CS 573: Assignment 3

Tiantu Xu

March 8, 2019

1. **Preprocessing**

   In Question 1, data pre-processing is run on the command line below:

   ```
   $ python preprocess-assg3.py dating-full.csv dating.csv
   ```

   The output from my code is

   ```
   Mapped vector for female in column gender:  [1]
   Mapped vector for Black/African American in column race:  [0, 1, 0, 0]
   Mapped vector for Other in column race_o:  [0, 0, 0, 0]
   Mapped vector for economics in column field:  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0]
   ```

2. **Implement Logistic Regression and Linear SVM**

   To run logistic regression, specify `sys.argv[3] = 1`:

   ```
   $ python lr_svm.py trainingSet.csv testSet.csv 1
   ```

   The output from my code is

   ```
   Training Accuracy LR: 0.65
   Test Accuracy LR: 0.65
   ```

   To run SVM, specify `sys.argv[3] = 2`:

   ```
   $ python lr_svm.py trainingSet.csv testSet.csv 2
   ```

   The output from my code is

   ```
   Training Accuracy SVM: 0.56
   Test Accuracy SVM: 0.55
   ```

3. **Learning Curves and Performance Comparison**

   (a) K-fold cross-validation is run on the command line below.

   ```
   $ python cv.py
   ```

   The model perforamnce on 3 models (NBC, LR, and SVM) is shown below.

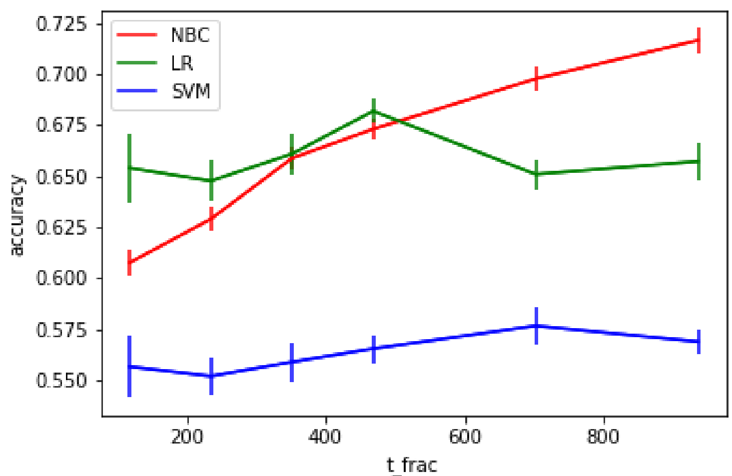

Figure 1: The model performance of NBC, LR, and SVM

   (b) The hypothesis testing is formulated as

   H_0: `LR and SVM model performances do not differ significantly.`
   H_1: `LR and SVM model performances differ significantly.`

   Assume I have a significance level of $\alpha = 0.05$. ttest is run on the performance numbers obtained in the above cross-validation. The output from the ttest is shown below

   ```
   Ttest_indResult(statistic=15.415948619949303, pvalue=2.6874121443687926e-08)
   ```

   It turns out that p-value $< \alpha$, so that we reject H_0 and accept H_1 that LR and SVM performances differ significantly.

4. **Bonus Question** To fine tune the hyperparameters, I made the following changes compared to Question 2:

   - Further divided the whole dataset into training set (60%), validation set (20%), and test set (20%). I trained on each model on the training set, validate the regularization and learning rate on the validation set, and finally test the best pair of parameters on the test set. **None of the three sets overlapped.**
   - I initialized the weights with have a normal distribution $N \sim (0, 1)$, instead of starting from weights with all zeros.
   - I normalized each attribute value to be in [0,1] for Linear Regression.
   - The set of regularization and learning rate is shown below:

   ```
   regularization = [0.01, 0.005, 0.002, 0.001, 0.0005, 0.0002, 0.0001]
   step_size = [0.01, 0.005, 0.002, 0.001, 0.0005, 0.0002, 0.0001]
   ```

To run my code,

```
$ python bonus_lr_svm_fine_tune.py
```

The output from my code is

```
Have max test accuracy: 0.74 when regularization = 0.0005 and step_size = 0.0001
Have max test accuracy: 0.64 when regularization = 0.002 and step_size = 0.0001
```

The result might be different when the grader run my code, because I started from a $N \sim (0, 1)$, so each time the initial weights are different. It may get even higher test accuracy with stochastic gradient descent, but I kept using the algorithm on the instructor's slides related to LR and SVM.