LING572 Hw3 (Naive Bayes)
Due: 11:45pm on Jan 28, 2010

The example files are under dropbox/09-10/572/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 (30 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 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**).
- prior_delta is the δ used in add-δ smoothing when calculating the 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**).
  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**). Except for the comment lines that start with %, all the other lines have 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**). It has the same format as **acc_file** in Hw2.
- As always, **model1**, **sys1**, and **acc1** are NOT gold standard. These files were created with a much smaller training dataset.

Q3 (10 points): Run build_NB1.sh with **train.vectors.txt** as the training data, **test.vectors.txt** as the test data, and class_delta set to 0. Fill out Table 1 with different values of cond_prob_delta.

Q4 (30 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.

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

cond_prob_delta	Training accuracy	Test accuracy
0.1		
0.5		
1.0		
2.0		

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

cond_prob_delta	Training accuracy	Test accuracy
0.1		
0.5		
1.0		
2.0		

Q5 (10 points): Binarize features first and then run build_NB2.sh for training and testing. You can use the tool created for Q2 in Hw2 to binarize features. Fill out Table 3.

Table 3: Results of your **multinomial** NB model with **binary** features

cond_prob_delta	Training accuracy	Test accuracy
0.1		
0.5		
1.0		
2.0		

Q6 (15 points):

(a): What conclusion can you draw from Tables 1-3?

(b): From the value of $P(f \mid c)$ in the model produced by the Bernoulli NB model, can you tell whether or not a feature is important? Why or why not?

(c): Answer Question (b) for the multinomial NB model.

(d): Between the two models, does one runs faster at the test time than the other one? If so, why?

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 hw3.*, include your answers to Q1-Q6, and any notes that you want the TA to read.
- Shell scripts for Q2 and Q4, and related source and binary code.

- The data files produced in Q1-Q6 (e.g., acc_file.2 is the acc_file when the cond_prob_delta is 2). Store the data files from Q1 under a subdirectory q1/, and so on.