Keystroke Analysis - Data Mining course project

Paulin Loïs, Badin de Montjoye Xavier, Gaziello Yannis, Menet Hugo

11/05/2017

Abstract

User authentification via biometrics measures is a vast field. Mostly divided into two parts, physiological features (such as iris or fingerprints recognition) and behavioral features (speech or handwriting analysis for instance) analysis. Surprisingly the second one, though less stable, can be used to provide solid and reliable user authentification. Keystrokes analysis falls in this category. Our project aimed at studying the scope and result in this domain, before constructing our own algorithm and samples data base.

Contents

1 Introduction

Keystrokes analysis for biometrics recognition. Why?
First, the material needed is widespread and easy to access, for instance to authentify while sending a mail or login into a computer. Then it is simple, in computation and parametrization time, moreover the results obtained in most studies present it as a quite reliable method. One more advantage is the simlicity of most algorithm and procedures developped and studied here.

2 What is the scope of keystroke analysis?

The scope of keystroke analysis both in the learning and authentication phase is quite open, and provides nice results with appropriate protocols for training and validation. The first way is fixed text. The user profile is learned with the user typing several time a same fixed text (wisely chosen in accord to the learning technique), with recognition done using that same fixed text. Studies also been made with free text, even with text typed in different language, different keyboards or with a long time between learning and testing.

3 State of the art

The general algorithm is based on several steps :

- Compute distances between two typing sessions
- Consider time sequences of keystroke as patterns

However it is not applicable for small free texts, which can be seen as a consequence of the curse of dimensionality.

4 Our work

To understand the problem from a simple point of view, and witness the scope and power of keystroke analysis, we decided to look upon it only as a problem of classification, several profiles registered and returning the closest one when given a new input (a new typed text). Which differentiate from authentication, which return yes or no given a profile in a set of profiles and an input.

4.1 First Observations

In order to have a good overview of the project, we started by testing our own algorithm.

4.1.1 Measurements

First of all, we asked ourselves the question, what can be calculated while typing. Using the script that was given to us, we had the moment a key was press and the moment it was release. With that, we decided to take into account two parameters : the fly time and the pressure length. The fly time correspond to the time between when a key is release and when the next is pressed. The pressure length correspond to the interval of time during when the key is pressed.

Thus, we had a matrix which counted the time needed to go from key i to key j and a list corresponding to the pressure length.

4.1.2 Value to consider

In fact, when we are parsing, we store every fly value and every pressure length that we come upon. Thus, we have for instance a list of fly time to go from letter "a" to letter "c".
How to exploit those ?
Our first idea is to consider the mean of all those value. Thus, we would be able to compare two matrix with some sort of distance, and decide, if they are close enough to be considered as being the work of the same person. We have a few idea to find such a number :

- Mean
- Mean without the first and last quartile

  Here, we want to tackle a problem linked to the speed of typing of someone : the fly time is lesser than 0.3 seconds. Thus, if the writer stop to think what he should write next, we have a value of several seconds that max a fly time by an order of magnitude.
- Geometric Mean
- Geometric Mean without the first and last quartile.

  With this solution, we made the value to be closer to 0, which could complicate the differentiation of two matrices.

4.1.3 Distance

We will define the distance d between two matrices. To do so, let's consider M_1, M_2 two matrices of same dimension. We call S_i the set of coordinates u, v such that $(M_i)_{u,v}$ is not NULL. We define d here :

$$d(M_1, M_2) = \begin{cases} 0 & if \quad S_1 = S_2 = \emptyset \\ 1 & if \quad |S_1 \cap S_2| = \emptyset and |S_1 \cup S_2| \neq \emptyset \\ \frac{1}{|S_1 \cap S_2|} \sum_{u,v \in |S_1 \cap S_2|} |(M_1)_{u,v} - (M_2)_{u,v}| & if \quad |S_1 \cap S_2| \neq \emptyset \end{cases} \tag{1}$$

We note that d is not a distance in the mathematical sense. Indeed, we have the symmetry and separation property, but we don't have the triangle inequality. However, if we consider a set of matrices with the same S_i, then we have defined a distance. The normalization term in the expression is here to be able to compare distance between two matrices, even if they have different S_i.

4.1.4 Results

We present here our first result : The test was done between Rédouane and Xavier. They both wrote a text enter the dictation of Loïs. Then they typed the first sentences that came spontaneously to their mind. It was done on this purpose to be the closest to their "real" way of typing.

By just considering the fly time, we obtain :

X_l : learning, X_t : test

- Distance $(X_l, R_t) = 459$
- Distance $(R_l, R_t) = 240$
- Distance $(X_l, X_t) = 43$
- Distance $(R_l, X_t) = 153$

Here we notice that when we inverse the role of learn and test, then the algorithm make a mistake. Thus, with this simple example, we notice that we can obtain pretty nice results, with just a 43ms difference on average between the test and the train for Xavier, but it does not work well when the training set is small.

4.2 The new algorithm

4.3 Experimental Protocol

We constructed a data set on 7 persons.

A sample being composed of :

- 6 entry of a same fixed phrase.
- 2 free text of around 60 words.

The phrase we decided to type as a fixed text was the following one:
"Je certifie que cette soumission est le fruit de mon propre travail effectué en accord avec la Charte Anti-Plagiat."

This phrase has the particularity of containing some capital letters in the middle of it, for our way of typing a capital letter is relatively characteristic of ourselves. Nevertheless it is a quite common and short phrase, with no other intentional features.

We then developed two quite simple and standard protocols, one to analyze the use of a fixed phrase and the other the free text. In protocol 1, our algorithm creates a user profile with 3 instances of the fixed phrases. Then the test are done, when all 7 profiles have been constructed, on the remaining 3 fixed phrases for each individual. For the free text, we had only 4 individuals, with 2 texts each, so we constructed the profile with one, and test with the other.

Even, if our database is not really consequent, if considered as an authentication problem, we would have had multiple intruders attack to simulate (once all profiles are created, for each profile, trying to authenticate with all test typed texts).

4.4 Our implementation

For each pair of letters, the training gave us, for each profile a mean and standard deviation for this profile and pair of letter, and so for each pair of letters in the test text we give a point to a profile if the time for the pair of letter is around the mean more or less the standard deviation time a parameter we have to tune. The profile with the highest score is returned.

for *all apparition of a pair of letter in the test text* **do**
  for *all profiles p* **do**
    μ training mean for this pair and profile;
    σ training standard deviation for this pair and profile;
    d time for this apparition of this pair in the test text;
    δ parameter of the algorithm, 1.5 by default;
    if $d = \mu \pm \delta\sigma$ **then**
      $S_p = S_p + 1$ for this profile;
    end
  end
end
Return profile with highest score

Algorithm 1: Closest profile

4.5 Results

For protocol 1, an example of a correct and an incorrect result, we give the score attributed for each profiles given a text from the profile Lois and from the profile Yannis.

Name	lois:	Yannis :
lois	28	22
Yannis	16	23
Xavier	15	19
Angele	13	19
hugo	20	24
Emile	23	23
red	21	24

The results from protocol 1, with variation on the parameter delta

delta	correct/21
1	12
1.4	15
2	16
2.2	13

For the second protocol, we had 3 tests rightly classified on 4, so few but encouraging results, even with a really open protocol (free and quite short text with no restriction at all).

Conclusion

So we obtained some encouraging results which need more testing and as well as narrowing our approach on the problem.
One of the possible amelioration we could implement is that if two scores are close, we could rerun with a different δ (in front of the standard deviation).
The next step would be to adapt our implementation to authentication and provides some more precise testing results.

References

[1] Gunetti, Daniele and Picardi, Claudia Keystroke Analysis of Free Text *ACM Trans. Inf. Syst. Secur.*, 8(3):312–347, 2005.

[2] Maas, Andrew and Heather, Chris and Do, Chuong (Tom) and Brandman, Relly and Koller, Daphne and Ng, Andrew Offering Verified Credentials in Massive Open Online Courses: MOOCs and Technology to Advance Learning and Learning Research *Ubiquity*, 2:1–2:11, 2014.