Exercises from the subject Biometrics
Exercise 0: keystroke dynamics

Eduard Bakštein, eduard.bakstein@fel.cvut.cz

October 5, 2023

Introduction

The aim of the exercise is to review in a short theoretical introduction the basic concepts of hypothesis testing and to put them in the context of biometrics. The introductory Exercise 0 is an example of statistical evaluation of a biometric system based on keystroke dynamics.

Exercise 0: Keystroke dynamics

One of the more experimental biometric indicators is the keystroke dynamics (i.e. the timing between individual keyboard hits), which is characteristic of each person. There are basically two variants of recording such data: i) Typing a known text (e.g. a password - it allows to verify that the person entering the password is not an impostor), and ii) typing of a free text, for example during common work on a PC (it allows to increase security by verifying the activity of an already authenticated user).

In this task, we use real data on strike dynamics obtained by researchers at Queen Mary University of London (`http://www.eecs.qmul.ac.uk/~ccloy/downloads_keystroke100.html`). The database contains data from 100 measured subjects, each entering the same password „try4-mbs“ ten times. This password was known in advance to all participants in the experiment and they were asked to practice entering the password. In the task, we will work with latencies, i.e. the times between keystrokes.

Tasks:

1. Download the dataset `keystrokes.mat` View the data (the vector „ids“ contains the identification of the persons to the corresponding rows in „latencies“)

2. Create a simple function `d=compareKeystrokes(a,b)` that calculates the distance between two vectors. You can start with a simple Euclidean distance. The function will be used for simple comparison of latencies between each other.

3. Create a program that uses the function you created in the previous section to compare the latencies in the database against each other (i.e. „each to each“). Store the calculated distances in two vectors according to whether the latencies from the same or different people were compared.

4. Plot normalized histograms (=add up to one) of data in both vectors in one graph.(You can use the function `hist` with two output parameters and then the function `bar` to plot them). Think about what the displayed waveforms tell us about the task and our comparison method.

5. Use the function `plotroc` to display the ROC curve for all the data together. What can we learn from it? What will be the FRR (false reject rate) at FAR=0.01? Comment on the result.

Bonus tasks:

1. Design your own method to compare latency vectors. Try to compare the ROC curves of your proposed method and the original Euclidean distances.

2. Calculate the AUC (Area Under (ROC)Curve) and compare the two comparison methods using it. Which one is more accurate?

3. Test how the AUC of the two methods would change if you used only 5 elements (shorter password) instead of the 7-element latency vectors.