COMP 309 — *Machine Learning Tools and Techniques*

Assignment 4: Performance Metrics and Optimisation

16% of Final Mark — Due: 11:59pm Monday 23 Sep 2019

1 Objectives

The main goal of this assignment is to use a popular machine learning tool, i.e. scikit-learn, to investigate two important factors for the success of machine learning applications, which are performance metrics and metric optimisation through a series of light coding practices. Two supervised learning scenarios will be used, namely classification and regression, and simple coding will also be involved to prepare for more coding work in the final project. The specific objectives of this assignment are:

- To write simple code and debug in Python using algorithms implemented in a toolbox, mainly in scikit-learn. Scikit-learn is a toolbox with solid implementations of a bunch of state-of-the-art machine learning algorithms and makes it easy to plug them into existing applications. Scikit-learn is probably the most popular machine learning tool nowadays.
- Be able to perform classification using different classification methods implemented in scikit-learn, such as kNN, support vector machines, decision tree, random forest, AdaBoost, gradient boosting, linear discriminant analysis, and logistic regression.
- Compare the performance of different classification methods using a number of popular performance metrics, e.g., accuracy, precision, recall, confusion matrix, area under the receiver operating characteristic curve (AUC under ROC), and analyse the results.
- Be able to use methods in scikit-learn to perform regression, such as linear regression, k-neighbors regression, Ridge regression, decision tree regression, random forest regression, gradient Boosting regression, stochastic gradient descent regression, support vector regression (SVR), linear SVR, and multi-layer perceptron regression.
- To write simple code of common optimisation methods, such as batch gradient decent (GD), mini-batch gradient descent, and stochastic GD (SGD).
- Be able to use existing (complex) optimisation methods to optimise given performance metrics.
- Compare and analyse advantages and disadvantages based on the results of different performance metrics for a given regression task.
- Be able to use exploratory data analysis (EDA) tools to understand and find insights of the given dataset.
- To analyse and visualise the EDA results to choose appropriate methods for data preprocessing in order to improve its quality.

These topics are (to be) covered in week 7 and week 8, but will also involve content from previous weeks. Research into online resources for AI and machine learning is encouraged. You are required to complete the following questions. For each part, make sure you finish reading all the questions before you start working on it, and your report for the whole assignment should *not exceed **12** pages* with font size no smaller than 10.

2 Questions

2.1 Part 1: Performance Metrics in Regression [35 marks]

This part focuses on performance metrics in regression. The task is to use different regression methods and different performance metrics to understand their differences and choose the most appropriate performance metric.

The given *Diamonds* data set, diamonds.csv, is to predict the price of round cut diamonds. This is a regression task with 10 features (the first 10 columns of diamonds.csv) as the input variables and the feature price (the last column of diamonds.csv) as the output variable. The task here is to learn a regression model to discover the relationship between the output variable and the 10 features/input variables. As we discussed in the lectures/tutorials, to use scikit-learn for regression, you may need the following seven steps:

- Step 1. Load Data
- Step 2. Initial Data Analysis
- Step 3. Preprocess Data
- Step 4. Exploratory Data Analysis
- Step 5. Build classification (or regression) models using the training data
- Step 6. Evaluate models by using cross validation (Optional)
- Step 7. Assess model on the test data.

Requirements

You are required to use "309" as the random seed to split the data into a training set and a test set, with 70% as the training data and 30% as the test data.

You should use the following 10 regression algorithms implemented in scikit-learn to perform regression. These 10 algorithms are very popular regression methods: (1) linear regression, (2) k-neighbors regression, (3) Ridge regression, (4) decision tree regression, (5) random forest regression, (6) gradient Boosting regression, (7) SGD regression, (8) support vector regression (SVR), (9) linear SVR, and (10) multi-layer perceptron regression. You are encouraged to read the documentation (and provided references if you would like to know more details) about these methods from scikit-learn, e.g. linear regression is implemented in *sklearn.linear_model.LinearRegression*.

Note that you may need to tune the parameters for some of these 10 regression methods to make them work properly or to achieve better performance.

You are required to submit the code of your program, a "readme.txt" file that describes clearly how to run the code, and a report answering the following questions using your own words:

- If you tune any parameter(s), report which algorithm(s), which parameter(s) and the parameter value(s).
- Based on exploratory data analysis, discuss what preprocessing that you need to do before regression, and provide evidence and justifications.
- Please report the results (keep 2 decimals) of all the 10 regression algorithms on the *test* data in terms of mean squared error (MSE), root mean squared error (RMSE), R-Squared, mean absolute error (MAE), and execution time. You should report them in a table.
- Compare the performance of different regression algorithms in terms of MSE, RMSE, R-Squared, and MAE, then analyse their differences and provide conclusions.

2.2 Part 2: Performance Metrics in Classification [35 marks]

The given *Adult* dataset is a popular classification data set from the UCI machine learning repository, and the task is to determine whether a person earns a salary of over $50K a year. Separate training and test sets are provided, as adult.train and adult.test, respectively.

You are recommended to follow the steps that we discussed in the lectures/tutorials and listed in Part 1.

Requirements

You are required to use 10 classification algorithms implemented in scikit-learn to perform classification. These 10 algorithms are very popular classification methods from different paradigms of machine learning: (1) kNN, (2) naive Bayes, (3) SVM, (4) decision tree, (5) random forest, (6) AdaBoost, (7) gradient Boosting, (8) linear discriminant analysis, (9) multi-layer perceptron, and (10) logistic regression. You are encouraged to read the documentation (and provided references if you would like to know more details) about these methods from scikit-learn, e.g. kNN is implemented in *sklearn.neighbors.KNeighborsClassifier*. We assume that class $> 50K$ is the positive class.

You are required to submit the code of your program, a "readme.txt" file that describes clearly how to run the code, and a report answering the following questions using your own words:

- Based on exploratory data analysis, discuss what preprocessing that you need to do before classification, and provide evidence and justifications.
- Please report the results (keep 2 decimals) of all the 10 classification algorithms on the given *test* data in terms of classification accuracy, precision, recall, F1-score, and AUC. You should report them in a table.
- Is accuracy the best performance metric to evaluate a classifier? and why?
- Find the two best algorithms according to each of the four performance metrics, Are they the same? Explain why.

2.3 Part 3: Optimisation Methods [30 marks]

This part focuses mainly on using different optimisation methods to optimise performance metrics. A code/project template is provided, which implements the batch gradient descent method to optimise MSE during the linear regression learning process, which we name the BGD+MSE method. The template also includes the code for drawing graphs. You are required to modify the code/project template to complete the given questions.

This part of the assignment is based on a regression problem, where the input variable is $height(inches)$, and the output variable is $weight(lbs)$. You are given two sets of data, i.e. Part2.csv with contains 500 examples without any outlier, and Part2Outliers.csv contains 502 examples with two outliers.

Requirements

You are required to modify the code/project template to:

1. Implement the mini-batch gradient descent optimiser (mini_batch_size = 10) based on the given template to optimise MSE for linear regression learning. We name this approach MiniBatchBGD+MSE.
2. Use the particle swarm optimisation (PSO) algorithm, which has been implemented in the provided code, to optimise MSE for linear regression. We name this method PSO+MSE.
3. Use PSO to optimise MAE for linear regression. We name this method PSO+MAE.

By writing the above coding, you now have four methods, i.e. BGD+MSE, MiniBatchBGD+MSE, PSO+MSE, and PSO+MAE. Perform experiments on the provided datasets to answer the following questions and write a report. You are required to submit the code, a "readme.txt" file and the report:

1. On the dataset without outliers, i.e. Part2.csv, use "309" as the random seed to split the dataset into a training set and a test set, with 70% of the data as the training set and 30% as the test set. Run each of the BGD+MSE, MiniBatchBGD+MSE, PSO+MSE, and PSO+MAE methods on the training data to learn a linear regression model and test the learnt model on the test data.

(a) Plot the paths of gradient descent of BGD+MSE and MiniBatchBGD+MSE, then discuss their differences and justify why.

(b) Report the results (keep 2 decimals) of the four learnt models over the MSE, R-Squared, and MAE performance metrics on the test set. Compare their results and discuss the differences. You can report them in a table.

(c) Generate a scatter plot with the regression line learnt by PSO+MSE and PSO+MAE and the data points in the test set.

(d) Compare the computational time of the BGD+MSE, MiniBatchBGD+MSE and PSO+MSE methods, find out the fastest one and slowest one, and explain why.

2. On the dataset with outliers, i.e. Part2Outliers.csv, split the data and run the PSO+MSE, and PSO+MAE methods in the same way as on the Part2.csv dataset in Question 1. Then:

(a) Generate the scatter plot with the regression line learnt by PSO+MSE and PSO+MAE and the data points in the test set.

(b) Compare the above two plots with the two plots you draw in Question 1(c), and discuss which of the two methods (PSO+MSE or PSO+MAE) is less sensitive to outliers and explain why.

(c) Discuss whether we can use gradient descent or mini-batch gradient descent to optimise MAE? and explain why.

3 Relevant Data Files and Program Files

A soft copy of this assignment, the relevant data and program files are available from the Assignments page of the course website.

4 Assessment

We will endeavour to mark your work and return it to you as soon as possible, hopefully in 2 weeks. The tutor(s) will run a number of helpdesks to provide assistance to answer any questions regarding what is required.

5 Submission Guidelines

5.1 Submission Requirements

1. Programs for all individual parts. To avoid confusion, all the individual parts should use directories `part1/`, `Part2/`, `...` and all programs should be stored in their corresponding directories. Within each directory, please provide a `readme file` that specifies how to compile and run your programs on the ECS School machines. A script file called `sampleoutput.txt` should also be provided to show how your program run properly. If you programs cannot run properly, you should provide a `buglist` file.

2. A document that consisting of the reports of all the individual parts. The document should mark each part clearly. The document can be written in PDF.

5.2 Submission Method

You are required to submit both the program *code* and the PDF version of the *report*, which should be submitted through the web submission system from the COMP309 course web site **by the due time**.

There is NO required hard copy of the documents.

KEEP a backup and receipt of submission.

Submission should be completed on School machines, i.e. problems with personal PCs, internet connections and lost files, which although eliciting sympathies, will not result in extensions for missed deadlines.

5.3 Late Penalties

The assignment must be handed in on time unless you have made a prior arrangement with the lecturer or have a valid medical excuse (for minor illnesses it is sufficient to discuss this with the lecturer). The penalty for assignments that are handed in late without prior arrangement is one grade reduction per day. Assignments that are more than one week late will not be marked.