COMS 4701 Artificial Intelligence

Homework 4 - Coding

Due date: **SUNDAY** November 21st, 2021

We ask you to implement three small and separate machine learning models. We provide sample input/output file pairs for your reference. The file `plot_db.py` contains two functions for plotting data which you may choose to use, modify, reference and steal ideas from at will.

1. Perceptron
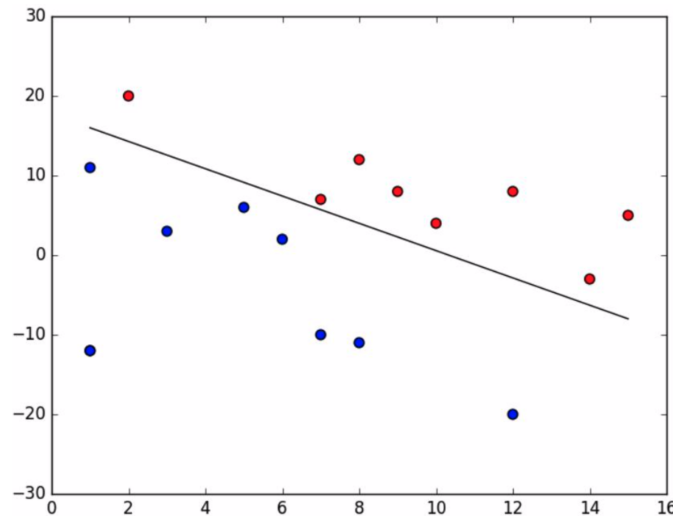
2. Linear Regression

Note: The Python <u>Pandas</u> library helps simplify a lot of the intermediate steps we ask from you below. Key functions include <u>reading</u> and <u>writing</u> to CSVs, matrix operations, and visualizing data. Examples are provided in `plot_db.py`.

1. **Perceptron**

   Implement the perceptron learning algorithm (PLA) for a linearly separable dataset. Your starter code includes `data1.csv` , where each line contains `feature_1`, `feature_2`, `label`. All values are numeric with labels 1 or −1. Take a look at the data input file. We suggest using `matplotlib` or <u>`pandas.DataFrame.plot`</u> (in `plot_db.py`) to view data, and <u>`pandas.DataFrame.describe`</u> to see stats.

   Write your PLA in `pla.py` in Python 3. Your program takes a csv of input data and a location to write the output csv with the weights from each iteration of your PLA, written as `weight_1`, `weight_2`, `b`. The weights in the last line of your output csv defines the **decision boundary** computed for the given dataset. Formatting is shown in `results1.csv`.

   NOTE: `results1.csv` values are purely filler and not representative of the values you should get. Feel free to visualize and include a screenshot of your final decision boundary in your README, like the one below:

   We will execute your code as follows: `$ python pla.py data1.csv results1.csv`

2. **Linear Regression**

   Use gradient descent to build a linear regression model for predicting height (m) using age (yr) and weight (kg), using data derived from CDC growth charts data.

Data Preparation and Normalization Load and understand the data from `data2.csv` [age(years), weight(kg), height(m)] remembering to add a vector column for the intercept at the front of your matrix. You'll notice the features are not on the same scale. What is the mean and standard deviation of each feature? Scale each feature (i.e. age and weight) by its standard deviation, so each scaled feature has a mean of zero. You do not need to scale the intercept. For each feature column, x, use the following formula:

$$x_{scaled} = \frac{x - \mu(x)}{\sigma(x)}$$

Gradient Descent Implement gradient descent to find a regression model in `lr.py`. Initialize your βs to zero. Recall the empirical risk and gradient descent rule as follows:

$$R(\beta) = \frac{1}{2n} \sum_{i=0}^{n} (f(x_i) - y_i)^2$$

$$\forall_j \ \beta_j := \beta_j - \alpha \frac{1}{n} \sum_{i=0}^{n} (f(x_i) - y_i) x_{ij}$$

You will run gradient descent with these nine learning rates: $\alpha \in 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10$ **exactly 100 iterations per α-value**, plus a tenth rate and number of iterations of your choice. To pick the tenth rate and loop count, observe how α affects the rate of convergence for the nine rates listed, then pick a rate and loop count you believe will perform well. Briefly explain your choice in your README.

The program should generate an output file containing ten lines, one for each (α, num_iters) hyperparameter pair. Each line contains: α, `num_iters`, `bias`, `b_age`, `b_weight`, expressed to your choice of decimal places (see example `results2.csv` in your starter code). Each line of this file defines the regression model your gradient descent method computed on the given data and hyperparameters.

We will execute your code as follows: `$ python lr.py data2.csv results2.csv`

Optional: Visualize the result of each linear regression model in three-dimensional space. You can plot each feature on the xy-plane, and plot the regression equation as a plane in xyz-space. Include these visualizations in your README file and label each one appropriately. Ex:
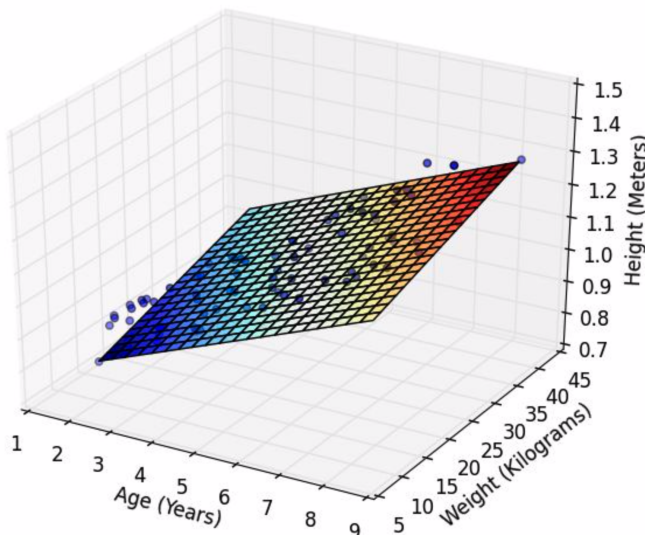
3. **SUBMISSION INSTRUCTIONS**

   Submit all the required files separately on Gradescope:

   - pla.py
   - lr.py