Assignment 2
Cogsci 188 Summer
Sentiment Classification
September 7th at 06:00 PM

Motivation:

Using K-nearest neighbor, Perceptron, Bayesian Classification algorithms for sentiment classification in the text data obtained from the Internet site, imdb.com and investigate the effect of specific parameters for each classifier in the classification accuracy.

Task:

In total you will write three functions and a PDF report

```
The four functions will be named as follows:
readTrainTestData
sentimentClassKNN
sentimentClassPerceptron
sentimentClassMyOwn
```

1. The first function is `readTrainTestData` that takes as input `train and test file names`, and returns four variables: `train_vec, train_lab, test_vec, test_lab`. You will use these four variables as inputs to subsequent functions.

   ```
   def readTrainTestData(trainFile, testFile)
   ```

2. The second functions will take in six arguments, shown below and will return one value as the accuracy ranging from 0.0 to 1.0.

   ```
   def sentimentClassKNN (train_vec, train_lab, test_vec, test_lab,
   k, distanceMeasure)
   ```

   `sentimentClassKNN` will return the accuracy using K-nn classifier using the distanceMeasure specified by the string in the variable “distanceMeasure” input argument and using the *k* as the k input argument.

3. The third functions will take in five input arguments, shown below and will return one value as the accuracy ranging from 0.0 to 1.0.

   ```
   def sentimentClassPerceptron (train_vec, train_lab, test_vec,
   test_lab, n)
   ```

   `SentimentClassPerceptron` will return the accuracy using Perceptron algorithm and will loop through the data points *n* times.

4. The fourth functions will take only four input arguments and will return a list of 3 numbers that show the accuracy of 3 different combinations of classifiers that you yourself come up with. These classifiers may be different parameters of the knn and perceptron or can involve processing of the input data by running it through PCA or another algorithm from the toolbox like k-means (somehow used for classification) etc. Please make sure your models are all different (don't use one model with different parameters). Also, please make sure your accuracy is higher than 80%. The test cases given in the notebook both give accuracy higher than 80%. Try to do better than the test cases.

```
def sentimentClassMyOwn (train_vec, train_lab, test_vec, test_lab)
```

The main goal of the last function is to be able to write about your own experiments and their results. Please use **Python3**. You are allowed to use the library **sklearn 0.19** (http://scikit-learn.org/0.19/)

Data:

You will be given two data files, one training set and one test set.

Each line in the dataset starts with a label of the vector, either a 1 or a 0, followed by the vector.

Each vector represents a review and the label represents whether the review is positive or negative. The vector is obtained based on the word2vec model. You need to search for some information about the word2vec model and include a simple description in your report.

Your job is to train your model by the training set and use the model to predict the vectors in the test sets to see whether it is a positive or negative review.

You can evaluate your model by the prediction accuracy, which is defined as the number of correct predicted label divided by the size of the test set.

Report:

Name your report "Assignment2_Report.pdf"
Include both your name and your partner's name at the top.
Your report should be 3 to 5 pages and will have the following parts. Report should not exceed 5 pages.

(1) Introduction of the models that you defined and use.
(2) Input Data: report the dimension of the data and some basic info on word2vec.
(3) Results from K-nn with different K and distance measure, include the time complexity of your model.
(4) Results from Perceptron with different n. Also report the time complexity.
(5) Results of my own experiments
    a. Experiment 1
    b. Experiment 2
    c. Experiment 3

Includes the following:

- The reason of why you choose these three models.

- The parameters you use for each model, and the reason you choose these parameters.
- The time each model takes to classify.
- The pros and cons of these three models for this dataset.

(6) Conclusion

We will run your code with the same data as we submit to you but the order of feature vectors can be different in the file (which should not matter too much). We will compare the results we obtain with our own results and results stated in your report. The results will not be exactly the same but should be comparable.

Grading:

The grading will be as follows:

10 points. Accurate implementation of `sentimentClassKNN`,
10 points. Accurate implementation of `sentimentClassPerceptron`
30 points. Accurate implementation of `sentimentClassMyOwn`
40 points. Report complete with all the parts including the write up of the three experiments.
10 points. The quality of the experimental results and conclusion. Even if your results are not good, you should be able to analyze and conclude something useful from the results. This last part is more to make sure that you understood what you were doing and had fun doing it! ☺
Total 100 points.

Assignment submission details

The assignments can be done in groups of up to two students. Following is the turning process.

1. login to your student account at datahub.ucsd.edu;
2. spawn your server under **cogs188;**
3. click the Assignments tab;
4. Fetch the Assignment 2;
5. You can now access the notebooks under your File tab inside the assignment folder;
6. Put your name and your partner name at the top by editing the second cell;
7. Fill in the function and run the simple tests provided;
8. You can't edit the test cells, they are read-only;
9. Once you are ready, you can do a "Run All" to make sure you pass all the tests;
10. Upload your report to the Assignment 2 folder and Gradescope;
11. Go back to the Assignments tab and click submit;