EE6550 Machine Learning HW3 Adaboost README

Adaboost classifer with the shallow decision tree (depth 1) for binary classfication

- Author: Yu-Chun (Howard) Lo
- Email: howard.lo@nlplab.cc

User Manual

Dev Environment

- Developed under Anaconda 4.3.0 (x86_64).
- Require Numpy for matrix operations.
- Tested on Python 3.6.0.

File Structure

- `dataset/`: **(Important)** The program reads datasets from this folder and performs training and testing on the specified training and test data files. (See the *Dataset Format* section below)
- `logs/`: **(Important)** Output reports are stored in this folder.
- `/hypothesis`: **(Important)** Output hypotheses are stored in this folder. (There is **timestamp** at the end of the file name, which corresponds to the report)
- `tree.py`: Shallow decision tree used by `adaboost.py` as weak classfier.
- `adaboost.py`: Adaboost classifer model.
- `utils.py`: Some utilities used by this program, such as loading dataset, normalize labels, etc.
- `main.py`: **(Important)** The main program. User should train a Adaboost classfier by running this program.

Dataset Format

- Currently, the program only supports reading `.csv` file.
- The class label of each item should locate at the first column. The class labels should only be **binary**, e.g. `{+1, -1}`, `{1, 0}` or `{'+', '-'}`, etc.
- **(Important)** If you want to train your Adaboost classifier with your own dataset, please be sure that you've followed the required format described above, and have placed your own training and test data files in the `dataset/` folder.

Getting Started

Train your Adaboost classifer by running `python main.py` in terminal. Be sure that your terminal is under the same directory as `main.py`.

Note that we've set default values for required input arguments. Run `python main.py --help` to view input arguments information shown below.

```
usage: main.py [-h] [--train_filename TRAIN_FILENAME]
               [--test_filename TEST_FILENAME] [--K K] [--T T]

Binary Adaboost classifer.

optional arguments:
  -h, --help            show this help message and exit
  --train_filename TRAIN_FILENAME
                        Training dataset csv. (Default:
                        "alphabet_DU_training.csv")
  --test_filename TEST_FILENAME
                        Training dataset csv. (Default:
                        "alphabet_DU_testing.csv")
  --K K                 Denotes for "K"-fold cross-validation for determine
                        the optimal hyper-parameters for adaboost classifer.
                        (Default: None)
  --T T                 The maximum number of classifers at which boosting is
                        terminated.
```

For Grading Session

Here we show some guides for different test scenarios:

- Place the training data file(e.g. `xxx_training.csv`), testing data file(e.g. `xxx_testing.csv`) in the `/dataset` folder before running `main.py` with specified `--train_filename` and `--test_filename`.
- For performing K-fold cross-validation, for example, specify `--K=5`.
- All the required output information, such as class label mapping, cross-validation history, optimal hyper-parameters, etc., are stored at the `logs/` folder. Note that the log file name indicates what number of K-fold you choosed, and when you run the program. This naming convension aims to help graders to choose which report to check after running the program.
- For running a specific `T`, for example, specify `--T=5`.
- Note that the csv file name of hypothesis in the `hypothesis/` folder is concatenated with the timestamp. This aims to let the graders know which hypothesis file to choose to test after running the program. (You may remove the timestamp for grading)
- Summing up, you may want to run the following commands in the different test scienarios:

```
 (For 5-fold cross-validation)
>> python main.py --train_filename="xxx_training.csv"
--test_filename="xxx_testing.csv" --K=5

 (For specifying hyper-parameters)
>> python main.py --train_filename="xxx_training.csv"
--test_filename="xxx_testing.csv" --T=5
```

Report

All required output information are stored at `logs/`. We've run on 50 different hyper-parameters to select a optimal hyper-parameter. Note that we've also print the **readable** format of Adaboost classifer in our report file.

- For 5-fold cross-validation results, check `logs/adaboost-5-fold-[HH:MM:SS]`
- For 10-fold cross-validation results, check `logs/adaboost-10-fold-[HH:MM:SS]`

Note that the corresponding hypothesis csv file is indicated by the `HH:MM:SS` timestamp, which is the time that the program finished training.