Assignment 2

Due on November 10, 2024 (23:00:00)
Programming Language: Python 3

Instructions There are two parts on this assignment. The first part involves a series of theory questions and the second part involves coding.

PART I: Theory Questions

1. One of the workers in an emergency call center is charged with calculating priorities of the cases that is reported to the department. He/She determines the priority of the case according to the probabilities, higher probability the case has, higher priority it has. Three phone calls arrive at the same time interval and all of the callers are calling from different neighborhoods but about the same topic: There is a smoke in their neighborhood and they are worried if the smoke caused due to a dangerous fire but they are too scared to observe the case from a nearby spot. The statistics are as follows, sort the cases according to their priority. **Do not forget to make explanations about your approach.**

    - The statistics show you that probability of the dangerous fires at the first neighborhood is rare (1%), however, barbecue smoke is quite common (20%), moreover smoke occurs 80% of the dangerous fires.
    - The statistics show you that probability of the dangerous fires at the second neighborhood is common (35%) and smoke due to factory near that neighborhood is quite rare (10%), moreover, smoke occurs 1% of the dangerous fires.
    - The statistics show you that probability of the dangerous fires at the third neighborhood is seldom (10%), however smoke due to usage of coal instead of natural gas is very common (80%), moreover, smoke occurs 30% of the dangerous fires.

2. There are two boxes, the first one contains 5 red and 3 blue balls while the second one contains 7 red and 4 blue balls. One of the boxes is selected with a bias, where probability of selecting the first box is 40%, and a ball is drawn from the selected box. Calculate the following with explanations:

    - The probability of blue ball is drawn.
    - The probability of second box has been selected if it is known that the blue ball is drawn.

3. Fill the blanks with T (True) or F (False) for the statements below, do not forget to make explanations:

    - Text classification is the primary application for Naïve Bayes classifier methods. (_)
    - When an attribute value in the testing record has no example in the training set, the total posterior probability in a Naïve Bayes algorithm will be zero. (_)

PART II: Movie Review Classification System

People are always in need of the others experiences as two heads are better than one, you can learn from other people's faults and gain the wisdom about it without suffering from the pain of the fault itself via experiences of the others, one of the experience transfer method is giving/reading reviews about the experiences itself, for example, people can leave reviews about the movies/series without giving spoilers, so that the others can have a information about the quality and the detailed genre of the movie/series. IMDb is one of the important review platform about movies, in short, it collects reviews about the movies/series from its users, the parts that users has to fill is rating (1-10) and comment about the content, according to this reviews, IMDb can create Top Movies/Series list and more to help movie and series lovers.

Dataset

The dataset contains movie reviews along with their associated binary sentiment polarity labels. It is intended to serve as a benchmark for sentiment classification. This document outlines how the dataset was gathered, and how to use the files provided. The core dataset contains 50,000 reviews split evenly into 25k train and 25k test sets. The overall distribution of labels is balanced (25k pos and 25k neg). It also includes an additional 50,000 unlabeled documents for unsupervised learning. In the entire collection, no more than 30 reviews are allowed for any given movie because reviews for the same movie tend to have correlated ratings. Further, the train and test sets contain a disjoint set of movies, so no significant performance is obtained by memorizing movie-unique terms and their associated with observed labels. In the labeled train/test sets, a negative review has a score ≤ 4 out of 10, and a positive review has a score ≥ 7 out of 10. Thus reviews with more neutral ratings are not included in the train/test sets. In the unsupervised set, reviews of any rating are included and there are an even number of reviews > 5 and ≤ 5. [1] [2] You are supposed to deal with the supervised part of the dataset, which means you just have to use "neg" and "pos" folders at the "train" and "test" folders, the rest of the dataset is given to you just for better understanding of yours.

Natural Language Processing Terminology

- **N-Gram:** An n-gram is a contiguous sequence of n items, items can be word, letter, or any of them, in Natural Language Processing domain it is mostly referred to words, and n can be any positive integer. For example every word can be considered 1-gram (which is usually called as unigram) and every string that contains two words can be considered as 2-gram (which is usually called as bigram). Considering the sentence "Lorem ipsum dolor sit amet, consectetur adipiscing elit.", unigrams are "Lorem", "ipsum", "dolor", "sit", "amet", "consectetur", "adipiscing", and "elit". Also punctuation marks can be considered as word for better understanding of data, in this manner "," and "." are also unigrams for this sentence. The bigrams of it is (including punctuation marks) "Lorem ipsum", "ipsum dolor", "dolor sit", "sit amet", "amet ,", ", consectetur", "consectetur adipiscing", "adipiscing elit", "elit .".

- **Stopwords:** Common words like "the," "is," and "and" that are frequently eliminated during text processing jobs since they usually don't have any relevance for analysis are known as stopwords.
- **TF-IDF (Term Frequency-Inverse Document Frequency):** A numerical metric called TF-IDF is used to assess a word's significance in a document in relation to a corpus, or group of documents. It considers a word's inverse frequency across all papers in the corpus as well as its frequency within a document (term frequency).

Steps to follow

1. Import and visualize the data in any aspects that you think it is beneficial for the reader's better understanding of the data.
2. Use BoW (Bag of Words) methodology to extract relevant information for your very own Naïve Bayes Algorithm (you must implement it from scratch) from the train data, you must use both unigram and bigram, you can also use trigram etc. for further implementation. Note that you may be in need of following to not fail with your implementation:
    - Use logarithmic probabilities instead of the raw format as it may cause numerical underflow. **Keep in mind that multiplication is addition in logarithmic domain according to the log(a*b)=log(a)+log(b) equation.**
    - Deal with the words that are not seen during training stage. (You may use Laplace Smoothing and unknown word handling via assuming that words that occur very rare are unknowns.)
    - You must implement a dictionary for your BoW approach, you must implement it from scratch.
3. Finally compute performance of your model to measure the success of your Naïve Bayes based classification algorithm for each setting you have used (unigram, bigram etc.):

    $$\textbf{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \qquad \textbf{Precision} = \frac{TP}{TP+FP} \qquad \textbf{Recall} = \frac{TP}{TP+FN}$$

    In addition to Accuracy, Precision, and Recall measures, F1-Score must be reported too:

    $$\textbf{F1-Score} = \frac{2*Precision*Recall}{Precision+Recall}$$

Bonus

How does the performance of the model get effected if "Word Embedding" with Logistic Regression is used instead of Bag of Words and n-gram approach at the previous part. You can use any extra library for this purpose, Word2Vec and Glove Techniques are one of the methodologies you may use. Feel free to play with data and generate new results, the points that you will get from bonus will be determined according to your effort, the better you do, the higher you get. Moreover, you can also use extra libraries, such as NLTK, for this part to achieve better accuracies for the BoW approach to compare three of them: Your approach from strach, your approach with NLTK, your approach with Word Embedding. **Keep in mind that you cannot use NLTK for the assignment itself, you can only use it for bonus part to compare and justify your results at the assignment. Moreover, you cannot get any bonus points if you have not completed both parts of the assignments, firstly you must complete assignment itself (it does not mean that you have to score 100 but it means that you must complete the assignment with considerable solutions).**

What to Hand In

You are required to submit all your code in a Jupyter notebook, along with a report in ipynb format, which should also be prepared using Jupyter notebook. The code you submit should be thoroughly commented. Your report should be self-contained and include a concise overview of the problem and the details of your implemented solution. Note that your report also has to contain necessary libraries to be installed (!pip install commands are preffered). Feel free to include pseudocode or figures to highlight or clarify specific aspects of your solution. Submission hierarchy must be as follows:

```
- b<StudentID>.zip
    - assignment2.ipynb
    - *.(jpg|jpeg|png|gif|tif|tiff|bmp|svg|webp) (optional)
```

Do not send the dataset.
P.S.: Please divide your Jupyter Notebook into two main parts for the two parts of this project, so, in short, you are supposed to give your answers to the theoretical questions in the first part at your Jupyter Notebook too.
Note that submission format is crucial and submit system is set to give you score as one if you follow the submission hierarchy, which is really easy (there might be some issues for the MacOS users but it can be overcomed via the mini guide that is shared along with this assignment itself at the Ed Platform). If you do not score one from the submit system **you will penalized by 20% even if your submission hierarchy is correct**.

Grading

- Part I : 20 Points
- Part II: 80 Points
- Bonus: There is not a specific maximum points that can be gathered for the bonus part, your work will be evaluated according to your effort.

P.S.: You can use libraries for visualization and explanation, but you must implement Naïve Bayes, BoW, Word Embedding, Logistic Regression, and things related to their mechanics from scratch, which means any library usage is forbidden except for the visualization and explanation purposes. Note that you can use NumPy, Pandas and libraries related to holding data in memory, it is an exception.

Note: Preparing a good report is important as well as the correctness of your solutions! You should explain your choices and their effects on the results. You can create a table (or any content you believe that it is beneficial to show your all work) to report your results.

Late Policy

You may use up to five extension days (in total) over the course of the semester for the three problem sets you will take. Any additional unapproved late submissions 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.

References

[1] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[2] Large movie review dataset. `https://ai.stanford.edu/~amaas/data/sentiment/` (Last access: 23.10.2024).