Exercise Sheet VIII

Submission Deadline: June 21th, 23:59

1 Text Classification using Naive Bayes

Exercise 1.1: Topic Categorization of News Documents (10 points)

In this exercise, you will work on the AG news documents dataset in `exercise8_corpora` and build a text classifier based on the Naive Bayes classification algorithm. Each instance in this dataset is a short text document that has been manually categorized into one of four possible categories: World, Sports, Business, Sci/Tech. Each line in the `.csv` files of this dataset is structured as `topic_ID, document`.

(a) Using a suitable tokenizer (e.g., `NLTK toktok`), preprocess the the training portion of the dataset and construct a vocabulary $\mathcal{V}$ of the words that occur at least 2 times in the entire training collection (lower-casing is recommended).

(b) From the preprocessed text, collect the following statistics

- D: the total number of documents
- D_k: the total number of documents labelled with class k
- $N_k(w_t)$: the frequency of word w_t in the documents of class k

(c) Estimate the priors $\mathbf{P}(C_k)$ as

$$\mathbf{P}(C_k) = \frac{D_k}{D}$$

(d) Estimate the likelihoods $\mathbf{P}(w_t|C_k)$ as

$$\mathbf{P}(w_t|C_k) = \frac{1 + N_k(w_t)}{|\mathcal{V}| + \sum_{w' \in \mathcal{V}} N_k(w')}$$

here, add-one smoothing is used to deal with zero-value probabilities.

(e) Design and implement a Python structure `NB_Classifier` that encapsulates the model parameters mentioned above, namely the priors and the likelihoods. The designed structure should also include a method `predict_class` that takes as an input a test document represented as a sequence of words d, and returns a predicted class of this document. This method should return the category label predicted as

$$\hat{C} = \text{argmax}_C \quad \log_2 \mathbf{P}(C) + \sum_{i=1}^{\text{len}(d)} \log_2 \mathbf{P}(w_i|C) \tag{1}$$

(f) Apply the same prepossessing steps that you have used on the training portion on the test portion. Using your method in part (e), predict the class for every test document. Evaluate the performance of Naive Bayes classifier using the accuracy metric

$$\text{accuracy} = \frac{\text{number of correct predictions}}{\text{total number of documents in the test set}} \tag{2}$$

(g) To get insights into the classification task, construct a confusion matrix (4×4 table) given the predictions in part (f). What are the two categories that the classifier gets confused between the most?

(h) What is the main idea behind Naive Bayes classification and what are the assumptions it makes? Starting from Bayes theorem, show the steps in order to get to the class prediction shown in Equation (1).

For this exercise, you are expected to provide a well-documented and efficient Python solution to solve the problem. To earn up to 4 extra points, you could investigate the impact of more prepossessing steps (stemming, lemmatization, etc.) or a different estimation method (e.g., other smoothing technique) on the classifier's performance.

Exercise 1.2: Sentiment Classification of Movie Reviews (6 points)

In this exercise, you will work on a dataset of movie reviews from Internet Movie Database (IMDb) where each review is categorized as either positive or negative with respect to the sentiment it expresses. You will find this dataset in `exercise8_corpora`. Contrary to the dataset in exercise 1.1, this dataset does not come with a 'split' for the train and test portions. Therefore, your task would be to use the k-fold cross-validation approach to evaluate the performance of the naive Bayes classifier. Proceed in this exercise as follows

(a) Preprocess the data as in Exercise 1.1 (a), then randomly shuffle the instances, where each instance is the pair $\langle \text{class}, \text{document} \rangle$.

(b) Split the datasets into 10 equally-sized partitions, or 'folds'. Then for each partition

1. Take this partition as a held-out test set
2. Take the remaining partitions as a training set
3. Estimate the model parameters using the training set
4. Evaluate the performance of the naive Bayes classifier on the test set

Keep in mind that each instance remains in the same partition during the entire procedure and each partition should be used as held-out set only once.

(c) Generate a table of the accuracy values obtained for each fold and show a summary statistics (e.g., mean, standard deviation) of these values. Briefly explain your observations.

(d) What is the main advantage of using k-fold cross-validation when building predictive models and evaluating model's performance?

(e) Compared to the topic categorization task, how difficult is sentiment classification for computational approaches? Justify your answer using the evaluation you performed in this exercise as well as Exercise 1.1.

The main goal in this exercise is to computationally build the k-fold cross validation procedure. Thus, you are advised to use the same code developed for Exercise 1.1 to estimate the model's parameters, class prediction, and evaluation but on the movie review dataset. You are expected to submit the source code and report your findings.

2 Instance-based Text Classification

Exercise 2.1: k-Nearest Neighbours Classification (4 points)

(a) Explain the main idea behind the k-nearest neighbours algorithm for text classification and discuss how it differs from naive Bayes classification.

(b) Write a simple pseudo-code (not more than 10 lines) for the k-nearest neighbours algorithm. Your pseudo-code should show the inputs and the output of the algorithm.

(c) How does the k-nearest neighbours algorithm learn the model's parameters from the data?

Submission Instructions

The following instructions are mandatory. Please read them carefully. If you do not follow these instructions, the tutors can decide not to correct your exercise solutions.

- You have to submit the solutions of this exercise sheet as a team of 2-3 students.
- Except for the tokenization and preprocessing, `NLTK` modules are not allowed, and not necessary, for this assignment.
- You do not need to include the distributed corpora within your submission.
- Make a single `ZIP` archive file of your solution with the following structure
  - A `source_code` directory that contains your well-documented source code and a `README` file with instructions to run the code and reproduce the results.
  - A `PDF` report with your solutions, figures, and discussions on the questions that you would like to include.
  - A `README` file with group member names, matriculation numbers and emails.
- Rename your `ZIP` submission file in the format

  `exercise08_id#1_id#2_id#3.zip`

  where `id#n` is the matriculation number of every member in the team.
- Your exercise solution must be uploaded by only one of your team members to the course management system (CMS). Once the grading is done, your assignment grade will be distributed to each team member by one of the tutors.
- If you have any problems with the submission, contact `babdullah@lsv.uni-saarland.de` before the deadline.