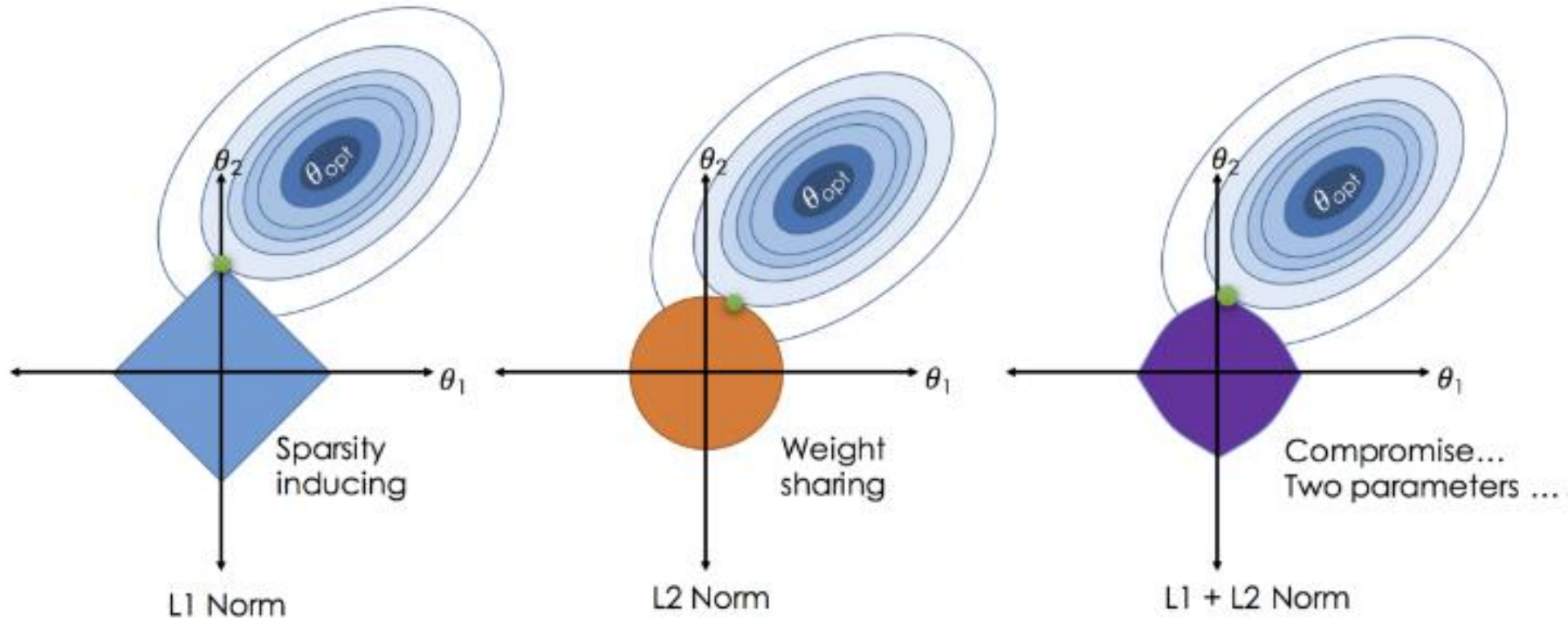
➔ LASSO regression, behind the scene?

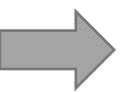
Why is it that the lasso, unlike ridge regression, results in coefficient estimates that are exactly equal to zero?

$$\underset{\beta}{\text{minimize}} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 \quad \text{subject to} \quad \sum_{j=1}^p |\beta_j| \leq s$$



➔ LASSO vs Ridge vs Elastic Net, behind the scene?





Class Modules

- Module 1- Introduction to Deep Learning
- Module 2- Setting up Machine Learning Environment
- Module 3- Linear Regression (Econometrics approach)
- Module 4- Machine Learning Fundamentals
- Module 5- Linear Regression (Machine Learning approach)
- **Module 6- Penalized Regression (Ridge, LASSO, Elastic Net)**
- Module 7- Logistic Regression
- Module 8- K-Nearest Neighbors (KNN)
- Module 9- Classification and Regression Trees (CART)
- Module 10- Bagging and Boosting
- Module 11- Dimensionality Reduction (PCA)
- Module 12- Clustering (KMeans – Hierarchical)





Class Modules

- ✓ Module 1- Introduction to Deep Learning
- ✓ Module 2- Setting up Machine Learning Environment
- ✓ Module 3- Linear Regression (Econometrics approach)
- ✓ Module 4- Machine Learning Fundamentals
- ✓ Module 5- Linear Regression (Machine Learning approach)
- ✓ Module 6- Penalized Regression (Ridge, LASSO, Elastic Net)
- Module 7- Logistic Regression
- Module 8- K-Nearest Neighbors (KNN)
- Module 9- Classification and Regression Trees (CART)
- Module 10- Bagging and Boosting
- Module 11- Dimensionality Reduction (PCA)
- Module 12- Clustering (KMeans – Hierarchical)

