CSCI 3320: Fundamental of Machine Learning **2019-2020 Term 2**

Programming Assignment 2

Instructor: Prof. John C.S. Lui Due: 23:59 on Friday, May. 1, 2020

1 Introduction

This programming assignment consists of two parts.

- The first part will guide you to write a logistic linear discriminator for binary classification and solve it by gradient descent.
- Secondly, you will learn how to implement decision tree in python. It contains gini index calculation, binary decision tree building and a decision tree depth experiment.

2 Binary Logistic Classification

In this section, we will use logistic discriminate to do a binary classification task.

In `ex1.py`, we generate two clusters of data points and split the data to training and test data with the following script:

```
n_samples = 1000
centers = [(-1, -1), (5, 10)]
X, y = make_blobs(n_samples=n_samples, n_features=2, cluster_std=1.8,
                  centers=centers, shuffle=False, random_state=42)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)
```

Note: please do **not** change the parameter `random_state` in the file `logistic_clf.py`.

2.1 Logistic Function

The logistic function is

$$L(x) = \frac{1}{1 + e^{-x}}.$$

Complete `logistic_func()` in `ex1.py`.

2.2 Gradient Descent Update Rule

Set $g(x|\mathbf{w}, w_0) = \mathbf{w}^T \cdot x + w_0$ as the linear function. The update rule of logistic regression is as follows:

$$\begin{aligned} w_0 &\leftarrow w_0 + \eta \cdot \sum_{d \in \mathcal{D}} (y_d - L(g(x_d))) \\ w_i &\leftarrow w_i + \eta \cdot \sum_{d \in \mathcal{D}} (y_d - L(g(x_d)))x_d(i) \end{aligned}$$

Complete `train()` in `ex1.py`.

Hint: the convergence of gradient descent can be measured by weight's change, like $|w^{i+1} - w^i| < 10^{-4}$.

2.3 Gradient Descent Update Rule in Matrix Form

There is also a matrix form of gradient descent update:

$$\mathbf{W} \leftarrow \mathbf{W} + \eta \cdot (\mathbf{y} - L(\bar{\mathbf{X}}\mathbf{W}))^T \bar{\mathbf{X}},$$

Where $\mathbf{W}^T = [w_0, \mathbf{w}^T]$, $\bar{\mathbf{X}} = [\mathbf{1}, \mathbf{X}]$ is the train feature with an all 1 vector, the logistic function L is applied to each entries of its input vector.

Complete `train_matrix()` in `ex1.py`.

2.4 Prediction Rule

Use the prediction rule of logistic classification, for input x:

$$C(x) = \begin{cases} 1, & p(x) \geq 0.5 \\ 0, & otherwise \end{cases},$$

where $p(x) = Logistic(g(x|\mathbf{w}, w_0))$.

Complete `predict()` in `ex1.py`.

2.5 Experiments

Use *ex1.py* to test both `train()` and `train_matrix()` function. Copy down both figures and number of wrong predictions to `Assignment2.pdf`.

3 Decision Tree Classification

3.1 Calculate Gini Index of a Split

Gini index is used in CART algorithm. The Gini index of a set measures the set's *impurity*:

$$Gini(S) = 1 - \sum_{i=1}^{C} p_i^2,$$

where C is the number of classes, p_i is the prior probability of class i in the set. When we split a set S into S_1 and S_2, the Gini index of this split is the summation of weighted Gini index of sets by the size of set:

$$Gini(split) = Gini(S_1)\frac{|S_1|}{|S|} + Gini(S_2)\frac{|S_2|}{|S|},$$

where $|\cdot|$ is the size of a set.

Complete the function `gini_index()` of `ex2.py`.

3.2 Split A Set

The `get_split()` function of `ex2.py` find the optimal split plane of a set S and split it to left set S_l and right set S_r. They are two children of the set. We select the optimal split plane (for example, $x = 1.2$ or $y = 2$) from the feature values of the set's data points.

Complete the function `get_split()` of `ex2.py`.

Hint: the optimal split of a set is the one with the smallest gini index.

3.3 Build up Decision Tree

There are two criterion for stopping split a set:

- The depth (height) of the decision tree is more than `max_depth`;
- The number of point in the set is no more than `min_size`.

Once the set meets any of the conditions, we don't split it anymore. The set is a `leaf`.

Complete function `split()` of `ex2.py`.

3.4 The Depth of Decision Tree

Use *ex2.py* to test the influence of decision tree's depth in classification. Try `max_depth=3,5,7`. Copy down these three figures and number of wrong predictions to `Assignment2.pdf`. Compare three figures and their wrong predictions times. Write down possible reasons of the result.

4 Submission

Instructions for the submission are as follows. **Please follow them carefully.**

1. Make sure you have answered all questions in your report.
2. Test all your Python scripts before submission. Any script that has syntax error will not be marked.

3. Zip all Python script files, i.e., the *.py files in asgn2.zip (Please do not change the filenames of the scripts.) and your report (Assignment2.pdf) into a single zipped file named <student-id>_asgn2.zip, where <student-id> should be replaced with your own student ID. e.g., 1155012345_asgn2.zip.

4. Submit the zipped file <student-id>_asgn2.zip via Blackboard System no later than 23:59 on Friday, May. 1, 2020.