Lecture 5: Introduction to Machine Learning

Yonghao Lee

December 4, 2025

1 Linear Hypothesis Classes

A linear hypothesis takes the form of a linear decision rule. In a binary classification setting, we predict positive if:

$$\mathbf{w} \cdot \mathbf{x} + b > 0 \tag{1}$$

Visually, this represents a hyperplane separating the data space.

1.1 Convexity

To find the best $\mathbf{w}$ and b, we optimize a loss function. We prefer **convex functions** (shaped like a bowl) because a local minimum in a convex function is guaranteed to be a global minimum.

2 Optimization: Gradient Descent

Since we cannot always find the optimal weights $\mathbf{w}$ by simple guessing, we use an iterative algorithm called Gradient Descent.

2.1 The Algorithm

We start with an initial $\mathbf{w}_{init}$ and iteratively move in the direction opposite to the gradient (the direction of steepest descent).

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \nabla_{\mathbf{w}} L(\mathbf{w}_t) \tag{2}$$

Where:

- $\nabla_{\mathbf{w}} L$: The gradient of the loss function.
- α: The step size (learning rate).

2.2 Step Size Intuition

The step taken is proportional to the gradient magnitude.

- **Steep slope (large gradient):** We take a large step.
- **Flat slope (small gradient):** We take a small step to settle into the minimum.

2.3 Stochastic Gradient Descent (SGD)

Computing the gradient over the entire training set (N_{train}) is computationally expensive.

- **SGD:** Approximates the gradient using a *single* sample per step.
- **Mini-batch GD:** Approximates the gradient using a small batch of samples. This is a compromise that is less noisy than SGD but faster than full Batch GD.

3 Linear Regression

The task of regression is to predict a continuous scalar output y given an input $\mathbf{x}$.

3.1 Matrix Notation and Bias

For convenience, we often absorb the bias term w_0 into the weight vector. We define $\mathbf{w} = (w_0, w_1, \dots, w_d)^T$ and pad each sample $\mathbf{x}$ with a 1, such that $\mathbf{x} = (1, x_1, \dots, x_d)^T$. The hypothesis then becomes a simple dot product: $f(\mathbf{x}) = \mathbf{x}^T \mathbf{w}$.

3.2 Squared Loss (MSE)

We minimize the Empirical Risk, defined as the Mean Squared Error (MSE):

$$L(S_{train}, \mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \mathbf{w})^2 \tag{3}$$

Note: In the realizable case, we only need $d + 1$ samples to calculate the weights correctly.

3.3 Polynomial Fitting

Linear regression assumes a linear relationship between features and the target, but data is often non-linear. We can still use linear regression by transforming the feature space.

Let $\psi(\mathbf{x})$ be a non-linear mapping (e.g., polynomial basis functions). For a polynomial of degree k, we transform a scalar x into a vector:

$$\psi(x) = (1, x, x^2, \dots, x^k)^T \tag{4}$$

We then solve for $\mathbf{w}$ such that $y \approx \psi(\mathbf{x})^T \mathbf{w}$. This allows us to fit curves while keeping the optimization problem linear with respect to the weights.

4 L2 Regularization (Ridge Regression)

To prevent overfitting and control model complexity, we introduce L2 Regularization. We penalize the magnitude of the weights.

4.1 Regularized Loss Function

The new loss function includes a penalty term weighted by hyperparameter α:

$$L_{total}(\mathbf{w}) = \underbrace{\frac{1}{2N_{train}} \sum (y - \mathbf{w} \cdot \mathbf{x})^2}_{\text{Data Term}} + \underbrace{\frac{\alpha}{2} \|\mathbf{w}\|^2}_{\text{Regularization}} \tag{5}$$

4.2 The Analytical Solution

Unlike standard gradient descent, Ridge Regression allows for a **closed-form solution**. The optimal weights $\mathbf{w}^*$ are:

$$\mathbf{w}^* = (X^T X + \alpha I)^{-1} X^T \mathbf{y} \tag{6}$$

Note: The term αI ensures the matrix is invertible (non-singular).

5 Classification

5.1 Logistic Regression

Instead of predicting a raw scalar, we predict the probability $P(y = 1|\mathbf{x})$. We use the **Sigmoid Function** to squash the score into $[0, 1]$:

$$\sigma(s) = \frac{1}{1 + e^{-s}} \tag{7}$$

The loss function is the **Negative Log Likelihood** (Log Loss).

5.2 Ridge Classification

A simpler alternative to Logistic Regression. We treat the binary labels as real numbers and apply Ridge Regression.

6 Ensemble Methods

Ensemble methods combine multiple models (”prophets”) to improve performance and reduce overfitting.

6.1 Committees and Majority Vote

Instead of relying on a single classifier h_1, we train a committee of k classifiers $\{h_1, \ldots, h_k\}$. The final prediction is determined by a majority vote:

$$y = \text{argmax}_{y \in Y} \sum_{i=1}^{k} \mathbb{I}(h_i(\mathbf{x}) = y) \tag{8}$$

Committees work best when the individual models are accurate (risk < 0.5) and uncorrelated.

6.2 Bagging and Random Forests

To ensure the models in the committee are uncorrelated, we cannot train them on the exact same data.

- **Bagging (Bootstrap Aggregating):** We train each model on a random subset of the data sampled *with replacement* (bootstrapping).
- **Random Forests:** An extension of bagging applied to Decision Trees. To further reduce correlation, at each split in the tree, only a random subset of features is considered.

6.3 Boosting

Boosting builds a committee iteratively rather than independently. At each step p, we add a new weak learner h_p^* that focuses on minimizing the errors made by the previous ensemble of models. This effectively combines many ”weak” models into a single ”strong” predictor.