CS464 Introduction to Machine Learning
Fall 2022
Homework 1

Due: Oct 30, 2022, 11:59 PM

Instructions

- Submit a soft copy of your homework of all questions to Moodle. Add your code at the end of your homework file and upload it to the related assignment section on Moodle. Submitting a hard copy or scanned files is NOT allowed. You must prepare your homework digitally(using Word, Excel, Latex etc.).
- This is an individual assignment for each student. You are NOT allowed to share your work with your classmates.
- For this homework, you may code in any programming language you prefer. When submitting the homework, please package your file as a gzipped TAR file or a ZIP file with the name: `CS464_HW1_Section#_Firstname_Lastname`.

  As an example, if your name is Sheldon Cooper and you are from Section 1 for instance, then you should submit a file with name `CS464_HW1_1_sheldon_cooper`. Do NOT use Turkish letters in your package name.

  Your compressed file should include the following:

  - report.pdf : The report file where you have written your calculations, plots, discussions, and other related work.
  - q2main.*: The main code file of your work. It should be in a format that is easy to run and must include a main script serving as an entry point. The extension of this file depends on the programming language of your choice. For instance, if you are using Python, your code file should end with ".py" extension. You can also submit an IPython notebook file (.ipynb). If you are using MATLAB, your file should end with extension ".m". For other programming languages, your file should have the extension of the main executable file format for that language.
  - README.txt : You must also provide us with a README file that tells us how we can execute/call your program. The README file should include which parameters are the default values, the terminal command to execute your file, and how to read the outputs. For notebook files, a simple description of your implementation will suffice.
- You are NOT allowed to use machine learning packages, libraries, or toolboxes for this assignment (such as scikit-learn, TensorFlow, Keras, theano, MATLAB Statistics, and Machine Learning Toolbox functions, e1071, nnet, kernlab, etc.). However, packages allowing you to do efficient matrix multiplications (NumPy), tabular operations (pandas), and plotting utilities (seaborn, matplotlib) are allowed.
- We will evaluate your codes in terms of efficiency as well. Ensure you do not have unnecessary loops and apparent inefficient calculations in your code.
- If you do not follow the submission routes, deadlines, and specifications (codes, report, etc.), it will lead to a significant grade deduction.

- If you have any questions regarding this homework, you can contact:
  - a.yildirim@bilkent.edu.tr for Q1
  - yusuf.dalva@bilkent.edu.tr for Q2

1 The CS 464 Case [30 pts]

As a student taking the CS 464 course, people are asking you if they can get high grades in this course if they take it next semester. To answer this question, you decide to calculate probabilities using historical statistics. Students can be divided into two groups considering their study habits: the motivated (S_M) and the unmotivated (S_U). Even though it can be anticipated that there is a correlation between the grades and the students' motivation, there are some exceptions too.

The statistics show 87% of the high (H) grades, 21% of the low grades (L), and 4% of the failing (F) grades are received by the motivated students. When the distribution of the grade types in the data is calculated, it is found that 64%, 24%, and 12% of the students got high, low, and failing grades, respectively.

Considering the statistics, you aim to calculate the following probabilities:

Note: It is strongly suggested to use a calculator in the following questions. The results can be rounded to 4 decimal places.

Question 1.1 [10 pts] What is the probability that a student is motivated [$\mathbf{P}(S_M)$]?

Question 1.2 [10 pts] If a student is motivated, what is the probability that he/she will get a high grade [$\mathbf{P}(H|S_M)$]?

Question 1.3 [10 pts] If a student is unmotivated, what is the probability that he/she will get a high grade [$\mathbf{P}(H|S_U)$]?

2 Sports News Classification [70 pts]

For this question, you will adopt the role of a data scientist at BBC News. Your job is to develop an efficient model that classifies a sports news article into one of the following categories: football, cricket, athletics, rugby, and tennis.

Dataset

For this task, you will use the BBC Sports News dataset [1], provided by the company. This dataset contains 737 articles. We preprocessed them so that each instance is represented in a tabular form in BOW format. Namely, each row includes the number of occurrences of words the article contains. As the last column of the tabular data, the class labels are included, which are one of the following: athletics (0), cricket (1), football (2), rugby (3), and tennis (4).

We already provided you a data split containing a training set and a validation set to eliminate bias resulting from randomization. Your primary task in this question is to find the model (following the required algorithm) that performs the best in the given validation set. Remember that we select a model based on our validation data and only use test data to report our final metrics. Thus, for this homework, assume that there is a test set you are unaware of but has similar characteristics to the validation set.

Using the word frequencies and the document classes, you will use the following `csv` files as your training and validation set:

- `bbcsports_train.csv`

- `bbcsports_val.csv`

Each of these data files is in a tabular format with 4614 columns. The first 4163 columns indicate the word frequencies for the given article, whereas the last column shows the class label as a numeric value. Stop words are removed while preprocessing the dataset [1]. Each row in the given dataset contains a feature vector specifying the occurrences of vocabulary words, followed by the class label. The j^{th} element of the feature vector given in row i indicates the number of occurrences of j^{th} word of the vocabulary in the i^{th} news article. The class label for row i is given as the final value in that row, which is one of the following values: 0 (athletics), 1 (cricket), 2 (football), 3 (rugby), and 4 (tennis). There are no missing values in the dataset in feature vectors and data labels.

You can use external libraries to process these tabular files and to do operations on matrices and vectors. You need to specify such external libraries in your report if you used any. Remember that you cannot use libraries considered as ML-specific packages (e.g., scikit-learn, MATLAB Statistics, and Machine Learning Toolbox).

Bag-of-Words Representation and Multinomial Naive Bayes Model [2]

Recall the bag-of-words document representation assumes that the probability that a word appears in a news article is conditionally independent of the word position, given the class of the article. If we have a particular document D_i with n_i words in it, we can compute the probability of D_i being an instance of class y_k as:

$$\mathbf{P}\left(D_i \,|\, Y = y_k\right) = \mathbf{P}\left(X_1 = x_1, X_2 = x_2, .., X_{n_i} = x_{n_i} \,|\, Y = y_k\right) = \prod_{j=1}^{n_i} \mathbf{P}\left(X_j = x_j \,|\, Y = y_k\right) \tag{2.1}$$

In Eq. (2.1), X_j represents the word at j^{th} position in the document D_i and x_j represents the actual word that appears in the j^{th} position in that article, whereas n_i represents the number of positions in document D_i. As a concrete example, let us have two examples of news articles (documents):

- The first article (D_1) contains 350 words ($n_1 = 350$). The document might be an article about tennis, corresponding to the class label $y_k = 4$. Additionally, the 17^{th} position in the article might have the word "well" (x_{17} = "well" where $j = 17$).
- The 5^{th} article (D_5) contains 800 words ($n_5 = 800$). This time, the document might be an article about athletics, corresponding to the class label $y_k = 0$. Additionally, the 110^{th} position in the article might have the word "you" (x_{110} = "well" where $j = 110$).

In the above formulation, the length of the feature vector for document i, $\vec{X}_i$, depends on the number of words in the article n_i. That means that the feature vector for each article will be of different sizes. Also, the above formal definition of a feature vector $\vec{x}$ for an article says that $x_j = w_c$ if the j-th word in this article is the c-th word in the dictionary. This does not exactly match our feature files, where the j-th term in a row i is the number of occurrences of the j-th dictionary word in article i. As shown in the lecture slides, we can slightly change the representation, which makes it easier to implement:

$$\mathbf{P}\left(D_i \,|\, Y = y_k\right) = \prod_{j=1}^{|V|} \mathbf{P}\left(X_j \,|\, Y = y_k\right)^{t_{w_j,i}} \tag{2.2}$$

Here, V is the vocabulary and $|V|$ is the vocabulary size, X_j represents the appearance of the j-th vocabulary word, and $t_{w_j,i}$ denotes how many times word w_j appears in article D_i. For example, we might have a vocabulary of size 2300, $|V| = 2300$. The second article might be about cricket ($y_k = 1$). For this document, the word "should" might appear three times in the article, and it is the 50^{th} word in the vocabulary (w_{50} = "should"). This means that $t_{w_{50},2} = 3$ where $j = 3$ for the third article. Contemplate on why these two

[1] http://mlg.ucd.ie/files/datasets/stopwords.txt

models (Eq. (2.1) and Eq. (2.2)) are equivalent.

In the classification problem, we are interested in the probability distribution over the article classes (athletics, cricket, football, rugby, tennis) given a particular news article D_i. We can use Bayes Rule to write:

$$\mathbf{P}\left(Y = y_k | D_i\right) = \frac{\mathbf{P}\left(Y = y_k\right) \prod_{j=1}^{|V|} \mathbf{P}\left(X_j \,|\, Y = y_k\right)^{t_{w_j,i}}}{\sum_k \mathbf{P}\left(Y = y_k\right) \prod_{j=1}^{|V|} \mathbf{P}\left(X_j \,|\, Y = y_k\right)^{t_{w_j,i}}} \tag{2.3}$$

Note that, for classification, we can ignore the denominator here and write:

$$\mathbf{P}\left(Y = y_k | D_i\right) \propto \mathbf{P}\left(Y = y_k\right) \prod_{j=1}^{|V|} \mathbf{P}\left(X_j \,|\, Y = y_k\right)^{t_{w_j,i}} \tag{2.4}$$

$$\hat{y}_i = \arg\max_{y_k} \mathbf{P}\left(Y = y_k \,|\, D_i\right) = \arg\max_{y_k} \mathbf{P}\left(Y = y_k\right) \prod_{j=1}^{V} \mathbf{P}\left(X_j \,|\, Y = y_k\right)^{t_{w_j,i}} \tag{2.5}$$

Probabilities are floating point numbers between 0 and 1, so when you are programming, it is usually not a good idea to use actual probability values as this might cause numerical underflow issues. As the logarithm is a strictly monotonic function on [0,1] and all of the inputs are probabilities that must lie in [0,1], using the logarithm of the probability values instead of the actual probability does not change the decision of which class gives a better score for the given document D_i. Taking the logarithm gives us the following:

$$\hat{y}_i = \arg\max_{y_k} \left(\log \mathbf{P}\left(Y = y_k\right) + \sum_{j=1}^{|V|} t_{w_j,i} * \log \mathbf{P}\left(X_j \,|\, Y = y_k\right) \right) \tag{2.6}$$

Here, $\hat{y}_i$ is the predicted label for the i-th example.

Question 2.1 [10 points] If the ratio of the classes in a dataset is close to each other, it is a "balanced" class distribution; i.e. it is not skewed. Regarding the class imbalance problem, answer the following questions:

1. How are the classes distributed in the training set? Give the number of instances in each class with a suitable plot. Any plot that shows the class distribution is acceptable.
2. Is the training set balanced or skewed towards one of the classes? Do you think having an imbalanced training set affects your model? If yes, please explain how it affects the Naive Bayes model and propose a possible solution if needed.
3. Does the validation set, and the training set have similar data distributions? Regardless of the answer, if we had a bad split where validation set and the training set have different characteristics, which term in the Naive Bayes algorithm would be misleading?
4. If your dataset is skewed towards one of the classes, does this affect your reported accuracy? If yes, to what extent the reported accuracy is misleading in such an unbalanced dataset?

We provided you the MLE estimators for the parameters of the Multinomial Naive Bayes Model as follows. Here, article classes are labeled as y_k:

$$\theta_{j \,|\, y=y_k} \equiv \frac{T_{j,y=y_k}}{\sum_{j=1}^{|V|} T_{j,y=y_k}}$$

$$\pi_{y=y_k} \equiv \mathbf{P}\left(Y = y_k\right) = \frac{N_{y_k}}{N}$$

- T_{j,y_k} is the number of occurrences of the word j in documents with label y_k in the training set, including the multiple occurrences of the word in a single article.
- N_{y_k} is the number of articles having document label y_k in the training set.
- N is the total number of articles in the training set.

- $\pi_{y=y_k}$ estimates the probability that any particular document has class label y_k.
- $\theta_{j\,|\,y=y_k}$ estimates the probability that a particular word in document with label y_k will be the j-th word of the vocabulary, $\mathbf{P}\left(X_j \,|\, Y = y_k\right)$

For all questions after this point, you will assess your models using your validation set. Assume that your executives will test the final model you propose on a test set unknown to you.

Question 2.2 (Coding*) [30 points] Train a Multinomial Naive Bayes model on the training set and evaluate your model on the validation set given. Find and report the accuracy and the confusion matrix for the validation set, as well as how many wrong predictions were made.

In estimating the model parameters use the above MLE estimator. If it arises in your code, define $\log 0$ as it is, that is, -inf. In the case of ties, you should predict the class that is lower in order (the class index with the lower value).

Be aware that, for this part, you may find low accuracy values. To ensure that your algorithm works correctly, pay attention to your confusion matrix and try to elaborate on why your algorithm works in this way.

Hint: To simulate the behavior of the number '-inf', you can assign an arbitrarily small number to this value (like it is defined as `np.nan_to_num(-np.inf)` in NumPy), to handle overflow issues. If you make such an assumption, indicate it in your report. Due to the nature of the dataset, we suggest you use small enough values (closer to $-\infty$).

Question 2.3 (Coding*) [25 points] Extend your classifier so that it can compute a MAP estimate of θ parameters using a fair Dirichlet prior. This corresponds to additive smoothing. The prior is fair because it "hallucinates" that each word appears additionally α times in the train set.

$$\theta_{j\,|\,y=y_k} \equiv \frac{T_{j,y=y_k}+\alpha}{(\sum_{j=1}^{|V|} T_{j,y=y_k})+\alpha*|V|}$$
$$\pi_{y=y_k} \equiv \mathbf{P}\left(Y = y_k\right) = \frac{N_{y_k}}{N}$$

For this question set $\alpha = 1$. Train your classifier using the training set and have it classify the validation set and report validation-set classification accuracy and the confusion matrix.

Question 3.4 [5 points] Comparing the two models you trained, how does the Dirichlet prior α effects your model? Also, interpret the structure of the dataset. Given that the dataset does not include stop words, why are the two models different? Explain by giving references to your results. You can also benefit from the statistical structure of the feature matrix.

Hint: You don't need to know all of the statistical properties of the matrix. Think about how word occurrences are distributed. What about the words that occur rarely? How does these words effect your results?

References

[1] Derek Greene and Pádraig Cunningham. "Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering". In: *Proc. 23rd International Conference on Machine learning (ICML'06)*. ACM Press, 2006, pp. 377–384.

[2] Andrew Ng and Michael Jordan. "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes". In: *Advances in neural information processing systems* 14 (2001).