CS464
Introduction to Machine Learning Course
Spring 2020 Homework 1
Report

Ömer Faruk Akgül
21703163 CS
Section 1

30/03/2020

Question 3.1

For this part of the question, two different methods are written to see the effects of different approaches. The first one approaches the problem in a way that

$$\frac{P(Y/X)=P(X/Y)P(Y)}{P(X)} \quad (1)\ [1]$$

, where X is a sequence of amino acids such as “GPRKAWLA”.

To do so, the probability of occurring at the i^{th} position of all amino acids are calculated by simply finding the ratio between the number of ones at that row and the total number of samples. This operation is carried by the mean function of the NumPy library. Then, since the important point is to predict which class given amino acid sequence(0 or 1) belongs to, the denominator is ignored. To be able to find the value of conditional probability $P(X_i \mid Y)$, the training set is split into two parts according to their labels. Though $P(X \mid Y)$ is not equal to $P(X_1 \mid Y)*P(X_2 \mid Y)*...*P(X_n \mid Y)$, it is assumed as they are independent to simplify the calculation and decrease the calculation time. Since multiplying small numbers may cause loss of information, the log of each $P(X_i \mid Y)$ is summed up. Labels of the test set data are predicted by comparing the magnitudes of the values of $P(Y=1 \mid X)$ and $P(Y=0 \mid X)$. Then, accuracy is calculated by finding the true prediction rate.

The second approach is that

$$\frac{P(Y/X)=P(X/Y)P(Y)}{P(X)} \quad (2)$$

, where X is representative of one-hot encoding. Thus, X consists of 160 parameters.

The remaining part of the algorithm is the same. I simply used the Naive Bayes approach given in the homework description. By using prior, P(Y), the posterior is calculated for both $P(Y=1 \mid X)$ and $P(Y=0 \mid X)$. After comparing obtained values, given sequences are labeled as 1 or 0.

After training the model by using the training set, the label of the test set sequences are predicted. For both approaches given above, the accuracy rate is found as **94.886** %.

During the calculations, log0 value is obtained because $\log(P(X_i \mid Y) = 0)$ for some sequences and this value is defined as it is, -inf.

Short Description of the Algorithm:

- Reading training and test files, and transform it into matrices of the form n*161.
- Splitting training data into two parts according to their labels
- Finding priors by simply calculating the means of the columns, (for P(Y=0), 1 - P(Y=1))
- Reversing the training matrix($0 \rightarrow 1, 1 \rightarrow 0$)
- Multiplying each row of the original matrix by P(Y=1)
- Multiplying each row of the reversed matrix by P(Y=0)
- Summing up the original and reversed matrix
- Taking the log of the matrix and summing up row-wise
- Comparing obtained values of split training data
- Predicting the label and calculating the accuracy

Question 3.2

For this part of the question, the given text file is read and the given protein is split into 493 parts. 8-mers are one-hot encoded as given for the previous part of the question. The Naive Bayes model trained for question 3.1 is given the created array of the size 493*160 as the test set. Then, the indexes of sequences which are predicted as 1 are put into a new set containing values between 0 and 492. To find the exact location of cleavage, indexes are concatenated with 3. Corresponding locations are found as

“The indices where cleavage may occur: ['5-6', '40-41', '42-43', '60-61', '79-80', '131-132', '168-169', '183-184', '196-197', '215-216', '295-296', '315-316', '320-321', '341-342', '342-343', '362-363', '366-367', '376-377', '447-448', '465-466', '467-468', '482-483']”.

Question 3.3

For the third part of the question, two return values of the Naive Bayes are searched, “log_condition_one_sum” and “log_condition_zero_sum”. The first one keeps the obtained ratio for P(Y=1 | X). Similarly, the second one keeps the calculated ratio for P(Y=0 | X). The maximum ratio found in the first array corresponds the 8-mer **ARVLAEAM**. The minimum ratio in the second set points to the 8-mer **RKKGCWKC**. In other words,

“Model assigns class 1 with the highest probability is ARVLAEAM
Model assigns class 0 with the lowest probability is RKKGCWKC”

There are two important reasons why these two sequences are different:

- To predict the label, the important issue was to find whether P(Y=1 | X) has a higher probability than P(Y=0 | X). As remembered, the denominator was ignored because it is same for both situations. However, ignoring causes loss of information because, though denominator(P(X)) is different for different amino acid sequences, we compared retrieved ratios as they all have same P(X) value. In order to be able to find a more reliable result, MAP estimator given above should be used without throwing a part of the formula away.

- The second reason of the error is that components of $P(X_1X_2...X_N \mid Y)$ is thought of as independent values and written as $P(X_1 \mid Y)*P(X_2 \mid Y)*...*P(X_N \mid Y)$. However, proteins are not composed of random sequences so these components are dependent.

Question 3.4

For this part of the question, the prior distribution for parameters is changed by introducing additive or Laplace smoothing. This time, 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 so that log0 error is solved, as well. To illustrate,

$$\theta_{i,j/y=0} = \frac{T_{i,j/y=0} + \alpha}{\sum_{j=0}^{19} T_{i,j/y=0} + 2\alpha} \quad (3)$$

,where α takes values {0,1,2,3,4,5,6,7,8,9,10}. In denominator, 2α is added because label takes two values {0,1}. θ 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.P (X j | Y = 0).

Since priors are changed, there existed some changes on posterior probabilities. Calculated accuracies are

“Accuracies obtained for the alpha values from 0 to 10 [0.94886364 0.94886364 0.9469697 0.9469697 0.94886364 0.94886364 0.94507576 0.94507576 0.9469697 0.94318182 0.94318182]”

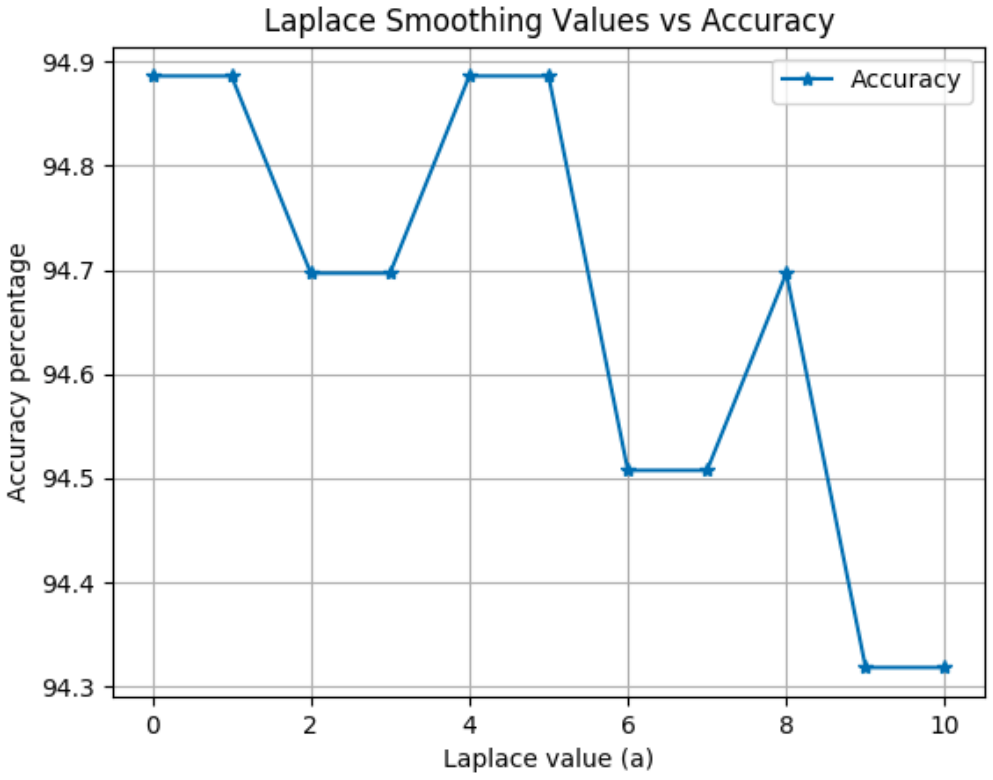


Figure 1

As seen from the graph, the best accuracy is found at **α = 0, 1, 4, 5**

For the second part of question 3.4, the model is trained by using only the first 75 rows of the training set. The remaining part of the process is the same. The results are as follows.

"[0.84848485 0.8655303 0.81628788 0.81628788 0.81628788 0.81628788
0.81628788 0.81628788 0.81628788 0.81628788 0.81628788]"

As seen from the set, the best accuracy is found at **α = 1**. The values obtained for the given set of α are visualized in the following graph.

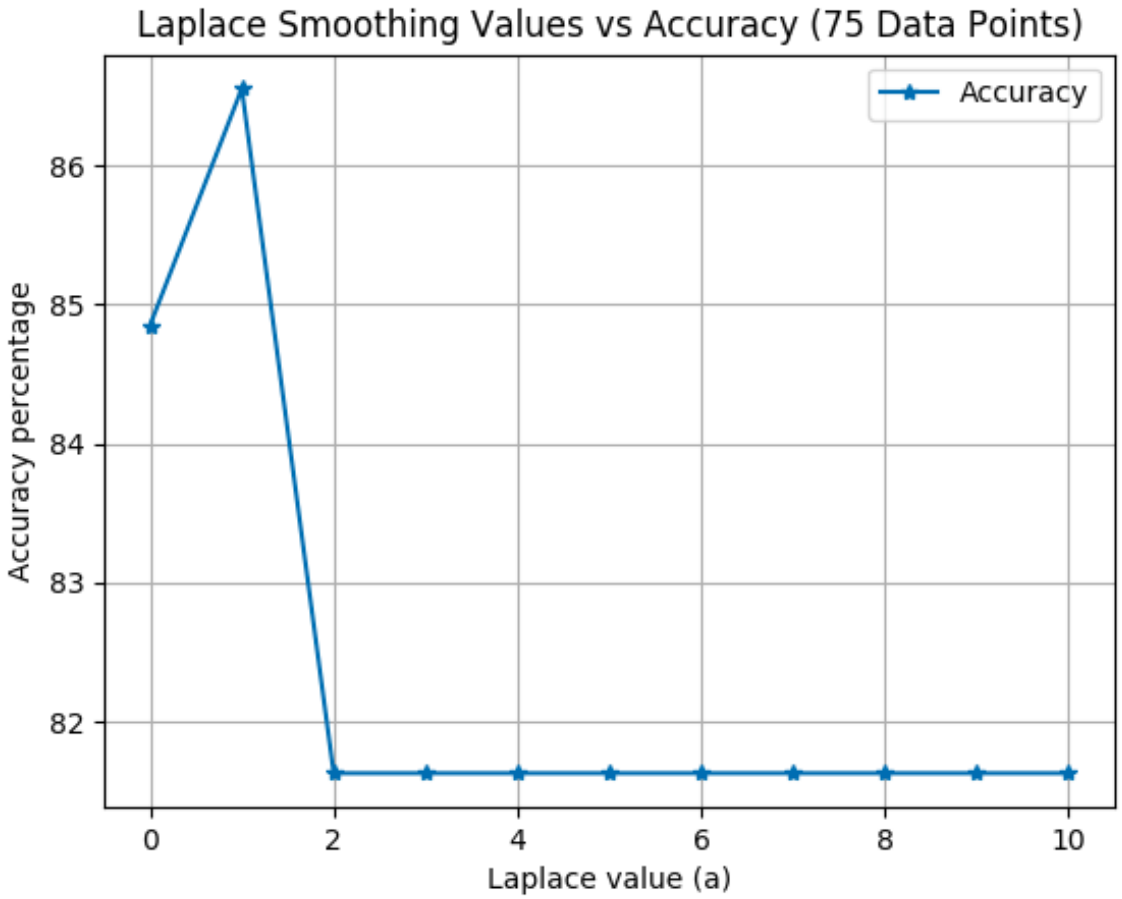


Figure 2

As understood from the graphs, adding small α values such as 1 and 2 may improve the accuracy because before adding that α, some of the conditional probabilities $P(X_i \mid Y)$ were equal to 0, which caused log0 error, that is -inf. Adding relatively small α values helps handling that error.

However, increasing the value of α destroyed the actual probability distribution that resulted in lower accuracy values. The reason why better accuracy could not be obtained for the first part might be there are only 2 such 0s that did not affect the prediction as thought.

The second accuracy set shows what happens when the otal number of the data is insufficient. Since increasing the number of samples helps model learn better, an insufficient number of training data means limited ability to predict unseen data. Moreover, when the amount of data is small, the reliability decreases because data might not exactly reflect the actual distribution. As a result, little training data results in a poor approximation.

Question 3.5

In this part of the question, we are supposed to sort features according to information they contain about the class. In other words, the perfect indicator of the class is looked for. To sort features, the following formula is used.

$$I=\frac{N_{11}}{N}\log_2\frac{NN_{11}}{N_{1.}N_{.1}}+\frac{N_{01}}{N}\log_2\frac{NN_{01}}{N_{0.}N_{.1}}+\frac{N_{10}}{N}\log_2\frac{NN_{10}}{N_{1.}N_{.0}}+\frac{N_{00}}{N}\log_2\frac{NN_{00}}{N_{0.}N_{.0}} \quad (4)$$

, N_{ij} represents the total number of data where the feature has value i while belonging to class j [2].

By the formula above, mutual information of the features are calculated. According to descending value of information, indices are listed below:

“[43 69 64 68 116 126   1 18 84 88 159 33 100 63 41 103 97 79
101 72 92 80 66 89 117 121 58 152 95   3 112 54 74 65 40   4
105 98 55 146 83 102 94 23 76 153 96 109 115 21 28 118 35   5
149 132 143 154 53 111 56 44 12 34 60 136 113 45 25 141 108 38
91 62 70 120 71 87 49 158 134   2 81 77 122 14 110 150 51 39
48   8 15 16 50 82 67 20 128 135   6 99 156 10 46 36 107 93
75   9 86 129 106 22 155 147 124 73 140 30 145 114 127 142 130 148
26 37 32 144   0 151 85 78 104 24 31 123 57 17 42 138   7 29
131 119 47 157 139 11 137 19 27 125 90 133 13 59 61 52]”

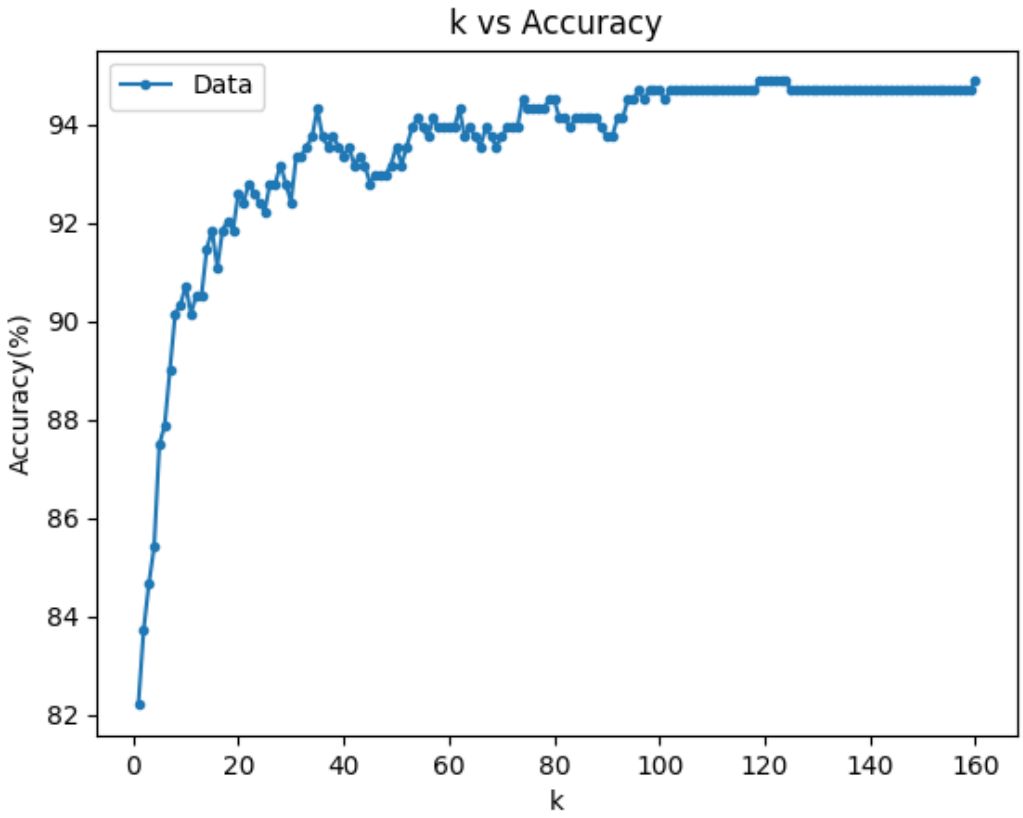
Then, the Naive Bayes method written for the first question is given first k features with the highest mutual information. Corresponding accuracies are recorded. The following list displays sorted k values in a descending order according to obtained accuracy values.

“[160 124 123 122 121 120 119 112 125 118 117 116 115 114 113 111 128 110
109 108 107 106 105 104 103 102 100 99 98 96 126 127 129 144 158 157
156 155 154 153 152 151 150 149 148 147 146 145 143 136 130 131 132 133
134 135 137 138 139 140 141 142 159 94 74 79 80 95 101 97 75 35
62 76 77 78 54 92 57 93 81 86 82 87 88 85 84 67 59 60
61 64 55 89 53 83 71 72 73 58 38 56 36 63 34 65 91 68
90 70 37 39 41 33 69 50 52 66 31 43 40 32 44 42 28 49
51 47 46 48 45 22 29 27 26 20 23 24 21 30 25 18 19 17
15 14 16 10 13 12   9 11   8   7   6   5   4   3   2   1]”

Best accuracy value corresponds to k=160 through 119 in the list above. Thus, we can use most important 119 of the 160 features to obtain exactly the same results.

Maximum accuracy value is : 94.88636363636364% and the corresponding k value is 119.

Following graph visualizes the relationship between selected features and acccuracies.



Question 3.6

To implement Principal Component Analysis on the given data set, the labels(161st column) is discarded. Then, the data is zero centered. Singular values, U and V^T matrices are calculated by using Singular Value Decomposition method. Since the question asks to find first three principal components, original data set is multiplied by first three vectors in the set V^T ((6062*160) * (160*3) forms a matrix with size 6062*3). Though dimensions with the highest variances are chosen, the proportion of variance explained is found as **3.194%**. Therefore, these three dimensions are not enough to represent the whole data set. Moreover, applying PCA is not reasonable for this data set. The graph in Figure 3 visualizes the first three principal components.

Because singular values obtained by using SVD are positive semi-definite, errors caused by complex numbers are prevented. However, using numpy.linalg.svd caused my program to spend extra time. While the program of first five parts of the question is spending less than 3 seconds, SVD part spends 8 seconds on its own. In order to see the efficiency of SVD, another method is written. After

finding eigenvalues of the covariance matrix, the most important three of them are chosen. Then, the data set is multiplied by eigenvectors corresponding to the chosen eigenvalues. Though this method spends ignorable time compared to the previous one, it caused some errors because of complex numbers. By this method, the proportion of variance explained is found as **6.98%**.

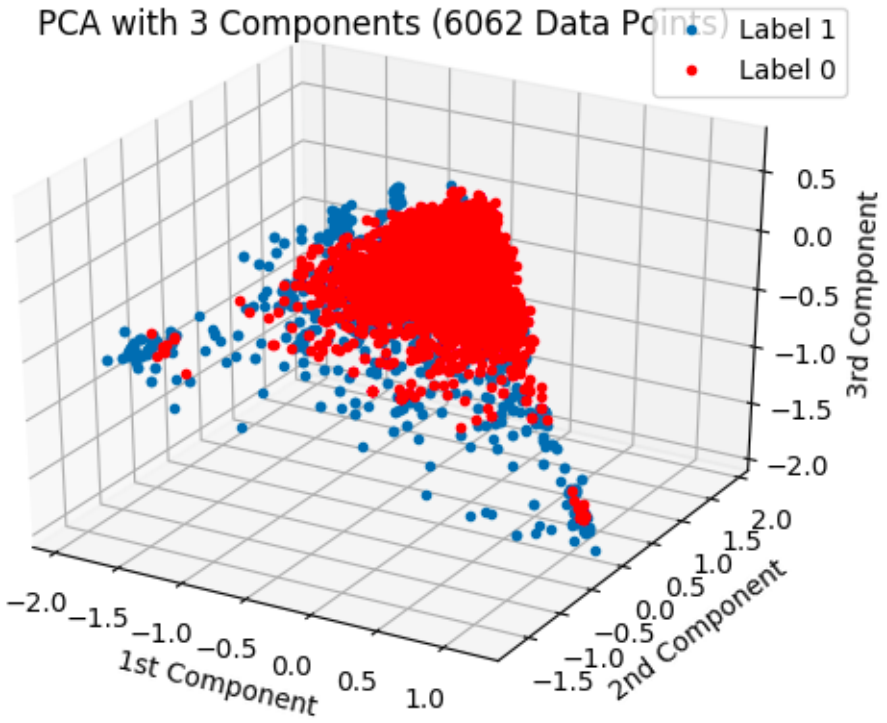


Figure 3

REFERENCES

[1] Murphy, Kevin P. *Machine Learning: a Probabilistic Perspective*. MIT Press, 2013.

[2] *Mutual Information*, nlp.stanford.edu/IR-book/html/htmledition/mutual-information-1.html.