Bias-Variance Tradeoff in Machine Learning

1 Introduction

The bias-variance tradeoff is a fundamental concept in machine learning that reflects the balance between model complexity and prediction accuracy. This document explores this concept through polynomial basis expansion for regression and the role of regularization techniques (L1 and L2) in managing this tradeoff.

2 Polynomial Basis Expansion for Regression

Polynomial basis expansion is a method of extending a standard linear regression model by including higher-order polynomial terms. Recall that, a simple linear regression model is expressed as:

$$y = w_0 + w_1 x + \epsilon,$$

where y is the dependent variable, x is the independent variable, w_0, w_1 are the weights or parameters of the model, and ϵ is the error term.

By incorporating polynomial terms, the model becomes:

$$y = w + w_1 x + w_2 x^2 + \cdots + w_p x^p + \epsilon,$$

where p is the degree of the polynomial.

Polynomial Regression vs. Linear Regression Consider a dataset where the true relationship between x and y is nonlinear. A linear regression model may underfit the data due to its simplicity (high bias), while a polynomial regression model can capture more nuances due to its higher flexibility (low bias).

However, as the polynomial degree p increases, the model gains more degrees of freedom. While this reduces bias, it increases variance, making the model more sensitive to small variations in the data. For example:

- A degree-1 polynomial (linear regression) may fail to capture curvature.
- A degree-10 polynomial may overfit, capturing noise as part of the pattern.

2.1 Multivariate Polynomial Basis Expansion

For multiple input features, $\mathbf{x} = \mathbb{R}^D$, a polynomial regression model of degree p includes all monomials of degree 0 to p. The model becomes:

$$y = w_0 + \sum_{j=1}^{D} w_j x_j + \sum_{j=1}^{D} \sum_{k=j}^{D} w_{jk} x_j x_k + \cdots + w_{1,\ldots,D} \prod_{j=1}^{D} x_j^{p_j} + \epsilon,$$

where $\sum_{j=1}^{D} p_j \leq p$.

The total number of terms, including interaction terms, is given by:

$$\text{Number of terms} = \binom{D+p}{p}.$$

2.2 Scalability Challenges

The polynomial basis expansion suffers from the curse of dimensionality. As the number of input features D or the degree p increases:

- The number of terms grows exponentially, leading to high computational cost.
- The model becomes highly prone to overfitting due to the large number of parameters (high variance).
- Data sparsity becomes an issue, as the available data points may not sufficiently cover the expanded feature space.

For instance, with $D = 10$ and $p = 5$, the number of terms is:

$$\binom{10+5}{5} = 3003,$$

which is computationally expensive and requires significant data to avoid overfitting.

Small Example However, for reasonably small D and p polynomial basis expansion can be quite useful, for example, for $D = 2$ and $p = 2$ we can express the full model as,

$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2 + \epsilon$$

2.3 Bias-Variance Tradeoff in Polynomial Regression

Increasing the polynomial degree reduces bias but increases variance. A model with a higher degree of complexity is more flexible and can closely fit the training data, but it may overfit, capturing noise instead of the underlying pattern.

This tradeoff is closely related to **Occam's Razor**, a principle suggesting that among competing models that explain the data, the simplest model (with the fewest assumptions) should be preferred. In machine learning, simpler models are often less prone to overfitting and tend to generalize better to unseen data. Overly complex models, such as high-degree polynomial regressions, violate this principle by adding unnecessary complexity, leading to higher variance and poorer generalization.

To manage this tradeoff, regularization techniques such as L1 and L2 regularization are often employed.

3 Regularization Techniques

Regularization introduces a penalty term to the loss function to constrain model complexity. Two common types of regularization are L1 and L2, which are applicable to both linear and logistic regression.

3.1 L1 Regularization (Lasso)

In L1 regularization, a penalty proportional to the sum of the absolute values of the coefficients is added to the loss function. For linear regression, the loss function becomes:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 + \lambda\sum_{i=0}^{D}|w_i| = \frac{1}{2N}\sum_{i=1}^{N}(\mathbf{x}_i^T\mathbf{w} - y_i)^2 + \lambda\sum_{i=0}^{D}|w_i|$$

where λ controls the strength of regularization.

L1 regularization promotes sparsity in the coefficients, often driving some of them to zero, which is useful for feature selection.

3.2 L2 Regularization (Ridge)

In L2 regularization, the penalty is proportional to the sum of the squares of the coefficients. The loss function becomes:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 + \lambda\sum_{i=0}^{D}(w_i)^2 = \frac{1}{2N}\sum_{i=1}^{N}(\mathbf{x}_i^T\mathbf{w} - y_i)^2 + \lambda\mathbf{w}^T\mathbf{w}$$

L2 regularization prevents large coefficients, promoting smoother models while retaining all features.

3.3 Logistic Regression with L1 and L2 Regularization

For logistic regression, the objective is to minimize the negative log-likelihood with regularization:

- **L1 Regularization:**

$$-\log L(\mathbf{w}) = -\sum_{i=1}^{N}\left[y_i \log\sigma(\mathbf{w}^T\mathbf{x}_i) + (1-y_i)\log\left(1-\sigma(\mathbf{w}^T\mathbf{x}_i)\right)\right] + \lambda\sum_{i=0}^{D}|w_i|$$

- **L2 Regularization:**

$$-\log L(\mathbf{w}) = -\sum_{i=1}^{N}\left[y_i \log\sigma(\mathbf{w}^T\mathbf{x}_i) + (1-y_i)\log\left(1-\sigma(\mathbf{w}^T\mathbf{x}_i)\right)\right] + \lambda\mathbf{w}^T\mathbf{w}$$

3.4 Impact of Regularization on the Bias-Variance Tradeoff

Regularization increases bias by constraining model flexibility but reduces variance, leading to improved generalization. The choice of λ is crucial and is typically determined via cross-validation.