LING572 Hw3 (Naive Bayes)
Due: 11pm on Jan 30, 2019

The example files are under dropbox/18-19/572/hw3/examples/.

Q1 (5 points): Run the Mallet NB learner (i.e., the trainer's name is NaiveBayes) with **train.vectors.txt** as the training data and **test.vectors.txt** as the test data. In your note file, write down the training accuracy and the test accuracy.

Q2 (35 points): Write a script, **build_NB1.sh**, that implements the Multi-variate Bernoulli NB model. It builds a NB model from the training data, classifies the training and test data, and calculates the accuracy.

- The learner should treat all features as binary; that is, the feature is considered present iff its value is nonzero.
- The format is: build_NB1.sh training_data test_data class_prior_delta cond_prob_delta model_file sys_output > acc_file
- training_data and test_data are the vector files in the text format (cf. **train.vectors.txt**).
- class_prior_delta is the δ used in add-δ smoothing when calculating the class prior $P(c)$; cond_prob_delta is the δ used in add-δ smoothing when calculating the conditional probability $P(f \mid c)$.
- model_file stores the values of P(c) and $P(f \mid c)$ (cf. **model1**).
  Comment lines start with "%". The line for P(c) has the format "classname P(c) logprob", where logprob is 10-based log of P(c).
  The line for $P(f \mid c)$ has the format "featname classname P(f|c) logprob", where logprob is 10-based log of $P(f \mid c)$.
- sys_output is the classification result on the training and test data (cf. **sys1**). Each line has the following format:
  instanceName true_class_label c1 p1 c2 p2 ..., where $p_i = P(c_i \mid x) = \frac{P(c_i,x)}{P(x)}$. The (c_i, p_i) pairs should be sorted according to the value of p_i in descending order.
- acc_file shows the confusion matrix and the accuracy for the training and the test data (cf. **acc1**).
- As always, **model1**, **sys1**, and **acc1** are NOT gold standard. These files were created with a much smaller training dataset.

Run build_NB1.sh with **train.vectors.txt** as the training data, **test.vectors.txt** as the test data, and class_prior_delta set to 0:

- Fill out Table 1 with different values of cond_prob_delta.
- Store the model_file, sys_output and acc_file for the second row (when cond_prob_delta is 0.5) under q2/.

Table 1: Results of your **Bernoulli** NB model

cond_prob_delta	Training accuracy	Test accuracy
0.1		
0.5		
1.0		

Q3 (35 points): Write a script, **build_NB2.sh**, that implements the multinomial NB model. Other than the modeling (e.g., the features in the multinomial NB model are real-valued), everything else (e.g., the input/output files) is the same as in Q2.

- Fill out Table 2.
- Store the model_file, sys_output and acc_file for the second row (when cond_prob_delta is 0.5) under q3/.

Table 2: Results of your **multinomial** NB model

cond_prob_delta	Training accuracy	Test accuracy
0.1		
0.5		
1.0		

Submission: Submit the following to Canvas:

- Your note file *readme.(txt | pdf)* that includes Table 1 and 2, and any notes that you want the TA to read.
- hw.tar.gz that includes all the files specified in dropbox/18-19/572/hw3/submit-file-list, plus any source code (and binary code) used by the shell scripts.
- Make sure that you run **check_hw3.sh** before submitting your hw.tar.gz.