# HW 2 Solutions

*Author:*
Fateme Noorzad

*StudentID:*
810198271

Machine Learning
Fall 2020

# Contents

# Chapter 1

# Question 1

To find the value for $\underline{W}$ such that the given cost function is maximized, we need to find the derivation of the cost funtion with respect to $\underline{W}$. By letting the resulting equations to zero, we can find $\underline{W}$ that let us achieve our goal.

$$\frac{\partial J(\underline{W})}{\partial \underline{W}} = 0$$
$$\rightarrow \frac{\partial J}{\partial \mu_1}\frac{\partial \mu_1}{\partial \underline{W}} + \frac{\partial J}{\partial \mu_2}\frac{\partial \mu_2}{\partial \underline{W}} + \frac{\partial J}{\partial \delta_1^2}\frac{\partial \delta_1^2}{\partial \underline{W}} + \frac{\partial J}{\partial \delta_2^2}\frac{\partial \delta_2^2}{\partial \underline{W}} = 0$$

Therefore, we are now required to calculate each term separetly to find the best $\underline{W}$.

$$\begin{aligned}
\mu_i &= E\{Y|w_i\} \\
&= E\{\underline{W}^T\underline{X}|w_i\} \\
&= \underline{W}^T E\{\underline{X}|w_i\} \\
&= \underline{W}^T\underline{\mu}_i \\
&\rightarrow \boxed{\frac{\partial \mu_i}{\partial \underline{W}} = \underline{\mu}_i}
\end{aligned}$$

$$\begin{aligned}
\delta_i &= Var\{Y|w_i\} \\
&= E\{(Y-\mu_i)^2|w_i\} \\
&= E\{(Y-\mu_i)(Y-\mu_i)^T|w_i\} \\
&= E\{(\underline{W}^T\underline{X} - \underline{W}^T\underline{\mu}_i)(\underline{W}^T\underline{X} - \underline{W}^T\underline{\mu}_i)^T|w_i\} \\
&= E\{\underline{W}^T(\underline{X}-\underline{\mu}_i)(\underline{X}-\underline{\mu}_i)^T\underline{W}|w_i\} \\
&= \underline{W}^T E\{(\underline{X}-\underline{\mu}_i)(\underline{X}-\underline{\mu}_i)^T|w_i\}\underline{W} \\
&= \underline{W}^T\mathbf{\Sigma}_i\underline{W} \\
&\rightarrow \boxed{\frac{\partial \delta_i}{\partial \underline{W}} = 2\mathbf{\Sigma}_i\underline{W}}
\end{aligned}$$

Also based on the given cost function we have:

$$\frac{\partial J}{\partial \mu_1} = 2\frac{\mu_1 - \mu_2}{\delta_1^2 + \delta_2^2}$$

$$\frac{\partial J}{\partial \mu_2} = -2\frac{\mu_1 - \mu_2}{\delta_1^2 + \delta_2^2}$$

$$\frac{\partial J}{\partial \delta_i^2} = -(\frac{\mu_1 - \mu_2}{\delta_1^2 + \delta_2^2})^2$$

Therefore, in total we get:

$$2\frac{\mu_1 - \mu_2}{\delta_1^2 + \delta_2^2}\ \underline{\mu}_1 - 2\frac{\mu_1 - \mu_2}{\delta_1^2 + \delta_2^2}\ \underline{\mu}_2 - (\frac{\mu_1 - \mu_2}{\delta_1^2 + \delta_2^2})^2\ (2\mathbf{\Sigma}_1\underline{W}) - (\frac{\mu_1 - \mu_2}{\delta_1^2 + \delta_2^2})^2\ (2\mathbf{\Sigma}_2\underline{W}) = 0$$

$$\rightarrow 2\frac{\mu_1 - \mu_2}{\delta_1^2 + \delta_2^2}(\underline{\mu}_1 - \underline{\mu}_2) = 2(\frac{\mu_1 - \mu_2}{\delta_1^2 + \delta_2^2})^2\ (\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)\underline{W}$$

$$\rightarrow \underline{W}^* = \frac{(\underline{\mu}_1 - \underline{\mu}_2)(\delta_1^2 + \delta_2^2)}{(\mu_1 - \mu_2)(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)}$$

Now, we take a look back at the given cost function, as can be seen, if instead of $\underline{W}$, $\underline{W}^{'} = k\underline{W}$ is used where $k$ is a constant, we get:

$$\begin{aligned}
\mu_i &= E\{Y|w_i^{'}\} \\
&= E\{\underline{W'}^T\underline{X}|w_i'\} \\
&= k\underline{W}^T E\{\underline{X}|kw_i\} \\
&= k^2\underline{W}^T\underline{\mu}_i
\end{aligned}$$

$$\begin{aligned}
\delta_i &= Var\{Y|w_i'\} \\
&= E\{(Y-\mu_i)^2|w_i'\} \\
&= E\{(Y-\mu_i)(Y-\mu_i)^T|w_i\} \\
&= E\{(\underline{W'}^T\underline{X} - \underline{W'}^T\underline{\mu}_i)(\underline{W'}^T\underline{X} - \underline{W'}^T\underline{\mu}_i)^T|w_i'\} \\
&= E\{\underline{W'}^T(\underline{X}-\underline{\mu}_i)(\underline{X}-\underline{\mu}_i)^T\underline{W'}|w_i'\} \\
&= k\underline{W}^T E\{(\underline{X}-\underline{\mu}_i)(\underline{X}-\underline{\mu}_i)^T|w_i'\}k\underline{W} \\
&= k^2\underline{W}^T\mathbf{\Sigma}_i\underline{W}
\end{aligned}$$

Therefore, the cost function with these values will be:

$$J(\underline{W'}) = \frac{(k^2\underline{W}^T\underline{\mu}_1 - k^2\underline{W}^T\underline{\mu}_2)^2}{(k^2\underline{W}^T\mathbf{\Sigma}_1\underline{W})^2 + (k^2\underline{W}^T\mathbf{\Sigma}_2\underline{W})^2}$$

$$= \frac{(\underline{W}^T\underline{\mu}_1 - \underline{W}^T\underline{\mu}_2)^2}{(\underline{W}^T\mathbf{\Sigma}_1\underline{W})^2 + (\underline{W}^T\mathbf{\Sigma}_2\underline{W})^2}$$

$$= \frac{(\mu_1 - \mu_2)^2}{\delta_1^2 + \delta_2^2}$$

$$= J(\underline{W})$$

The above result shows that if the a weight is scaled with a constant number as $k$, the cost function does not change. As a result, the calculated optimum weight which has $\frac{\delta_1^2+\delta_2^2}{\mu_1-\mu_2}$ as the constant part, can be simplified into:

$$\boxed{\underline{W}^* = \frac{\underline{\mu}_1 - \underline{\mu}_2}{\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2} = (\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)^{-1}(\underline{\mu}_1 - \underline{\mu}_2)}$$

# Chapter 2

# Question 2

## 2.1 Part A

: Based on the definition of $\mathbf{S}_B$ as below:

$$\mathbf{S}_B = \sum_{k=1}^{C} N_k(\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T$$

where

$$\mathbf{m}_k = \frac{1}{N_k} \sum_{n C_k} \mathbf{x}_n$$

$$\mathbf{m} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n = \frac{1}{N} \sum_{k=1}^{C} N_k \mathbf{m}_k$$

$$N = \sum_{k} N_k$$

It is clear that $\mathbf{S}_B$ is composed of the sum of $C$ matrices, each of which is an outer product of two vectors and therefore has rank 1. In addition, only $(C-1)$ of these matrices are independent as a result of the above equations. Thus, $\mathbf{S}_B$ has rank at most equal to $(C-1)$. With this fact in mind, the given formulation for $\mathbf{S}_B$ while it is projected in the new space, results in the conclusion for $\mathbf{S}_B$'s rank to be at most equal to $(C-1)$.

In order to achieve $(C-1)$ as the rank of this matrix, $\mu_k$s need to be linearly independet from one another. This condition occurs when dimensions of $x$ are independent. In addition, its dimension is required to be at least $C$.

## 2.2 Part B

On the grounds of knowing the fact number of non-zero eigenvalues of a matrix is the same as rank of it, we need to find the rank of the given matrix.

Based on the formulation for $\mathbf{S}_W$, it is clear that it's rank is $d$. With keeping the conclution regarding the rank of $\mathbf{S}_B$ in mind, rank of $\mathbf{S}_W^{-1}\mathbf{S}_B$ will at most be, $min\{rank(\mathbf{S}_W^{-1}), rank(\mathbf{S}_B)\}$, which itself is at most $(C-1)$. Therefore, maximum number of non-zero eigenvalues of $\mathbf{S}_W^{-1}\mathbf{S}_B$ is $(C-1)$.

## 2.3 Part C

Based on the definition of $\mathbf{S}_T$:

$$\mathbf{S}_T = \sum_x (x-m)(x-m)^T = \sum_{i=1}^{C} \sum_{x \in D_i} (x - m - m_i + m_i)(x - m - m_i + m_i)^T$$
$$= \sum_{i=1}^{C} \sum_{x \in D_i} (x - m_i)(x - m_i)^T + \sum_{i=1}^{C} \sum_{x \in D_i} (m - m_i)(m - m_i)^T = \mathbf{S}_W + \mathbf{S}_B$$

# Chapter 3

# Question 4

In order to calculate the expected value of the Parzen window, we need to do the below procedure:

$$\begin{aligned}\overline{p}_n(x) &= E\{\frac{1}{nh_n}\sum_{i=1}^{n}\varphi(\frac{x-x_i}{h_n})\}\\ &= \int \frac{1}{h_n}\varphi(\frac{x-\tau}{h_n})p(\tau)d\tau\\ &= \frac{1}{h_n}\int_{\tau\leq x} exp(-\frac{x-\tau}{h_n})p(\tau)d\tau\end{aligned}$$

Now, based on what $x$ is the above integral's solution differs:

- $\mathbf{x} < 0$ :

$$\rightarrow \tau < 0 \rightarrow p(\tau) = 0 \rightarrow \overline{p}_n(x) = 0$$

- $0 \leq \mathbf{x} < a$:

$$\begin{aligned}\rightarrow 0 \leq \tau < x \rightarrow p(\tau) &= \frac{1}{a}\\ \rightarrow \overline{p}_n(x) &= \frac{1}{a}exp(\frac{-x}{h_n})\int_0^x \frac{1}{h_n}exp(\frac{\tau}{h_n})d\tau\\ &= \frac{1}{a}exp(\frac{-x}{h_n})(exp(\frac{x}{h_n})-1)\\ &= \frac{1}{a}(1-exp(\frac{-x}{h_n}))\end{aligned}$$

- $\mathbf{x} > a$:

$$\begin{aligned}\rightarrow 0 \leq \tau < a \rightarrow p(\tau) &= \frac{1}{a}\\ \rightarrow \overline{p}_n(x) &= \frac{1}{a}exp(\frac{-x}{h_n})\int_0^a \frac{1}{h_n}exp(\frac{\tau}{h_n})d\tau\\ &= \frac{1}{a}exp(\frac{-x}{h_n})(exp(\frac{a}{h_n})-1)\end{aligned}$$

# Chapter 4

# Question 5

In this problem, we are asked to draw a decision tree to find out under the given conditions, whether their costumers wait until a table becomes available or not. To do so, we are required to calculate information gain. Based on the formulation of information gain as below:

$$\text{IG(A)} = \text{I(Before Choosing A) - Average[I(After Choosing A)]} = I(A) - \sum_k \frac{|A_k|}{|A|} I(A_k)$$

where $I(.)$ represents the entropy of an action. Therefore, in order to calculate information gain, we are required to evaluate entropy. Entropy can be calculated as below:

$$I(S) = -\sum_{i-1}^{n} p(s_i) log_2 p(s_i)$$

Entropy of the tree:

$$I(\text{tree}) = -\frac{6}{12} log_2(\frac{6}{12}) - \frac{6}{12} log_2(\frac{6}{12}) = 1$$

The entropy for three features are calcualted as below:

- **Pat**:

$$I(\text{Full}) = -\frac{2}{6} log_2(\frac{2}{6}) - \frac{4}{6} log_2(\frac{4}{6}) = 0.92$$
$$I(\text{Some}) = -\frac{4}{4} log_2(\frac{4}{4}) - \frac{0}{4} log_2(\frac{0}{4}) = 0$$
$$I(\text{None}) = -\frac{2}{2} log_2(\frac{2}{2}) - \frac{0}{2} log_2(\frac{0}{2}) = 0$$
$$\rightarrow IG(\text{Pat}) = 1 - \frac{4}{12} * 0 - \frac{6}{12} * 0.92 - \frac{2}{12} * 0 = 0.54$$

- **Bar**:

$$I(\text{Yes}) = -\frac{3}{6} log_2(\frac{3}{6}) - \frac{3}{6} log_2(\frac{3}{6}) = 1$$
$$I(\text{No}) = -\frac{3}{6} log_2(\frac{3}{6}) - \frac{3}{6} log_2(\frac{3}{6}) = 1$$
$$\rightarrow IG(\text{Bar}) = 1 - \frac{6}{12} * 1 - \frac{6}{12} * 1 = 0$$

- **Price**:

$$I(\$) = -\frac{3}{7}log_2(\frac{3}{7}) - \frac{4}{7}log_2(\frac{4}{7}) = 0.98$$
$$I(\$\$) = -\frac{2}{2}log_2(\frac{2}{2}) - \frac{0}{2}log_2(\frac{0}{2}) = 0$$
$$I(\$\$\$) = -\frac{1}{3}log_2(\frac{1}{3}) - \frac{2}{3}log_2(\frac{2}{3}) = 0.92$$
$$\rightarrow IG(\text{Price}) = 1 - \frac{7}{12} * 0.98 - \frac{2}{12} * 0 - \frac{3}{12} * 0.92 = 0.2$$

And we do the same procedure for all the other attributes. The result is as below:

- $IG(\text{Alt}) = 0$
- $IG(\text{Fri}) = 0.02$
- $IG(\text{Hun}) = 0.2$
- $IG(\text{Rain}) = 0.02$
- $IG(\text{Res}) = 0.02$
- $IG(\text{Type}) = 0$
- $IG(\text{Est}) = 0.21$

Based on these calculations, the best feature to be as the root of the tree will be *Pat*. The same procedure goes on, and the resulting tree is depicted as below:

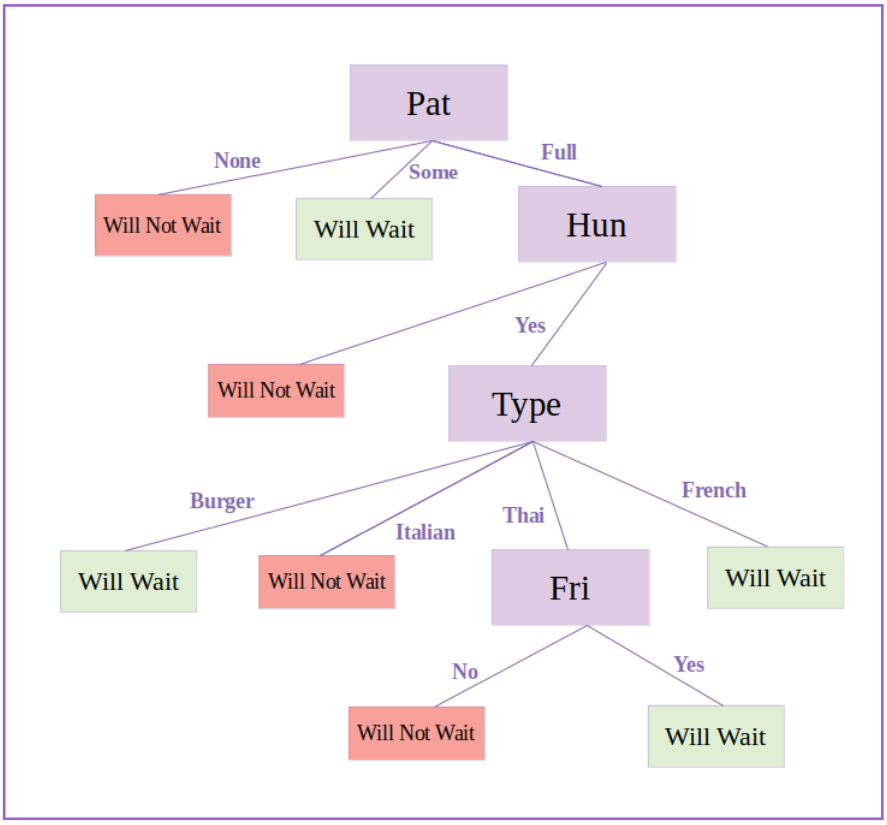



FIGURE 4.1: Restaurant Data Decision Tree

Note that, in the attached notebook, the code for the tree implementation is included both from scatch and by sklearn.

# Chapter 5

# Question 8

First of all, the images are read. In order to make sure they are read accurately, three random samples of them are plotted.

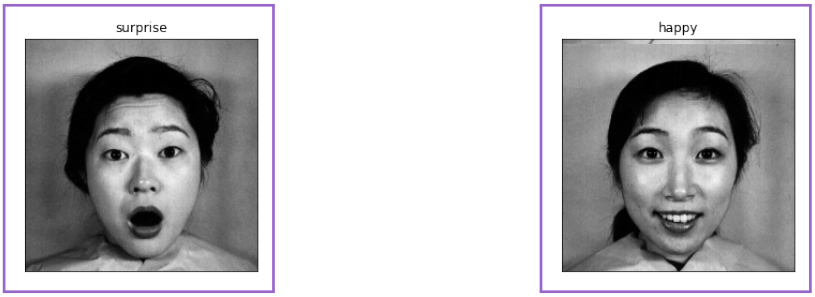



FIGURE 5.1: Images in Given Dataset

## 5.1 Part A

In this part, the eigen values of the covariance matrix is found for various values of principle components and their values are plotted. The resulting figure is depicted as below:

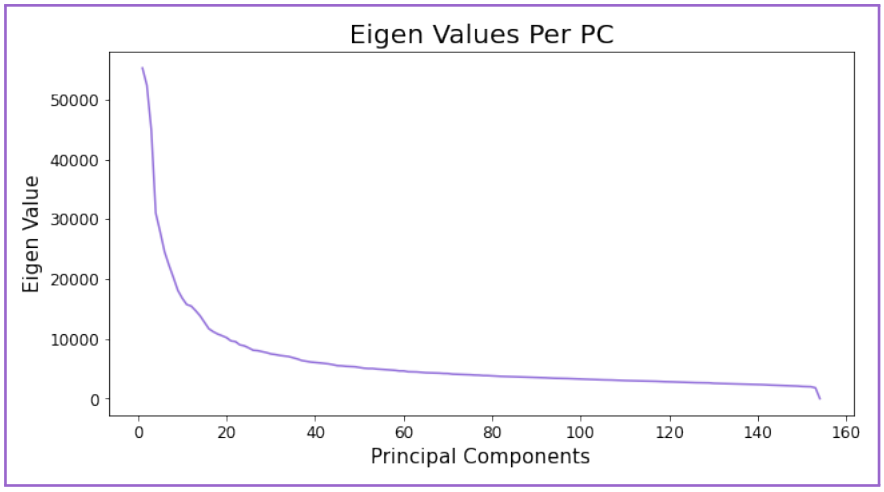



FIGURE 5.2: Plot of Eigen Values of Covariance Matrix of Training Dataset with Respect to Various PCs.

As can be seen in the above plot, the value of eigen values decreases with the increament in PCs. In order to select the optimum value of PCs, we need to eliminate the ones which are very small with respect to the others or check the cumulative sum.

In the second method, the place where %90 of data lies, is selected. Therefore, we the cumulative sum is plot is:

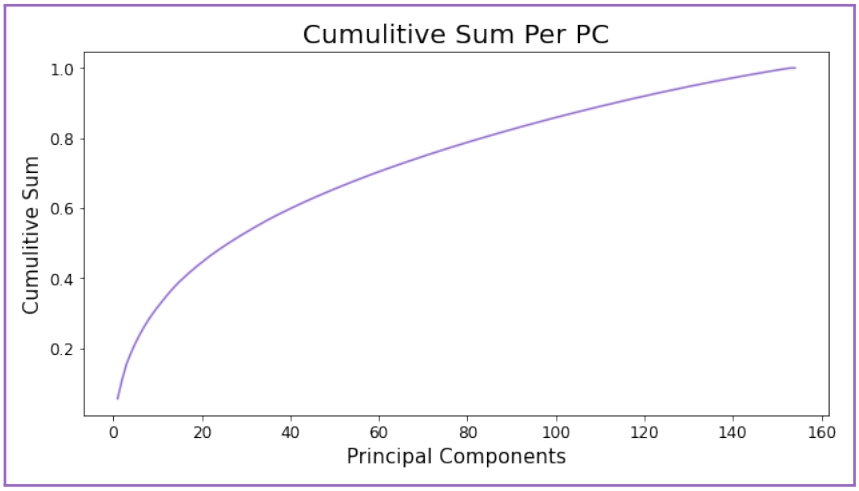



FIGURE 5.3: Cumulative Sum based on Various Values of PCs.

```
Explained variation per principal component based on eigen values for the first 112 PCs is %89.654
```

Therefore, the number of principle components is 112.

## 5.2 Part B

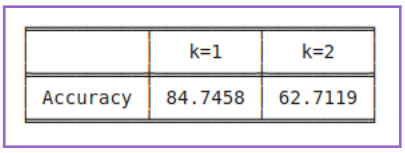
| | | | | |
|---|---|---|---|---|
| First Four Eigen Values | 55237.6 | 52353.3 | 44910.8 | 31017.6 |
| Last Four Eigen Values | 2.664e-11 | 1808.82 | 1987.66 | 2004.61 |

FIGURE 5.4: Desired Eigen Values

As can be seen, the last eigen values are smaller than the first four ones. Specially, the last one is almost zero. Therefore, it shows that the last one can be eliminated for sure. However, the other three can be eliminated as well, since they are about 30 times smaller that the first four eigen values.

## 5.3 Part C

By employing the optimum value found in the first part of this problem, the dimension of training dataset is reduced. Afterward, the dimension of test dataset is increased to have the same one as training dataset. Although this increament is done and the labels of the test dataset is predicted, in KNN, test and training dataset is not needed. The result is shown as below:



| | k=1 | k=2 |
|---|---|---|
| Accuracy | 84.7458 | 62.7119 |

FIGURE 5.5: Table of Accuracy of Model with Various Values of $k$ on the Reduced Dataset.

# Chapter 6

# Question 12

## 6.1 Part A, B

Generally, in Parzen Windows, we seek to shrink the regions as some function of data samples represented as $n$, such as $V_n = \frac{1}{\sqrt{n}}$. Therefore, in these windows, for a particular value of $n$, $V_n$ is calculated and fixed.

Therefore, the probability (distribution) is estimated as below:

$$P_n(\underline{X}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{V_n}\varphi(\frac{\underline{X}-\underline{X}_i}{h_n})$$

Afterward, the resulting probability can be used in the Bayesian classifier as the posterior probability. (In both window methods, the prior probability is calculated based on the given dataset.)

In order to implement the asked windows, a Parzen window class is created, in which based on the given method as its input, the Gaussian or rectangular one is selected.

### 6.1.1 Rectangular Window

In rectangular window, the window is created based on the below definition:

$$\varphi(\underline{U}) = \begin{cases} 1 & |u_j| \leq 0.5 \\ 0 & O.W. \end{cases} \qquad j = 1,\ 2,\ ...,\ d$$

Based on the above definition, the window is created and implemented. The analysis of this window with 3 different values of $h$ can be seen in the seen below. In the first plot the accuracy for various vlaues of $h$ is shown and afterward, the confusion matrix for each is plotted.

| | h=1 | h=5 | h=10 |
|---|---|---|---|
| Accuracy | 9.76 | 79.8 | 22.52 |

FIGURE 6.1: Table of Accuracy of the Model Based on Various Values of $h$

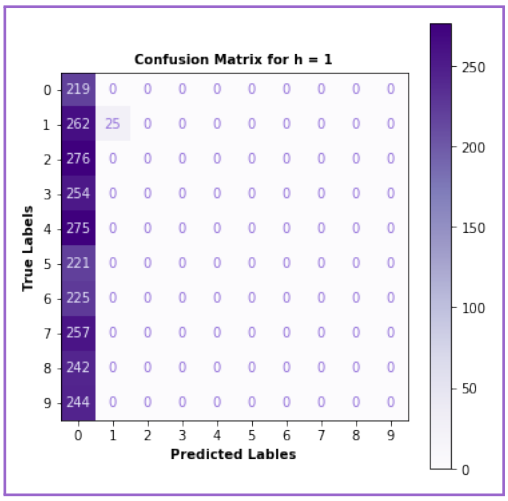



FIGURE 6.2: Confusion Matrix for $h = 1$ for Rectangular Parzen Window

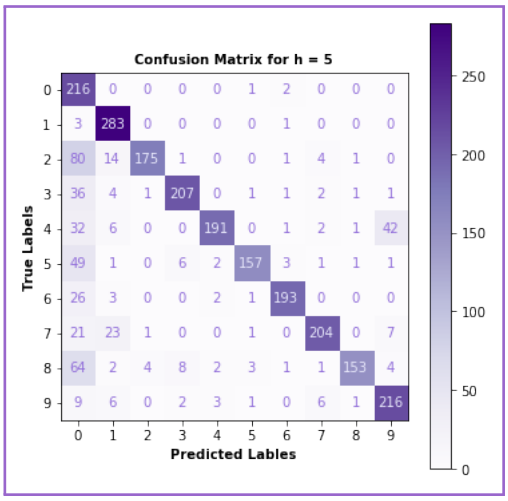



FIGURE 6.3: Confusion Matrix for $h = 5$ for Rectangular Parzen Window

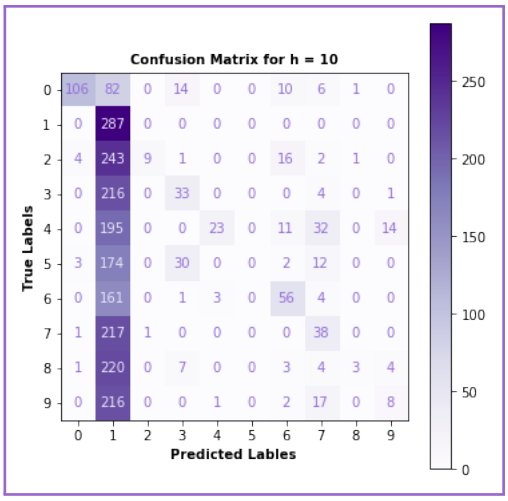



FIGURE 6.4: Confusion Matrix for $h = 10$ for Rectangular Parzen Window

As the above results suggest, the best value for $h$ is 5. This is due to the fact that too big windows include too many data. On the other hand, when windows dimension is too small, not enough data are selected to be processed. Hence, the low accuracy of model.

### 6.1.2 Gaussian Window

In this method, the window is created based on the definition of Gaussian PDF distribution. Since the variations in this method is smoother than the previous one, it is assumed that the results are better as well.

The analysis of this window with 3 different values of $h$ can be seen in the seen below. In the first plot the accuracy for various vlaues of $h$ is shown and afterward, the confusion matrix for each is plotted.

| | h=0.1 | h=5 | h=10 |
|---|---|---|---|
| Accuracy | 91.76 | 28.8 | 12.2 |

FIGURE 6.5: Table of Accuracy of the Model Based on Various Values of $h$

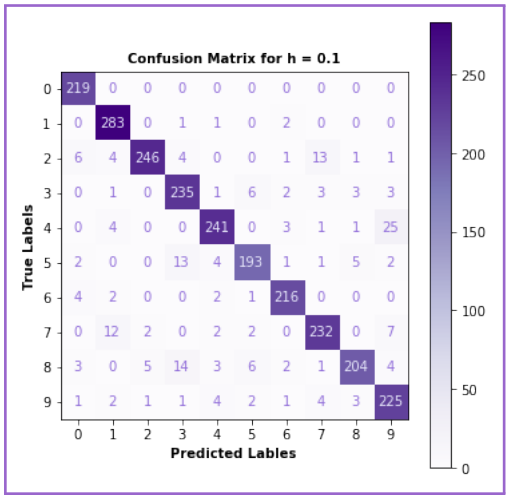



FIGURE 6.6: Confusion Matrix for $h = 0.1$ for Rectangular Gaussian Window

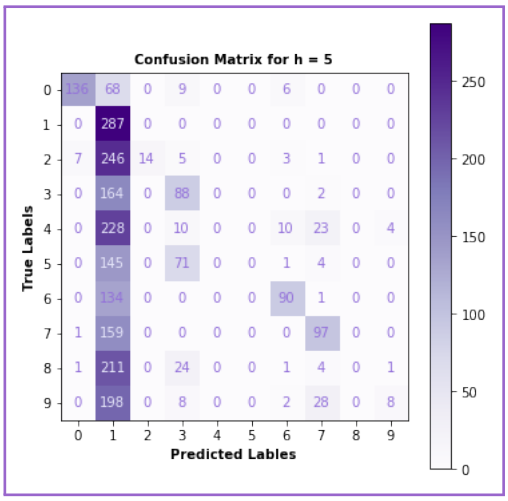



FIGURE 6.7: Confusion Matrix for $h = 5$ for Rectangular Gaussian Window

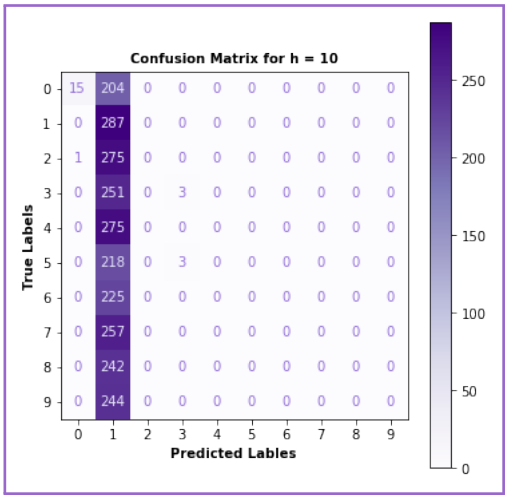



FIGURE 6.8: Confusion Matrix for $h = 10$ for Rectangular Gaussian Window

As was assumed, since the variations in the Gaussian window is much smoother, with smaller windows we can get a better accuracy. However, in both cases, when the window's size is not suitable for the given dataset, it can be seen that decision is almost based on the prior probabilities. Therefore, the classifier's performance is poor.

# Chapter 7

# Question 13

## 7.1 Part A

In this part the $k^{\text{th}}$ Nearest Neighbor PDF estimator method is implemented. In order to do so, the Euclidean distance of each point to its $k$ neighbors are calcualted. The label with the maximum repeatation is selected as the predicted label.
Note that based on the nature of KNN, besides $k$ which is a hyperparameter and can be found by methods such as cross-validation, there is no learning. Therefore, train-test dataset is not needed. However, since for this problem, the number of data is more than enough and concating train and test dataset, results in a colossal dataset, both of them, separatly, are used.
The results of the simulation for various values of $k$ is depicted as below:

| | k=1 | k=5 | k=15 | k=25 |
|---|---|---|---|---|
| Accuracy | 91.84 | 90.12 | 86.92 | 84.72 |

FIGURE 7.1: Table of Accuracy of the Model Based on Various Values of $k$

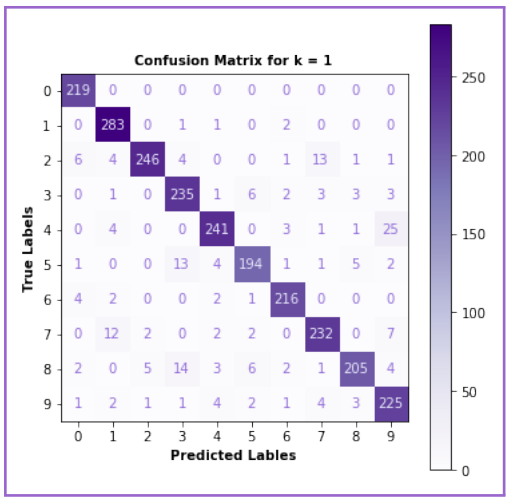



FIGURE 7.2: Confusion Matrix for Model Based on the Nearest Neighbor

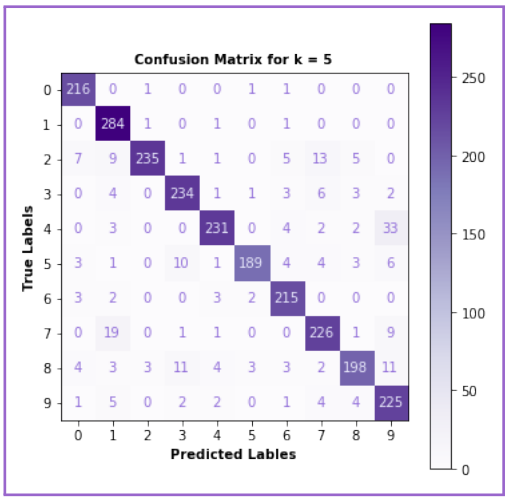



FIGURE 7.3: Confusion Matrix for Model Based on Five Nearest Neighbor

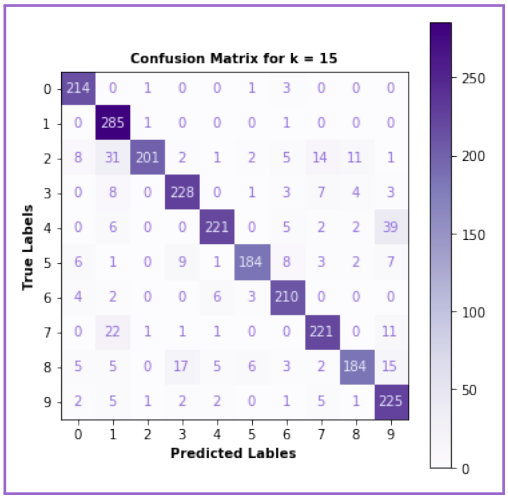



FIGURE 7.4: Confusion Matrix for Model Based on Fifteen Nearest Neighbor

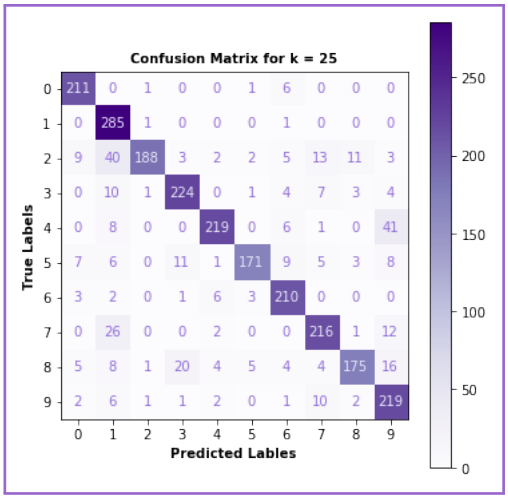



FIGURE 7.5: Confusion Matrix for Model Based on Twenty-Five Nearest Neighbor

Based on the above results, it is crystal clear that if the number of neighbors near a sample data is increased, the accuracy of the model is decreased.

## 7.2 Part B

Both of these two methods, in their best cases, approximately, behave as if they are a Optimal Bayes Classifier. As the results of the last problem suggests, the best condition of the Parzen Gaussian Window occured while $h$ was 0.1. The best accuracy for KNN has taken place when only one neighor is considered. In addition, for the case where $k = 1$, and there are enough samples, it is known that:

$$E^* \leq E_{1NN} \leq 2E^*$$

where $E^*$ is the Optimal Bayes' error. Therefore, KNN and Parzen's accuracy, in their best cases, (as for this problem it is mentioned what they are) can be same as Bayes' accuracy.