**CS 430/530 - Assignment 3**

**Due date – Friday, October 13th, 2023, on or before 11:59 pm.**

**Assigned - Monday, October 2nd, 2023**

**Please answer all the questions. No late submissions will be accepted. The solutions for all questions except the programming project should be neatly typed or written on a document and submitted through Canvas on or before the deadline. It can also be submitted in class in hard copy before the deadline.**

**Problem 01: 10 points**

Please mention which library function you are using to optimize the cost function and why you are choosing this function.

We chose to use scipy’s optimize.minimize function, with the BFGS algorithm. We choose BFSG because it is a quasi-Newton method that uses the first and second derivatives of the function to find the minimum. It seemed like a good improvement over gradient descent, while still being in the same family of algorithms.

**Problem 02 (Programming project) [190 points]**

**This problem needs to be submitted separately through email. See submission section below.**

# Problem statement: Logistic Regression

In this exercise, you will implement logistic regression and apply it on the supplied dataset. The supplied data set is one of the classic datasets used in the literature on classification methods and widely used in statistics and machine learning. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are not linearly separable from each other.

You will build a logistic regression model to predict the class of a given iris plant. Your task is to build three classification models that estimate the class of a given test/validation sample consisting of the four features measured on an iris plant.

The link for the dataset and its description is shared below:

<https://archive.ics.uci.edu/dataset/53/iris>

Also the data is provided on canvas as “iris.data”

**Implementation:**

### **Implementing sigmoid function**

Before you start with the actual cost function, recall that the logistic regression hypothesis is defined as:

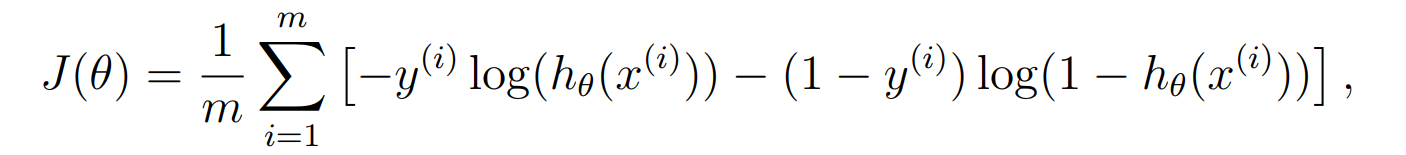
h*θ*(x) = g(θ*T* x),

where function g is the sigmoid function. The sigmoid function is defined as:

Your first task is to implement this function in sigmoid.py so that it can be called by the rest of your program. When you are finished, try testing a few values by calling sigmoid(x)using the Python command line. For large positive values of x, the sigmoid should be close to 1, while for large negative values, the sigmoid should be close to 0. Evaluating sigmoid(0)should give you exactly 0.5. Your code should also work with vectors and matrices. **For a matrix, your function should perform the sigmoid function on every element.**

### **Cost function and gradient**

Now you will implement the cost function and gradient for logistic regression. Recall that the cost function in logistic regression is:



and the gradient of the cost is a vector of the same length as θ where the jth

*θ*

*θ*

element (for j = 0, 1, . . . , n) is defined as follows:

A picture containing text, clock, watch

Description automatically generated

**Initialize the parameters θ to zero.**

1. **Learning parameters using python library function for unconstrained optimization**

In the previous assignment, you found the optimal parameters of a linear regression model by implementing gradient descent. You wrote a cost function and calculated its gradient, then took a gradient descent step accordingly. This time, instead of coding gradient descent steps yourselves, you will use python library functions for optimization (for example you can look at SciPy library). **Please note that** **you are not allowed to use a library function for logistic regression directly.** Whichever optimization function you use, you need to supply at the most the following information:

* The initial values of the parameters we are trying to optimize.
* The implementation of the cost function and the gradients of the cost function.

﻿

For logistic regression, you want to optimize the cost function J(θ) with parameters θ. Concretely, you are going to use the library function to find the best parameters θ for the logistic regression cost function, given a fixed dataset (of X and y values). Depending on the library function you are using you may or may not need to set the learning rate. **In case you need to set a learning rate, please mention which value you use for the final results.** After learning the parameters, you can use the model to predict the class of a given iris plant.

**The targeted tasks are as follows:**

1. **Read the data description carefully. In this assignment you will create a training set with 80% of the data and a validation set with the remaining 20% of the data. Note that you are supposed to do a random split of the dataset for this homework. You need to make sure that the validation set contains elements from all the three classes.**
2. **You will train three classifiers on the training set for the following:**
   1. **Classifying 1 against 2 and 3**
   2. **Classifying 2 against 1 and 3**
   3. **Classifying 3 against 1 and 2**
3. **Computing validation accuracy of the classifier**

Once you have found θ using the optimization method of your choice, you will predict the target classes based on the cases mentioned above for the samples in the validation set. In this part your task is to write a code to predict the class on the validation set using the estimated parameters. The code should produce “1” if the class is predicted correctly and “0” otherwise. **For all three above mentioned cases, you will report the confusion matrix, accuracy, and precision for all the three classifiers.**

#### **Confusion matrix:**

The confusion matrix, also known as the error matrix, is depicted by a matrix describing the performance of a classification model on a set of test data.

A diagram of a positive and false negative

Description automatically generated

**True positive (TP):** Observation is predicted positive and is actually positive.

**False positive (FP):** Observation is predicted positive and is actually negative.

**True negative (TN):** Observation is predicted negative and is actually negative.

**False negative (FN):** Observation is predicted negative and is actually positive.

#### **Accuracy:**

Accuracy gives the proportion of the total number of predictions that were correct:

Accuracy=

#### **Precision:**

Precision or the positive predictive value, is the fraction of positive values out of the total predicted positive instances. In other words, precision is the proportion of positive values that were correctly identified:

Precision=

**Programming languages and formatting**

You should use preferably python or MATLAB and should implement the algorithms yourself.

Please do NOT use any library function which provides an implementation of logistic regression cost function and gradients. You will get ZERO otherwise.

**Submission Guidelines**

Your submission must have the following:

* A **README file** that describes how the code can be compiled and run. Also list any external dependencies that need to be satisfied for compiling and running the code. If your code fails to compile YOU GET A ZERO.

Please e-mail your submission to [hs0111@uah.edu](mailto:hs0111@uah.edu) by 11:59 PM on Friday, October 13th, 2023. DO NOT SUBMIT THROUGH DROPBOX or CANVAS. They will not be accepted and will result in late penalty. Put all your materials for the PROGRAMMING project ONLY in a folder with your name and then create a zip file out of it. Email the zip file to the TA.

**Point Breakdown**

For loading the data: 5 points

Splitting the data into test set and validation set: 15 points

For implementing sigmoid function: 10 points

For correct implementation of cost function: 10 points

For correct optimization: 20x3=60 points

Computing optimal theta and reporting: 10x3=30 points

Reporting confusion matrix: 10x3=30 points

For computing accuracy: 3x5=15

For computing precision: 3x5=15