Programming Assignment 1

(Submit via Blackboard by the 20th Feb. 2020)

In this assignment, you will build a naïve Bayes and a logistic regression classifier for sentiment classification. We are defining sentiment classification as two classes: positive and negative. Our data set consists of airline reviews. The zip directory for the data contains training and test datasets, where each file contains one airline review tweet. You will build the model using training data and evaluate with test data. Each of training data and test data contains 4182 reviews.

1. Build your naive Bayes classifier

a. Create your Vocabulary: Read the complete training data word by word and create the vocabulary V for the corpus. You must not include the test set in this process. Remove any markup tags, e.g., HTML tags, from the data. Lower case capitalized words (i.e., starts with a capital letter) but not all capital words (e.g., USA). Keep all stop words. Create 2 versions of V: with stemming and without stemming. You can use appropriate tools in nltk[1] to stem. Tokenize at white space and also at each punctuation. In other words, "child's" consists of two tokens "child and 's", "home." consists of two tokens "home" and ".". Consider emoticons in this process. You can use an emoticon tokenizer, if you so choose. If yes, specify which one.

b. Extract Features: Convert documents to vectors using Bag of Words (BoW) representation. Do this in two ways: keeping frequency count where each word is represented by its count in each document, keeping binary representation that only keeps track of presence (or not) of a word in a document.

c. Training: calculate the prior for each class & the likelihood for each word|class.
Note that if you want to experiment with different stemmers or other aspects of the input features, you must do so on the training set, through cross-validation. You must **not** do such preliminary evaluations on test data. Once you have finalized your system, you are ready to evaluate on test data.

d. Evaluation: Compute the most likely class for each document in the test set using each of the combinations of stemming + frequency count, stemming + binary, no-stemming + frequency count, no-stemming + binary.

e. Ignore any words that appear in the test set but not the training set.

f. Save your results in a.txt or .log file.

2. Build your logistic regression classifier.

a. Create your Vocabulary: Use the same V that you generated for your Naïve Bayes implementation.

b. Extract Features: Use the same features that you generated for your Naïve Bayes implementation. However, this time, use word frequency rather than binary representation as your input.

c. Training: Initialize weights to zeros. Use cross-entropy as the loss function and stochastic gradient ascent as the optimization algorithm. Set the sigmoid threshold to 0.5. Predict labels for each sample. Compute the cross entropy and gradient of predictions against the gold standard labels. Update weights with the gradient of the score function using an appropriate learning rate. Iterate until performance converges.

[1] nltk.org

Note that if you want to experiment with different learning rates or other aspects of the input features, you must do so on the training set, through cross-validation. You must **not** do such preliminary evaluations on test data. When you have finalized your system, features, and parameters, you can evaluate on test data.

d. Evaluation: Report your selected learning rate and the number of iterations. Run your logistic regression, as trained and tuned above, through each sample in the test data.
e. Ignore any words that appear in the test set but not the training set.
f. Save your results in a.txt or .log file.

3. Evaluation Methods. For each of your classifiers, compute and report accuracy. Accuracy is number of correctly classified reviews/number of all reviews in test. Also create a confusion matrix for each classifier. Save your output in a .txt or .log file.

   Bonus point: how would the results change if you used term frequency instead of binary representation for both logistic regression and Naïve Bayes (1 point)? What about term frequency x inverse document frequency (1 point)? How do your results change if you regularize your logistic regression (1 point)?

4. Documentation

a. Start all your files with a description of your code. Example description is shown below.
   "Author: Jamie Lee
   Description: AIT 726 Homework 1
   Command to run the file:
   (i.e. python naive_bayes.py [inputarg1] [inputarg2])
   (i.e. python naive_bayes.py) - if no input argument required
   Detailed Procedure: # show the flow of the code with little description with it

   i. main
   ii. Def run
   iii. Def load data
      1. ...
   i. Def training
      1. Calculating score function
      2. ...
   i. Def test
      1. ...

b. Write short description of each function on top of it

3. Deliverables
   Please submit a zip file named with student1[firstname initial][lastname]_student2[firstname initial][lastname]_[hw#].zip (i.e. student 1 jamie lee, student 2 kahyun lee: jlee_klee_hw1.zip).
   Zip file should include the following:

a. Your code(s)

b. .log file or .txt file that contains your output. You can choose whatever is convenient for you. .log can be created using logging library. .txt file can be created using simpleio library.

Reference:

Potts, Christopher. 2011. On the negativity of negation. In Nan Li and David Lutz, eds., Proceedings of Semantics and Linguistic Theory 20, 636-659.