Deep Learning III: Evaluation

Swati Mishra

Applications of Machine Learning (4AL3)

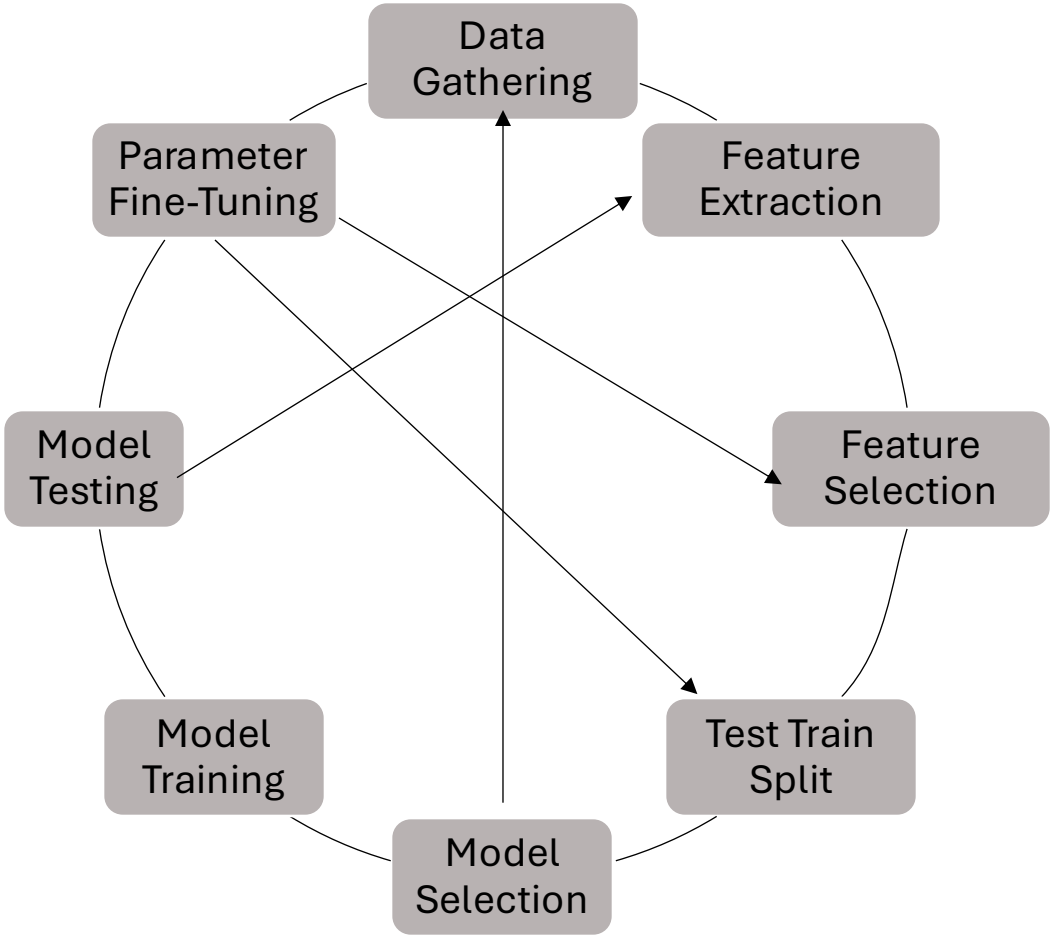
Fall 2024

McMaster University
ENGINEERING

Review

- Neural Network as a Computation Graph
- Backpropagation, Chain Rule
- Loss Functions
- Training and Graph traversal (Forward and Backward Pass)

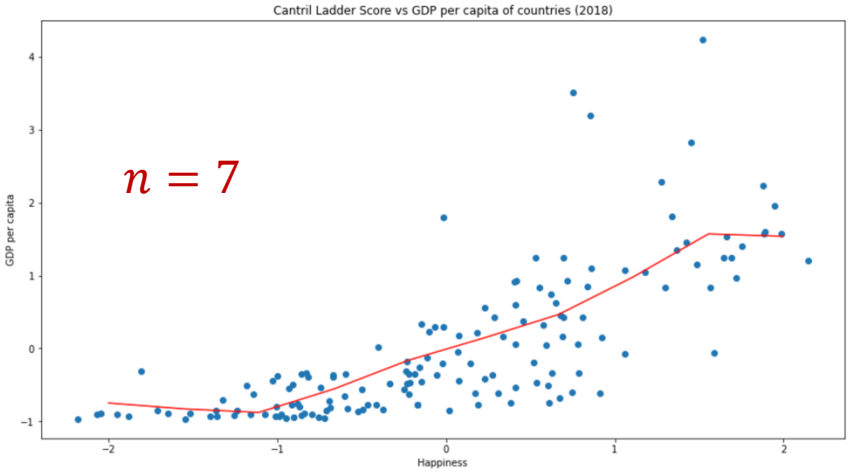
Model Building Pipeline

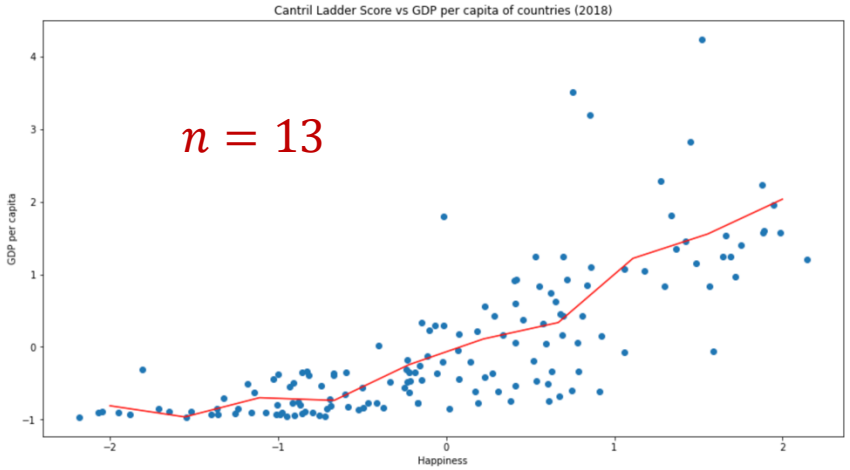


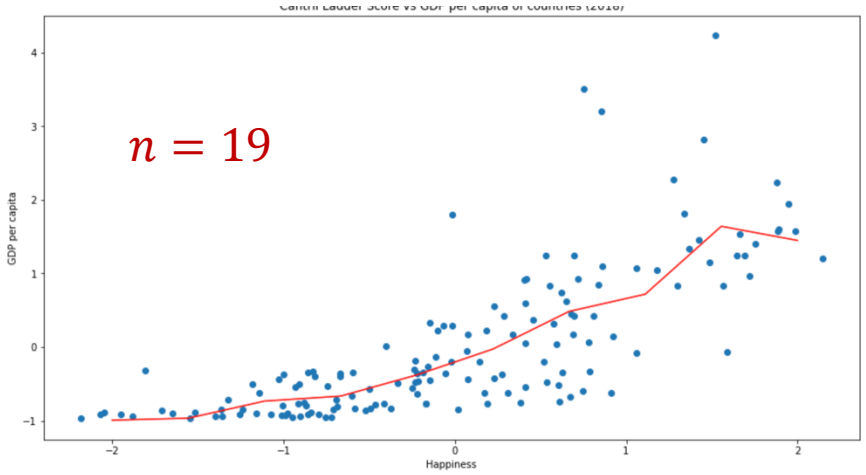
Review: Polynomial Regression

$$y' = \beta_0 + \beta_1 * x_1 + \beta_2 * {x_1}^2 + \beta_3 * {x_1}^3 + \cdots + \beta_n * {x_1}^n + \epsilon$$

n= degree of polynomial

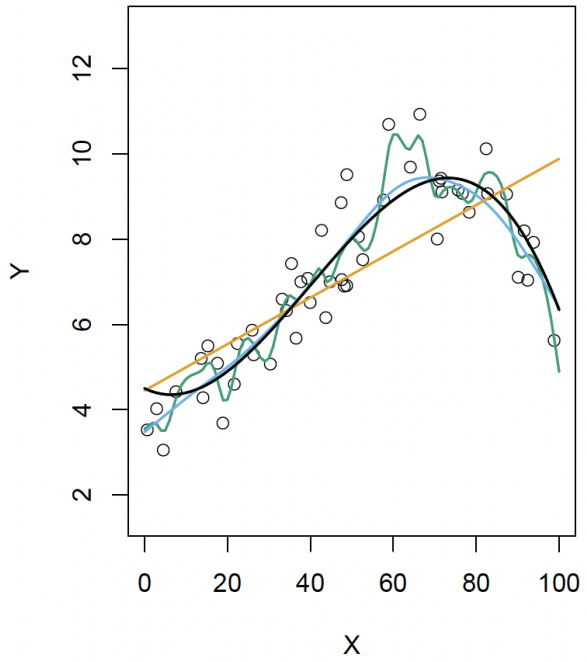


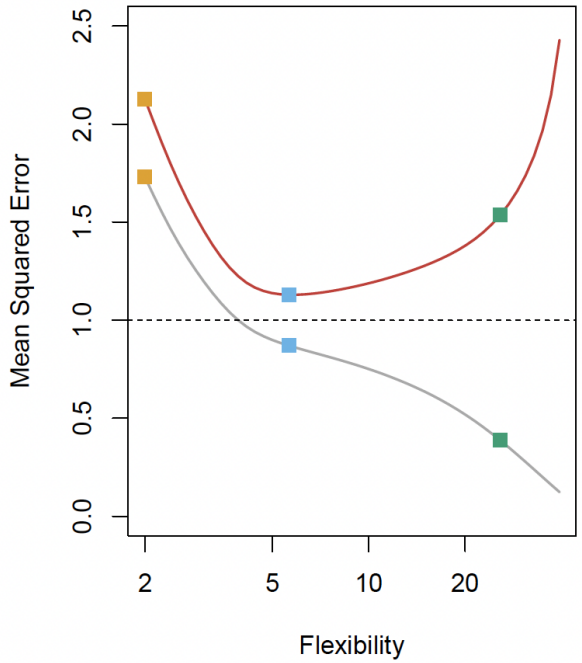




McMaster University

Review: Bias-Variance Trade Off



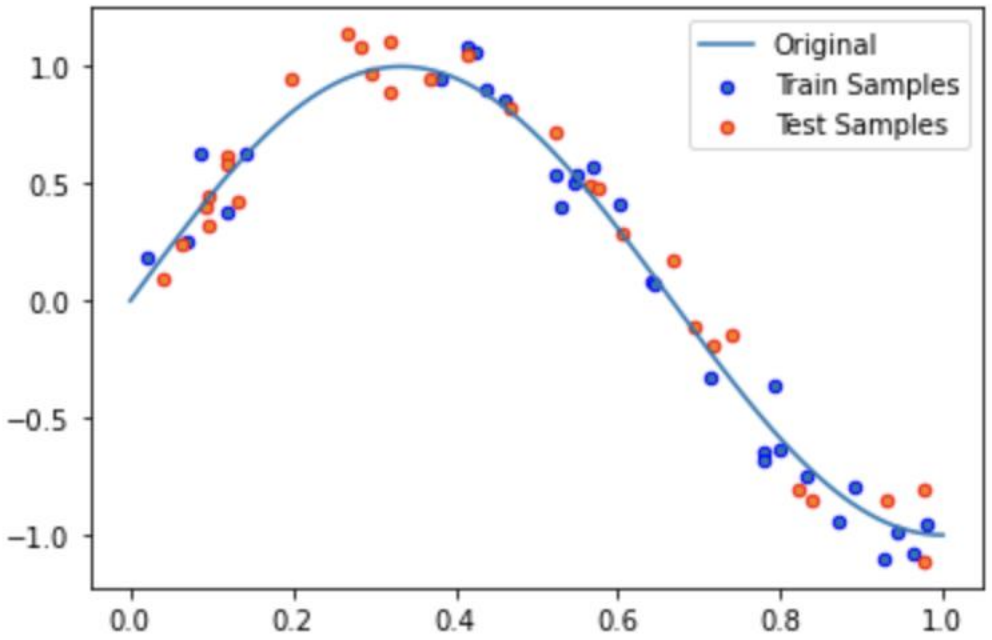


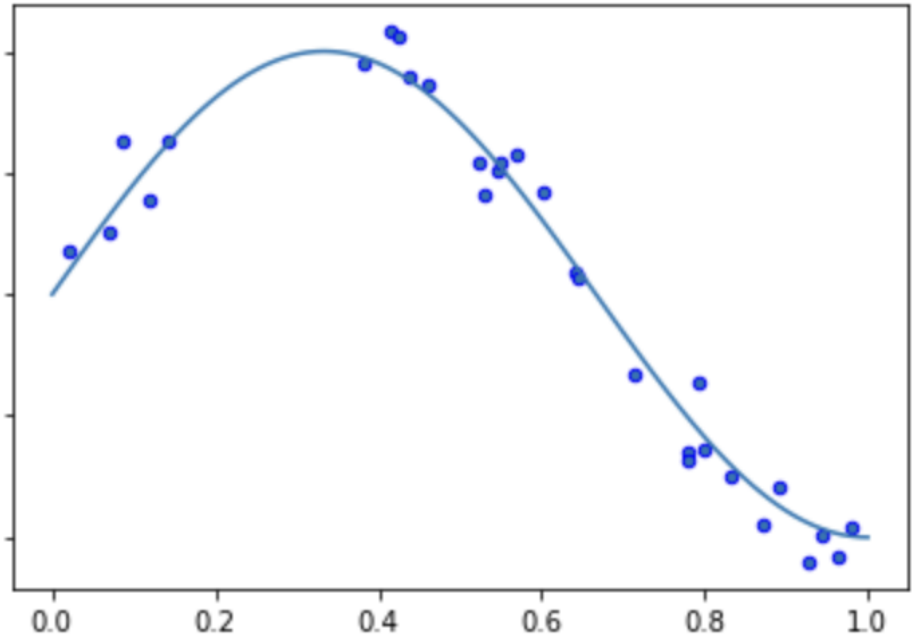
Remember: Underfitting occurs when a model is too simple, which can be a result of a model needing more training time, more input features, or less regularization.

Remember: We always expect the training MSE to be smaller than the test MSE, overfitting is when less flexible model would have yielded a smaller test MSE

Expected Test MSE = Variance of y' + Bias of y' + Variance of ϵ

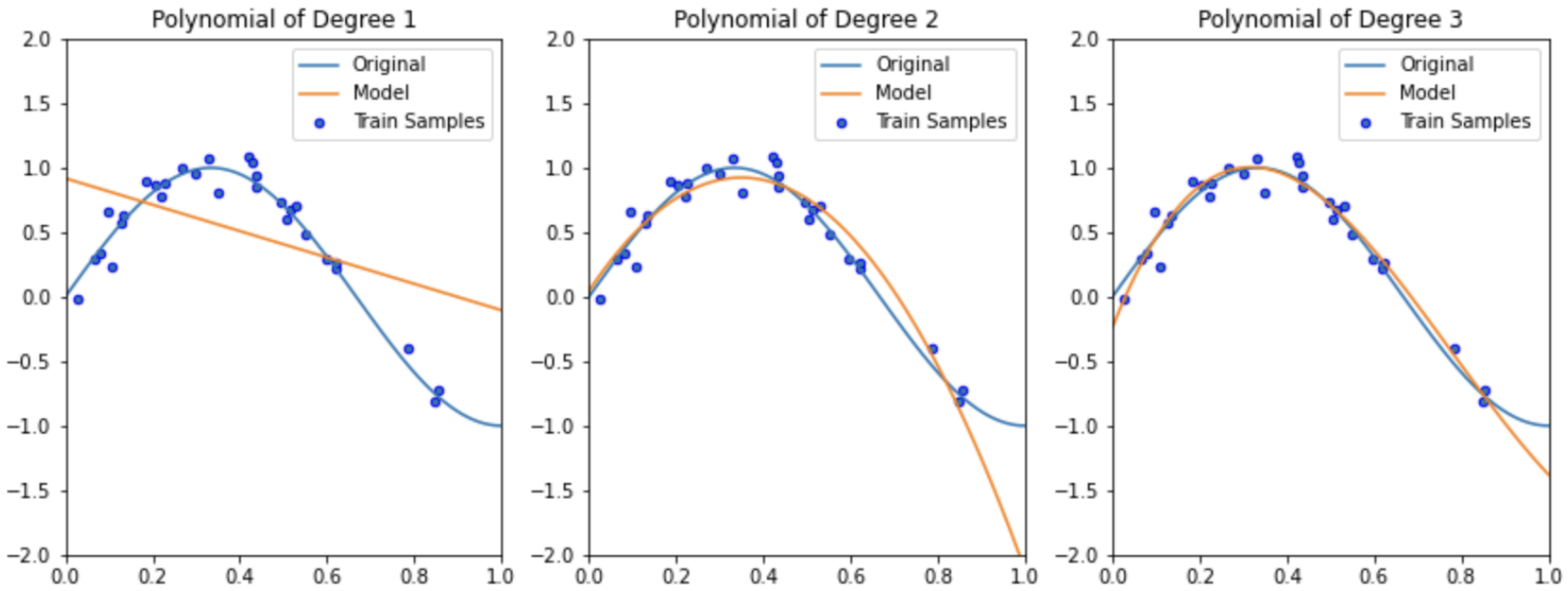
A Similar Example





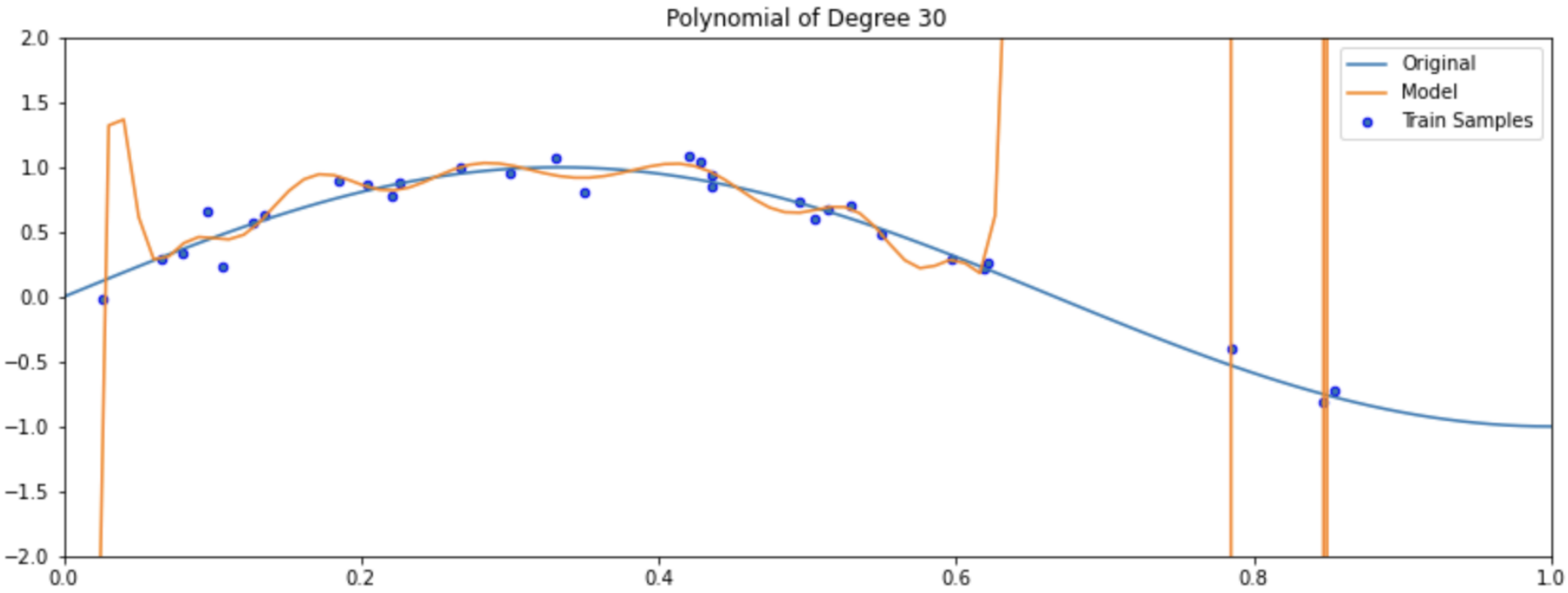

```
def sinfunc(X):
    return np.sin(1.5 * np.pi * X)
```

A Similar Example




```
polynomial_features = PolynomialFeatures(degree=degrees[i],include_bias=False)
linear_regression = LinearRegression()
```

A Similar Example




```
polynomial_features = PolynomialFeatures(30,include_bias=False)
linear_regression = LinearRegression()
```

Evaluation Goal

- Our goal is to find the model that performs best on test set.
  - The model should have high variance.
  - The model should have low bias.
  - It should have high predictive accuracy on the test set.
  - It should be interpretable in the sense, only relevant features should matter.

Evaluating Model Performance

- Since complex models may overfit, we can use several strategies to evaluate them.
  - Subset Selection
    - Throwing in all features seems easy but is not always correct (or even scientific)
    - This approach involves identifying a subset of features that contribute directly to the predictions.

Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using using the prediction error on a validation set, C_p (AIC), BIC, or adjusted R^2. Or use the cross-validation method.

Source: ISLP Book

Evaluating Model Performance

- Since complex models may overfit, we can use several strategies to evaluate them.
  - Subset Selection
    - Throwing in all features seems easy but is not always correct (or even scientific)
    - This approach involves identifying a subset of features that contribute directly to the predictions.

Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using using the prediction error on a validation set, C_p (AIC), BIC, or adjusted R^2. Or use the cross-validation method.

How many models to train when $p = 20$?

Evaluating Model Performance

- Since complex models may overfit, we can use several strategies to evaluate them.
  - Subset Selection
    - Throwing in all features seems easy but is not always correct (or even scientific)
    - This approach involves identifying a subset of features that contribute directly to the predictions.

Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using using the prediction error on a validation set, C_p (AIC), BIC, or adjusted R^2. Or use the cross-validation method.

How many models to train when $p = 20$?

2^p

Evaluating Model Performance

- Since complex models may overfit, we can use several strategies to evaluate them.
  - Subset Selection
  - Forward step-wise selection
    - Begin with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all the predictors are in the model.

Algorithm 6.2 *Forward stepwise selection*

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   (a) Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   (b) Choose the *best* among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using the prediction error on a validation set, C_p (AIC), BIC, or adjusted R^2. Or use the cross-validation method.

Source: ISLP Book

Evaluating Model Performance

- Since complex models may overfit, we can use several strategies to evaluate them.
  - Subset Selection
  - Forward step-wise selection
    - Begin with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all the predictors are in the model.

Algorithm 6.2 *Forward stepwise selection*

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   (a) Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   (b) Choose the *best* among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using the prediction error on a validation set, C_p (AIC), BIC, or adjusted R^2. Or use the cross-validation method.

How many models to train when $p = 20$?

Evaluating Model Performance

- Since complex models may overfit, we can use several strategies to evaluate them.
  - Subset Selection
  - Forward step-wise selection
    - Begin with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all the predictors are in the model.

Algorithm 6.2 *Forward stepwise selection*

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   (a) Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   (b) Choose the *best* among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using the prediction error on a validation set, C_p (AIC), BIC, or adjusted R^2. Or use the cross-validation method.

How many models to train when $p = 20$?

$$1 + \frac{p(p+1)}{2}$$

Evaluating Model Performance

- Since complex models may overfit, we can use several strategies to evaluate them.
  - Subset Selection
  - Forward step-wise selection
  - Backward step-wise selection
    - Begin with full model containing all p predictors, and then iteratively remove the least useful predictor, one-at-a-time.

Algorithm 6.3 *Backward stepwise selection*

1. Let $\mathcal{M}_p$ denote the *full* model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the *best* among these k models, and call it $\mathcal{M}_{k-1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using the prediction error on a validation set, C_p (AIC), BIC, or adjusted R^2. Or use the cross-validation method.

Source: ISLP Book

Evaluating Model Performance

- Since complex models may overfit, we can use several strategies to evaluate them.
  - Subset Selection
  - Forward step-wise selection
  - Backward step-wise selection
    - Begin with full model containing all p predictors, and then iteratively remove the least useful predictor, one-at-a-time.

Algorithm 6.3 *Backward stepwise selection*

1. Let $\mathcal{M}_p$ denote the *full* model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the *best* among these k models, and call it $\mathcal{M}_{k-1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using the prediction error on a validation set, C_p (AIC), BIC, or adjusted R^2. Or use the cross-validation method.

How many models to train when $p = 20$?

Evaluating Model Performance

- Since complex models may overfit, we can use several strategies to evaluate them.
  - Subset Selection
  - Forward step-wise selection
  - Backward step-wise selection
    - Begin with full model containing all p predictors, and then iteratively remove the least useful predictor, one-at-a-time.

Algorithm 6.3 *Backward stepwise selection*

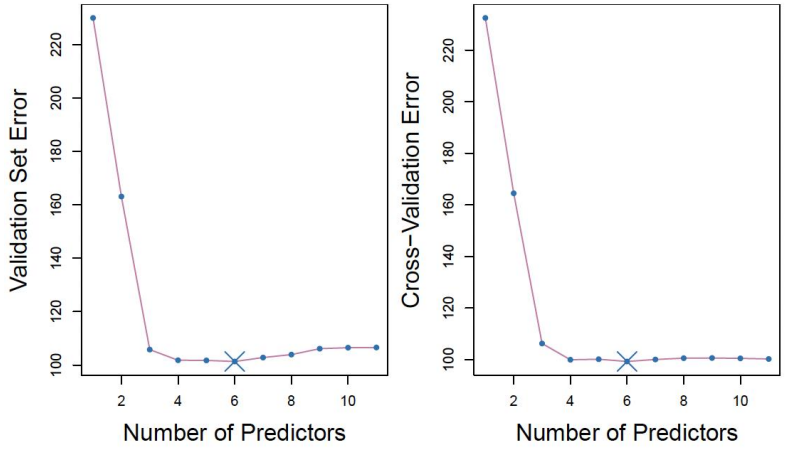
1. Let $\mathcal{M}_p$ denote the *full* model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the *best* among these k models, and call it $\mathcal{M}_{k-1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using the prediction error on a validation set, C_p (AIC), BIC, or adjusted R^2. Or use the cross-validation method.

How many models to train when $p = 20$?

$$1 + \frac{p(p+1)}{2}$$

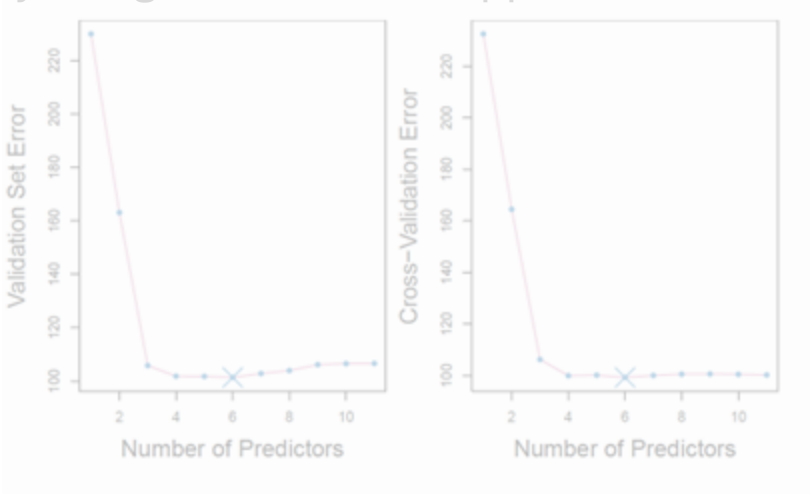
Evaluating Model Performance

- Since complex models may overfit, we can use several strategies to evaluate them.
  - Subset Selection
  - Forward step-wise selection
  - Backward step-wise selection
- Using above approaches, you get multiple models, to select the best model:
  - Indirectly estimate the test error by making adjustment to the training error
  - Directly estimate the test error by using validation set approach or cross-validation approach.



Evaluating Model Performance

- Since complex models may overfit, we can use several strategies to evaluate them.
  - Subset Selection
  - Forward step-wise selection
  - Backward step-wise selection
- Using above approaches, you get multiple models, to select the best model:
  - Indirectly estimate the test error by making adjustment to the training error
  - Directly estimate the test error by using validation set approach or cross-validation approach.



How does this work for complex models like Neural Networks?

Regularization

- Goal in Machine Learning:
  - Minimize the test error, possibly at the expense of increased training error.

- Regularization is defined as any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error.
  - One strategy is adding restrictions on parameter values.

- Regularization of an estimator works by trading increased bias for reduced variance.

- The best fitting model is a large model that has been regularized appropriately.

Regularization

- Regularization is defined as any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error.
  - One strategy is adding restrictions on parameter values.

$$J(\beta) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, f_\beta(x_i)) + \lambda R(\beta)$$

- $L =$ Loss Function
- $R =$ Regularization Function or Parameter norm penalty
- $\lambda =$ Strength of R

Source: ISLP Book

Regularization

- Regularization has effect on all aspects of training.
- If training objective is : $\tilde{J}(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}) = \frac{\alpha}{2}\boldsymbol{w}^\top \boldsymbol{w} + J(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}),$
- Then the parameter gradient is : $\nabla_{\boldsymbol{w}} \tilde{J}(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}) = \alpha \boldsymbol{w} + \nabla_{\boldsymbol{w}} J(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}).$
- The updated weights with gradient decent are: $\boldsymbol{w} \leftarrow (1 - \epsilon\alpha)\boldsymbol{w} - \epsilon \nabla_{\boldsymbol{w}} J(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}).$

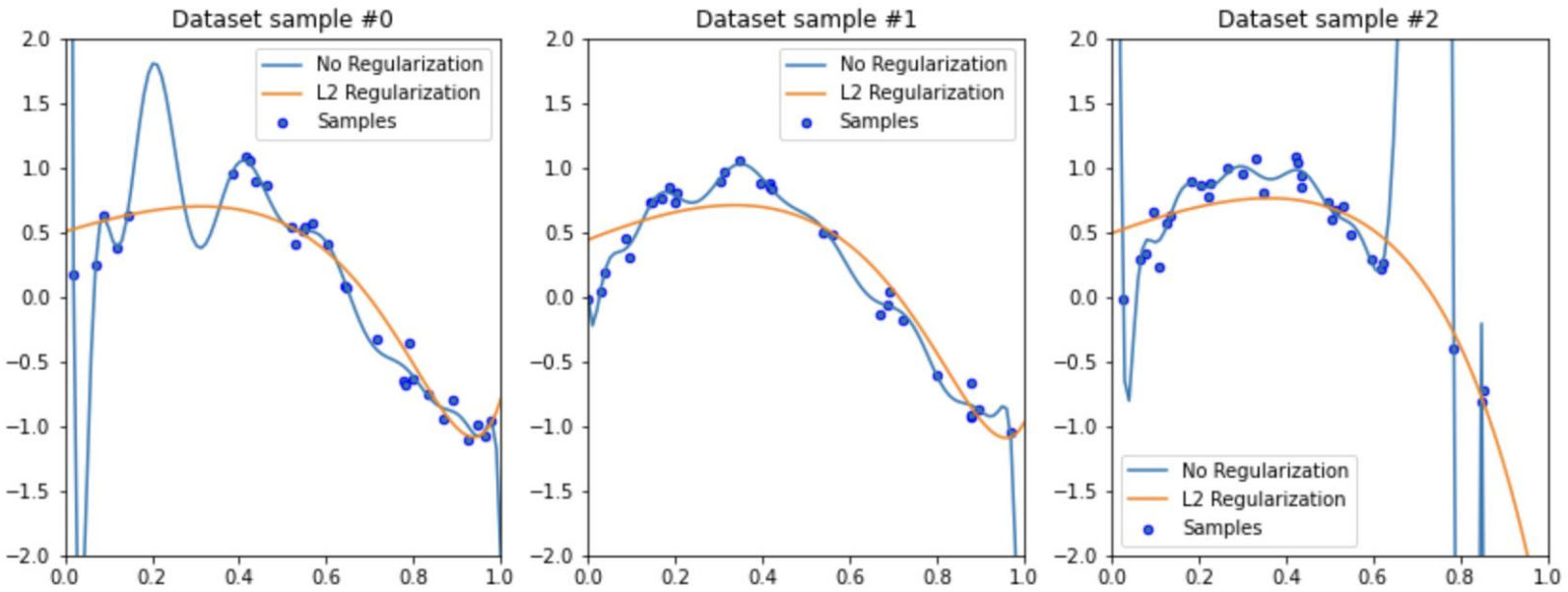
L2 -Regularization

- For many models like linear and neural networks, L2 Regularization is used.

$$J(\beta) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, f_\beta(x_i)) + \frac{\lambda}{2}.||\beta||_2^2 \qquad \text{where,} \quad ||\beta||_2^2 = \sum_{j=1}^{d} \beta_j^2$$

- For neural network, we only impose penalty on the weights, not on biases.
- The regularizer penalizes large parameters
- Prevents model from over-relying on any single feature
- Penalizes wildly irregular solutions.

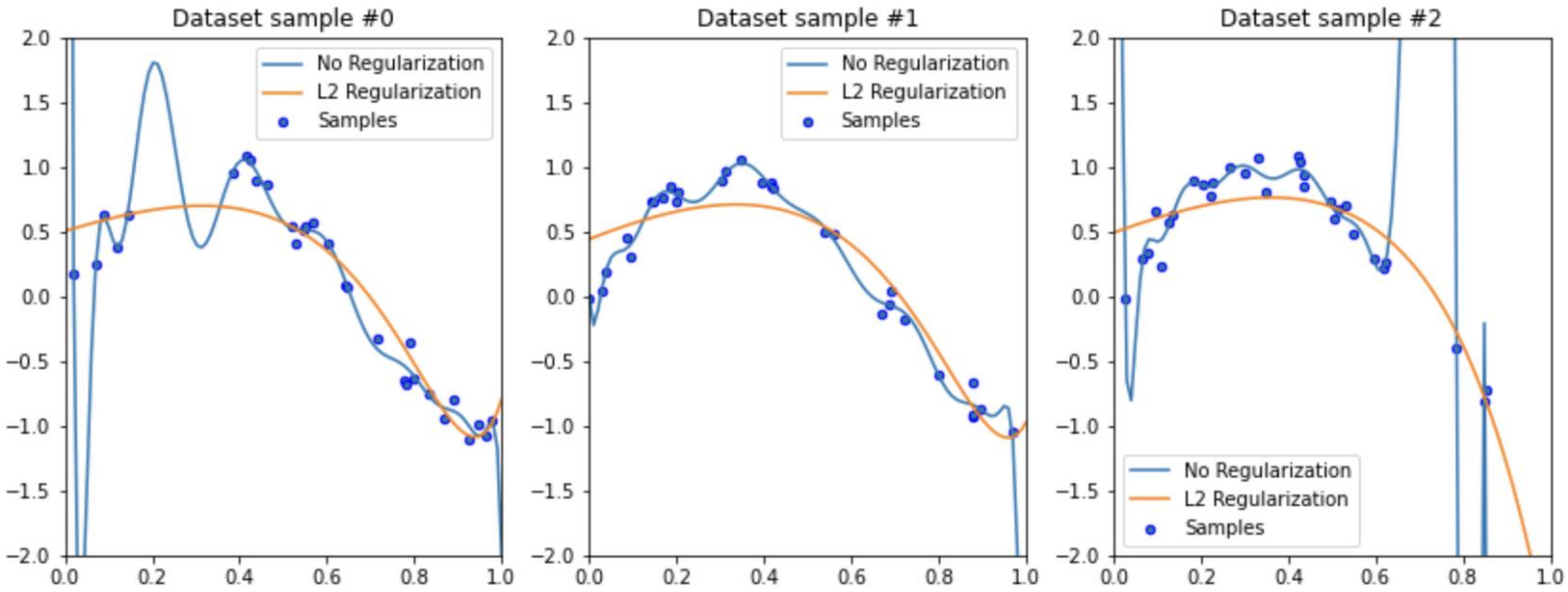
L2 -Regularization




```
polynomial_features = PolynomialFeatures(degree=15,include_bias=False)
linear_regression = Ridge(alpha=0.1)
```

```
polynomial_features = PolynomialFeatures(degree=15,include_bias=False)
linear_regression = LinearRegression()
```

L2 -Regularization




```
Non-regularized weights
[-3.01896363e+03  1.16538776e+05 -2.44723584e+06  3.20285728e+07]

By regularizing the weights to be small
[ 1.2311481  -0.76787277 -1.09819257 -0.90944181]
```

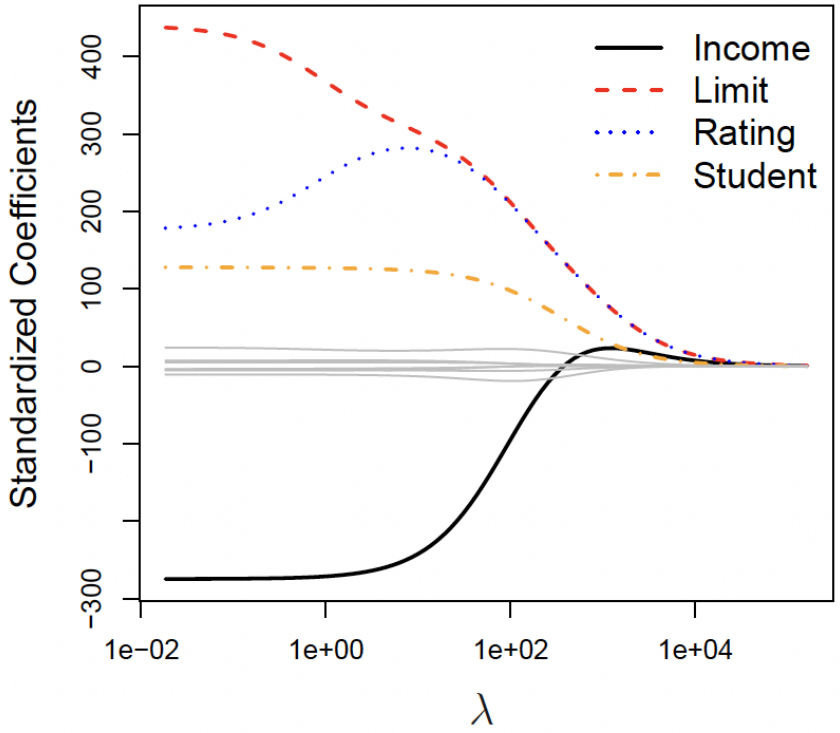
$$J(\beta) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, f_\beta(x_i)) + \frac{\lambda}{2}.||\beta||_2^2$$

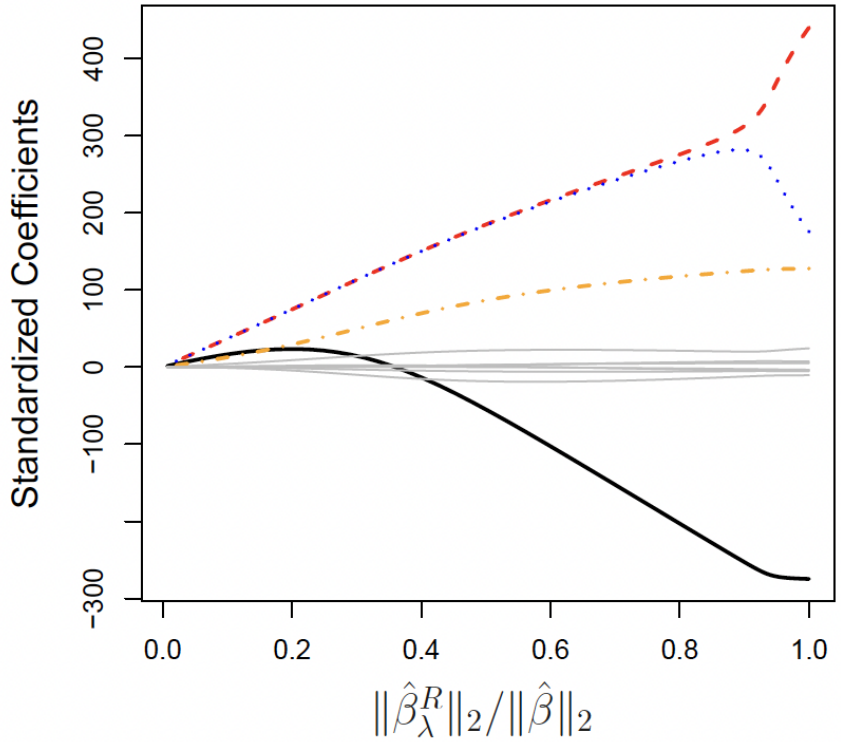
where, $||\beta||_2^2 = \sum_{j=1}^{d} \beta_j^2$

L1 -Regularization

L2

- Credit Card rating prediction using 20 variables



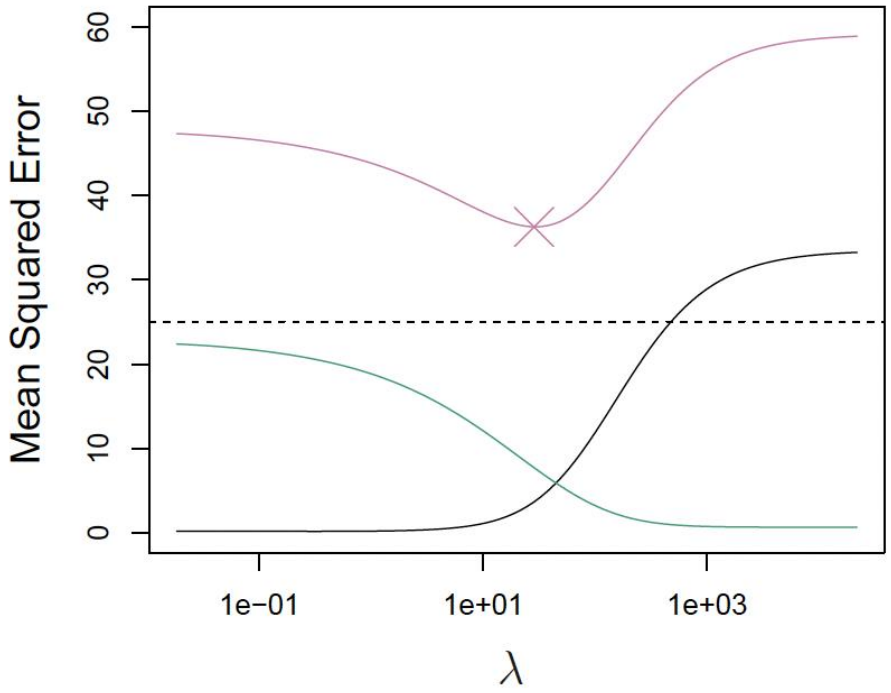


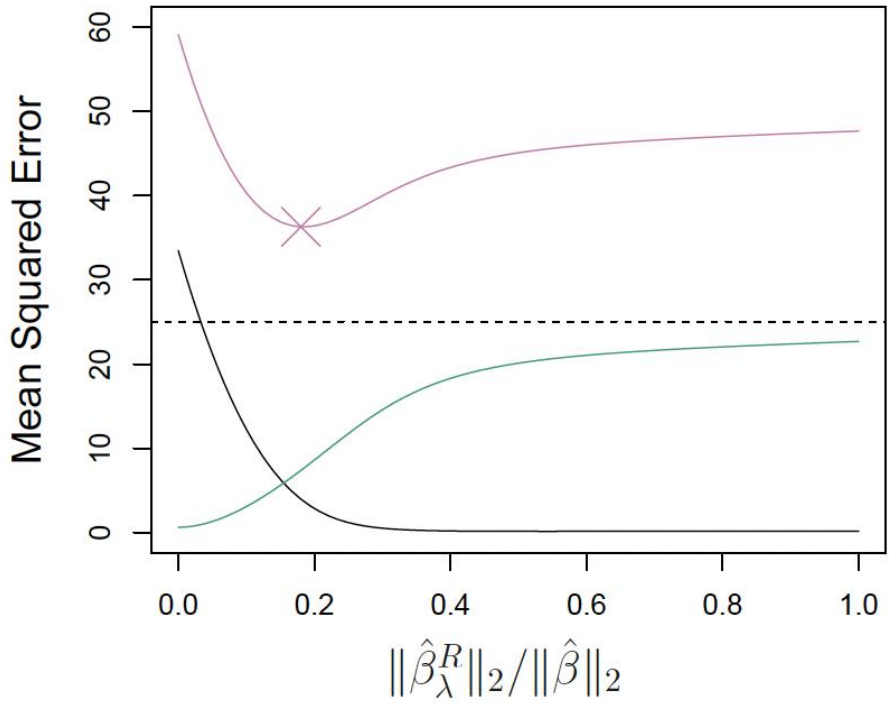
Source: ISLP Book

L1 -Regularization

L2

- Credit Card rating prediction using 20 variables





Source: ISLP Book

L2 -Regularization

- For many models like linear and neural networks, L2 Regularization is used

$$J(\beta) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, f_\beta(x_i)) + \frac{\lambda}{2}.||\beta||_2^2$$

where, $||\beta||_2^2 = \sum_{j=1}^{d} \beta_j^2$

- For neural network, we only impose penalty on the weights, not on biases.
- The regularizer penalizes large parameters
- Prevents model from over-relying on any single feature
- Penalizes wildly irregular solutions.

How do we chose λ?

L2 -Regularization

- For many models like linear and neural networks, L2 Regularization is use

$$J(\beta) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, f_\beta(x_i)) + \frac{\lambda}{2}.||\beta||_2^2$$

where, $||\beta||_2^2 = \sum_{j=1}^{d} \beta_j^2$

- For neural network, we only impose penalty on the weights, not on biases.
- The regularizer penalizes large parameters
- Prevents model from over-relying on any single feature
- Penalizes wildly irregular solutions.

How do we chose λ?

Chose the value that results in the best performance on a held-out validation set.

L1 -Regularization

- For many models like linear and neural networks, L2 Regularization is use

$$J(\beta) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, f_\beta(x_i)) + \frac{\lambda}{2}||\beta||_1$$

where, $||\beta||_1 = \sum_{j=1}^{d} |\beta_j|$

- The regularizer also penalizes large weights.
- It forces more weights to decay to zero.

How do we chose between L1 or L2?

L1 -Regularization

- For many models like linear and neural networks, L2 Regularization is use

$$J(\beta) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, f_\beta(x_i)) + \frac{\lambda}{2}||\beta||_1$$

where, $||\beta||_1 = \sum_{j=1}^{d} |\beta_j|$

- The regularizer also penalizes large weights.
- It forces more weights to decay to zero.

How do we chose between L1 or L2?

The answer lies in sparsity.

Readings

Required Readings:

Introduction to Statistical Learning
- Chapter 2 – Section 2.2 Page 27-34
- Chapter 6 – Section 6.1 and 6.2 Page 231-235

***Supplemental Readings*:**

Deep Learning
- Chapter 7 – page 228 – 236

Introduction to Statistical Learning
- Chapter 6 – Section 6.2 Page 240-244

Thank You

McMaster University