**Project 1**

Merryn Marderosian

CSE 410 – Biometrics and Pattern Recognition

October 9, 2019

**Note: It can be assumed that all calculations, graphs, and other work have been generated with the Python program included in the appendix of this report.**

**Question 1**

Based on the data given for the fingerprint matcher and the hand matcher, the following counts have been calculated for each matcher’s genuine and impostor scores as shown in the table below:

*Table 1: Total genuine and impostor scores count for each matcher (fingerprint, hand)*

|  |  |  |
| --- | --- | --- |
|  | Fingerprint Matcher | Hand Matcher |
| Genuine Scores Count | 450 | 450 |
| Impostor Scores Count | 450 | 450 |

**Question 2**

The maximum and minimum scores generated by each matcher are as follows.

*Table 2: Maximum and minimum scores for both the fingerprint and hand matcher*

|  |  |  |
| --- | --- | --- |
|  | Fingerprint Matcher | Hand Matcher |
| Minimum Score | 0.0 | 0.0 |
| Maximum Score | 996.0 | 626.0 |

**Question 3**

The mean and variance for each matcher’s set of genuine and impostor scores can be found below.

*Table 3: Calculated mean and variance for both the genuine and impostor scores relating to the fingerprint matcher.*

|  |  |  |
| --- | --- | --- |
|  | Genuine Scores | Impostor Scores |
| Mean | 306.582 | 7.971 |
| Variance | 40825.043 | 90.806 |

*Table 4: Calculated mean and variance for both the genuine and impostor scores relating to the hand matcher.*

|  |  |  |
| --- | --- | --- |
|  | Genuine Scores | Impostor Scores |
| Mean | 50.644 | 144.436 |
| Variance | 1516.274 | 6925.659 |

**Question 4**

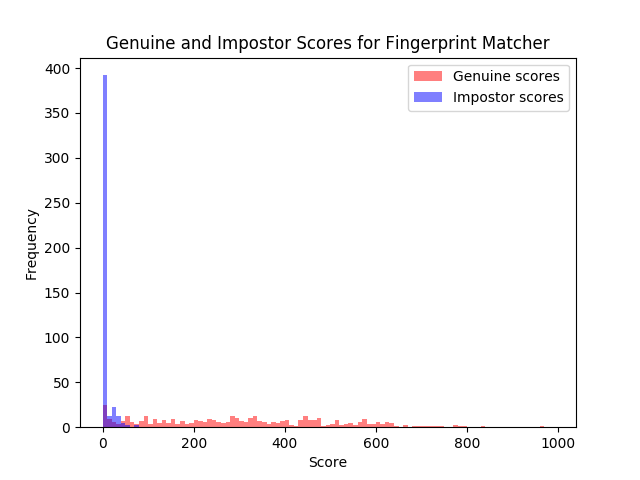
The d’-values for each matcher can be seen below. A higher d’-value is generally associated with a more efficient matcher.

|  |  |  |
| --- | --- | --- |
|  | Fingerprint Matcher | Hand Matcher |
| d’-value | 2.088 | 1.444 |

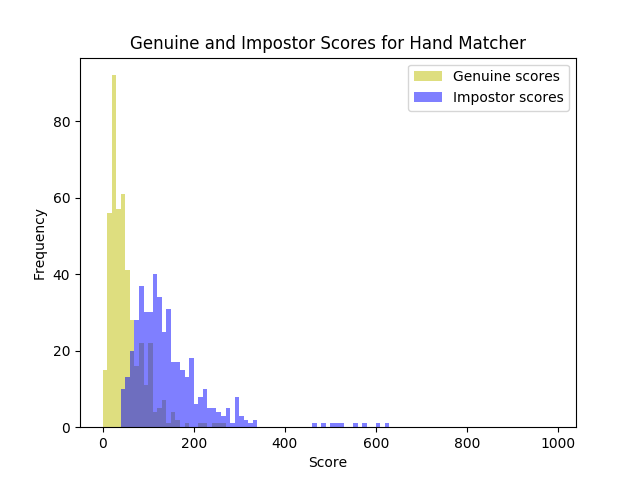
*Table 5: Calculated d’-value for the fingerprint matcher and the hand matcher*

**Question 5**

Below, the histograms for each matcher can be found, with genuine and impostor scores plotted in the same graph.



*Figure 1: Histogram recording the frequency of each genuine and impostor score for the fingerprint matcher. Impostor scores have been plotted in blue, and genuine scores have been plotted in red.*

**

*Figure 2: Histogram recording the frequency of each genuine and impostor score for the hand matcher. Impostor scores have been graphed in blue, and genuine scores have been plotted in yellow.*

**Question 6**

Using a program that allows a threshold (η) to be input manually, the following False Non-Match Rates (FNMR) and False Match Rates (FMR) were calculated for each matcher using an η value of 45.

|  |  |  |
| --- | --- | --- |
| **η = 45** | Fingerprint Matcher | Hand Matcher |
| FNMR | 1.3% | 0.4% |
| FMR | 10.4% | 43.6% |

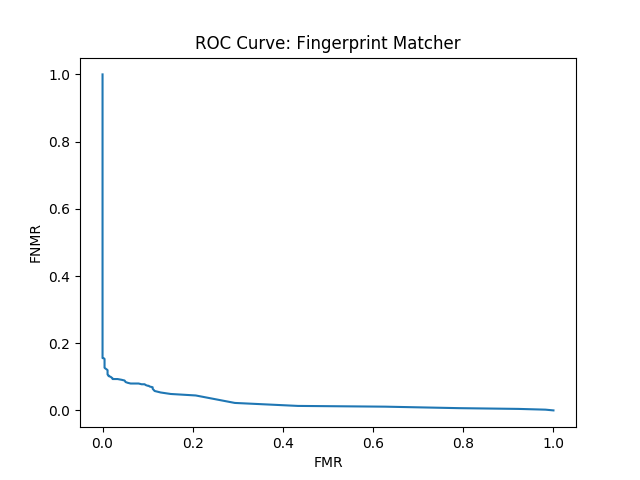
*Table 6: False Match Rate (FMR) and False Non-Match Rate calculated for both the fingerprint and hand matchers using and η value of 45.*

**Question 7**

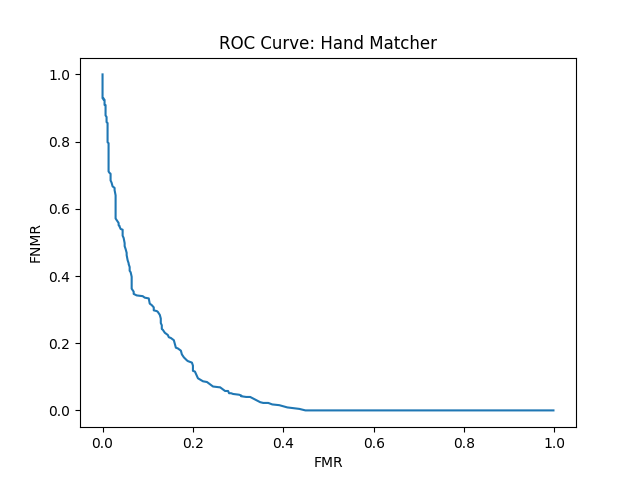
The ROC (Receiver Operating Characteristic) curve for both matchers can be seen below, along with the Are Under the Curve (AUC) and the Equal Error Rate (EER) for each. The EER is rounded to one decimal point due to unforeseen errors in calculating this value for both matchers’ ROC curves.

*Table 7: AUC and EER for each matcher, based on the ROC curves found in Figure 3 and Figure 4 below.*

|  |  |  |
| --- | --- | --- |
|  | Fingerprint Matcher | Hand Matcher |
| AUC | 0.025 | 0.085 |
| EER | ~0.1 | ~0.2 |



*Figure 3: ROC curve for the fingerprint matcher. FMR is plotted on the x-axis, while FNMR is plotted on the y-axis. The FMR and FNMR values for this curve were calculated with η values ranging from 0 to 1000*

*Figure 4: ROC curve for the hand matcher. FMR is plotted on the x-axis, while FNMR is plotted on the y-axis. The FMR and FNMR values for this curve were calculated with η values ranging from 0 to 1000.*

**Question 8**

For each matcher, FNMR values were determined based on a given FMR value. This was calculated based on the ROC curve in Figure 3 and Figure 4.

*Table 8: FNMR values for each matcher, calculated at the FMR values of 10%, 5%, and 1%.*

|  |  |  |
| --- | --- | --- |
|  | Fingerprint Matcher | Hand Matcher |
| FNMR @ FMR = 10% | 7.56% | 33.33% |
| FNMR @ FMR = 5% | 8.89% | 51.33% |
| FNMR @ FMR = 1% | 12.22% | 90.89% |

**Question 9**

It is clear that, based on the calculated data in the tables and figures pictured above, that the best-performing matcher is the fingerprint matcher. There are a number of reasons why this is the case.

Referencing Table 1 and Table 2, although both matchers have the same number of scores recorded, the fingerprint matcher has a larger range in comparison to the hand matcher (0 to 996 versus 0 to 626). This means that there is a better data spread and it is clear that the matcher can confidently identify a match, whereas there are no “confident” matches with the hand matcher (highest score is 626, which out of 1000 is relatively low).

Another note can be made when looking at the d’-values in Table 5. It is known that a higher d’ value is associated with a better matcher, and the fingerprint matcher’s d’-value is nearly 50% higher than that of the hand matcher.

Looking at the histograms in Figures 1 and 2, there is more overlap between the hand matcher’s genuine and impostor scores. This could mean that it has less distinction between a match and a non-match, in comparison to that of the fingerprint matcher, which has a much clearer separation of scores.

In Table 6, the FMR and FNMR for each matcher at an η value of 45 is displayed. While the hand matcher’s FNMR rate is lowest, its FMR rate is significantly higher than that of the fingerprint matcher. Having a lower FMR rate is the most important in this case, because a false non-match is less detrimental to a user’s privacy than a false match would be.

The area under the ROC curve as well as the EER are other determining factors to look at when comparing the reliability of two matchers. A smaller AUC means that the ROC curve is closer to the vertex and indicates a more efficient matcher, and a lower EER indicates the same thing. Looking at Table 7, the fingerprint matcher has both a lower AUC and EER than the handprint matcher.

Finally, looking at Table 8, we see the corresponding FNMR values based on FMR values of 10%, 5%, and 1%. It is better to see the FNMR values being relatively close to the chosen FMR values since they are low. It is ideal to have both low FMR and FNMR values at the same time. This is seen in the case of the fingerprint matcher, whereas the hand matcher’s corresponding FNMR values are significantly higher than the given FMR values.

Based on the above observations, it can safely be determined that the fingerprint matcher performed better than the hand matcher in a vast number of ways.

Appendix

The following program was used in all calculations seen in this report, as well as generating graphical representations of data.

########################################################################  
# #  
# Project 1 - CSE 402: Biometrics and Pattern Recognition #  
# #  
# Calculates the following based on the datasets from two matchers: #  
# Fingerprint and Hand. #  
# #  
# - Genuine and impostor score counts #  
# - Maximum and minimum scores for each matcher #  
# - Mean and variance for each matchers' set of genuine and #  
# impostor scores #  
# - Histogram displaying genuine and impostor score distribution on #  
# the same graph #  
# - FNMR and FMR given an input eta value for each matcher #  
# - ROC curve, as well as AUC and EER for each matcher #  
# - Using values plotted in the ROC curve, determines the FNMR #  
# at a given FMR value (10%, 5%, 1%) #  
# #  
########################################################################  
  
import numpy as np  
import math  
import matplotlib.pyplot as plt  
import matplotlib.pyplot as ROC\_plt  
import sklearn.metrics  
  
  
def newline\_remove(a\_file):  
 *"""Opens a data file, removes new lines, converts each point to a float and returns as a list of values"""* a\_list = open(a\_file).readlines()  
 return [float(item.rstrip('\n')) for item in a\_list]  
  
  
def find\_max(a\_list):  
 *"""Finds the maximum value in a list of numbers"""* a\_list.sort()  
 return a\_list[-1]  
  
  
def find\_min(a\_list):  
 *"""Finds the minimum value in a list of numbers"""* a\_list.sort()  
 return a\_list[0]  
  
  
def find\_mean(a\_list):  
 *"""Finds the mean value in a list of numbers"""* length = len(a\_list)  
 total = sum(a\_list)  
  
 return total/length  
  
  
def d\_prime(gen, imp):  
 *"""Given a list of genuine and impostor scores, calculates and returns d' value of the matcher"""* gen\_mean = find\_mean(gen)  
 imp\_mean = find\_mean(imp)  
  
 numerator = math.fabs(gen\_mean - imp\_mean)  
  
 sum\_vars = np.var(gen) + np.var(imp)  
 denominator = math.sqrt(sum\_vars/2)  
  
 return numerator/denominator  
  
  
def FMR(eta, a\_list):  
 *"""Calculates the False Match Rate given an eta value and list of impostor scores"""* false\_matches = 0  
 total\_scores = len(a\_list)  
  
 for item in a\_list:  
 if item > eta:  
 false\_matches += 1  
  
 return false\_matches/total\_scores  
  
  
def FNMR(eta, a\_list):  
 *"""Calculates the False Non-Match Rate given an eta value and a list of genuine scores"""* false\_non\_matches = 0  
 total\_scores = len(a\_list)  
  
 for item in a\_list:  
 if item < eta:  
 false\_non\_matches += 1  
  
 return false\_non\_matches / total\_scores  
  
  
def plot\_ROC(gen\_data, imp\_data, title):  
 *"""Plots the ROC curve of a matcher"""* FNMR\_list = []  
 FMR\_list = []  
 ROC\_title = "ROC Curve: " + title  
  
 for eta\_value in range(1001):  
 FNMR\_list.append(FNMR(eta\_value, gen\_data))  
 FMR\_list.append(FMR(eta\_value, imp\_data))  
  
 # giving a title to my graph  
 ROC\_plt.title(ROC\_title)  
  
 # naming the x axis  
 ROC\_plt.xlabel("FMR")  
 # naming the y axis  
 ROC\_plt.ylabel("FNMR")  
  
 ROC\_plt.plot(FMR\_list, FNMR\_list)  
  
 ROC\_plt.show()  
  
 print()  
  
 AUC = sklearn.metrics.auc(FMR\_list, FNMR\_list)  
 print("Area under the curve for the %s: %.3f" % (title, AUC))  
  
 EER\_val = EER(FNMR\_list, FMR\_list)  
 print("Equal Error Rate (EER) for the %s: %.1f" % (title, EER\_val))  
  
  
def EER(FNMR\_vals, FMR\_vals):  
 *"""Calculates the Equal Error Rate of a matcher's ROC curve"""* EER\_val = None  
  
 for i in range(1001):  
 FMR = round(FMR\_vals[i], 1)  
 FNMR = round(FNMR\_vals[i], 1)  
  
 if FMR == FNMR:  
 EER\_val = FMR  
  
 return EER\_val  
  
  
def FNMR\_at\_FMR(FMR\_val, gen\_data, imp\_data):  
 *"""Calculates a matcher's FNMR at a given FMR value"""* FNMR\_list = []  
 FMR\_list = []  
  
 FNMR\_val = 0  
  
 for eta\_value in range(1001):  
 FNMR\_list.append(FNMR(eta\_value, gen\_data))  
 FMR\_list.append(FMR(eta\_value, imp\_data))  
  
 for loc, score in enumerate(FMR\_list):  
 if round(score, 2) == FMR\_val:  
 FNMR\_val = float(FNMR\_list[loc])  
  
 FMR\_val \*= 100  
 FNMR\_val \*= 100  
  
 print("The FNMR value when FMR is %.2f%% is: %.2f%%." % (FMR\_val, FNMR\_val))  
  
  
# All score sets for each matcher  
finger\_gen\_scores = newline\_remove("proj01\_q1\_match\_scores/finger\_genuine.score")  
finger\_imp\_scores = newline\_remove("proj01\_q1\_match\_scores/finger\_impostor.score")  
hand\_gen\_scores = newline\_remove("proj01\_q1\_match\_scores/hand\_genuine.score")  
hand\_imp\_scores = newline\_remove("proj01\_q1\_match\_scores/hand\_impostor.score")  
  
  
# Code relating to problems 1 through 3  
finger\_scores = finger\_imp\_scores + finger\_gen\_scores  
hand\_scores = hand\_imp\_scores + hand\_gen\_scores  
  
print("1. NUMBER OF SCORES")  
print()  
print("Fingerprint Matcher:")  
print("Number of Genuine Scores: %d" % len(finger\_gen\_scores))  
print("Number of Impostor Scores: %d" % len(finger\_imp\_scores))  
print()  
print("Hand Matcher:")  
print("Number of Genuine Scores: %d" % len(hand\_gen\_scores))  
print("Number of Impostor Scores: %d" % len(hand\_imp\_scores))  
print()  
  
print()  
print("2. MAXIMUM AND MINIMUM SCORES")  
print()  
print("Fingerprint Matcher:")  
print(" - Maximum: %.3f" % find\_max(finger\_scores))  
print(" - Minimum: %.3f" % find\_min(finger\_scores))  
print()  
print("Hand Matcher:")  
print(" - Maximum: %.3f" % find\_max(hand\_scores))  
print(" - Minimum: %.3f" % find\_min(hand\_scores))  
print()  
  
print()  
print('3. MEAN AND VARIANCE')  
print()  
print("Fingerprint Matcher")  
print("Genuine Scores:")  
print(" - Mean: %.3f" % find\_mean(finger\_gen\_scores))  
print(" - Variance: %.3f" % np.var(finger\_gen\_scores))  
print()  
print("Impostor Scores:")  
print(" - Mean: %.3f" % find\_mean(finger\_imp\_scores))  
print(" - Variance: %.3f" % np.var(finger\_imp\_scores))  
print()  
print("Hand Matcher")  
print("Genuine Scores:")  
print(" - Mean: %.3f" % find\_mean(hand\_gen\_scores))  
print(" - Variance: %.3f" % np.var(hand\_gen\_scores))  
print()  
print("Impostor Scores:")  
print(" - Mean: %.3f" % find\_mean(hand\_imp\_scores))  
print(" - Variance: %.3f" % np.var(hand\_imp\_scores))  
print()  
  
print()  
print("4. D-PRIME VALUES")  
print()  
print("Fingerprint Matcher: %.3f" % d\_prime(finger\_gen\_scores, finger\_imp\_scores))  
print("Hand Matcher: %.3f" % d\_prime(hand\_gen\_scores, hand\_imp\_scores))  
  
# Code relating to problem 5: Histogram plotting  
print()  
print("5. SEE HISTOGRAMS")  
# Set number of bins; constant among each matchers' scores  
num\_bins = [i for i in range(0, 1000, 10)]  
  
# Plotting fingerprint matcher's genuine and impostor scores on same graph  
plt.figure(0)  
plt.title("Genuine and Impostor Scores for Fingerprint Matcher")  
plt.xlabel("Score")  
plt.ylabel("Frequency")  
plt.hist(finger\_gen\_scores, bins=num\_bins, alpha=0.5, label='Genuine scores', color='r')  
plt.hist(finger\_imp\_scores, bins=num\_bins, alpha=0.5, label='Impostor scores', color='b')  
plt.legend(loc='upper right')  
plt.show()  
  
# Plotting fingerprint matcher's genuine and impostor scores on same graph  
plt.figure(1)  
plt.title("Genuine and Impostor Scores for Hand Matcher")  
plt.xlabel("Score")  
plt.ylabel("Frequency")  
plt.hist(hand\_gen\_scores, bins=num\_bins, alpha=0.5, label='Genuine scores', color="y")  
plt.hist(hand\_imp\_scores, bins=num\_bins, alpha=0.5, label='Impostor scores', color="b")  
plt.legend(loc='upper right')  
plt.show()  
  
# Code relating to problem 6  
print()  
print("6. FMR and FNMR")  
print()  
  
# Inputs for eta  
try:  
 eta\_fingerprint = int(input("Input a threshold value for the fingerprint matcher: \n"))  
 eta\_hand = int(input("Input a threshold value for the hand matcher: \n"))  
 print()  
  
# input is not a number  
except ValueError:  
 print("Please try another input (numeric value 0 - 1000)")  
 eta\_fingerprint = int(input("Input a threshold value for the fingerprint matcher: \n"))  
 eta\_hand = int(input("Input a threshold value for the hand matcher: \n"))  
 print()  
  
# Format and print data  
print("FINGERPRINT MATCHER")  
print("FMR: %.3f" % FMR(eta\_fingerprint, finger\_imp\_scores))  
print("FNMR: %.3f" % FNMR(eta\_fingerprint, finger\_gen\_scores))  
print()  
print(".............................................................")  
print()  
print("HAND MATCHER")  
print("FMR: %.3f" % FMR(eta\_hand, hand\_gen\_scores))  
print("FNMR: %.3f" % FNMR(eta\_hand, hand\_imp\_scores))  
  
# Code relating to problems 7 and 8  
print()  
print("7. ROC CURVES, AUC & EER")  
  
# ROC curve for finger matcher  
plot\_ROC(finger\_gen\_scores, finger\_imp\_scores, "Fingerprint Matcher")  
  
print()  
  
# for hand matcher, swap impostor and genuine scores, for dissimilarity  
plot\_ROC(hand\_imp\_scores, hand\_gen\_scores, "Hand Matcher")  
  
print()  
  
# FMR values we are testing at  
percent\_vals = [.10, .05, .01]  
  
print("8. FNMR VALUES AT SPECIFIC FMR VALUES")  
print("Fingerprint Matcher")  
for val in percent\_vals:  
 FNMR\_at\_FMR(val, finger\_gen\_scores, finger\_imp\_scores)  
print()  
print("Hand Matcher")  
for val in percent\_vals:  
 FNMR\_at\_FMR(val, hand\_imp\_scores, hand\_gen\_scores)