LING572 Hw4 (kNN)
Due: 11pm on Feb 1, 2022

The example files are under dropbox/21-22/572/hw4/examples/ and hw4/example_output/.

Q1 (40 points): Write a script, **build_kNN.sh**, that implements the kNN algorithm. It classifies a test instance x by letting the k nearest neighbors of x vote.

- The learner should treat features as real-valued.
- Use majority vote; that is, each of the k nearest neighbors has one vote.
- The format is: build_kNN.sh training_data test_data k_val similarity_func sys_output > acc_file
- training_data and test_data are the vector files in the text format (cf. **train.vectors.txt**).
- k_val is the value of k; i.e., the number of nearest neighbors chosen for classification.
- similarity_func is the id of the similarity function. If the variable is 1, use Euclidean distance. If the value is 2, use cosine similarity. **Notice that Euclidean distance is a dissimilarity measure; that is, the longer the distance between two instances is, the more dissimilar (i.e., the less similar) the instances are.**
- While some packages include functions for calculating Euclidean distance and cosine similarity, you should implement your own functions.
- sys_output and acc_file have the same format as the one specified in Hw3, and they should include the classification results for both training and test data. When choosing k nearest neighbors for a training instance x, one of such neighbors can be x itself. Since the other k-1 neighbors could have labels different from that of x, the training accuracy could be lower than 100%.
- For each line of sys_output, remember to sort the (c_i, p_i) pairs by the value of p_i in **descending order**. If two class labels have the same probability, either order of two (c_i, p_i) pairs is ok.

Run build_kNN.sh with **train.vectors.txt** as the training data and **test.vectors.txt** as the test data. Fill out Table 1 with different values of k and similarity function, and submit sys_output and acc_file with **k_val=5** and **similarity_function=2**.

Table 1: Test accuracy using **real-valued** features

k	Euclidean distance	Cosine function
1		
5		
10		

Q2 (35 free points): Free points so you have time to read papers on MaxEnt etc. No need to turn in anything for this.

Submission: Submit the following to Canvas:

- Your note file *readme.(txt | pdf)* that includes Table 1 and any notes that you want the TA to read.
- hw.tar.gz that includes all the files specified in dropbox/21-22/572/hw4/submit-file-list, plus any source code (and binary code) used by the shell scripts.
- Make sure that you run **check_hw4.sh** before submitting your hw.tar.gz.