EMORY UNIVERSITY
Department of Computer Science
CS 334 Section 2 — Machine Learning
Fall 2024

Homework 1, Issued: Fri. 08/30, Due: Fri. 09/13 at 11:59pm

Submission Instructions: The homework is due on Gradescope in two parts.

- **Upload PDF to HW1-Written**: Create a single high-quality PDF with your answers to the non-coding problems. Your submission may be typed or handwritten (can be scanned or from a note-taking software), but the pages must be tagged with the relevant questions appropriately on Gradescope. You do not need to include code in the PDF unless otherwise instructed.
- **Submit code to the HW1-Code**: Questions marked with 🛠 on the left margin are graded by a Python autograder. Your submission must include only the following files: `sumsquare.py`, `palmer.py`, `README.txt` (no data files please). You may submit multiple times but be sure to upload *ALL* files when you re-submit; only the latest submission will be considered. The `README.txt` file must contain *a **SIGNED** honor code statement* that reads as follows:

```
THIS CODE IS MY OWN WORK, IT WAS WRITTEN WITHOUT CONSULTING CODE
WRITTEN BY OTHER STUDENTS OR LARGE LANGUAGE MODELS SUCH AS CHATGPT.
/* Your_Name_Here */

I collaborated with the following classmates for this homework:
<names of classmates>
```

1 Vectors & Planes Review (15 pts)

(a) (3pts) Given two 3-dimensional vectors $\vec{x}^{(1)} = [a_1, a_2, a_3]^T$ and $\vec{x}^{(2)} = [a_1, a_3, -a_2]^T$ in $\mathbb{R}^3$, write down an expression for calculating the angle between them. When is $\vec{x}^{(1)}$ orthogonal to $\vec{x}^{(2)}$?

(b) (3pts) Consider a hyperplane p in $\mathbb{R}^d$ that passes through the origin, which includes all points $\vec{x}$ such that $\theta_1 x_1 + \theta_2 x_2 + ... + \theta_d x_d = 0$. We can say that the d-dimensional vector $\vec{\theta} = [\theta_1, ..., \theta_d]^T$ describes p. Give one other d-dimensional vector that describes p. How many such alternative possible vectors are there? Explain.

(c) (4pts) Consider an arbitrary d-dimensional point $\vec{x}$ in $\mathbb{R}^d$, and a hyperplane through the origin described by normal vector $\vec{\theta} = [\theta_1, \ldots, \theta_d]^T$. The **signed distance** of $\vec{x}$ from the plane is the perpendicular distance between $\vec{x}$ and the hyperplane, multiplied by $+1$ if $\vec{x}$ lies on the same side of the plane as the vector $\vec{\theta}$ points and by -1 if $\vec{x}$ lies on the opposite side.

(i) Let p_1 be the plane consisting of the set of points for which $x_1 - 2x_2 = 0$. What is the signed distance of point $\vec{a} = [-1, 1]^T$ from p_1?

(ii) Let p_2 be the plane consisting of the set of points for which $x_1 - 2x_2 + 5 = 0$. What is the signed distance of point $\vec{a} = [-1, 1]^T$ from p_2?

(d) Consider a hyper-plane in a d-dimensional space, $\vec{\theta} \cdot \vec{x} + b = 0$ (that does not necessarily pass through the origin).

- (1pt) If we project a point onto this plane using orthogonal vector projection, can we uniquely recover the original point by "projecting" it back to the original space?
- (4pts) If so, write down the formula for performing the back-projection. If not, write down the formula for all such points that are projected to the same point $\vec{x}^{(0)}$ on the hyper-plane.

2 Linear Algebra Review (15 pts)

(a) Let $\vec{v}$ represent an m-dimensional column vector.

$$\vec{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}$$

i. (1pt) Express $\sum_{i=1}^{m} v_i v_i$ in terms of $\vec{v}$ and $\vec{v}^T$.

ii. (1pt) Write down an expression that is equivalent to (i) in terms of the norm $\|\cdot\|$.

iii. (2pts) Express matrix C, where $c_{ij} = v_i v_j \ \forall i, j$, in terms of $\vec{v}$ and $\vec{v}^T$.

(b) (5pts) Express

$$\sum_{i=1}^{n} \left(\vec{\theta} \cdot \vec{x}^{(i)} - y^{(i)}\right)^2$$

in terms of $\vec{\theta}$, X and $\vec{y}$, where $X = [\vec{x}^{(1)}, \vec{x}^{(2)}, \ldots, \vec{x}^{(n)}]^T$ and $\vec{y} = [y^{(1)}, y^{(2)}, \ldots, y^{(n)}]^T$.

(c) (3pts) Let A and B be two matrices compatible with multiplication (i.e., both AB and BA are well defined). Is it true that $(AB)^T = A^T B^T$? Prove or give a counterexample.

(d) (3pts) Let $A \in \mathbb{R}^{n \times p}$, $B \in \mathbb{R}^{p \times n}$, and $C = AB \in \mathbb{R}^{n \times n}$. Suppose $p < n$. Is it possible that $\text{rank}(C) = n$? Why or why not?

3 Vectorization Comparison (25 pts)

In this problem, we will perform a speed comparison of two different implementations of the same function, one using standard for-loops and another using vectorized numpy operations. The two functions should return approximately the same result (there may be subtle differences past 5 decimal places due to floating point errors). The template for the problem is in `sumsquare.py`.

(a) (3pts) Implement `gen_random_samples` which, given an input parameter `n`, generates a `np.array` of `n` random numbers using a normal (Gaussian) distribution of mean 0 and variance 1. Hint: you might find `numpy.random.randn` useful.

(b) (4pts) Fill in the function `sum_squares_for`. This function must compute the sum of squares using a `for` loop for each element of the array.

(c) (4pts) Fill in the code for `sum_square_np`. Compute the sum of squares using `numpy.dot`.

(d) (6pts) Implement the function `time_ss` which given an input list of integers (e.g., [1, 50, 100]), will yield a Python dictionary where the keys are `'n'`, `'ssfor'`, `'ssnp'` and the values are lists. The list associated with `'n'` should be the same as the input. The lists associated with `'ssfor'` and `'ssnp'` should be the time in seconds for the `sum_squares_for` and `sum_square_np` method calls, respectively. In other words, if the input is the list `[1,5,100]`, the return will have the format of `{'n':[1,5,100], 'ssfor':[5,6,7], 'ssnp':[2,3,4]}`.[1] You will find it useful to use the `timeit` module where you can calculate the execution time of a function in the following manner:

```
start = timeit.default_timer()
function_to_time()
elapsed = timeit.default_timer() - start
```

(e) (3pts) Implement the function `timess_to_df` which given an input dictionary formatted as specified by the output in (d) yields a `pandas.DataFrame` that contains 3 columns `n`, `ssfor`, `ssnp` (named exactly as such and in this order) that contains the time comparison between `sum_squares_for` and `sum_square_np`.

(f) (5pts) Use (d) and (e) to compares the two types of code for n ranging from 10 to 10M. Using the dataframe, create a *single plot* using Python that shows the data points for the two methods. Each set of points (related to either the for loop or the numpy loop) should have a different shape and color, with the legend visible in the plot. Make sure to label both axes of your plot as well. **Attach both your code and the generated plot in the Written Gradescope submission**.

4 Pandas DataFrame (15 pts)

For this problem, you will become familiar with the basic functions of `pandas.DataFrame`. We will be using the Palmer Penguins dataset, `penguins.csv`, for this and the following problem. The template code can be found in `palmer.py`. The dataset contains measurements from 344 penguins collected from the Palmer Archipelago in Antarctica. There are 3 species of penguins: Adelie, Chinstrap, and Gentoo. More details of the dataset can be found at https://github.com/allisonhorst/palmerpenguins?tab=readme-ov-file. For this problem, make sure you are not hardcoding your solution to the Palmer Penguins dataset, it should work for any comma-separated value (CSV) file.

(a) (2pts) Fill in `load_csv` that imports the file located at `inputfile` as a pandas frame. *Do not hardcode the filename*, so if inputfile is a different file, your function will load it as a pandas dataframe.

(b) (4pts) Implement the function `remove_na` that given an input dataframe and column name removes the rows with the NaN values and returns the dataframe. If there are no rows with an NaN for the column, your function should still return a valid dataframe.

(c) (4pts) Fill in the function `onehot` that will take in a dataframe and a column name and performs one-hot encoding of the column specified. The function should return a dataframe that does not contain the specified column and includes the one-hot encoding of that column. One-hot encoding takes a

[1]These are not actual times from the solution, simply just an example with made-up numbers.

categorical variable with k categories and converts it to k new binary variables where the value in the column represents the boolean (is it that value or not). For example, one-hot encoding of the feature `sex` with 2 categories, `male` and `female`, will result in 2 new features, `male`, `female` where (1,0) will be the values for a male penguin and a (0, 1) will be the values for the female penguin in the 2 new columns.

(d) (5pts) Implement the function `to_numeric` where given an input dataframe will drop the non-numeric features and convert the dataframe to a `numpy.ndarray` object. You can assume that NaNs have been removed and any categorical variable that should be kept has been one-hot encoded.

5 Visualization Exploration (30 pts)

For this problem, we will explore the visualization capabilities in Python using the Palmer Penguins dataset and the functions you wrote in the above problem. Create your own Python file that contains the commands to load the dataset and plot the features. Make sure to submit this file to HW1-Code and that it is not a part of any of the files above to avoid breaking the autograder. Make sure your code is well-documented. You may want to consider `pandas.DataFrame.boxplot` and `pandas.DataFrame.plot` for the visualization in this problem. Other alternative packages are `matplotlib` or `seaborn`. Each plot should contain sufficient information alone to be able to interpret it. This means there should be appropriate axis names, labels (i.e., axis tick marks with labels at the tick locations), titles, and legends if there are different colors or shapes.

(a) (6pts) Create 2 histograms for the counts of the species: (1) Use the original data (from a) and (2) Drop the rows with a NaN in the numeric features (from b). Does dropping data with incomplete measurements (i.e., NaN) change the species distribution? Make sure you justify your answer. For full credit, your written assignment **must have the 2 histograms, the answer to the question, and a justification for your answer based on the 2 histograms**.

(b) (8pts) For each numeric feature (i.e., culmen length, culmen depth, flipper length, body mass), plot the boxplot of the distribution of the feature as a function of the species type. In other words, for culmen length, there should be one plot with 3 boxes (i.e., Adelie, Chinstrap, and Gentoo). For full credit, your written assignment **must have 4 boxplots** that have appropriate axes labels, legends (if applicable), and titles.

(c) (10pts) Explore the correlation between all the different pairs of the 4 numeric features by plotting a scatterplot of each feature versus the other 3 features with each species containing a different color or shape (and the legend appropriately labeled). Note that you only need to produce 6 different scatterplots. For full credit, your written assignment **must have 6 plots (or subplots)** that have appropriate axes labels, legends (if applicable), and titles.

(d) (6pts) Based on your exploration of the data in parts b and c, come up with a set of "rules" that could classify the species type. You **must justify your rules based on the results from b and c** to receive full credit.

REMEMBER: Submit your completed assignment by 11:59pm on Sept. 13th to Gradescope.