EE6550 Machine Learning HW2 README

Linear Support Vector Machine for Binary Classification trained with Sequential Minimal Optimization

- 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 2.7.12 and 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)
- `svm.py`: Support vector machine (SVM) 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 SVM by running this program.
- `<K>-fold-result-[HH:MM:SS].txt`: **(Important)** Logs, containing cross validation errors with corresponding hyper-parameter (`C` in our case), optimal hyper-parameter, optimal hypothesis (`weight vector` □ `w` and `bias term` □ `b` is our case) and performance. This will be generated when running the program without specified hyper-parameter. Note that the file name `<K>` is the number of "K"-fold cross-validation and `HH:MM:SS` is the time you run the 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.
- If you want to train your SVM 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 SVM 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] [--C C]

Linear support vector classifer.

optional arguments:
  -h, --help            show this help message and exit
  --train_filename TRAIN_FILENAME
                        Training dataset csv. (default:
                        "messidor_features_training.csv")
  --test_filename TEST_FILENAME
                        Training dataset csv. (default:
                        "messidor_features_testing.csv")
  --K K                 Denotes for "K"-fold cross-validation for determine
                        the optimal value C for SVM. (default: 5)
  --C C                 If C is specified, disable cross-validation, train SVM
                        on this specified C (default: None)
```

Here we show some running examples for different test scenarios:

- For inputting a training data file, which contains a labeled training sample. This labeled training sample is used to train the SVM-learning algorithm which will return a hypothesis after K-fold cross-validation, **place the training data file in the `/dataset` folder**.
- For inputting a testing data file, which contains a labeled testing sample. This labeled testing sample is used to evaluate the performance of the returned hypothesis from the SVM-learning algorithm based on the labeled training sample, **place the testing data file in the `/dataset` folder**.
- For inputting a positive integer **K** to perform K-fold cross-validation to determine the optimal value C of the free parameter C to minimize the cross-validation error, **specify the argument `--K` when running `main.py`** (default value is **5**).
- Summing up the above, you can also run, for example, `python main.py --train_filename="xxx.csv" --test_filename="xxx.csv" --K=5`. Note that training data and test data files must be specified.
- For checking the obtained optimal value C, optimal hypothesis and performance, see the generated log file `<K>-fold-result-[HH:MM:SS].txt`.

Next, we describe reported results in the *Report* section.

Report

Note that training on a single hyper-parameter `C` takes about 1~3 minutes. The larger `C` is, the more time it is needed to train a SVM.

Experiment Set

Tested on 20 different hyper-parameters `C`

```
[0.1, 0.15, 0.19, 0.24, 0.29, 0.34, 0.38, 0.43, 0.48, 0.53, 0.57, 0.62, 0.67, 0.72,
0.76, 0.81, 0.86, 0.91, 0.95, 1.0]
```

5-Fold Cross-validation

- ~2 hours to perform the experiment.
- The cross-validation errors over 20 different `C` is shown as follows:

C	Cross-validation Error
0.1	0.270808929632
0.15	0.269552669553
0.19	0.279976232917
0.24	0.26301672184
0.29	0.274789915966
0.34	0.268253968254
0.38	0.260402342755
0.43	0.262999745353
0.48	0.277353365589
0.53	0.278652066887
0.57	0.264306934895
0.62	**0.257813428402**
0.67	0.268236991766
0.72	0.277327900857
0.76	0.263050674815
0.81	0.269544181309
0.86	0.266929802224
0.91	0.268236991766
0.95	0.269552669553
1.0	0.272158560394

- Optimal C: `0.62`
- Weight vector: `[0.41376851, -1.1783888, 0.76335893, -0.2726952, -0.37456999, -0.15391376, 0.02827039, 0.00133731, 0.0055075, -0.00362205, -0.02664064, -0.03642613, -0.02312384, 0.52445173, 0.92577699, 0.57587795, 0.20615108, -0.24830772, -0.0214078]`
- Bias term: `-0.553474057815`
- Performance:
  - training error: `0.247395833333`
  - test error: `0.240208877285`

10-Fold Cross-validation

- ~5 hours to perform the experiment.
- The cross-validation errors over 20 different `C` is shown as follows:

C	Cross-validation Error
0.1	**0.25517771702**
0.15	0.270830485304
0.19	0.265447710185

C	Cross-validation Error
0.24	0.256459330144
0.29	0.266866028708
0.34	0.262918660287
0.38	0.268147641832
0.43	0.270745044429
0.48	0.279887218045
0.53	0.257740943267
0.57	0.260269993165
0.62	0.261637047163
0.67	0.27462406015
0.72	0.259039644566
0.76	0.264200273411
0.81	0.269480519481
0.86	0.256425153794
0.91	0.261637047163
0.95	0.266814764183
1.0	0.265516062884

- Optimal C: `0.1`
- Weight vector: `[0.2, -0.70799666, 0.73837806, -0.24449533, -0.38217573, -0.13843935, -0.00088726, 0.02371915, 0.00757344, -0.01023123, -0.01775169, -0.00509908, 0.00981942, 0.17055394, 0.21350525, 0.11907159, 0.03387551, -0.06414395, -0.08501416]`
- Bias term: `-0.721050592969`
- Performance:
  - training error: `0.255208333333`
  - test error: `0.22454308094`