CS464 Introduction to Machine Learning
Spring 2020
Homework 1

Due: March 30, 2020, 11:55 PM

Instructions

- Submit a soft copy of your homework of all questions to Moodle. Add your code at the end of the your homework file and upload it to the related assignment section on 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.).
- 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.
- If you do not follow the submission routes, deadlines and specifications (codes, report, etc), it will lead to significant grade deduction.

1 Birthday Paradox [0 pts]

This question is changed to be an exercise question.

Assume that you are a student in an auditorium. Your math professor, David Caine, starts the semester with an interesting question: What is the probability that any two student in the auditorium has the same birthday?

Question 1.1 **[0 pts]** Ignoring the existence of leap years, find and plot the probability that at least two students have the same birthday in an auditorium of n students where $n \in \{2, 3, 5, 10, 20, 30, 40, 50, 75, 100\}$. Do not forget to add proper axis titles and legend (if necessary).

Question 1.2 **[0 pts]** Ignoring the existence of leap years, what is the minimum number of students to make sure that the probability of any two student have the same birthday will be 100%? Explain your reasoning explicitly.

2 Medical Diagnosis [0 pts]

This question is changed to be an exercise question.

Assume that, as a clinic worker, you are asked to conduct lab tests for diagnosis of a disease, namely ML464-HW1. From experiments, it is known that any person in the population either has the disease (positive), or has not (negative), i.e. there is no carrier. Over the entire population of people only 1.1% have this disease and the lab test returns a correct positive result in only 94% of the people in which the disease is actually present and a correct negative result in only 98% of the people in which the disease is not present. The state of the patient is represented with random variable S and the test results are represented with random variable T.

Question 2.1 **[0 pts]** Obtain and report the following probabilities:

- $\mathbf{P}(S = disease)$
- $\mathbf{P}(S = healthy)$
- $\mathbf{P}(T = positive \,|\, S = disease)$
- $\mathbf{P}(T = negative \,|\, S = disease)$
- $\mathbf{P}(T = positive \,|\, S = healthy)$
- $\mathbf{P}(T = negative \,|\, S = healthy)$

Question 2.2 **[0 pts]** Suppose a new patient comes to the clinic and the test returns a positive result. Show whether the patient should be diagnosed as having the disease or not. Explain your reasoning explicitly.

Question 2.3 **[0 pts]** Assume that the person is tested multiple times to make sure the result is confident if the test comes positive. What is the minimum number of tests you should conduct to diagnose the patient in the previous part as sick? Explain your reasoning explicitly.

3 HIV-1 Protease Cleavage [100 pts]

Dataset

A protease is an enzyme that is responsible for cleaving proteins into polypeptides or single amino acids. A specific protease, namely HIV-1 PR, cleaves polyproteins into smaller proteins that is essential for an HIV virion and allows the HIV virus to be infectious.

k-mers are subsequences of length k contained within any sequence. As an example assume that you have the sequence $ABCDEABCD$, a table for possible k-mers of this sequence is given below.

k	k–mers
1	A, B, C, D, E, A, B, C, D
2	AB, BC, CD, DE, EA, AB, BC, CD
3	ABC, BCD, CDE, DEA, EAB, ABC, BCD
4	ABCD, BCDE, CDEA, DEAB, EABC, ABCD
5	ABCDE, BCDEA, CDEAB, DEABC, EABCD
6	ABCDEA, BCDEAB, CDEABC, DEABCD
7	ABCDEAB, BCDEABC, CDEABCD
8	ABCDEABC, BCDEABCD
9	ABCDEABCD

Your task for this question is to identify amino acid 8-mers that can be cleaved by HIV-1 PR in the central position (between amino acids 4 and 5). Your dataset is taken from [1], which consist of 6590 labeled 8-mers in total. We have merged these datasets and created train and test sets of sizes 6062 and 528, respectively. You will use the following files for this question:

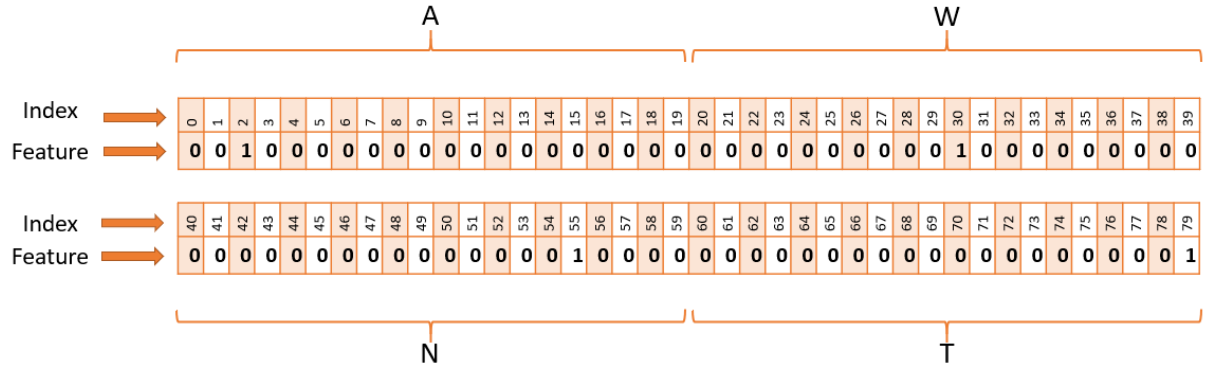
- *q2_train_set.txt*
- *q2_test_set.txt*
- *q2_gag_sequence.txt*

Each amino acid in the sequence is labeled using one-hot encoding, and since there are 20 amino acids in total, we have 160 features for an 8-mer. Each line in given files corresponds to a single 8-mer. First 20 columns denote the first amino acid, the second 20 columns denote the second amino acid, and so forth. The amino acids are enumerated as follows:

- G (Glycine), index: 0
- P (Proline), index: 1
- A (Alanine), index: 2
- V (Valine), index: 3
- L (Leucine), index: 4
- I (Isoleucine), index: 5
- M (Methionine), index: 6
- C (Cysteine), index: 7
- F (Phenylalanine), index: 8
- Y (Tyrosine), index: 9
- W (Tryptophan), index: 10
- H (Histidine), index: 11
- K (Lysine), index: 12
- R (Arginine), index: 13
- Q (Glutamine), index: 14
- N (Asparagine), index: 15
- E (Glutamic Acid), index: 16
- D (Aspartic Acid), index: 17
- S (Serine), index: 18

– T (Threonine), index: 19

For a 4-mer $AWNT$ for instance, there would be 80 features and all columns would be 0 except the columns with indices $2, 30, 55$ and 79, as represented in the figure below.



The final column in each row represents the ground truth label for the corresponding 8-mer. The label for any 8-mer is either *"1"*, meaning it can be cleaved by HIV-1 PR in the middle, or *"0"*, meaning HIV-1 PR cannot cleave this 8-mer in the middle.

Bernoulli Naive Bayes Model

The naive assumption allows us to think of all features as conditionally independent from each other so that we can multiply each $\mathbf{P}(X_i = x_i | Y = y_k)$, instead of considering their joint probabilities.

$$\mathbf{P}(S_i \,|\, Y = y_k) = \mathbf{P}(X_1 = x_1, X_2 = x_2, ..., X_n = x_n) = \prod_{j=1}^{n_i} \mathbf{P}(X_j = x_j \,|\, Y = y_k) \tag{3.1}$$

In Eq. 3.1, X_j represents a binary one-hot encoded feature that is described above, whose value is either 1 or 0. S_i denotes the i^{th} 8-mer sequence. Y represents the ground truth value for S_i. We have 2 classes, therefore y_k can be 0 or 1. We would like to find $\mathbf{P}(Y = 0 \,|\, X_1 = x_1, X_2 = x_2, ..., X_n = x_n)$ and $\mathbf{P}(Y = 1 \,|\, X_1 = x_1, X_2 = x_2, ..., X_n = x_n)$ so that the classification can be performed. Using the naive assumption and Bayes' equation together, we obtain the following relation:

$$\mathbf{P}(Y = y_k \,|\, X_1 = x_1, X_2 = x_2, ..., X_n = x_n) = \frac{\prod_{j=1}^{n_i} \mathbf{P}(X_j = x_j \,|\, Y = y_k)\, \mathbf{P}(Y = y_k)}{\prod_{j=1}^{n_i} \mathbf{P}(X_j = x_j)} \tag{3.2}$$

In Eq. 3.2, n_i denotes the total number of features. Please note that for both $y_k = 0$ and $y_k = 1$, the denominator of Eq. 3.2 remains the same. Therefore, to compare probabilities for each class, we can ignore the denominator value and define $\mathbf{P}(Y = y_k | S_i)$ as

$$\mathbf{P}(Y = y_k | S_i) \propto \mathbf{P}(Y = y_k) \prod_{j=1}^{n_i} \mathbf{P}(X_j \,|\, Y = y_k) \tag{3.3}$$

Probabilities are floating point numbers and multiplying many probability values could cause numerical inaccuracies or overflows. Therefore, it is a good practise apply $log(x)$ on these probability values to avoid

any computational issues. After taking log, the predicted label can be found using the equation below.

$$\hat{y}_i = \underset{y_k \in \{0,1\}}{\arg\max} \left(\log \mathbf{P}\left(Y = y_k\right) + \sum_{j=1}^{n_i} \log \mathbf{P}\left(X_j \,|\, Y = y_k\right) \right) \tag{3.4}$$

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

The parameters to learn and their MLE estimators are as follows:

$$\theta_{i,j\,|\,y=0} \equiv \frac{T_{j,y=0}}{\sum_{k=0}^{19} T_{i,k,y=0}}$$
$$\theta_{i,j\,|\,y=1} \equiv \frac{T_{j,y=1}}{\sum_{k=0}^{19} T_{i,k,y=1}}$$
$$\pi_{y=0} \equiv \mathbf{P}\left(Y = 0\right) = \frac{N_0}{N}$$
$$\pi_{y=1} \equiv \mathbf{P}\left(Y = 1\right) = \frac{N_1}{N}$$

- $T_{i,j,0}$ is the number of occurrences of the amino acid j at i^{th} position in 0 labeled instances of the training set.
- $T_{i,j,1}$ is the number of occurrences of the amino acid j at i^{th} position in 1 labeled instances of the training set.
- N_0 is the number of instances with label 0 in the training set.
- N_1 is the number of instances with label 1 in the training set.
- N is the total number of 8-mers in the training set.
- $\pi_{y=0}$ estimates the probability that any particular 8-mer will belong to class 0.
- $\pi_{y=1}$ estimates the probability that any particular 8-mer will belong to class 1.
- $\theta_{i,j\,|\,y=0}$ estimates the probability that a particular 8-mer will contain the amino acid j at i^{th} position in "0" labeled instances of the training set.$\mathbf{P}\left(X_j \,|\, Y = 0\right)$
- $\theta_{i,j\,|\,y=1}$ estimates the probability that a particular 8-mer will contain the amino acid j at i^{th} position in "1" labeled instances of the training set.$\mathbf{P}\left(X_j \,|\, Y = 1\right)$

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.

Question 3.1 [20 points] Train a Bernoulli Naive Bayes model on the training set and evaluate your model on the test set given. Find and report the accuracy for the test set.

If it arises in your code, define $\log 0$ as it is, that is -inf. In case of ties, you should predict "0". Report your test set accuracy in your report.

Question 3.2 [20 points] HIV-1 protease cleaves polyproteins such as "Group-specific antigen (Gag)". A sample gag polyprotein amino acid sequence of length 500 is given to you in a file with name *q2_gag_sequence.txt* [2].

– Use this file to create 493 8-mers by sliding a window of size 8 and one-hot encode amino acids in an 8-mer using the enumeration given in the Dataset section to obtain a dataset of size 493×160 for the gag polyprotein.

– Using your trained model, find all the locations where HIV-1 PT would cleave this polyprotein. Report your result by giving the exact amino acid indices for the whole 500 amino acid sequence where cleavage would happen (starting from index 0). For instance, if we had a polyprotein of 20 amino acids and our model labeled our 4^{th} and 10^{th} 8-mers as 1, then the cleavage would happen between amino acid indices 7-8 and 13-14, resulting with 3 smaller polypeptides in the end of lengths 8, 6 and 6 amino acids. Please remember that when your model label an 8-mer as 1, the cleavage would happen between the 2 amino

acids at the center. $GYWRKDIM$ would be cleaved between R and K amino acids if it is labeled as 1.

Question 3.3 [5 points] Using your trained model, find the 8-mer for which your model assigns it to class 1 with highest probability. Similarly, find the 8-mer for which your model assigns it to class 0 with lowest probability. Discuss your findings.

Question 3.4 [20 points] For Bernoulli Naive Bayes model, MLE estimation of the parameters are obtained using Bernoulli distribution. Now, let us change the prior distribution for parameters by introducing additive or Laplace smoothing.

Extend your classifier so that it can compute MAP estimates of θ parameters using a fair Dirichlet prior. This corresponds to additive smoothing. The prior is fair in the sense that it presumes that each amino-acid appears additionally α times at a specific position in all train set.

This time the parameters to learn and their MAP estimators are as follows:

$$\theta_{i,j\,|\,y=0} \equiv \frac{T_{i,j,y=0}+\alpha}{\sum_{j=0}^{19} T_{i,j,y=0}+2*\alpha}$$

$$\theta_{i,j\,|\,y=1} \equiv \frac{T_{i,j,y=1}+\alpha}{\sum_{j=0}^{19} T_{i,j,y=1}+2*\alpha}$$

$$\pi_{y=0} \equiv \mathbf{P}\,(Y=0) = \frac{N_0}{N}$$

$$\pi_{y=1} \equiv \mathbf{P}\,(Y=1) = \frac{N_1}{N}$$

For this question, use α values in $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and train your classifier using all of the training set and have it classify all of the test set and report test set classification accuracy for all given α values. Plot α vs. test set accuracy values and put your plot in your report. In addition, use the same set of α values given above to train your model using only first 75 rows of the training set and have it classify all of the test set. Report accuracy value and plot α values vs. test set accuracy once more. Do not forget to add proper axis names, plot title and legend to your plots. Compare your results from this question with the results from Question 3.1 and discuss the effect of additive smoothing. What happens when the amount of data you have is insufficient ? How does MAP estimates of parameters affect your performance when the amount of training data is small ? Explain your findings clearly.

Question 3.5 [20 points] For this part of the question, you are instructed to obtain mutual information values between each of the 160 features and the labels. You can check [4] to gain an insight about mutual information. Calculate mutual information of each feature with respect to label. If you encounter with a division by zero error, simply set the corresponding mutual information value to +inf. After finding mutual information values, perform the following operations in order.

- Sort mutual information values in descending order and also keep the original indices.
- Train a model (using MLE estimation for learning its parameters) using all training set instances, but only use first k features with the highest mutual information. Repeat this step for $\forall k \in \{1, 2, 3, ..., 160\}$. For each run, also calculate the test set accuracy and save them.
- After the step above is complete, find the maximum accuracy value and the corresponding k value for that accuracy value, and report them. Is the maximum accuracy you found higher than the accuracy value you obtained for Question 3.1 ?

Question 3.6 [15 points] For this part of the question, you are expected to apply principal component analysis (PCA) to the given dataset.

- Apply PCA on amino acid 8-mers, X_i, to obtain first 3 principal components for each X_i. Plot 3D projections of amino acid 8-mers by using these components.
- Report proportion of variance explained (PVE) for the obtained principal components. Discuss your results.
- Discuss whether it is feasible to apply PCA to given data set. Explain your reasoning.

References

1. UCI - HIV-1 protease cleavage Data Set https://archive.ics.uci.edu/ml/datasets/HIV-1+protease+cleavage#
2. Gag protein sequence https://www.uniprot.org/uniprot/O93182
3. "On Discriminative vs. Generative Classifiers: A comparison of logistic regression and Naive Bayes" by Andrew Ng and Michael I. Jordan.
4. Manning, C. D., Raghavan, P., and Schutze, H. (2008). Introduction to information retrieval. New York: Cambridge University Press.
http://nlp.stanford.edu/IR-book/html/htmledition/mutual-information-1.html
5. CMU Lecture Notes.
http://www.cs.cmu.edu/~epxing/Class/10701-10s/Lecture/lecture5.pdf