Homework 2

Due: Friday, February 10, 2022 at 12:00pm **(Noon)**

Written Assignment

Problem 1: Halfspaces

(8 points)

Let $\mathcal{X} = \{0,1\}^d$ and $\mathcal{Y} = \{-1,1\}$. Define a halfspace $h : \mathcal{X} \to \mathcal{Y}$ for the following functions:

1. **Conjunction**: Output is 1 if and only if all d attributes are 1.
2. **Majority**: Output is 1 if and only if more than half of the d attributes are 1.

Make sure to explain why you have chosen your weights and why they exhibit the desired behavior. You can assume that there is a bias term and that you can express your weights and bias in terms of d.

Note: The sign function can be considered as

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \text{ (Should not happen to achieve 100\% accuracy)} \\ -1 & \text{if } x < 0 \end{cases}$$

for all problems on this homework.

Problem 2: More Halfspaces

(4 points)

Consider the function $h_{\text{equiv}} : \{0,1\}^2 \to \{-1,1\}$ (ie. $x_1, x_2 \in \{0,1\}$) defined as

$$h_{\text{equiv}}(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 = x_2, \\ -1 & \text{otherwise.} \end{cases}$$

Show that h_{equiv} cannot be represented as a halfspace with a bias term. *Hint:* In class, we showed that XOR cannot be represented with a halfspace. You can use a similar argument here.

Problem 3: Decision Boundaries

(6 points)

Consider an arbitrary halfspace classifier of the form $h_{\mathbf{w}}(\mathbf{x}) = \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle)$ where $h : \mathbb{R}^n \to \{-1,1\}$, $w, x \in \mathbb{R}^n$, and there is no bias term. Prove that the distance from an arbitrary example $\mathbf{x}$ to the decision boundary defined by $\mathbf{w}$ is:

$$\frac{|\langle \mathbf{w}, \mathbf{x} \rangle|}{\|\mathbf{w}\|_2}$$

The distance from an example $\mathbf{x}$ to the decision boundary is defined as the distance from the example to the point $\mathbf{a}$ on the decision boundary with minimum distance to the example $\mathbf{x}$. $\|\mathbf{w}\|_2$ denotes the l^2 norm for distance (i.e. Euclidean distance).

Reminder: The decision boundary of a halfspace classifier is the set of points in $\mathcal{X}$ where the classifier's output changes from -1 to 1, i.e., the solution to $\langle \mathbf{w}, \mathbf{x} \rangle = 0$.

Programming Assignment

Introduction

After months of studying to become a pastry chef, Steve decided to take some time off and host a socially distant Zoom wine night for his friend. Unfortunately, with all the time it takes to make desserts, he hasn't had time to become a wine connoisseur, so he's asked you to help them choose which wines to stock, with the help of machine learning!

From his group of friends, we've collected quite a lot of data, and need your help to electronically determine how each person rated a wine on a scale of 1 to 10.

In this assignment, you'll implement linear regression and use your model to predict wine quality. The book sections relevant to this assignment are 9.2 on page 123, and 9.3 on page 126.

Stencil Code & Data

You can find the stencil code and dataset for this assignment on github classroom at this link. For more details, please see the download/submission guide.

We have provided the following stencil code:

- `main.py` is the entry point of program which will read in the dataset, run the model, and print the results.
- `models.py` contains the `LinearRegression` model you will be implementing.

You should not modify any code in the `main.py`. If you do for debugging or other purposes, please make sure any additions are commented out in the final handin. The autograder will run on an unmodified version of `main.py`. All the functions you need to fill in reside in `models.py`, marked by `TODO`s. You can see a full description of them in the section below. To run the program, run `python main.py` in a terminal with the course environment set up.

Datasets

UCI Wine Quality

For the Linear Regression model, you will be using the UCI Wine Quality Dataset,[1] which contains information about various attributes of a wine and its corresponding quality rating (out of 10). It includes 4898 examples, which will be split into training and testing datasets using the `sklearn` library (you do not have to worry about this, as we have implemented it for you – you will be doing this in the next assignment!). Each example contains 12 attributes, and your model will train on the first 11 attributes (and a 12th bias term) to predict the last value. More information about the dataset can be found at https://archive.ics.uci.edu/ml/datasets/Wine+Quality.

The Assignment

In `models.py`, there are three functions you will implement. They are:

- `LinearRegression`:

[1] P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis.
Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.

- **train_solver()** uses matrix inversion to find the optimal set of weights for the given data.
- **predict()** predicts the values of test data points using the trained weights.
- **squared_error()** is a helper function that calculates the sum squared difference between two arrays.

In addition, three methods are provided for you. You should not change them.

- `train()` calls `train_solver`, which will train your model based on your implementation of matrix inversion.
- `loss()` computes the squared error loss of the predicted labels over a dataset.
- `average_loss()` computes the average squared error loss per prediction over a dataset.

Note: You are not allowed to use any off-the-shelf packages that have already implemented these models, such as scikit-learn or any linear regression functions. We're asking you to implement them yourself.

Linear Regression

Linear Regression learns linear functions of the inputs:

$$h_{\mathbf{w}}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$$

Note: the bias term can be included here by padding the data with an extra feature of 1. This has been already done for you in the stencil.

As we are using squared loss, the ERM hypothesis has weights

$$\mathbf{w} = \text{argmin}_{\mathbf{w}} \sum_{i=1}^{m} (y_i - h_{\mathbf{w}}(\mathbf{x}_i))^2$$

Squared Loss

For this assignment, we will be evaluating and training the Linear Regression model using mean squared loss (or L2 loss). Recall that the L2 loss function is defined as:

$$L_S(h_{\mathbf{w}}) = \frac{1}{m} \sum_{i=1}^{m} (y_i - h_{\mathbf{w}}(\mathbf{x}_i))^2$$

where y_i is the target value of i^{th} sample and $h_{\mathbf{w}}(\mathbf{x}_i)$ is the predicted value of that sample given the learned model weights. Your model should seek to minimize this loss through matrix inversion to learn the ERM hypothesis.

Matrix Inversion

You can assume that the examples in the training data are linearly independent. In that case, we showed in lecture that you can use matrix inversion to compute the vector of weights $\mathbf{w}$ that minimizes the squared loss. The equation to find $\mathbf{w}$, for a set of data points X and their labels $\mathbf{y}$ is

$$\mathbf{w} = (X^T X)^{-1} X^T \mathbf{y}$$

Note: X here is a matrix of examples stacked row-wise (i.e. $\mathbf{x_1}$ is the first row of X and so on). In lecture we saw that with X as a matrix of examples stacked column-wise (i.e. $\mathbf{x_1}$ is the first column of X and so on), the equation is equivalently:

$$\mathbf{w} = A^{-1}\mathbf{b} = (XX^T)^{-1} X\mathbf{y}$$

Implement the solution for $\mathbf{w}$ in `LinearRegression`, using the `np.linalg.pinv` function to calculate matrix inverses.

Project Report

Each programming assignment in this course will be accompanied by a short report in which you will answer a few guiding questions. These questions are included to promote critical thinking about the results of your algorithms, both mathematically and societally. By the end of this course you will not only be able to implement common machine-learning algorithms but also develop intuition as to how the results of a given algorithm should be interpreted and can impact the world around you.

The next section outlines some guiding questions that you should answer in your report. We ask that your final report be a PDF file named `report.pdf`, which must be handed in along with your code. You may use any program to create the PDF file, but we highly recommend using LaTeX. We have provided an example report available on our course website to get you started.

In future reports, you may need to write new code that generates some output (potentially a value or graph) that you will want to include in your writeup. For example, you may be asked to contrast the results of running the same algorithm on different datasets or to explore the effect of changing a certain hyperparameter in your algorithm. Please leave any code that you use in your final handin but make sure that it is **not** run by default when your program is run.

Guiding Questions

- Linear regression analysis makes several assumptions. For one, all observations in the data must be independent of each other (e.g., the data should not include more than one observation on any individual/unit). Furthermore, the data should avoid including extreme values since these will skew the results and create a false sense of relationship in the data. In general, linear regression gives more weight to cases that are far from the average. Can you think of any examples or datasets in which this might pose an issue?
- In this discussion post, the argument is posed that there is no ethically neutral statistical method, with specific reference to linear regression. What is the basis for this argument? Do you agree or disagree? Why?
- Suppose a machine learning researcher at a company learns a model to automate the hiring process. This company sells software based on this model to other companies looking to expedite their hiring. However, it is later discovered that the algorithm heavily favors members of a certain class unfairly.
  a) In this situation, who should be to blame for the unfair hiring?
  b) On whom does the responsibility fall to check the fairness of automated systems?

Grading Breakdown

We expect that the `LinearRegression` model should have a training loss of less than 0.55 on the training dataset.

As always, you will primarily be graded on the correctness of your code and not based on whether it does or does not achieve the accuracy targets.

The grading breakdown for the assignment is as follows:

Written Assignment	36%
Linear Regression	38%
Report	26%
Total	100%

Handing in

You will hand in both the written assignment and the coding portion on gradescope, separately.

Make your written assignment and project report into a single pdf, and upload it to gradescope under "Homework 2". Submit your hw2 github repo containing all your source code on gradescope under "Homework 2 Code". For questions, please consult the download/submission guide.

If you have questions on how to set up or use Gradescope, ask on Edstem! For this assignment, you should have written answers for Problems 1, 2, and 3.

Anonymous Grading

You need to be graded anonymously, so do not write your name anywhere on your handin.

Obligatory Note on Academic Integrity

Plagiarism — don't do it.

As outlined in the Brown Academic Code, attempting to pass off another's work as your own can result in failing the assignment, failing this course, or even dismissal or expulsion from Brown. More than that, you will be missing out on the goal of your education, which is the cultivation of your own mind, thoughts, and abilities. Please review this course's collaboration policy and if you have any questions, please contact a member of the course staff.