Assignment 2

Due on November 24, 2018 (23:59:59)

Click here to accept your Assignment 2

Instructions. There are two parts in this assignment. The first part involves a series of theory questions and the second part involves coding. The goal of this problem set is to make you understand and familiarize with Naive Bayes algorithm.

Part I: Theory Questions

MLE

- Suppose you have N samples $x_1, x_2.....x_N$ from a univariate normal distribution with unknown mean μ and known variance σ^2. Derive the MLE estimator for the mean μ.
- Consider a dataset $(x^n, c^n), n = 1, ..., N$ of binary attributes, $x_i^n \in 0, 1, i = 1, ..., D$ and associated class label c^n. The number of datapoints from class $c = 0$ is denoted n_0 and the number from class $c = 1$ is denoted n_1. Estimate $p(x_i = 1|c) \equiv \theta_i^c$.
- Suppose that X is a discrete rrandom variable with the following probability mass function: where $0 \leq \theta \leq 1$ is a parameter. The following 10 independent observations were taken from such a distribution: (3,0,2,1,3,2,1,0,2,1). What is the maximum likelihood estimate of θ.

X	0	1	2	3
P(X)	$2\theta/3$	$\theta/3$	$2(1-\theta)/3$	$(1-\theta)/3$

Naive Bayes

A psychologist does a small survey on 'happiness'. Each respondent provides a vector with entries 1 or 0 corresponding to whether they answer 'yes' to a question or 'no', respectively. The question vector has attributes
x = (rich, married, healthy)
Thus, a response (1, 0, 1) would indicate that the respondent was 'rich', 'unmarried', 'healthy'. In addition, each respondent gives a value c = 1 if they are content with their lifestyle, and c = 0 if they are not. The following responses were obtained from people who claimed also to be 'content': (1, 1, 1), (0, 0, 1), (1, 1, 0), (1, 0, 1) and for 'not content': (0, 0, 0), (1, 0, 0), (0, 0, 1), (0, 1, 0), (0, 0, 0) .

- Using Naive Bayes, what is the probability that a person who is 'not rich', 'married' and 'healthy' is 'content'?
- What is the probability that a person who is 'not rich' and 'married' is 'content'? (That is, we do not know whether ot not they are 'healthy'.)

PART II: Detection of Fake News

In this part of the assignment, you will try to determine whether a headline is real or fake news[1]. You will implement a Naive Bayes classifier and verify its performance on A Million News Headlines dataset. As you learned in class, Naive Bayes is a simple classification algorithm that makes an assumption about the conditional independence of features, but it works quite well in practice.

Dataset
1298 fake news headlines (which mostly include headlines of articles classified as biased etc.) and 1968 real news headlines, where the fake news headlines are from *https://www.kaggle.com/mrisdal/fake-news/data* and real news headlines are from *https://www.kaggle.com/therohk/million-headlines* have been compiled. Data is cleaned by removing words from fake news titles that are not a part of the headline, removing special characters from the headlines, and restricting real news headlines to those after October 2016 containing the word trump. For your interest, the cleaning script is available at clean_script.py, but you do not need to run it. The cleaned-up data is available below:

- Real news headlines: clean_real.txt
- Fake news headlines: clean_fake.txt

Each headline appears as a single line in the data file. Words in the headline are separated by spaces, so just use `str.split()` in Python to split the headlines into words.

A dataset is provided for your training phase (fake news 1104 lines, real news 1673 lines). Test set will be provided later and announced from Piazza group.

Approach

1. **Understanding the data**
   You will be predicting whether a headline is real or fake news from words that appear in the headline. Is that feasible? Give 3 examples of specific keywords that may be useful, together with statistics on how often they appear in real and fake headlines.

[1] This assignment is adapted from https://www.teach.cs.toronto.edu// csc411h/winter/projects/proj3/

2. **Implementing Naive Bayes**
   You will represent your data with listed approaches and use them to learn a classifier via Naive Bayes algorithm. You have to implement your own Naive Bayes algorithm.

   - Features: You will use Bag of Words (BoW) model which learns a vocabulary from all of the documents, then models each document by counting the number of times each word appears. You will use BoW with two options:
     - Unigram: The occurrences of words in a document(frequency of the word).
     - Bigram: The occurrences of two adjacent words in a document.

   Note: You should compute the log probabilities to prevent numerical underflow when calculating multiplicative probabilities.
   You may encounter words during classification that you havent during training. This may be for a particular class or over all. Your code should deal with that. Hint: You can use Laplace smoothing.
   You have to use a dictionary for BoW representation. You can implement your own method to obtain BoW model or you can use scikit-learn library (`http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html`).

3. (a) Analyzing effect of the words on prediction

   - List the 10 words whose presence most strongly predicts that the news is real.
   - List the 10 words whose absence most strongly predicts that the news is real.
   - List the 10 words whose presence most strongly predicts that the news is fake.
   - List the 10 words whose absence most strongly predicts that the news is fake.

   You can narrow down your dictionary by choosing specific words for real and fake news. In other words, your classification results can be improved by selecting a subset of extremely effective words for the dictionary. TF-IDF (`http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html`) and Information Theory are good places to start looking. Reimplement the part2 and see the effct of using specific words on the task.

   State how you obtained those in terms of the the conditional probabilities used in the Naive Bayes algorithm. Compare the influence of presence vs absence of words on predicting whether the headline is real or fake news.

(b) Stopwords
You may find common words like a, to, and others in your list in Part 3(a). These are called stopwords. A list of stopwords is available in sklearn here. You can import this as follows:
from sklearn.feature_extraction.text import ENGLISH_STOP_WORDS
Now, list the 10 non-stopwords that most strongly predict that the news is real, and the 10 non-stopwords that most strongly predict that the news is fake.

(c) Analyzing effect of the stopwords

Why might it make sense to remove stop words when interpreting the model? Why might it make sense to keep stop words?

4. **Calculation of Accuracy**
You will compute accuracy of your model to measure the success of your classification method:

$$\textbf{Accuracy} = 100 * (\frac{\textbf{number of correctly classified examples}}{\textbf{number of examples}}) \qquad (1)$$

Submit

You are required to submit all your code (*all your code should be written in Jupyter notebook* long with a report in ipynb format (should be prepared using Jupyter notebook). The codes you will submit should be well commented. Your report should be self-contained and should contain a brief overview of the problem and the details of your implemented solution. You can include pseudocode or figures to highlight or clarify certain aspects of your solution. Finally, prepare a ZIP file named **name-surname-pset2.zip** containing

- report.ipynb (Jupyter notebook file containing your report)
- code/ (directory containing all your codes as Python file .py)

The ZIP file will be submitted via Github Classroom. Click here to accept your Assignment 2

NOTE: To enter the competition, you have to register kaggle in Class with your department email account. The webpage of the competition will be announced later. Top 5 assignment will earn extra points.

Grading

- Code (50): Part1: 5, Part2: 25, Part3: 15, Part4: 5
- Report(50): Theory part: 12 points, Analysis of the results for prediction: 38 points.

  Notes for the report: Preparing good report is important as well as your solutions! You should explain your choices (Unigram, Bigram or both of their use for Bow, or constraints on data) and their effects to the results.

Late Policy

You may use up to four extension days (in total) over the course of the semester for the three problem sets you will take. Any additional unapproved late submission will be weighted by 0.5. You have to submit your solution in (rest of your late submission days + 4 days), otherwise it will not be evaluated.

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.