Code Guidance

Vidhi Lalchand

Thesis title: A meta-algorithm for classifications using random recursive tree ensembles: A high energy physics application

This document summarizes the workings of the python code base submitted along with the thesis mentioned in the title. This is not a README file (a separate README file has been provided with the code). This document is written with the sole purpose of enhancing the usability of the code in addition to reproducibility of the main results published in the thesis. This code base is designed to run on the dataset publicly available to download from the CERN Open data portal at `http://opendata.cern.ch/record/328?ln=en`. The size of the unzipped data file is 195 MB.
The code base allows a user to run several tree ensemble classifiers and examine the results. The code base comes with a `settings.ini` file in the parent folder which is the only file a user needs to amend in order to run the code in different configurations. The `settings.ini` file is read by a config parser when `__main__.py` is called. The `settings.ini` file is divided into different sections and has extensive comments to guide the user with parameter setting. Here, I just mention some important points.

1. **paths**: The user must amend this section to give the *path/to/folder/where/data/is/stored/*, the datafile must be named *'higgs.csv'*.
2. **algorithms**: The user alters the algorithm acronym here to run different algorithms.
3. **pipeline**: Extensive comments in `settings.ini`
4. **algorithmName**: The user does not need to (and should not) change this section.
5. **userParams**: Extensive comments in `settings.ini`

Individual model parameters are specified under sections - **DT**, **BDT**, **RF**, **ET**, **BRF**, **BXT**.
Since they all use trees as primitive learners, they have a lot of common parameters.

Algorithm	n_estimators	criterion	max_features	max_depth	min_samples_split	min_sample_leaf	learning_rate
DT		•	•	•	•	•	
BDT	•						•
RF	•	•	•	•	•	•	
ET	•	•	•	•	•	•	
BRF	•						•
BXT	•						•

Table 1: Necessary parameters for the tree algorithms

These are not arbitrary letters, rather they are acronyms for the full names of algorithms which are too long to be peppered around in the code or in the text. For instance, BRF stands for Boosted Random Forests. The thesis provides background on what the parameters for each model mean in the context of trees and `settings.ini` provides some guidance.

1 Outputs

1.1 Log file

In each run a log file with the file name format `higgs_classification_pipeline_id_147*******` is generated in the `/Logs/` sub-directory. The final suffix starting with 147 of the file name is a unique

identifier, it is constructed by using time since epoch in seconds. This ensures that log files created in repetitive runs have unique (and in this case increasing) file name identifiers and do not overwrite existing logs. Apart from writing to the log file, the `sys.out` is also printed on the screen.

1.2 Performance Report

A performance report with the file name format `Classifier_Performance_Report_algorithm_xx.txt` is generated in each run and stored in the `Results/` sub-directory. The final suffix 'xx' of the file name is a random two-digit integer. This is to ensure that a moderate number of repetitive runs of the same algorithm will create new files and not overwrite existing ones. If you run the same algorithm 100 times, you will most certainly overwrite one of the existing reports.

1.3 Significance Curve (σ)

In the settings file there is a boolean variable (under pipeline) that governs if a significance curve should be generated and saved (in `/Graphs/`). This curve depicts the AMS (σ) at 1500 thresholds between 80 and 95 using a step size of 0.01. This curve is the definitive performance benchmark against which all algorithms are assessed and compared. If the setting to generate this curve is set to `False`, no curve will be generated and only a single AMS σ at the 85th percentile threshold is computed and written out to logs and in the performance report.

2 Testing

Using this code base a user can,

- Train and test a machine learning algorithm (specified by the user in the settings.ini) on the ATLAS higgs dataset, after selecting the chosen algorithm in the `settings.ini` file the user can can run `>> python __main__.py`. Before testing, the user must ensure the following:
  - To point to the correct version of Python (2.7.3).
  - Download the data (as mentioned above) and place it in the `/Data/` sub-directory, do not forget to rename it to *higgs.csv*.
  - Ensure all python packages specified in the README are installed.

  If there are no unforseen errors, running of the __main__.py should create a log file (in `/Logs/`), print out and store a performance report (in `/Results/`) and save the plot of the significance curve (in `/Graphs/`)
- The user can use the stand alone scripts in the `plotting_scripts` sub-directory to generate the plots in the thesis. The scripts have been enriched with docstrings to provide direction to the user. The most useful of these scripts are `compare_boosted_ensembles` and `compare_forest_models`, they rely on pre-trained pickled models.

Thank you