Building Interactive Intelligent Systems
Assignment 1

ELEC4010I / COMP4901I
Hong Kong University of Science and Technology, Spring 2019

Deadline: Mon 13:30, 4 March 2019

1 Introduction

The goal of this assignment is to perform twitter sentiment analysis for sentence-level text. Students will derive and implement logistic regression from scratch with gradient descent. You first need to analyze the data, i.e., clean the data and perform feature extraction. Furthermore, you need to learn basic training and testing concepts and implement a sentiment classifier. To start, you can clone the starter code and dataset at https://github.com/hkusthltc/regression

2 Data Preprocessing and Analysis

Students will get two datasets, one is *twitter_sentiment.csv* and another is *twitter_sentiment_testset.csv*. The former is for training and contains 20k sentences and the labels, the latter is for testing only and contains 5k sentences. You need to submit the test set with the labels you predicted and should **not** try to label the test set manually.

2.1 Data Statistics

In this section, the only file you need to modify and execute is the *preprocess.py* file. You need to read the .csv file first, and then build the vocabulary dictionary. Please list the following data statistics:

- number of sentences, number of words, number of vocabulary, maximum sentence length, top 10 most frequently words, etc.

You can run the following command to check the results.

```
$ python preprocess.py
```

2.2 Data Cleaning

Based on the previous statistics, you can find that the original dataset is quite noisy. Please try to clean the dataset your own, such as removing dummy strings and dealing with punctuations. Please list at least three main methods you have used and report the data statistics again after cleaning. You can run the following command to check the results.

```
$ python preprocess.py -c
```

2.3 Feature Extraction & Normalization

Now, you need to change each sentence into features by the function *feature_extraction_bow*() in *preprocess.py*. There are various ways to do so, the easiest way uses bag-of-word (BoW), which is a sentence representation method based on word counting but disregards grammar and even word order. Please check on Wiki link for examples, url: https://en.wikipedia.org/wiki/Bag-of-words_model.

In the class, we talked about the mean and variance of the feature influence on the training stability. Now, please normalize each dimension of your feature to zero mean and set the standard deviation to 1 by the function *normalization*() in *preprocess.py*.

3 Logistic Regression

In this section, you are going to derive the cross entropy loss and the gradient of $\mathbf{W}$ and b for backpropagation. Suppose our binary classification dataset is $D = [(\mathbf{x}_1, C_{pos}), (\mathbf{x}_2, C_{neg}), , ...(\mathbf{x}_M, C_{pos})]$, where $\mathbf{x}_i$ is the input feature vector and C_{pos} or C_{neg} is the corresponding sentiment class. Assume that the data is generated by a function $f(\mathbf{x}) = p(C_{pos}|\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x} + b)$, then the probability of $p(C_{neg}|\mathbf{x})$ can be expressed by $1 - f(\mathbf{x})$. $\sigma(x)$ is the sigmoid function with the form $\sigma(x) = \frac{1}{1+e^{-x}}$. Therefore, the probability to generate the whole dataset D is

$$L(\mathbf{W}, \mathbf{b}) = \prod P_i, P_i = \begin{cases} f(\mathbf{x}_i) & if \quad C_i == C_{pos} \\ 1 - f(\mathbf{x}_i) & if \quad C_i == C_{neg} \end{cases} \tag{1}$$

We want to find the parameter $\mathbf{W}^*, b^*$ that can maximize the probability, which is equivalent to minimize the $-logL(\mathbf{W}, b)$.

$$\mathbf{W}^*, b^* = \arg\max_{\mathbf{W},b} L(\mathbf{W}, b) = \arg\min_{\mathbf{W},b} -logL(\mathbf{W}, b) \tag{2}$$

3.1 Formula

- Let's assume $C_{pos} = 1$ and $C_{neg} = 0$. Please represent the loss $-logL$ with C_i and $f(\mathbf{x}_i)$.
- Now we have our loss and we need to minimize it. Please compute the derivative of $-logL$ over $\mathbf{W}$ or b, that is, $\frac{\partial(-logL)}{\partial W}$ and $\frac{\partial(-logL)}{\partial b}$. Represent your answer with C_i and $\mathbf{x}_i$. (Hint: $\frac{\partial\sigma(x)}{\partial x} = \sigma(x)(1 - \sigma(x))$.)

3.2 Implementation

Based on your computation in the previous section, you can now modify the *logistic_regression.py* file to train a sentiment classifier. The functions you need to complete are the following:

- sigmoid(), compute_loss(), back_prop(), compare(), predict()

You can run the following command to start training the model. The prediction test set based on your model will be generated automatically after training. Please try your best to achieve higher development accuracy. We suggest to limit the maximum vocabulary size for faster training.

```
$ python logistic_regression.py -c --mv=10000 -lr=0.1 -fn=myTest
```

[Note] -c: use the data after cleaning; -mv: maximum vocabulary size; -lr: learning rate setting; -fn: file name for saving learning curve (*.log), parameters (*.npy) and test set prediction (*.csv).

Afterward, please try to answer and analysis the following questions (with figure is preferred).

- What is the effect of the learning rate? What is the best learning rate you found?
- What is the best accuracy you can get on the training set? how about on the development set?

- Describe whether the data normalization helps. Why or why not?
- List at least 10 sentences in the dataset that your model gives you a wrong prediction (sentence should be positive but the prediction is negative, and vice versa). Try to explain why.

4 Bonus

- Implement bi-gram feature and concatenate with the BoW feature. Show the results and whether the bi-gram feature helps.
- Implement the momentum method and explain how it helps gradient descent. One can check the gradient descent with momentum on Coursera for details.
- State some training phenomenons or tips you found.

5 Submission

Turn this in by **Mon 13:30, 4 March 2019** by emailing a zip file to hkust.hltc.courses@gmail.com and pascale@ece.ust.hk, with subject BIIS-HW1. Remember to state your name and student ID in the email. The zip file should **only** include the following materials:

- Report of the homework in pdf format (maximum 3 pages)
- Your model prediction of the test set (5k sentences) in csv format. Check the *sample_submission.csv* for example.
- Two python files: *preprocess.py* and *logistic_regression.py*

6 Enjoy the Homework