COMP3314 Machine Learning

Programming Assignment 1:
Random Forest and Linear SVC

Start date: March 12, 2020
Due date: 11:59pm, April 3, 2020

Task:

This assignment is about the implementation of the random forest or support vector machine algorithms. Students are required to follow the lectures, re-implement either the random forest or the support vector machine algorithm, and test it on several toy datasets provided by UCI (url: https://archive.ics.uci.edu/ml/index.php).

Datasets:

Three classification datasets are provided for the students, and two of them should be selected and tested according to the nature of their algorithm. Students who choose to implement the random forest algorithm should test their models on the breast cancer dataset [1] and the car evaluation dataset [3]. Students who choose to implement the support vector machine algorithm should test their models on the breast cancer dataset [1] and the iris dataset [2]. We have split the datasets into training and testing subsets for evaluation purpose. **Students may download the corresponding .csv files in Moodle to train and test their models**.

[1] Breast cancer dataset

(url: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29)
Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image to decide if the cancer is benign (class 1) or malignant (class 0).

There are 30 input attributes: mean radius, mean texture, mean perimeter, mean area, mean smoothness, mean compactness, mean concavity, mean concave points, mean symmetry, mean fractal dimension, radius error, texture error, perimeter error, area error, smoothness error, compactness error, concavity error, concave points error, symmetry error, fractal dimension error, worst radius, worst texture, worst perimeter, worst area, worst smoothness, worst compactness, worst concavity, worst concave points, worst symmetry, worst fractal dimension.

There are 426 training samples and 143 testing samples. The training attributes and testing attributes are stored in '**cancer_X_train.csv**' and '**cancer_X_test.csv**' respectively. And the training and testing labels are stored in '**cancer_y_train.csv**' and '**cancer_y_test.csv**' respectively.

[2] Iris dataset

(url: https://archive.ics.uci.edu/ml/datasets/Iris)

The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. The 3 plants are Versicolour (class 0), Virginica (class 1) and Setosa (class 2) respectively.

There are 4 input attributes: sepal length, sepal width, petal length and petal width.

There are 100 training samples and 50 testing samples. The training attributes and testing attributes are stored in '**iris_X_train.csv**' and '**iris_X_test.csv**' respectively. And the training and testing labels are stored in '**iris_y_train.csv**' and '**iris_y_test.csv**' respectively.

[3] Car evaluation dataset

(url: https://archive.ics.uci.edu/ml/datasets/Car+Evaluation)

This model evaluates cars according to their status. And classify them as unacceptable (class 0), acceptable (class 1), good (class 2) and very good (class 3).

The 5 input attributes are buying buying price, maint price of the maintenance, number of doors, number of persons to carry, size of luggage boot, safety of the car.

There are 1209 training samples and 519 testing samples. The training attributes and testing attributes are stored in '**car_X_train.csv**' and '**car_X_test.csv**' respectively. And the training and testing labels are stored in '**car_y_train.csv**' and '**car_y_test.csv**' respectively.

Guidelines:

[1] Students are required to implement one of the following algorithms: random forest or linear support vector classifier with slack variables. Students must implement its core functions, including but not limited to random bagging or slack variables.
[2] It may be useful to use third-party numerical packages such as CVXOPT or numpy. But it is prohibited to simply call existing machine learning model interfaces of the random forest or support vector machine algorithm.
[3] The language is not specified, but the submitted source codes must be self-contained. A README file regarding how to run your code should be provided so that we can compile and run your code on our machine.
[4] Students are required to try different parameters in their implementation (e.g. number of trees, slack variables, etc.), and examine the final performance under different parameter settings. Discuss their implementations, obtained results, advantages and limitations of the implemented method in their reports.

Submission Instructions:

[1] One report in **pdf** describing the implementation and results of the model.
[2] Source codes and a README file packed in **zip** format that can be unzipped and compiled.