CS464 Introduction to Machine Learning
Fall 2023
Homework 1

Due: November 12, 2023, 23:59

Instructions

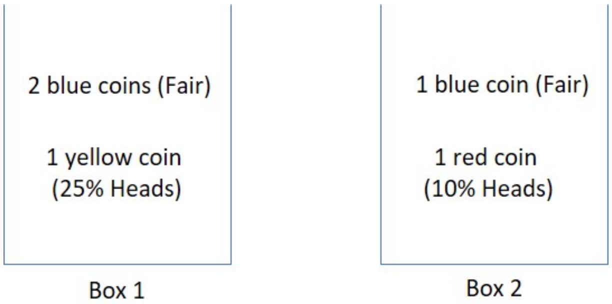
- Submit a soft copy of your homework of all questions to Moodle. Submitting a hard copy or scanned files is NOT allowed. You have to prepare your homework digitally(using Word, Excel, Latex etc.).
- This is an individual assignment for each student. That is, you are NOT allowed to share your work with your classmates.
- For this homework, you may code in any programming language you would prefer. In submitting the homework file, 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.
  - q3main.*: The main code file of your work. It should be in a format 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. If you are using a notebook editor, do not forget to save your file as a Python file at the end. 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. README file should include which parameters are the default values, what is the terminal command to execute your file and how to read the outputs.
- You are NOT allowed to use any 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, vector and matrix manipulation libraries (such as numpy, pandas etc.) and data visualization libraries (such as matplotlib, seaborn, plotly etc.) are allowed.
- Your codes will be evaluated in terms of efficiency as well. Make sure you do not have unnecessary loops and obvious inefficient calculations in your code. Execution time should not pass 5 seconds (excluding reading the dataset into memory).
- If you do not follow the submission routes, deadlines and specifications (codes, report, etc), it will lead to significant grade deduction.
- For questions regarding this homework, contact to hamza.pehlivan@bilkent.edu.tr or a.yildirim@bilkent.edu.tr.

1 Probability Review [10 pts]



There are 2 boxes in a room. The first box contains 2 blue coins and 1 yellow coin. The second box contains 1 blue and 1 red coin. The blue coins are fair. However, the yellow coin has 25% and red coin has 10% chance of landing heads.
You randomly select a coin from one of the boxes and toss it two times.
Question 1.1 **[4 pts]** What is the probability that you get two heads in a row.
Question 1.2 **[4 pts]** You toss the coin two times and got two heads in a row. What is the probability that the selected coin was fair?
Question 1.3 **[2 pts]** You toss the coin two times and got two heads in a row. What is the probability that the selected coin was the red one?
Note: Give your answers in 5 decimal points.

2 MLE and MAP [20 pts]

Suppose you have n data points $x_1, x_2....x_n$. After visualizing your data, you think that these data points are coming from a normal distribution. The probability density function for the normal distribution is given as:

$$f(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{2.1}$$

Question 2.1 **[8 pts]** Find MLE for the μ using the given data points.
Question 2.2 **[8 pts]** Suppose the prior distribution of μ is an exponential distribution with the parameter λ. Further assume that μ will be greater than 0. Find MAP estimate of μ. Probability density function of exponential function is given as:

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x}, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

Question 2.3 **[4 pts]** After completing the above steps, you decided the normal distribution represents your dataset quite well when $\mu = 1$ and $\sigma = 1$. Suppose you found a new data point x_{n+1}. What is the probability that this data point is equal to 1? After calculating the probability, you decided to measure data point's value and obtained 2. What is the likelihood of the data point $x_{n+1} = 2$ according to pdf of normal distribution?

3 BBC News Classification [70 pts]

As a data scientist working for BBC, your task is to develop a model that classifies news into 5 topics: Bussiness, Entertainment, Politics, Sport and Tech.

Dataset

For this task, your company provided you a dataset composed of 2225 real news [1]. Each of these news are labeled as Bussiness (0), Entertainment (1), Politics (2), Sport (3) and Tech (4). The dataset is already preprocessed such that each column indicates the number of occurrences of a given word for a given document instance. As a preprocessing stage, we dropped the column indicating instance numbers as they have no use. Additionally, data labels are given as a separate file.

The dataset has been split into two subsets: a 1668 news for training and 557 news for testing. For this task, treat your test set as a validation set and assume that there is another dataset that is not provided to you as a test set. Your company expects you to report the model that performs best on this given validation set and expects your model to behave similar in the test set that is not provided to you. To prevent any bias occurring from the order of the samples, the train-test split has been performed after shuffling the data.

Using the word frequencies, data files are generated for you. You will use the following files:

- `x_train.csv`
- `y_train.csv`
- `x_test.csv`
- `y_test.csv`

The files that start with `x` contain the features and the files starting with `y` contain the ground truth labels.

In the data files generated for you, each row contains a feature vector specifying the occurrences of vocabulary words. 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} document. There are 9635 words in your vocabulary, representing all of the words in all documents. The labels are represented with integer values from 0 to 4. In the dataset, there are no missing values both in feature vectors and data labels.

In the `csv` files provided to you, the data is organized in a tabular form. The feature files include a header specifying the words itselves. Label files do not have a header. The entities in the table are separated with space character.

Bag-of-Words Representation and Multinomial Naive Bayes Model

Recall the bag-of-words document representation makes the assumption that the probability that a word appears in a document is conditionally independent of the word position given the class of the document. 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:

$$\mathbf{P}\left(D_i \,|\, Y = y_k\right) = \mathbf{P}\left(X_1 = x_1, X_2 = x_2, .., X_{n_i} = x_{n_i} \,|\, Y = y_k\right) = \prod_{j=1}^{n_i} \mathbf{P}\left(X_j = x_j \,|\, Y = y_k\right) \tag{3.1}$$

In Eq. (3.1), X_j represents the word at j^{th} position in the news D_i and x_j represents the actual word that appears in the j^{th} position in the news, whereas n_i represents the number of positions in the news. As an example, let us have one of the examples from the news (documents):

- The first news (D_1) contains 341 words ($n_1 = 341$). The document is tech news, which corresponds to the class label $y_k = 4$. Additionally, the 17^{th} position in the news might has the word “well” ($x_{17} =$ “well” where $j = 17$).

In the above formulation, the length of the feature vector for document i ,$\vec{X}i$, depends on the number of words in the document n_i. That means that the feature vector for each document will be of different sizes. Also, the above formal definition of a feature vector $\vec{x}$ for a document says that $x_j = c$ if the j-th word in this document 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 that document i. As shown in the lecture slides, we can slightly change the representation, which makes it easier to implement:

$$\mathbf{P}\left(D_i \,|\, Y = y_k\right) = \prod_{j=1}^{|V|} \mathbf{P}\left(X_j \,|\, Y = y_k\right)^{t_{w_j,i}} \tag{3.2}$$

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 an news D_i. As an example, we might have a vocabulary of size 2300, $|V| = 2300$. The second news might be politics ($y_k = 2$). For this document, the word "should" might appear three times in the news and it is the 50^{th} word in the vocabulary (w_{50} = "should"). This means that $t_{w_{50},2} = 3$ where $i = 3$ for the third news. Contemplate on why these two models (Eq. (3.1) and Eq. (3.2)) are equivalent.

In the classification problem, we are interested in the probability distribution over the news classes (from 0 to 4) given a particular news D_i. We can use Bayes Rule to write:

$$\mathbf{P}\left(Y = y_k | D_i\right) = \frac{\mathbf{P}\left(Y = y_k\right) \prod_{j=1}^{|V|} \mathbf{P}\left(X_j \,|\, Y = y\right)^{t_{w_j,i}}}{\sum_k \mathbf{P}\left(Y = y_k\right) \prod_{j=1}^{|V|} \mathbf{P}\left(X_j \,|\, Y = y_k\right)^{t_{w_j,i}}} \tag{3.3}$$

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

$$\mathbf{P}\left(Y = y_k | D_i\right) \propto \mathbf{P}\left(Y = y_k\right) \prod_{j=1}^{|V|} \mathbf{P}\left(X_j \,|\, Y = y\right)^{t_{w_j,i}} \tag{3.4}$$

$$\hat{y}_i = \arg\max_{y_k} \mathbf{P}\left(Y = y_k \mid D_i\right) = \arg\max_{y_k} \mathbf{P}\left(Y = y_k\right) \prod_{j=1}^{V} \mathbf{P}\left(X_j \,|\, Y = y_k\right)^{t_{w_j,i}} \tag{3.5}$$

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, 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:

$$\hat{y}_i = \arg\max_{y} \left(\log \mathbf{P}\left(Y = y_k\right) + \sum_{j=1}^{|V|} t_{w_j,i} * \log \mathbf{P}\left(X_j \,|\, Y = y_k\right) \right) \tag{3.6}$$

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

Question 3.1 [10 points] If the ratio of the classes in a dataset is close to each other, it is called "balanced" class distribution; i.e it is not skewed. Regarding the class imbalance problem, answer the following questions:

1. What are the percentages of each category in the `y_train.csv` `y_test.csv`? Draw a pie chart showing percentages. [2.5 points]

2. What is the prior probability of each class? Write your answer to the report. [2.5 points]

3. 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 can affect the model briefly. [2.5 points]

4. How many times do the words "alien" and "thunder" appear in the training documents with the label "Tech", including multiple occurrences, and what is the log ratio of their occurrences within those documents, i.e, $\ln(P(alien \,|\, Y = Tech))$ and $\ln(P(thunder \,|\, Y = Tech))$? [2.5 points]

We provided you the estimators for the parameters of the Multinomial Naive Bayes Model as follows.

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

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

- T_{j,y_k} is the number of occurrences of the word j in news with class y_k in the training set including the multiple occurrences of the word.
- N_{y_k} is the number of news of class y_k in the training set.
- N is the total number of news in the training set.
- $\pi_{y=y_k}$ estimates the probability that any particular document will be class y_k.
- $\theta_{j \mid y=y_k}$ estimates the probability that a particular word in class y_k will be the j-th word of the vocabulary, $\mathbf{P}(X_j \mid Y = y_k)$

For all questions after this point, consider your test set as a validation set and assume that there is another test set that is not given to you. You will assess your models depending on how it performs on the validation set.

Question 3.2 (Coding*) [20 points] Train a Multinomial Naive Bayes model on the training set and evaluate your model on the test set given. Find and report the accuracy in **three** decimal points and report the confusion matrix for the test set.

In estimating the model parameters use the above estimator functions. If it arises in your code, define $\log 0$ as it is, that is -inf. In case multiple classes have the same score, you should favor the one with the lowest label id.
Hint: To simulate the behavior of the number '-inf', you can assign an arbitrarily small number to this value (like -10^{12}), to handle overflow issues. If you make such an assumption, indicate it in your report.
Hint 2: For this question you should get an accuracy around 0.242.

Question 3.3 (Coding*) [20 points] Extend your classifier so that it can compute an estimate of θ parameters using a fair Dirichlet prior. This corresponds to additive smoothing. The prior is fair in the sense that it "hallucinates" that each word appears additionally α times in the train set.

$$\theta_{j \mid y=y_k} \equiv \frac{T_{j,y=y_k} + \alpha}{\sum_{j=1}^{V} T_{j,y=y_k} + \alpha * V}$$

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

For this question set $\alpha = 1$. Train your classifier using all of the training set and have it classify all of the test set and report test-set classification accuracy in **three** decimal points and report the confusion matrix. Explicitly discuss your results and interpret on the effect of the Dirichlet prior α.

Hint 3: For this question you should get an accuracy around 0.977.

Bag-of-Words Representation and Bernoulli Naive Bayes Model

Question 3.4 (Coding*) [20 points] Train a Bernoulli Naive Bayes classifier using all of the data in the training set, and report the testing accuracy in **three** decimal points and report the confusion matrix. Compare your results with the previous ones.

Remember that this time, if the j-th word exists in the i-th news then the related term is set to 1, and 0 otherwise, that is, $t_j = 1$ when the word exists and $t_j = 0$ otherwise (a binary attribute, rather than the frequency of the given word in vocabulary). The formula for the estimated class is given in Eq. (3.7). In estimating the model parameters use the estimator equations, that are provided below. If it arises in your code, define $0 * \log 0 = 0$ (note that $a * \log 0$ is as it is, that is -inf), again you can use an arbitrarily small number instead of the value '-inf'. In the case of ties, you should predict the label with the lowest value. You should again use the additive prior with $\alpha = 1$ to deal with zero occurrences. Report your test set accuracy in your written report. What did your classifier end up predicting? Compare your results with the Multinomial Model. How does Bernoulli Naive Bayes model differ from Multinomial Naive Bayes? Discuss your findings and observations explicitly.

$$\hat{y}_i = \arg\max_y \left(\log \mathbf{P}\left(Y = y_k\right) + \log \left(\prod_{j=1}^{V} t_j * \mathbf{P}\left(X_j \,|\, Y = y_k\right) + (1 - t_j) * \left(1 - \mathbf{P}\left(X_j \,|\, Y = y_k\right)\right) \right) \right) \tag{3.7}$$

,where y_k is the predicted label for the i-th example and t_j indicates whether word j appears in the document.

The parameters to learn and their estimator functions are as follows:

$$\theta_{j\,|\,y=y_k} \equiv \frac{S_{j,y=y_k}+\alpha}{N_{y_k}+2\alpha}$$

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

- S_{j,y_k} is the number of occurrences of the word j in news with label y_k in the training set NOT including the multiple occurrences of the word in a single document.
- N_{y_k} is the number of news having class y_k in the training set.
- N is the total number of documents in the training set.
- $\pi_{y=y_k}$ estimates the probability that any particular news will be class y_k.
- $\theta_{j\,|\,y=y_k}$ estimates the fraction of the the j-th word of the vocabulary given that the class is y, $\mathbf{P}\left(X_j \,|\, Y = y_k\right)$. Note that this estimate should use additive smoothing.

Hint 4: For this question you should get an accuracy around 0.966.

References

1. BBC News Dataset http://mlg.ucd.ie/datasets/bbc.html