LING572 Hw4 (Feature selection and chi square)
Due: 11:45pm on Feb 4, 2010

The example files are under dropbox/09-10/572/hw4/examples/.

Q1 (30 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 nearest neighbor 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 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 function. **Notice that Euclidean distance is a dissimilarity measure; that is, the longer the distance is, the dissimilar the two vectors are.**
- sys_output and acc_file have the same format as the one specified in Hw3.

Q2 (10 points): 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.

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

k	Euclidean distance	Cosine function
1		
5		
10		

Q3 (5 points): Repeat Q2 but treating features as binary. Here, you can simply run build_kNN.sh with **train2.vectors.txt** as the training data and **test2.vectors.txt** as the test data. Fill out Table 2.

Table 2: Test accuracy using **binary** features

k	Euclidean distance	Cosine function
1		
5		
10		

Q4 (15 points): Write a script, rank_feat_by_chi_square, that ranks features by χ^2 scores.

Table 3: A contingency table for feature f_k

	c_1	c_2	c_3
$\bar{f}_k$	$a_1 - b_1$	$a_2 - b_2$	$a_3 - b_3$
f_k	b_1	b_2	b_3

- The format for command line is: cat input_file | rank_feat_by_chi_square > output_file
- input_file is a feature vector file in the text format
- The output_file has the format "featName score docFreq". The score is the chi-square score for the feature; docFreq is the number of documents that the feature occurs in. The lines are sorted by χ^2_score in descending order.
- For χ^2 calculation, treat each feature as binary; that is, suppose the input_file has a_i instances with class label c_i. Out of the a_i instances, b_i of them contain the feature f_k, then the corresonding contingency table for feature f_k is shown in Table 3.
- Run "cat train.vectors.txt | rank_feat_by_chi_square > feat_list", and use feat_list for Q5.

Q5 (20 points): Test the effect of feature selection:

- The original training and test data are **train.vectors.txt** and **test.vectors.txt**.
- Use the feat_list created in Q4 to filter out *unrelated* features in the original data.
- A feature is considered **related** if its χ^2-score is high enough that the Null hypothesis in the χ^2 test is rejected.
- Write a script to filter out unrelated features from the vector files. For the script, you can choose whatever format you like.
- After the filtering step, run the kNN learner with **k=1** and **cosine function** using the newly processed data.
- Fill out the test accuracy on Table 4. The p_0 in the first column is the significance level in the χ^2 test. The second column shows the number of features that are considered *related* based on the χ^2 test. The third column shows the test accuracy when only the related features are used by the kNN learner. The baseline is the result when using the original vector files (i.e., without the filtering step).

Q6 (10 points): The same as Q5 except that the features should be treated as binary (e.g., using **train2.vectors.txt** and **test2.vectors.txt**), and k should be set to 10. Fill out Table 5.[1]

Q7 (10 points): What conclusion can you draw from the experimental results shown in the tables?

[1]The column for the number of features is removed because it would be the same as in Table 4.

Table 4: Test accuracy using **real-valued** features, **k=1**, cosine function

p_0	Number of related features	Test accuracy
baseline		
0.001		
0.01		
0.025		
0.05		
0.1		

Table 5: Test accuracy using **binary** features, **k=10**, cosine function

p_0	Test accuracy
baseline	
0.001	
0.01	
0.025	
0.05	
0.1	

Submission: Submit a tar file via CollectIt. The tar file should include the following.

- If your team has two people, please submit only one copy. In your note file, please list the names of team members.
- In your note file hw4.*, include your answers to Q1-Q7, and any notes that you want the TA to read.
- Shell scripts for Q1 and Q4, and related source and binary code.
- The feat_list created in Q4.