User Classification for Keystroke Dynamics Authentication

Sylvain Hocquet, Jean-Yves Ramel, and Hubert Cardot

Université François Rabelais de Tours,
Laboratoire d'Informatique (EA 2101),
64 Avenue Jean Portalis, 37200 TOURS, France
{sylvain.hocquet,jean-yves.ramel,hubert.cardot}@univ-tours.fr

Abstract. In this paper, we propose a method to realize a classification of keystroke dynamics users before performing user authentication. The objective is to set automatically the individual parameters of the classification method for each class of users. Features are extracted from each user learning set, and then a clustering algorithm divides the user set in clusters. A set of parameters is estimated for each cluster. Authentication is then realized in a two steps process. First the users are associated to a cluster and second, the parameters of this cluster are used during the authentication step. This two steps process provides better results than system using global settings.

Keywords: keystroke dynamics, clustering, parameters adaptation.

1 Introduction

Since a few years, the need of more security for every day life has greatly increased. The biometric is a promising solution to answer this challenge. Biometric divides itself in two fields: the physical biometric and the behavioral biometric. The physical biometric methods (fingerprint, hand recognition...) are usually more accurate compared to methods based on the study of behavior (signature, voice, gait…). However, the behavioral methods are easier to implement, and better accepted by users. The problem of this kind of methods is the great variability in the user behaviors. In the case of an authentication problem, this variability implies difficulties for the setting of thresholds used by the system to separate authentic users from impostors. Most of the times, the parameters and thresholds are the same for all the users. This choice results in a great disparity of performances between users. The variability of some user profiles implies to accept impostors and at the opposite side, for some users with specific practice, all attempts including authentic ones are refused. Therefore, the automatic determination of the threshold for each user seems to be a solution to solve this problem. This paper first describes briefly classical methods used in keystroke dynamics. Next, our clustering algorithm is detailed and results are discussed in conclusion.

2 Keystroke Dynamics

Keystroke dynamics is the field of biometrics that studies the way a user interacts with a keyboard. To extract data from the striking of a user, the times between

S.-W. Lee and S.Z. Li (Eds.): ICB 2007, LNCS 4642, pp. 531–539, 2007.

keyboard events are used. One commercial application exist [1]. Keyboard events are the pressure and the release of a key. For a couple of successive keys, several different times are extracted:

- P-P (Press - Press) : time between two key pressures (T2-T1)
- P-R (Press - Release) : time between the pressure on a key and his release (T3-T1 and T4-T2)
- R-P (Release - Press) : time between the release of a key and the press on the next key (T3-T2)
- R-R (Release - Release) : time between the release of two successive keys (T4-T3)

For a sequence of strokes, the system extracts a feature vector for each type of time; to finally obtain vectors of four features (PP, PR, RP, and RR). The first test to differentiate people using the keystroke dynamics were carried out by Gaines et al. [2] in 1980. These first results were encouraging but inapplicable in real cases because of the low number of involved people and because of the length of the text used for the authentication. In the five last years, many studies took place on this subject, a summary of many of them is presented in [3] and a more recent review has been conducted in [4]. In this study, we restrict our investigations to the authentication or identity verification problem. Our goal is to compare a new observation with feature vector associated to only one profile and then to decide if the observation is from the same user or not. Therefore, we are limited to only a few observations from the user in the learning process. In addition, no impostor's data are available. A great variability of methods has been applied to solve this problem using similarity measure [5], one class support vector machine and genetic algorithm [6], hidden markov model [7], neural network [8]…

3 The Proposed Methods

To prove the performances of such systems, we have chosen to use a fusion of three different methods to decide if a new observation is corresponding or not to a given user. We have chosen these methods for their fair good performance, the low volume of data needed for training, and their easy implementation. The first one is an adaptation of a statistical method and uses the average and the standard deviation of each feature. The second is based on a measure of disorder between the different feature vectors, and the third one uses the concept of time discretization.

3.1 The Statistical Method

This method uses statistical measures extracted from the keystroke dynamics; that is to say the average and standard deviation of the different acquired times are computed. User profile is composed of the ten logins acquired during enrollment process. The profile contains the average and standard deviation of all times extracted from the striking sequences. To compute a score on an n length feature vector, with t_i

the ith time and μi, σ_i the associated average and standard deviation stored in the profile, the method use:

$$score_{statiscal} = 1 - \frac{1}{n}\sum_{i=1}^{n} e^{-\frac{|t_i - \mu_i|}{\sigma_i}}$$

3.2 Method Based on a Measure of Disorder

The second method studies the variation between the time ranks in the profile and in the tested sequences. This is corresponding in fact to the measurement of disorder between two vectors. To measure the difference between the ranks of the times, the times of each observation are reordered from the longest to the shortest. The information store in the profile for each time is the average rank computed according to the logins in the training set. To compute the distance the sum of the ranks difference in the profile and in the observation is used. With r_i^O the rank of time i in the observation and r_i^P the average rank of time i in the profile, n the number of times, the formula to compute a score is:

$$score_{disorder} = \frac{\sum_{i=1}^{n}(r_i^P - r_i^O)}{n * (n-1)}$$

3.3 Time Discretization Method

The third method uses a time discretization. Each time is put into a class according to its duration. To compare an observation with a profile, the difference between the indexes of the interval was chosen. The intervals are fixed for each class of times (for example between 0 and 30 ms for the 5th class). The class of each time i in the profile c_i^p are compared with the one in the tested sequences c_i^o Then the score is computed with:

$$score_{discretization} = \frac{1}{n}\sum_{i=1}^{n}\left|c_i^p - c_i^o\right|$$

3.4 Fusion of the Three Methods

Each one of the scores provided by these three methods must be normalized. Then, the fusion of the three results is computed by combining the scores. Previous experiments using fusion in biometric have shown that good results are obtained with a sum rule [9] and a z-score [10] normalization. To manage the differences of performances of the three methods, fusion weights (w_i) are associated to each of score from our method. So the final score becomes:

$$Final\ score(FSC) = \Sigma_i w_i * score_i$$

To normalize the final score, the sum of the weights has to be equal to 1, so only two weights must be estimated.

4 The User Classification Step

4.1 Parameters Personalization

Each classification method applied to an authentication problem needs in most cases a few parameters to be set in order to give the similarity score. In addition, a threshold is needed to take the final decision of authentication. All these parameters are often chosen according to all users after several experiments. However, such methods can cause some problems in one class problem, especially in the case of the behavioural biometrics. For example, if the threshold is the same for all the users, it can result in a great disparity of performances between users. To estimate these parameters for each user according to information present in their profile, a set of information can be extracted for each user from the set of ten sequences in their profile. This information includes:

- The length of one sequence, in characters (1 feature)
- The average, standard deviation, maximum and minimum of the times from the four extracted time vectors (PP, RP RR, PR) (4*4 features)
- The average and standard deviation of the total duration of the striking sequences (2 features)
- The average, standard deviation maximum and minimum of the three scores computed on the learning sequences by our three methods. In order to compute these scores, the leave-one-out method is used; nine sequences are included in the profile and the score is computed with the last one. The process is repeated with the other combinations of sequences.(3*4 features)

Finally, 31 features can be used, to characterize a profile containing ten learning sequences. In a previous work [11] we have decided to simplify the problem by making classes of users based on the proximity of their optimum parameters. So a set of parameters is no more associated to one user but to a class of users. The construction of the classes is made by auto associative methods and clustering algorithm on the optimum parameters found by using a private base. We obtain a good clustering result with a number of clusters equal to four. We finally use SVMs to compute the class of a new user. These SVMs were trained with the set of features extracted from user profiles.

In the following, we present a different approach; we tried to make classes using only the feature extracted from the user profile. Adequate parameters are then computed for each class. At the enrollment step, the class of the user is determined. The class whose center is near the user profile according to Euclidian distance is chosen. The parameters of this class will be used during the authentication process. Our first tests show that a clustering realized only on users training set give bad results. We observed that keystroke features evolve along the time. Therefore, we decide to continually recreate the classifiers with the last ten authentic sequences.

Therefore, users have not a single profile, but a profile for each acquisition they made after the enrollment.

4.2 Data Analysis

The clustering algorithm working on the 31 features extracted on the user profile may have difficulties to learn on this high dimensional space. So we decided to reduce the dimension of this space. It is impossible to a priori know wich features will best characterize a user. Therefore, data analysis methods are available to simplify the representation space. The method, we have used is a principal component analysis (PCA) [12]. The values of the first ten eigenvalues obtained after a PCA on the 31 features is presented in Table 1.

Table 1. Eigen values

Order	value	Cumulate value	% of inertia
1	8.6	8.6	38
2	4.2	12.8	56
3	2.7	15.6	68
4	2.3	17.9	78
5	1.3	19.3	84
6	0.9	20.2	88
7	0.6	20.9	90
8	0.5	21.4	93
9	0.4	21.8	95
10	0.3	22.1	96

The first five values are explaining more than 80 % of the inertia of the system. So we decided to keep only the first five factorial axes to represent a user profile.

A major advantage of factorial analysis is to represent on a small dimension, data that are defined in a high dimension space. On Figure 1, the first factorial plan is drawn. This plan explained only 56% of the initial inertia, but this is a fine first approximation. Each user profile is represented by a point. We can easily see on this figure, the division of user population into three apparent classes. One class contains the majority of users; we suppose these users will share common parameters. The other two classes regroup users with different behaviours and may need specific set of parameters. Our test has showed us, that this number of class still valid in the five dimensional spaces we have chosen.

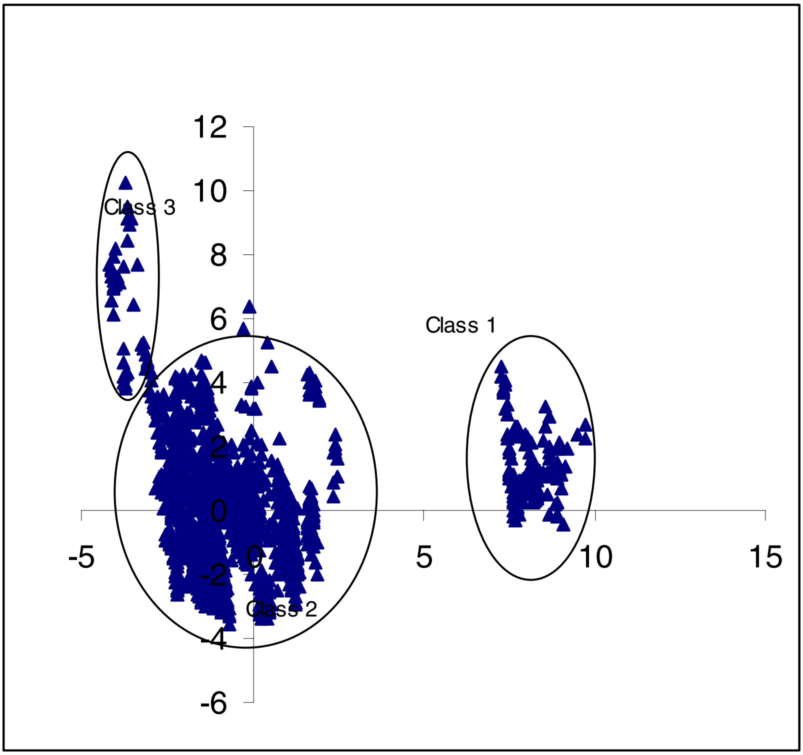

Fig. 1. First factorial plan and clustering on the feature extracted from user profile

The k-means algorithm is used to make clustering starting from the coordinates of the point on the first two axes, with k set to three. This simple algorithm gives a good classification using a low dimension space. The k-means with k equal to three will be used in the rest of our works, to create the class with the five retained values.

4.3 Computation of the Required Parameters

Our classification method required three parameters:

- A security threshold: this threshold needs to be determined according to the variability of the user profile. If the user has a stable profile, the security level can be high so the threshold has to be decrease. If the user profile has a lot of variation, the threshold must be increased to relax the security and allow him to authenticate itself.
- Two-fusion weights: these weights determinate the most accurate feature for a user

For each of our three classes the three parameters are computed by minimizing the sum of the FRR and FAR. The space of parameters is explored during an exhaustive search.

Table 2. Description of classes

	threshold	The weight of statistical method	The weight of disorder method	number of profiles in the classes
class 1	0.9	0.10	0.3	340
class 2	1	0	0.5	1322
class 3	1.2	0.2	0.4	156

Table 2 presents the characteristics of the three classes. The largest class is the class two. This class is characterized by an average threshold, and by the dual utilization of the disorder measure method and time discretization method, rather than the statistical method.

The class one regroups users with a low threshold. This class may contain profiles characterized by their stability. This class has the lowest weights for the disorder measure method.

The class three seems to represents users with great variations in the profiles as the threshold is high.

To verify such affirmation, we have proceeded to several experiments described in the following section.

5 Experiments and Results

Our private database is composed of 38 users. The keystroke sequences are corresponding to user names and passwords with different lengths (between 8 and 30 characters for the total length of the sequences). The data base is also containing impostor's attacks for each user. Each user has provided between 20 and 110 logins sequences and some people have been asked to try to reproduce some sequences between 20 and 100 times.

The different methods proposed to adapt the parameters (the security threshold and the fusion weights) for each user have been evaluated by using the leave one out method. We estimated the parameters of one user with a tool trained on all the other users.

Implementation of real life applications should also integrated our private database. This database will be considered as a training set and is supposed to be representative of the different classes of users. Results provided by a such application are presented in table 3.

Table 3 shows important improvements compared to the use of global parameters. Performances are improved for all the classes. The obtained error rates are very good for a keystroke dynamics method. However, these error rates hide the fact that the

Table 3. Results of user classification

	FAR% Global Parameter	FRR% Global Parameter	FAR %	FRR%
class 1	0	0.1	0	0
class 2	1.8	5.8	2.8	3.3
class 3	0	1.9	0	1.9
Total	1.8	5.3	1.7	2.1

error is computed on all profiles of a class. It tends to minimize the influence of low performance users, who has catastrophic results. We have identified three of this type of users in our base (EER>30%), they have given only a few sequences (between 20 and 40) so their influence is small. If we compute the average of the EER computed on each user we obtain 4.5%, corresponding to a fair performance.

This value points an other problem of our method: probably, because of the few numbers of problematic users, we are unable to achieve our second objective which was identifying them before the authentication with our clustering methods.

6 Conclusion

The works presented in this paper shows that the keystroke dynamics can be used to perform authentication or identification in real case applications (with an EER around 5%). Adaptation of thresholds and parameters of the system according to user behaviour is a promising way for improving the performances of keystroke dynamics. In addition, the combination of classifiers by adding a fusion step in the system architecture also improves performances. Our experiment shows important improvements even with simple classifiers. Our works on parameters adaptation and classification of user show also interesting results and other improvements remain possible. The authentication of problematic users is still a problem. Therefore, the keystroke dynamics is beginning reaching maturity even if, in real applications, a series of problems can occur: For example, how the systems will react when the keyboard changed? This problem is also present in other biometric systems. It can probably explain why behavioural biometric remains rather marginal in commercial applications.

References

1. Biopassword: Biopassword, http://www.biopassword.com
2. Gaines, R.S., al.: Authentication by Keystroke Timing: Some Preliminary Results. Rand Corporation (1980)
3. Ilonen, J.: Keystroke dynamics. In: Advanced Topics in Information Processing (2003)
4. Peacock, A., Ke, X., Wilkerson, M.: Typing Patterns: A Key to User Identification. IEEE: Security & Privacy Magazine 02(5), 40–47 (2004)
5. Monrose, F., Rubin, A.D.: Keystroke dynamics as a biometric for authentication. Future Generation Computer Systems 16(4), 351–359 (2000)
6. Yu, E., Cho, S.: Keystroke dynamics identity verification–its problems and practical solutions. Computers and Security 23(5), 428–440 (2004)
7. Chen, W., Chang, W.: Applying Hidden Markov Models to Keystroke Pattern Analysis for Password Verification. In: IEEE International Conference on Information Reuse and Integration, pp. 467–474. IEEE Computer Society Press, Los Alamitos (2004)
8. Revett, K., al.: Authenticating computer access based on keystroke dynamics using a probabilistic neural network. DSI - Sistemas de Computação e Comunicações, 2006 (to appear)
9. Kittler, J., al.: On Combining Classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(3), 226–239 (1998)
10. Jain, A.K., Nandakumar, K., Ross, A.: Score Normalization in Multimodal Biometric Systems. Pattern Recognition 38(12), 2270–2285 (2004)
11. Hocquet, S., Ramel, J.-Y., Cardot, H.: Estimation of User Specific Parameters in One-class Problems. In: 18th International Conference on Pattern Recognition, Hong Kong, pp. 449–452 (2006)
12. Jolliffe, I.T.: Principal Component Analysis. Springer, Heidelberg (1990)