

Homework 2

Instructions on homework submission. Please upload your solutions to Gradescope. You need to use this template for your written submission. You may use \LaTeX or print the template and hand-write your answers then scan it in. Failure to use the template may result in a penalty. Handwritten submissions should be clearly legible. If your writing is not legible, you will not be awarded marks. Please make sure your answers are fully included in the given space for each question.

Instructions on programming assignments. You need to submit your code for Question 2 and for Question 4 in the Homework 2 (programming) section on gradescope.

For Question 2, your submission will be graded by Autograder. Follow the detailed instruction in Question 2.

For Question 4, the plots generated by your code needs to be in the pdf file (written submission). However, **your answers for Question 4 in the written submission will not be graded if you do not submit the codes in the programming part.** You are required to use Python as the programming language. Please submit your .py file or .ipynb file as submission with instructions on how to execute your code.

For the integrity and conciseness of grading, please ask questions to:

Young Je: Problem 1, Problem 2, Problem 3

Parker: Problem 4

1. [Logistic Regression, Theory 20 pts] Consider the following logistic regression model for two-class classification

$$P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + w_1 X_1 + w_2 X_2)}$$

for classifying the data in Figure 1.

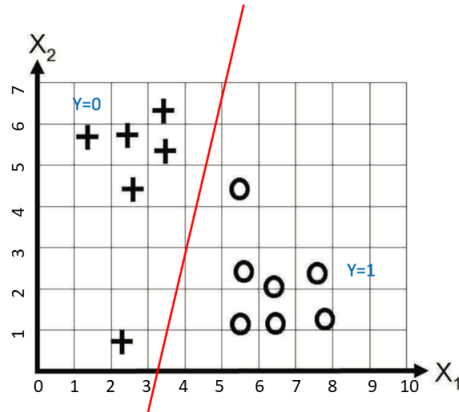


Figure 1: Training data

- (a) [2 pts] Find the training sample with the highest probability of class 1. Find the training sample with the highest probability of class 0. You may specify the sample using coordinates or the value range of X_1 and X_2 . Provide brief justification of your answer.

- (b) [5 pts] Use a linear equation to describe the decision boundary (red line) in Figure 1.

- (c) [3 pts] Consider modifying this classifier so that it uses only feature X_1 . Provide the equation for this decision boundary. Consider modifying this classifier so that it uses only feature X_2 . Provide the equation for this decision boundary. Which of the two classifiers will perform better on the training data? Provide justification.

- (d) [5 pts] Consider using decision tree on the data in Figure 1, instead of logistic regression. What is the advantage and disadvantage of decision tree over logistic regression?

- (e) [5 pts] Consider using Naive Bayes classifier and decision tree on the data in Figure 1. What is the advantage and disadvantage of Naive Bayes over logistic regression?

2. [Logistic Regression, Implementation 10 pts] This problem is graded with Autograder. Follow the below instruction, and implement in **reg.py**. Do not rename this files, and do not change anything other than the functions you are asked to implement. Otherwise, Autograder might unable to grade your work properly. You are only allowed to use Numpy in this problem.

In this part, you need to train a linear binary classification model $F(x) = \langle w, x \rangle$ with logistic regression (for simplicity, the bias term b is omitted here). For training samples $\{(x_i, y_i)\}_{i=1}^n$ where $x_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$, you need to find a vector $w \in \mathbb{R}^d$ that minimizes the following loss function:

$$\ell(w) = \frac{1}{n} \sum_{i=1}^n [\log(1 + e^{\langle w, x_i \rangle}) - y_i \cdot \langle w, x_i \rangle]$$

which is the negative log likelihood. Minimizing this loss function is a convex optimization problem, which means that in most cases, there exists a unique minimizer. Here, you need to find this minimizer with gradient descent, which runs as follows:

Algorithm 1 Gradient Descent

Require: Learning rate $\eta > 0$, stopping criterion $\rho > 0$

Initialize w

while True **do**

 Compute gradient $\nabla_w \ell(w)$

 If $\|\nabla_w \ell(w)\| < \rho$ then break

$w \leftarrow w - \eta \cdot \nabla_w \ell(w)$

end while

Task: In `reg.py`, implement three functions: `compute_loss` for computing $\ell(w)$, `compute_grad` for computing $\nabla_w \ell(w)$, and `train` for training the model with gradient descent.

3. [Decision Tree, 20 pts] A group of students ran several experiments to design a system to determine the effectiveness of initial drug screening campaign. Your friend took notes on the 8 experiments he has ran, and he wants to construct a decision tree that would predict whether or not drug has effect based on 3 binary features:

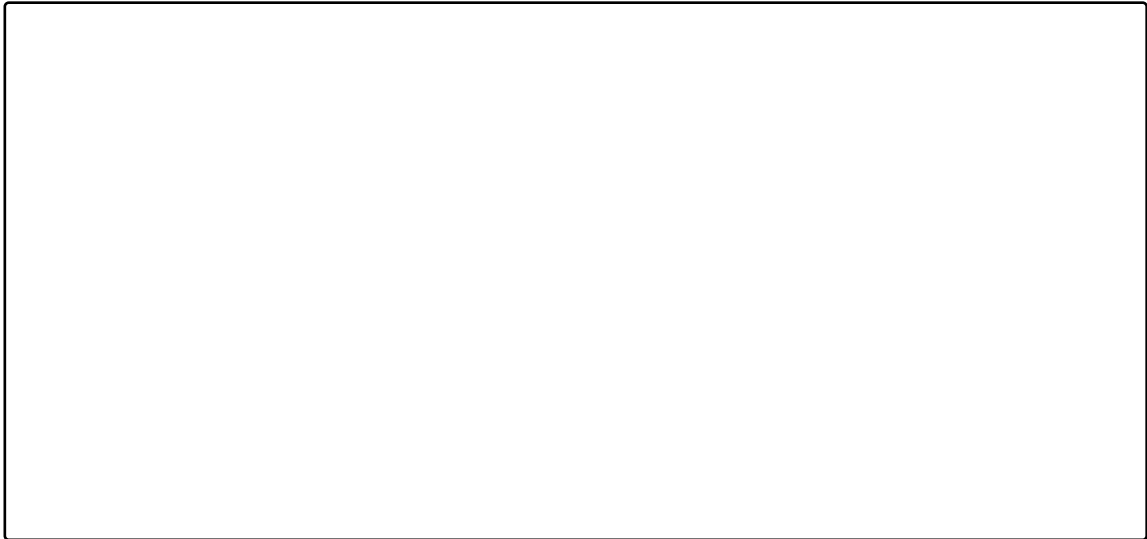
- x_1 = The drug molecule has benzene ring
- x_2 = The drug molecule has more than 2 negative ions
- x_3 = The drug molecule showed less than 2 angstrom distance from pocket

Movie	x_1	x_2	x_3	Like
$\bar{x}^{(1)}$	False	False	True	No
$\bar{x}^{(2)}$	True	False	True	Yes
$\bar{x}^{(3)}$	False	False	False	Yes
$\bar{x}^{(4)}$	False	True	True	Yes
$\bar{x}^{(5)}$	True	True	True	No
$\bar{x}^{(6)}$	True	False	False	Yes
$\bar{x}^{(7)}$	True	True	False	Yes
$\bar{x}^{(8)}$	False	True	False	Yes

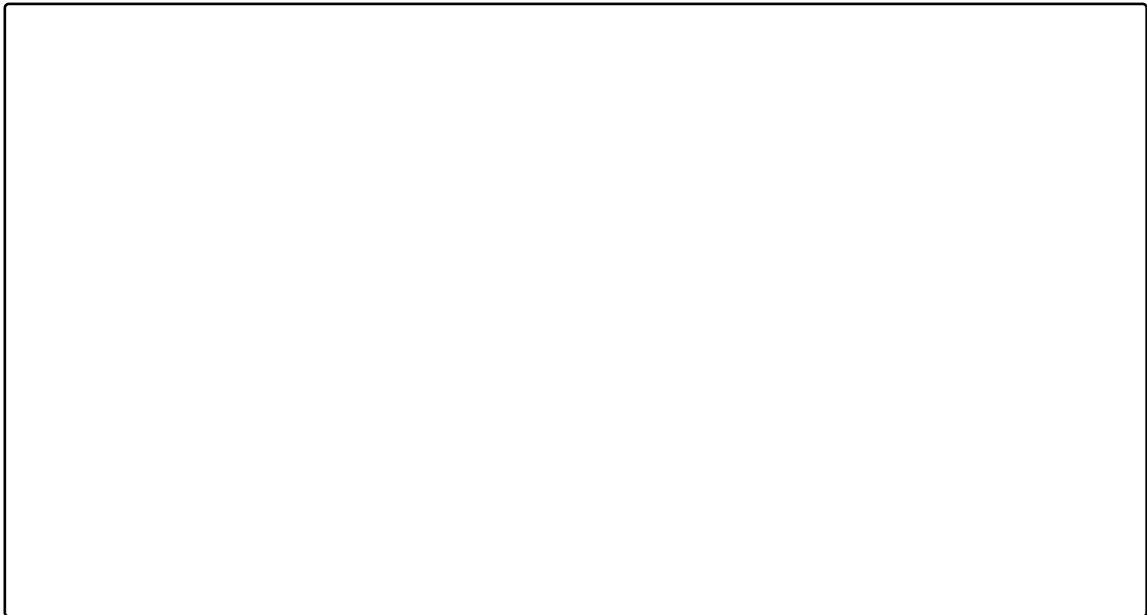
- (a) [5 pts] Use the above dataset to construct a decision tree, where each split is performed on the feature with the highest information gain. Use the following stopping conditions in the decision tree algorithm:

- Tree depth is 3
- Information gain is 0

Output majority label if any of the stopping criteria are met, or a positive label if both classes are equally represented. **NOTE:** To help you out, the TAs have decided to tell you that the split at the root made by this algorithm will be on the feature x_3 .



- (b) [5 pts] Does your decision tree from previous part minimize the training error? If not, report the smallest training error that a decision tree can achieve on this dataset and draw the decision tree.



- (c) [10 pts] During recitation, we introduced KL-Divergence to provide another view of information gain. Here we have formal definition of KL-Divergence:

Let p and q be two discrete probability distributions defined over the same event space. Formally, for $i \in \{1, \dots, n\}$, let $p(i)$ be the probability of event i occurring under distribution p and $q(i)$ be the corresponding probability under distribution q . The **Kullback-**

Leibler (KL) divergence between p and q is defined as

$$D(p||q) = \sum_{i=1}^n p(i) \log \frac{p(i)}{q(i)}$$

KL-Divergence is proven to have non-negative: $D(p||q) \geq 0$. Using this, we want to show that information gain is always non-negative. In detail,

In a sample S , the information gain of a variable Y with respect to a target variable X is the expected reduction in entropy of a target variable X after gaining insight into Y :

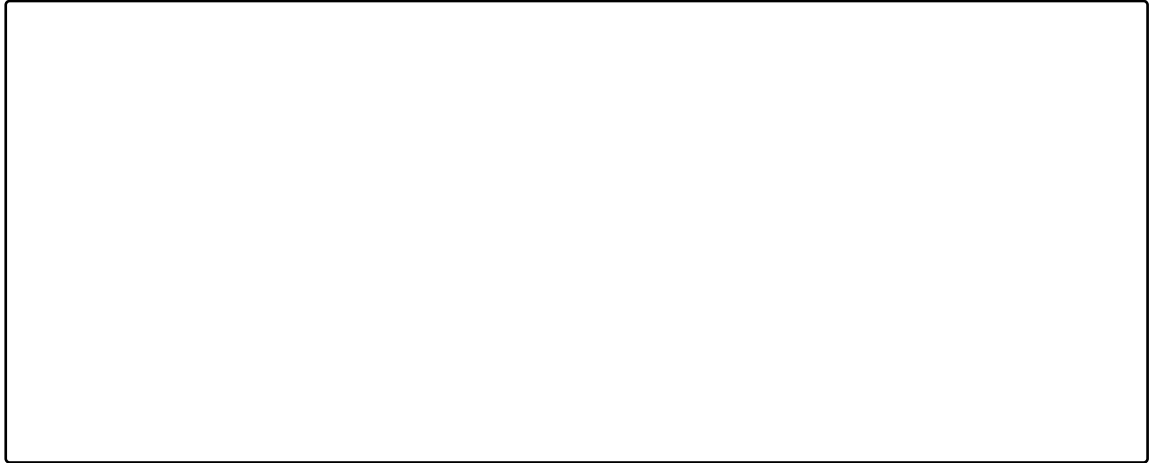
$$Gain(S, Y) = H_S(X) - H_S(X | Y)$$

Show that information gain is always non-negative. Show all steps of your work.

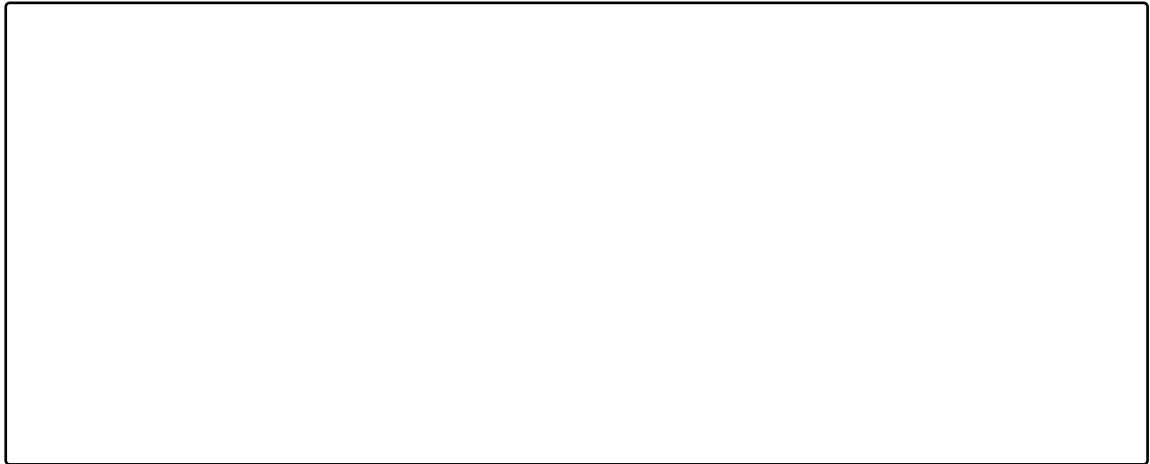
4. [K-Means Clustering, 50 pts] In this homework problem, you will implement K -means algorithm and apply it to cluster genes using the mouse gene expression data from three brain tissue types, HIP, PFC, and STR. For each tissue type, data are provided in two files, one for expression data, where the rows are for mice/samples and the columns are for genes, and the other file for gene names. Answer the following questions on the tissue type HIP. Explore PFC and STR in the bonus problem.

Note: For plotting the correlation coefficient matrix you can use the pyplot function `imshow()`. For parts (c) and (d) you should show all the correlation coefficient matrices in one plot. You can achieve this with the `pyplot.subplots()` function. Each subplot should be titled with the objective function value (parts (c) and d) and the K value (part (d) only).

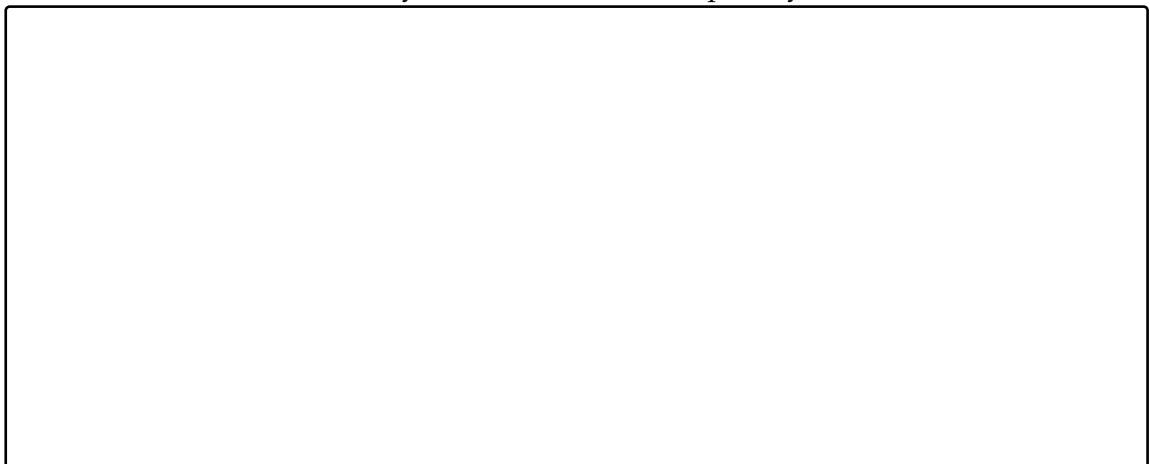
- (a) [15 pts] Run K -means algorithm with $K = 3$ with the initialization of K means given in 'test_mean.txt'. Plot the objective (y -axis) over iterations (x -axis). What is the objective value at convergence? (Make sure you use the given initialization. The initialization is given for the grading purpose.)



- (b) [10 pts] Run K -means algorithm with $K = 3$ with your own random initialization. Plot the correlation coefficient matrix of a) the raw data and b) the data after you group the columns of the expression matrix according to the clusters found by K -means. Plot these as an image of $N \times N$ correlation matrix for N genes.



- (c) [10 pts] With $K = 3$, try 10 random initializations. Show the correlation coefficient matrix as above and the corresponding objective at convergence for each initialization. Which of the 10 results would you choose for follow-up analysis?



- (d) [15 pts] Run K -means algorithm for $K = 3, \dots, 12$, with 10 random initializations for

each K . Show the correlation coefficient matrix and objectives at convergence for different K , with your best choice of the 10 runs for each K . What do you think is the best K ? Why?