Ling/CSC 439/539: Assignment #2 (75 pts)
(Graduate students)

Due by 11:59 P.M., September 24
(upload all materials to D2L)

Last modified on: 8/26/2017

Requirements for the submission

You must submit code and a written report for this assignment. The code and report must follow these requirements:

1. Each programming question must be answered through code that is **executed with a single command** in the terminal. Please include one command for each of the requirements stated below in the assignment. Submissions that must be run through an IDE (e.g., Eclipse, IntelliJ, etc.) are not accepted.
2. Similarly, if your code requires compilation (e.g., it is written in Java), **a single command line for compiling the code must be provided**.
3. If your code requires certain dependencies (e.g., specific libraries, version of the Python language), these have to be clearly stated with instructions for installation.
4. Your report must include **clear instructions for all the above issues**.
5. The code for this assignment **cannot use ML libraries** such as TensorFlow, but may use libraries such as numpy for the linear algebra in the assignment such as managing vectors, and NLP libraries such as NLTK (`http://nltk.org`) or spaCy (`https://spacy.io`) for the tokenization of the text.

Points will be taken off if the above requirements are not met. Additionally, your code must **compile** (if required by the programming language), **run**, and **produce the correct output**. Points will be taken off in any of these issues are violated.

Data

In this assignment you will use the `SMSSpamCollection` dataset (available in D2L). This dataset contains SMS messages labeled either as `spam` or `ham`. The files contain one message per line. Each line is composed by two columns: one with label (ham or spam) and other with the raw text. Here are some examples:

```
ham   What you doing?how are you?
ham   Ok lar... Joking wif u oni...
ham   dun say so early hor... U c already then say...
spam   FreeMsg: Txt: CALL to No: 86888 & claim your reward of 3 hours talk time
  to use from your phone now! ubscribe6GBP/ mnth inc 3hrs 16 stop?txtStop
spam   Sunshine Quiz! Win a super Sony DVD recorder if you canname the capital
  of Australia? Text MQUIZ to 82277. B
spam   URGENT! Your Mobile No 07808726822 was awarded a L2,000 Bonus
  Caller Prize on 02/09/03! This is our 2nd attempt to contact YOU! Call
  0871-872-9758 BOX95QU
```

The dataset was split by the instructor in 3 files:

- `SMSSpamCollection.train` – The partition to be used for the training of your learning algorithm;
- `SMSSpamCollection.devel` – The partition to be used for the tuning of hyper parameters; and
- `SMSSpamCollection.test` – The testing partition to be used solely for the evaluation of your algorithm.

Problem 1 (35 points)

Implement a binary logistic regression (LR) algorithm and use it to train a spam classifier using the `SMSSpamCollection` dataset. Your LR algorithm must contain implementations from scratch for the sigmoid/logistic function, as well as for stochastic gradient descent (SGD). That is, you are not allowed to use an optimizer from another machine learning library. You may use numpy vectors in your code.

Train your algorithm using only the training partition. For features use unigrams (i.e., individual words in the messages). Tune any hyper parameters you have (I expect you will have at least two: number of epochs, and size of the minibatch) on the development partition. Include functionality to save the model learned to disk (after training), and to load a pre-trained model (before testing).

Answer the following questions:

1. What were your best hyper parameters according to the analysis on the development partition?
2. What was the classification accuracy (i.e., percentage of messages classified correctly) of your algorithm on the test partition, using the best hyper parameters from the development partition?

To answer these questions, your submission must contain:

- Code to train, tune, and evaluate the LR algorithm. In the report, include: (a) a single command line to train the algorithm and save the resulting model; (b) a single command line to tune the algorithm, and (c) a single command line to run the algorithm on a test partition, using a pre-trained model, and specific hyper parameters.
- Include your best model in the submission, so the instructors can run the algorithm on a test partition without retraining.
- In the written report, include the best values for your hyper parameters (question 1), as well as data to support your choices (e.g., classification accuracy on the development partition as you vary the hyper parameters).
- Include the accuracy on testing in the written report.

Problem 2 (10 points)

Add bigrams (i.e., contiguous sequences of two words) to the features used by your algorithm. How does performance change after adding these features?

Include in your submission the same materials as above, but adapted for a classifier that uses both unigrams and bigrams as features.

Problem 3 (10 points)

Filter your features (i.e., unigrams and bigrams) based on their frequency in the training partition. For example, in one model keep only features seen more than 1 time in training, in another only features seen more than 2 times in training, etc. How does classification accuracy change for the different filtering thresholds (try at least five values for the filtering threshold)?

Include in your report a chart that plots the classification accuracy for the different feature threshold values. Your code must contain the functionality to filter features. Include in your report pointers to where this functionality is implemented.

Problem 4 (10 points)

What are the top 20 features associated with the `spam` class, and the top 20 features associated with the `ham` class? For this analysis, you may use the best model you obtained as a result of the first three problems. What do you observe in this analysis? Do you believe your algorithm is overfitting, or not?

Problem 5 (10 points)

Find a research paper in the ACL Anthology (`http://aclweb.org/anthology/`) that uses text categorization and summarize it in your written report. What task is the paper addressing? What is the approach implemented? What are the main results and how do they compare with other approaches in the same space? What are the limitations of the proposed work?