CSCI 1051 Problem Set 1

January 6, 2025

Submission Instructions

Please upload your solutions by **5pm Friday January 10, 2025.**

- You are encouraged to discuss ideas and work with your classmates. However, you **must acknowledge** your collaborators at the top of each solution on which you collaborated with others and you **must write** your solutions independently.
- Your solutions to theory questions must be written legibly, or typeset in LaTeX or markdown. If you would like to use LaTeX, you can import the source of this document here to Overleaf.
- I recommend that you write your solutions to coding questions in a Jupyter notebook using Google Colab.
- You should submit your solutions as a **single PDF** via the assignment on Gradescope.

Problem 1: Linear Regression

Consider a d-dimensional multivariate linear regression problem with n samples. Each sample i consists of a data vector $\mathbf{x}^{(i)} \in \mathbb{R}^d$ and a label $y^{(i)} \in \mathbb{R}$. From these samples, let $\mathbf{X} \in \mathbb{R}^{d \times n}$ be the data matrix where the ith row consists of the ith data vector. Similarly, let $\mathbf{y} \in \mathbb{R}^n$ be the target vector where the ith entry consists of the ith label.

When we solve the multivariate linear regression problem, we find the coefficients

$$\mathbf{w}^* = \arg\min_{\mathbf{w} \in \mathbb{R}^d} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2. \tag{1}$$

Part A: Computing the Optimal Solution

Using the strategy described in class, show that

$$\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}. \tag{2}$$

Part B: The Optimal Solution on Real Data

Load a real regression dataset and compute the optimal coefficients $\mathbf{w}^*$.

I recommend:

- coding in a Colab notebook,
- using the scikit-learn regression datasets such as the diabetes dataset, and
- adapting the example given in the documentation at the above link.

Part C: scikit-learn on Real Data

On the same regression dataset from Part B, use a linear regression solver to compute the optimal coefficients.

I (still) recommend:

- coding in a Colab notebook,
- using the scikit-learn linear regression solver,
- adapting the example given in the documentation at the above link.

Sanity Check: Ensure that both methods of solving the same regression problem result in the same coefficients.

Problem 2: Logistic Regression

Consider a d-dimensional multivariate logistic regression problem with a single labelled pair. The data vector is $\mathbf{x} \in \mathbb{R}^d$ and the label is $y \in \{0, 1\}$.

Consider a logistic regression model with weights $\mathbf{w} \in \mathbb{R}^d$. Recall that the sigmoid function is given by $\sigma(z) = \frac{1}{1+e^{-z}}$. Then the model is $\sigma(\mathbf{w} \cdot \mathbf{x})$. The cross entropy loss is given by

$$\mathcal{L}(\mathbf{w}) = -\left(y \log(\sigma(\mathbf{w} \cdot \mathbf{x})) + (1-y)\log(1-\sigma(\mathbf{w} \cdot \mathbf{x})\right) \tag{3}$$

Part A: Computing the Gradient

Show that

$$\nabla_w \mathcal{L}(\mathbf{w}) = (\sigma(\mathbf{w} \cdot \mathbf{x}) - y)\mathbf{x}. \tag{4}$$

Hint: First, show that $\frac{\partial}{\partial z}\sigma(z) = \sigma(z)(1-\sigma(z))$. Next, apply the chain rule.

Part B: Optimization?

Why can't we (easily) compute the exact solution in closed form? What should we do instead?

Part C: scikit-learn on Real Data

Load a real classification dataset and compute the optimal coefficients for logistic regression.

I (still) recommend:

- coding in a Colab notebook,
- using the scikit-learn logistic regression solver,
- adapting the example given in the documentation at the above link.