Classical ML

1 Gaussian Linear Regression via Maximum Likelihood Estimation

In this problem, we will derive the objective function for ordinary least squares (OLS) regression from the principle of **Maximum Likelihood Estimation (MLE)**.

What is MLE?. Recall from lecture that the core idea of MLE is to "learn" or estimate the value of the parameters, θ, that "pin down" the distribution from which our data came. We do this by finding the parameters θ_{MLE} that maximize the **Likelihood Function** $p(D|\theta)$. This function answers the question: "Given our data D, what set of parameters θ makes this data the *most likely* to have been observed?"

Likelihood vs. Loss Function ($\mathcal{L}$ vs. $\mathcal{L}$). This can be confusing, as the symbol $\mathcal{L}$ is used for two different concepts:

- In statistics, $\mathcal{L}(\theta) = p(D|\theta)$ is the **Likelihood Function**. It's a function of the parameters, θ. Our goal is to **maximize** this.
- In machine learning, $\mathcal{L}(y, \hat{y})$ is the **Loss Function**. It measures the "cost" or "error" of a prediction. Our goal is to **minimize** this.

MLE gives us a principled and widely-used way to derive a loss function. This exercise will show how *maximizing* the statistical Likelihood $\mathcal{L}(\theta)$ is mathematically equivalent to *minimizing* a specific ML Loss function. We will use $\log(\cdot)$ to denote the natural logarithm (base e), as is common in ML.

Set up. We have a training dataset $D = \{(x_i, y_i)\}_{i=1}^N$, where $x_i \in \mathbb{R}^d$ are the input features and $y_i \in \mathbb{R}$ are the continuous target labels.

We will use the discriminative approach, modeling $p(y|x)$ directly. We assume our data is generated from a linear model with Gaussian noise. This means the target y_i is a Gaussian random variable whose mean is a linear function of x_i, $w^\top x_i$, and it has some constant variance σ^2.

This assumption is written as:

$$p(y_i|x_i, w, \sigma^2) = \mathcal{N}(y_i|w^\top x_i, \sigma^2)$$

This is equivalent to writing $y_i = w^\top x_i + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. Our parameters to be estimated are $\theta = (w, \sigma^2)$.

(a) **Write the Likelihood of a Single Data Point**

The first step is to write down the probability of observing a single data point (x_i, y_i), given our parameters. This is the Gaussian PDF.

Write out the full probability density function $p(y_i|x_i, w, \sigma^2)$ in terms of y_i, x_i, w, and σ^2.

(b) **Write the Likelihood of the Entire Dataset**

Now we write the total likelihood for *all* our data D. Assuming the training examples are **i.i.d.** (identically and independently distributed), this means the total likelihood is the *product* of the individual likelihoods.

Let the likelihood of the entire dataset be $\mathcal{L}(w, \sigma^2) = p(D|w, \sigma^2)$. Write this function as a product $\prod_{i=1}^N$ using your answer from part (a).

(c) **Write the Log-Likelihood**

Working with a large product $\prod$ is difficult. It's much easier to work with a sum $\sum$. We can (and should!) switch to the **log-likelihood**, $\ell(w, \sigma^2) = \log \mathcal{L}(w, \sigma^2)$, because the $\log(\cdot)$ function is monotonic. Working with a sum is more convenient, and maximizing the log-likelihood is equivalent to maximizing the likelihood.

Derive the log-likelihood $\ell(w, \sigma^2)$ by taking the natural logarithm of your answer in part (b). Simplify the expression using log rules ($\log(ab) = \log a + \log b$ and $\log(e^x) = x$).

(d) **Find the Optimal Parameters** w_{MLE}

This is the key step. The goal of MLE is to find the parameters θ^* that *maximize* the log-likelihood. In contrast, the goal of a Loss Function is to *minimize* error. Let's see how they connect.

$$(w_{MLE}, \sigma^2_{MLE}) = \text{argmax}_{w,\sigma^2} \ell(w, \sigma^2)$$

We only care about finding the optimal weights w. Show that finding the w_{MLE} that **maximizes** your expression from part (c) is **equivalent** to finding the w that **minimizes** the sum of squared errors.

(Hint: Look at the two terms in your final expression for $\ell(w, \sigma^2)$. Which parts depend on w? Also, remember the key identity: $\text{argmax}_w f(w) = \text{argmin}_w - f(w)$.*)*

(e) **Conclusion: Connect to Least Squares**

The term $\sum_{i=1}^{N}(y_i - w^\top x_i)^2$ is the **Sum of Squared Errors (SSE)**. In matrix/vector form, this is $||Xw - y||_2^2$.

Based on your result from part (d), state the final optimization problem.

2 NumPy, Matplotlib, Scikit-learn

In this problem, you will work through a notebook that goes over linear, ridge, lasso, and logistic regression, as well as model evaluations, k-means clustering, random forests & gradient boosting, and PCA (link). Fear not, there is little-to-no coding in this notebook. Instead, we ask that you take your time to understand each problem and answer the analysis questions to the best of your ability.

3 What to Submit

Please submit a single .zip file that includes your written answers and completed notebook. Upload the .zip file to the corresponding homework assignment on Gradescope.

Contributors

- Patrick Mendoza