INF8953CE(Fall 2020) : Machine Learning — Given on: Oct 26, 10:00 pm

Programming Assignment #3 — Due on : Nov 16, 10:00 pm

- This is an individual assignment. However, you are allowed to discuss the problems with other students in the class. But you should write your own code and report.
- If you have any discussion with others, you should acknowledge the discussion in your report by mentioning their name.
- You have to submit the `pdf` copy of the report on gradescope before the deadline. If you handwrite your solutions, you need to scan the pages, merge them to a single `pdf` file and submit. Mark page 1 for outline items *'Late Submission'* and *'Verbosity'*.
- All the hyper-parameter details and other reporting details must be in the `pdf` report file only. You will have three types of files, 1) a single report (`pdf`), 2) dataset files (`.txt`), 3) a single code file (`.ipynb`). Submit all the text files and codes compressed to a single file `<your-matricule-ID>.zip` on **Moodle**.
  Note: gradescope doesn't accept .zip.
- Be precise with your explanations in the report. Unnecessary verbosity will be penalized.
- You are free to use libraries with general utilities, such as matplotlib, numpy and scipy for python. You can use pre-existing implementation of the algorithms available in scikit-learn package. Do not use NLTK or any other NLP libraries for pre-processing.
- If you have questions regarding the assignment, you can ask for clarifications in Piazza.

Sentiment Classification

In this assignment, we will design a sentiment classifier for classifying the sentiment of the reviews. This is a Natural Language Processing (NLP) task where the input is a natural language text and output is the sentiment label. We will consider two different review datasets: yelp reviews for restaurants and IMDB reviews for movies.

Yelp dataset

The Yelp dataset consists of 7000 reviews in the training set, 1000 reviews in the validation set, and 2000 reviews in the test set. This is a 5 class problem where each review is classified into one of the five ratings with rating-5 being the best score and rating-1 being the worst score.

IMDB dataset

IMDB dataset consists of 15000 reviews in training set, 10000 reviews in validation set, and 25000 reviews in test set. This is a 2 class problem with class 1 being positive sentiment and class 0 being negative sentiment.

1. Most of the algorithms described in the class expects input as a vector. However, the reviews are natural language text of varying number of words. So the first step would be to convert this varying length movie review to a fixed length vector representation. We will consider two different ways of vectorizing the natural language text: binary bag-of-words representation and frequency bag-of-words representation (as explained in the end of the assignment). Convert both the datasets into both these representations. Instruction for dataset submission is given in the end of the assignment (do not include the dataset in the report).

2. For this question, we will focus on the yelp dataset with binary bag-of-words (BBoW) representation. We will use the F1-measure as the evaluation metric for the entire assignment.

   (a) As a baseline, report the performance of the random classifier (a classifier which classifies a review into an uniformly random class) and the majority-class classifier (a classifier which computes the majority class in the training set and classifies all test instances as that majority class).

   (b) Now train Naive Bayes, Decision Trees, and Linear SVM for this task. [Note: You should do a thorough hyper-parameter tuning by using the given validation set. Also, note that you should use the appropriate naive Bayes classifier for binary input features (also called Bernoulli naive Bayes).]

   (c) Report the list of hyper-parameters you considered for each classifier, the range of the individual hyper-parameters and the best value for these hyper-parameters chosen based on the validation set performance[1].

   (d) Report training, validation, and test F1-measure for all the classifiers (with best hyper-parameter configuration).

   (e) Comment about the performance of different classifiers. Why did a particular classifier performed better than the others? What was the role of that hyper-parameter that fetched you the best results.

3. Now we will repeat question 2 but with frequency bag-of-words (FBoW) representation.

   (a) Train Naive Bayes, Decision Trees, and Linear SVM for this task. [Note: You should do a thorough hyper-parameter tuning by using the given validation set.

[1]Note that it is a good practise to discuss about your hyper-parameter tuning in any report or paper. Whenever you report the performance of an algorithm with hyper-parameters, you should mention all the hyper-parameters, the range of values you considered for each hyper-parameter and the final values you chose for the hyper-parameters to compute the test performance.

Also, note that you should use the appropriate naive Bayes classifier for real valued input features (also called Gaussian naive Bayes).]

(b) Report the list of hyper-parameters you considered for each classifier, the range of the individual hyper-parameters and the best value for these hyper-parameters chosen based on the validation set performance.

(c) Report training, validation, and test F1-measure for all the classifiers (with best hyper-parameter configuration).

(d) Comment about the performance of different classifiers. Why did a particular classifier performed better than the others? What was the role of that hyper-parameter that fetched you the best results.

(e) Compare the performance with the binary bag-of-words based classifiers. Why the difference in performance? Give a brief explanation comparing BBoW Naive Bayes and FBoW Naive Bayes and similarly for Decision Trees and Linear SVM.

(f) Which representation is better? Why?

4. Now we will repeat questions 2 and 3 but with IMDB dataset. For this question we will use IMDB dataset with BBoW representation.

   (a) As a baseline, report the performance of the random classifier (a classifier which classifies a review into an uniformly random class). Note that the IMDB dataset is a balanced dataset. Hence majority-class classifier doesn't make sense for this dataset, and you can omit it for this question.

   (b) Now train Naive Bayes, Decision Trees, and Linear SVM for this task. [Note: You should do a thorough hyper-parameter tuning by using the given validation set. Also, note that you should use the appropriate naive Bayes classifier for binary input features (also called Bernoulli naive Bayes).]

   (c) Report the list of hyper-parameters you considered for each classifier, the range of the individual hyper-parameters and the best value for these hyper-parameters chosen based on the validation set performance.

   (d) Report training, validation, and test F1-measure for all the classifiers (with best hyper-parameter configuration).

   (e) Comment about the performance of different classifiers. Why did a particular classifier performed better than the others? What was the role of that hyper-parameter that fetched you the best results.

5. Now we will consider IMDB dataset with frequency bag-of-words (FBoW) representation.

   (a) Train Naive Bayes, Decision Trees, and Linear SVM for this task. [Note: You should do a thorough hyper-parameter tuning by using the given validation set. Also, note that you should use the appropriate naive Bayes classifier for real valued features (also called Gaussian naive Bayes).]

(b) Report the list of hyper-parameters you considered for each classifier, the range of the individual hyper-parameters and the best value for these hyper-parameters chosen based on the validation set performance.

(c) Report training, validation, and test F1-measure for all the classifiers (with best hyper-parameter configuration).

(d) Comment about the performance of different classifiers. Why did a particular classifier performed better than the others? What was the role of that hyper-parameter that fetched you the best results.

(e) Compare the performance with the binary bag-of-words based classifiers. Why the difference in performance? Give a brief explanation comparing BBoW Naive Bayes and FBoW Naive Bayes and similarly for Decision Trees and Linear SVM.

(f) Which representation is better ? Why ?

(g) What did you observe about the relative performance of the classifiers when the dataset is changed? What aspects of the dataset aided or hampered the performance of that classifier.

Instruction for binary bag-of-words representation

1. First step is to construct the word vocabulary for a dataset. Given a dataset, we will only consider the training set and enumerate all unique words in the training set reviews[2]. To do this, we should do a bit of pre-processing to the raw text. For this assignment, we will do only 2 pre-processing steps: removal of punctuation, and lower-casing the words. For example, *(How old are you?)* would become *(how old are you)*.

2. Once you have the vocabulary, count the frequency of each word in the training set and sort the words in the vocabulary based on frequency (in descending order). Now pick top 10,000 words in the vocabulary and ignore the rest of the words. These 10,000 words will form the feature set for our classification process.

3. For any example to be classified, generate a 10,000 dimensional feature vector as follows: for each of the top 10,000 words, there is one corresponding dimension in the feature vector that is 1 if the example contains the word, and 0 otherwise. Make sure to use the same pre-processing steps as before for validation and test reviews.

Instruction for frequency bag-of-words representation

1. Follow steps 1-2 of the instructions for the binary bag-of-words.

2. Now, for each of the 10,000 words, the corresponding feature is the frequency of occurrence of that word in the given review. You can calculate this by summing the occurrences of words in a review into a histogram, and than divide by the sum of occurrences of all 10,000 words so that the vector for each example sums to 1.

[2]Think why should we not include validation and test set in the vocabulary construction process.

Instruction for dataset submission

For IMDB dataset, you should submit IMDB-train.txt, IMDB-valid.txt, IMDB-test.txt, IMDB-vocab.txt.

1. IMDB-vocab.txt file should contain the 10,000 words in the vocabulary, their corresponding id (starting from 1), and their frequency. Each line is a word, its numeric id, and its frequency all tab separated. Example:

   the 1 20456

   where *the* is the word, 1 is the id of the word, and 20456 is the frequency of the word.

2. For train/valid/test file, each line is a data point. Each review should be represented as space separated ids for corresponding words in the review[3] and the class label in mentioned in the end of the review as tab separated. Skip ids for words which are not there in top 10000. Just replace the word with the id (don't sort the ids in a row). Example:

   100 8 3 1034 0

   Here 0 is the class label and rest of the numbers represent a 4 word review.

Follow the same instructions for the yelp dataset as well. Add all files into the same zipfile as your code (see below).

Instruction for code submission

1. Submit a single zipped folder with your matricule id as the name of the folder. For example if your matricule ID is 12345678, then the submission should be 12345678.zip.

2. You can only use python and you must submit your solution as a jupyter notebook.

3. Make sure all the data files needed to run your code is within the folder and loaded with relative path. We should be able to run your code without making any modifications.

Instruction for report submission

1. You report should be brief and to the point. When asked for comments, your comment should not be more than 3-4 lines.

2. Do not include your code in the report!

[3] You can ignore words that are not in the vocabulary.