Project 2

ESE 545, Data Mining: Learning from Massive Datasets

October 17, 2017

Due at 11:59PM on **October 29, 2017**

This question consists of several parts. You are required to solve the problems using Python, Matlab or C and turn in your code. You are allowed to work in groups of at most two members. You should submit your code with a brief report containing responses to each part. Turn in one report and script per group. Upload a zipped file containing the report and the code on Canvas.

Question 1.1. For the first part you are required to download the Sentiment140 dataset found here (http://cs.stanford.edu/people/alecmgo/trainingandtestdata.zip). Unzip the file and read in the CSV with training data (it has 1.6 million entries). The file contains a table of Sentiment, UserID, Date, no-query, user id, the actual tweet. Read in only the sentiment and the tweet. You are welcome to use other values as additional features. Sentiment in the data file is classified as either 0 (negative emotion) or 4 (positive emotion). Convert these to -1 and 1 respectively. **5 points**

Question 1.2. cleaning up the data Perform the following operations on each of the tweets.

1. Convert all letters into lowercase.
2. Convert all occurences of 'www.website' or 'https:website//' to URL . For example $http://twitpic.com/2y1zl$ becomes URL.
3. Remove additional white spaces.
4. Remove all punctuation.
5. Convert @username to AT-USER
6. Convert
7. Replace duplicate words - e.g. 'very very' should be replaced by very.
8. After Steps 1-6 have been performed, check if the tweet has any of the words from stopwords.txt (provided on Canvas). If there are any other words that are stop words, ignore them. Whatever now remains of the original tweet is the feature vector for that tweet.

   Create a table of feature vectors and sentiment. For example for the first tweet in the dataset, which looks like "@switchfoot http://twitpic.com/2y1zl - Awww, that's a bummer. You shoulda got David Carr of Third Day to do it. ;D" the extracted feature is "aww". Since it is a csv file, we only read till the tweet till the first comma appears to make things

simpler, So the tweet "@*switchfoot* http://twitpic.com/2y1zl - Awww, that's a bummer ..." is only read in as "@switchfoot http://twitpic.com/2y1zl - Awww". On doing steps 1-6 you get AT-USER URL Aww. The stopwords.txt has the words AT-USER and URL which get removed.

The second tweet "is upset that he can't update his Facebook by texting it... and might cry as a result School today also. Blah!" after doing steps 1-5 yields the features ['upset', 'update', 'facebook', 'texting', 'cry', 'result', 'school'] **35 points**

Question 1.3. Extract unigram features from the bag of words. The bag of words here is the set of all words collected after performing steps 1-6. You are free to use any library to create unigram features from the bag of words. To give an example (from wikipedia) :
Consider we have the tweets "John likes to watch movies. Mary likes movies too." and the tweet "John also likes to watch football games". The bag of words (Steps 1-6) gives us
(John,likes,to,watch,movies,Mary,too,also,football,games.)
To now extract unigram features, in the tweet "John likes to watch movies. Mary likes movies too." count the number of times each word appears in the bag of words. Thus, the feature vector would look like [1, 2, 1, 1, 2, 1, 1, 0, 0, 0]. John occurs once in the tweet, likes occurs twice and so on. For more detail see the wikipedia article on bag of words. You are free to extract bi-gram/n-gram features.
Thus, at the end of this you should have a list of features and the sentiment attached to these features. **10 points**

Question 1.4. Use PEGASOS to train an SVM on the features extracted above. Make a plot of training error vs number of iterations. **20 points**

Question 1.5. Use AdaGrad to train a classifier on the features extracted above. Report a plot of training error vs. number of iterations for every 1000 iterations. Merge this plot with the one from the previous question. No libraries are allowed here. **20 points**

Question 1.6. Read in the test CSV dataset. Perform all steps in Question 1.1 and 1.2. An additional step required is that the sentiments in the test set are 0, 2 and 4. Convert the tweets with sentiment 2 to 4. Report test accuracy and a plot of test error vs number of iterations for classifiers trained in Part 1.4 and Part 1.5. **10 points**