CS461 Homework 3

Due: Nov. 17 11:59 pm

1. [Decision Tree] You will build a decision tree to determine whether or not a child goes out to play.

Day	Weather	Temperature	Humidity	Wind	Play
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No

1.1 Calculate the information gain for each feature and select the feature with the highest information gain to serve as the root of the decision tree. Draw a root and split the ten training data points into some groups based on the value of the selected root feature.

$$k* = \arg\max_k I(X_k; Y) = H(Y) - H(Y|X_k)$$

1.2 Repeat the two procedures: (1) selecting a feature and (2) splitting the data points until the leaf nodes of the tree achieve complete purity.

1.3 [Extra Points: 10 points] Try pruning your tree. You need to find a subtree (T') minimizing the criterion $C(T')$ below. Based on the criterion computation, do you think we need to prune the tree found in 1.2? Please show how different range of lambdas results in the different decisions on tree pruning.

$$C(T') = \sum_{\tau=1}^{T'} Q(\tau) + \lambda \cdot |\text{num of leaves in } T'|$$

$$Q(\tau) = \text{entropy of a leaf in } T' \text{ (measure of impurity)}$$

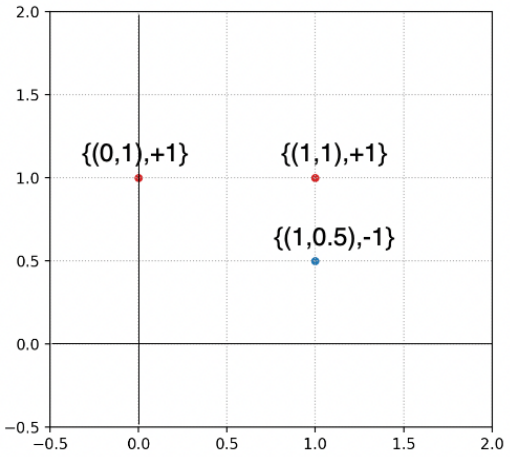
2.[Perceptron] The Perceptron algorithm finds a decision boundary for binary classification below.

$$\begin{cases} \delta_w(x_1, x_2) = +1 & w_1 x_1 + w_2 x_2 > 0 \\ \\ \delta_w(x_1, x_2) = -1 & w_1 x_1 + w_2 x_2 \leq 0 \end{cases} \tag{1}$$

2.1 Assume a data set consists only of a single data point $\{(x_1, x_2), +1)\}$. How many iterations will be required until it finds a decision rule when the initial $w_0 = (0, 0)$ and step size $\eta = 1$?

2.2 How many iterations will be required until it finds a decision rule if the initial weight vector w_0 was initialized randomly and not as the all-zero vector?

2.3 Suppose you have the three data points below. Please complete the iterative updates for w_i in Perceptron algorithm. The initial $w_0 = (0, 0)$ and step size $\eta = 1$; the point $(0, 1)$ is detected as misclassification at the first iteration so w is updated by the point: $w_1 = w_0 + 1 \cdot (0, 1)$



iteration	$\vec{w}$
0	$w_0 = (0, 0)$
1	$w_1 = (0, 0) + (0, 1) = (0, 1)$
2	...

3. (GDA: Gaussian Discriminant Analysis) We will build a binary classifier by using GDA.

3.1. Suppose someone gave you a decision rule (classifier) based on GDA, as shown below. Please use "./data_1/train.npz" and write the code "train3_1.py" to estimate the four statistics in the table below. $\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{(2\pi\sigma^2)}} \exp\{\frac{-1}{2\sigma^2}(x-\mu)^2\}$

$$\begin{cases} \delta(x) = +1 & \mathcal{N}(x|\mu_{pos},\sigma^2_{pos}) \geq \mathcal{N}(x|\mu_{neg},\sigma^2_{neg}) \\ \delta(x) = -1 & \mathcal{N}(x|\mu_{pos},\sigma^2_{pos}) < \mathcal{N}(x|\mu_{neg},\sigma^2_{neg}) \end{cases} \tag{2}$$

class	mean (μ)	var (σ^2)
class +		
class −		

3.2. Write the code "test3_2.py" to predict the class for the test data "./data_1/test.npz". What is the test accuracy?

3.3. How would you improve your classifier? Is the decision rule given above optimal? Write a new code "test3_3.py" modifying "test3_2.py" and report new test accuracy. (hint: recall MAP rule.)

3.4. Your will build a GDA classifier for 2D data. Use the data:"./data_2/train.npz" and write the code "train3_4.py" to estimate the four statistics and complete the table below.

class	mean ($\vec{\mu}$)	COV (Σ)
class +		
class −		

3.5 Write the code "test3_5.py" to predict the class for the test data: './data_2/test.npz". What is the test accuracy?

3.6 [Extra Points: 10 points] The 2D data is generated by the densities specified below. Write the code "test3_6.py" based on the information and compute the new accuracy. Determine whether there is any significant change compared to the accuracy obtained in 3.5. Based on your accuracy comparison, discuss how GDA provides a reasonable framework for classification.

$$\begin{cases} x_{pos} \sim \mathcal{N}([0,0]^t, I) \\ \\ x_{neg} \sim 0.5 \times \mathcal{N}([0,2]^t, I) + 0.5 \times \mathcal{N}([0,-2]^t, I) \end{cases} \tag{3}$$

4. [Logistic Regression] You will implement a spam mail detector by using logistic regression and enron data set (https://huggingface.co/datasets/SetFit/enron_spam/tree/main). The original data samples are plain texts, so a data preprocessing will be needed to convert text data to numerical values.

4.1 [Data Reprocessing] Run "preprocessing.py" to convert text data to numerical values. It will generate "spam_ham.csv". Please briefly explain the text vectorization process: tf-idf (Term Frequency-Inverse Document Frequency).

4.2 [Dimensionality Reduction] Write the code "data4_2.py" to perform PCA dimensionality reduction on the 2000-D the data in "spam_ham.csv". We will reduce the data dimension to 50-D and split $4,000$ data samples into train: $3,500$ and test: 500. Please save them as "train4_2.npz" and "test4_2.npz".

4.3 [Logistic Regression] Write the code "train4_3.py" to train a logistic regression classifier using only NumPy library. Implement a gradient descent algorithm to find the global minimum of the Negative Log Likelihood (NLL) function defined below. Hint: You may use a step size about $1.0e-4$ and detect convergence when the change in NNL is less than 10, but you are free to choose your own values.

$$J(w) = -\ln P(t|w) = \sum_{n=1}^{N} -t_n \ln \sigma(w^t x_n) - (1-t_n) \ln \left(1 - \sigma(w^t x_n)\right)$$

4.4 [Logistic Regression] Report the train and test accuracy of your spam detector.

4.5 [Extra Points: 10 points] "mail.txt" is the text extracted from a spam email. Please test the text using your classifier and report the result whether it is classified as a spam.