COMP3314_2C_2022 Machine Learning

Programming Assignment 1:

Logistic Regression and Random Forest

Start date: February 20, 2023
Due date: 11:59pm, March 12, 2023

1. Task

This assignment is about the implementations of multi-class logistic regression AND random forest algorithms. Students must implement these two methods using scikit-learn (https://scikit-learn.org/stable/) and test each on 2 classification datasets to deepen their understanding of overfitting/underfitting (refer to Pages 30-31 in Lec. 4 Classification Algorithms).

2. Dataset

In this assignment, we will use three datasets: Iris [1], Wine [2], and Breast Cancer Wisconsin [3]. Each dataset can be accessed using existing interfaces of scikit-learn (more details can be found in the provided python template). Specifically, the implemented logistic regression algorithm should be validated on Iris [1] and Wine [2], while the random forest algorithm needs to be validated on Iris [1] and Breast Cancer [3].

[1] Iris

(https://archive.ics.uci.edu/ml/datasets/Iris)
This dataset contains 3 classes (i.e., Setosa, Versicolor and Virginica) of 50 instances each, where each class refers to a type of iris plant. There are 4 input attributes: sepal length (cm), sepal width (cm), petal length (cm) and petal width (cm).

[2] Wine

(https://archive.ics.uci.edu/ml/datasets/wine)
These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from 3 different cultivars. There are 178 instances in total and each instance has 13 attributes (acquired via chemical analysis). Your job is to classify each instance into 3 types of wines based on its 13 attributes.

[3] Breast Cancer

(https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))
There are 569 instances in this dataset. Each instance contains 32 attributes that describe characteristics of the cell nuclei present in the image. The goal is to tell whether each instance is malignant or benign. Thus, this is a binary classification problem.

Guidelines:

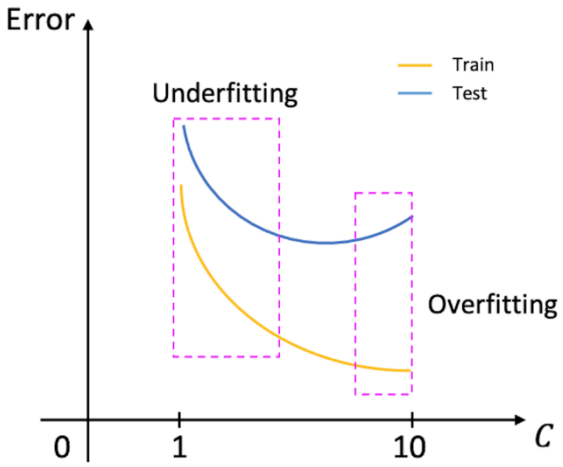
[1] Using Python as the programming language is strongly recommended because we will use scikit-learn (a library for python) in this assignment. Also, the provided `template.ipynb' should

be opened with jupyter notebook. Please refer to https://jupyter.org/install for how to install jupyter notebook.

[2] In logistic regression, you are required to tune the inverse regularization parameter C to generate underfitting and overfitting results. As for random forest, the maximum depth of the decision tree ('max_depth' in scikit-learn) and bootstrap sample size ('max_samples' in scikit-learn) are two parameters that need to be tuned.

[3] You need to control variables when tuning the target parameter. For instance, when you are trying to investigate the impact of 'max_depth' given a random forest model, it is necessary to fix values of `max_samples' and other parameters.

[4] As aforementioned, you need to submit a report that includes figures describing cases of underfitting and overfitting for each of the three parameters. Here is an example that illustrates the impact of the inverse regularization parameter C in logistic regression. Note that data in this figure are for demonstration purposes only (not real).



Besides figures, you also need to explain the key reasons behind the underfitting/overfitting phenomena. Along with the report, you also need to submit a piece of code that can reproduce your reported results.

[5] The submitted source code must be self-contained. A README file regarding how to run your code should be provided so that we can run your code on our local machine.

[6] In the submitted report, you should indicate each group member's contributions (in the form of fractions). Otherwise, we assume that two group members contribute equally. For those who work alone (i.e., your group only involves yourself), there is no need to indicate your contribution.

Submission Format:

A zip format is preferred, which should include three files: report, code, and README. The name of the zip file should be your group number. For example, if your group number is G30, you should submit a zip file named 'G30.zip'.