CPE 490 590: Machine Learning for Engineering Applications

07 Overfitting and Regularization

Rahul Bhadani

Electrical & Computer Engineering, The University of Alabama in Huntsville

Outline

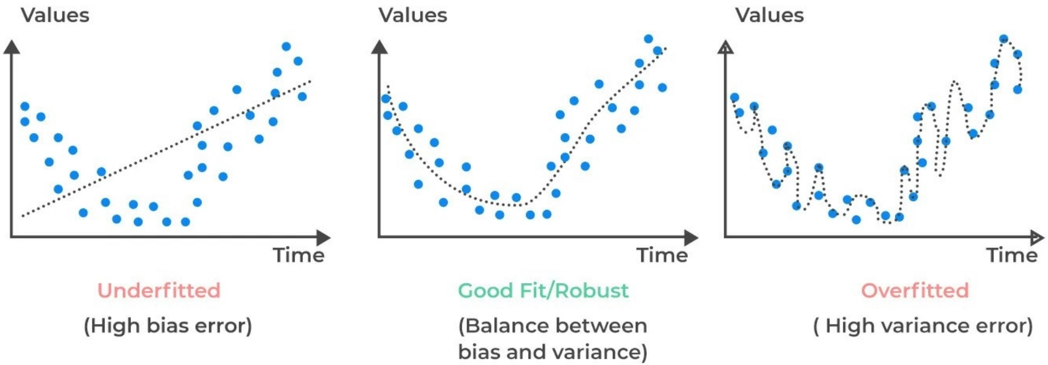
Logistics

Quiz

- Quiz 2: Opens on Feb 14, 2025 Midnight, Closes on Feb 16, 2025 11:59 PM
- Covers Topics on Linear Regression
- Duration: Approximately One Hour
- Requires you running Python Notebook while taking the quiz for some questions.

Motivation

Motivating Regularization



Underfitting

Underfitting occurs when a model is too simple to capture the underlying structure of the data. The model has high bias and low variance, leading to poor performance.

Overfitting

Overfitting occurs when a model is too complex and captures the noise along with the underlying structure of the data. The model has low bias and high variance, leading to good performance on training data but poor generalization to new data.

In overfitting the model describes random error or noise instead of the underlying relationship.

Bias-Variance Trade-off

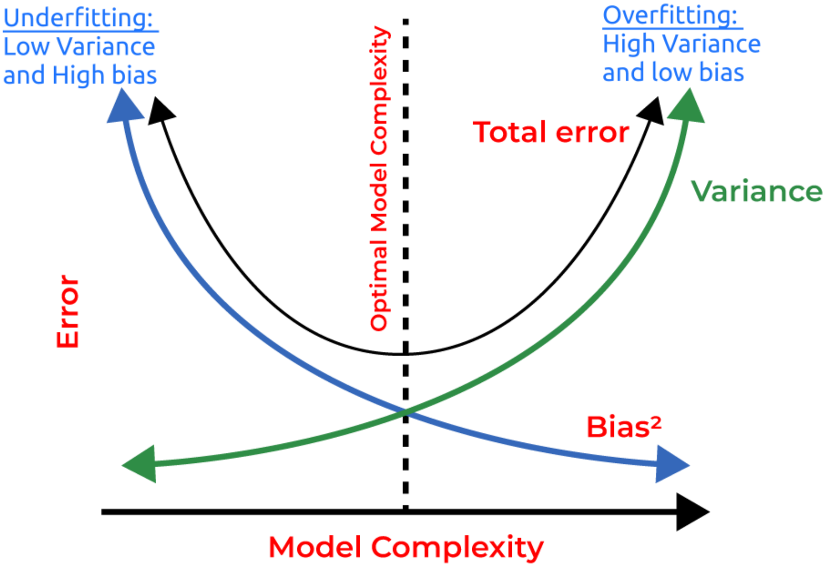
The bias-variance trade-off is the balance that must be found between the simplicity and complexity of a model. An optimal model complexity would result in low bias and low variance, minimizing total error.

Bias

Bias is an error introduced in the model due to oversimplification of the algorithm used where data doesn't fit properly into the model. High-base is the case of underfitting.

Variance

Variance is the error introduced in the model due to a too-complex algorithm. It performs well on the training dataset but fails to generalize to unseen datasets (such as the test dataset).


Underfitting:
Low Variance
and High bias
Optimal Model Complexity
Overfitting:
High Variance
and low bias
Total error
Variance
Error
Bias²
Model Complexity

Regularization

What is Regularization

Regularization is a technique used in machine learning to prevent overfitting by adding a penalty term to the loss function. The penalty term discourages the learning algorithm from assigning too much importance to any individual feature, helping to keep the model simpler and more general.

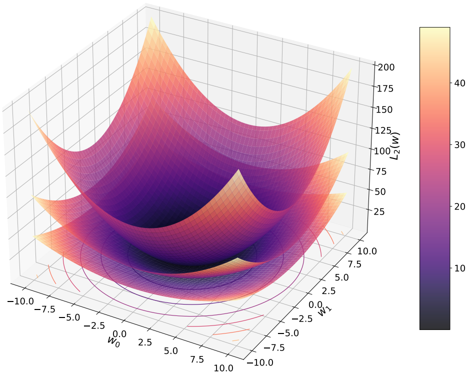
Intuition Behind Regularization

The intuition behind regularization is that simpler models are less likely to overfit. By adding a penalty term to the loss function, we encourage the learning algorithm to find simpler models that still fit the data well. This can lead to better generalization performance on unseen data.

Linear Regression with ℓ_2 Regularization

$$
\begin{aligned}
J(\mathbf{w}, w_0) &= (\mathbf{y} - \mathbf{x}\mathbf{w})^\mathsf{T}(\mathbf{y} - \mathbf{x}\mathbf{w}) + \lambda \|\mathbf{w}\|_2^2 \\
\frac{\partial J}{\partial \mathbf{w}} &= -\mathbf{x}^\mathsf{T}(\mathbf{y} - \mathbf{x}\mathbf{w}) + \lambda \mathbb{I}\mathbf{w} = 0 \\
(\mathbf{x}^\top \mathbf{x} + \lambda \mathbb{I})\mathbf{w} &= \mathbf{x}^\top \mathbf{y} \\
&\longrightarrow \mathbf{w} = (\mathbf{x}^\mathsf{T}\mathbf{x} + \lambda \mathbb{I})^{-1}\mathbf{x}^\mathsf{T}\mathbf{y}
\end{aligned}
$$

ℓ_2 Penalty with Different λ



- The loss increases quadratically as the weights w_0 and w_1 move away from zero. This encourages the optimization process to favor smaller weight values, effectively shrinking them toward zero.
- Smaller λ: The regularization term has little influence, and the optimization focuses more on minimizing the data fitting term. The ℓ_2 surface becomes flatter, allowing larger weights.
- Larger λ: The regularization term dominates, forcing the weights closer to zero. The ℓ_2 surface becomes steeper, strongly penalizing large weights.

ℓ_2 Regularization in In Python

```
def linreg(X, y, reg:float=0.0):
    return np.dot(np.dot(np.linalg.inv(np.dot(X.T, X) \
        + reg*np.eye(X.shape[1])), X.T), y)
```

Regularized Linear Models: $g(\mathbf{x}) = \mathbf{w}^\mathsf{T}\mathbf{x}$

Ridge Regression (Tikhonov regularization or ℓ_2 regularization)

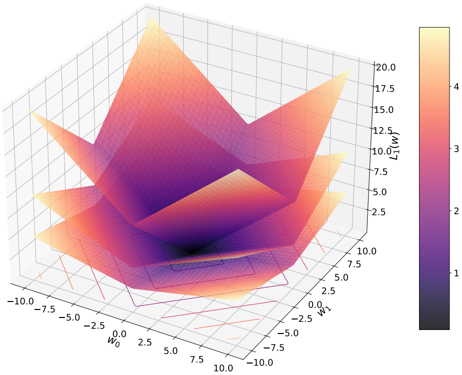
$$J_{RR} = \frac{1}{n}\sum_{i=1}^{n}(y_i - g(\mathbf{x}_i))^2 + \lambda\sum_{j=1}^{d} w_j^2$$

Regularized Linear Models: $g(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$

LASSO or ℓ_1 regularization

$$J_L = \frac{1}{n}\sum_{i=1}^{n}(y_i - g(\mathbf{x}_i))^2 + \lambda \sum_{j=1}^{d} |w_j|$$

ℓ_1 Penalty with Different λ



- The sharp edges and corners of the ℓ_1 surface encourage weights to become exactly zero, promoting sparse solutions (a situation where most weights are zero).
- ℓ_1 regularization is ideal for feature selection, as it eliminates irrelevant or less important features.

Elastic Nets

$\alpha \in [0, 1]$

$$J_{EN} = \frac{1}{n}\sum_{i=1}^{n}(y_i - g(\mathbf{x}_i))^2 + (1-\alpha)\lambda_2\sum_{j=1}^{d} w_j^2 + \alpha\lambda_1\sum_{j=1}^{d}|w_j|$$

Summary of Linear Regression

- Linear, ridge, LASSO and elastic net regression are widely used in many fields for constructing a linear model.
  - LASSO and elastic nets can be used for feature selection too since many of the w_j's will go to zero.
  - LASSO works best when $d \gg n$ and elastic nets should be used if $d > n$.
- Linear and ridge regression have closed for solutions; however, LASSO and elastic nets require that gradient-based methods be used to numerically find the solution.

Code Availability and Python Notebook

Code used for this chapter is available at
https://github.com/rahulbhadani/CPE490_590_Sp2025/blob/master/Code/Chapter_07_Overfitting_And_Regularization.ipynb

The End