CS229 Final Project
Personal Identification through Keystroke Dynamics

Stock Sawasdee

December 2017

Introduction

In computer securities, identification of a user is essential for a computer to authenticate some action or detect any suspicious activities. Popular techniques include passwords, PIN, and biometrics such as face recognition. Keystroke dynamics, unlike fingerprints or iris patterns, is a biometric verification that does not require additional hardware. When a person type on a keyboard, they tend to do so with unique stroke patterns, which could be used for identification. Keystroke dynamics can also work as a supplementary tool to passwords in that it can record the use of each user in real time and could detect whether other people are using their computers.

In this project, I applied two machine learning algorithms, k-nearest neighbors and softmax regression to the keystroke data. The output is a predicted user.

Related work

Keystroke dynamics developed its idea from the telegraph age. In World War II, military intelligence could identify the fist (the keying rhythm on the telegraph) of the sender, and make strategic decisions based on that information. In the age of computers, the same concept for telegraphs can be applied to keyboards.

There are a number of past studies on keystroke dynamics. Joyce and Gupta [2] used key latency as features, calculated the mean latency values for each user, and used 1-norm distance between the data and the references to classify. Monrose and Rubin [3] collected data for 11 months for 63 users and applied a similar technique as Joyce and Gupta; however, Monrose and Rubin also includes key durations as features and modified the distance to Euclidean norm. They also used factor analysis to determine feature sets and use k-nearest neighbors algorithm to cluster the data. Roy et al. [4] also discussed the performance of different algorithms and different distance schemes.

Dataset and Features

The dataset I used for this project was collected during November 18 – 26. I sent out to my friends, an HTML file with a script that records timestamps for any key pressed and released. Each person was asked to type a word, ‘Stanford’ for a total of 200 times. That will include a total of 10 keys (20 events), ‘Shift’, ‘s’, ‘t’, ‘a’, ‘n’, ‘f’, ‘o’, ‘r’, ‘d’, and ‘Enter’. I received the data from a total of 5 users, and each data point is labeled by the user ID (1 – 5). The total number of data points is 998.

From the 20 timestamps, I extracted key durations, down-down key latencies (the time between one key press and the next), and up-up key latencies (the time between one key release and the next). Sometimes when keys are not released in order, the up-up latencies can become negative, for example, some people release the ‘Shift’ key after the character key.

When a user type the word incorrectly, the data will show high latency because the timestamp will be delayed by the backspaces. I have tried to apply the algorithms to the data with (Figure 1a) and without (Figure 1b) backspace correction. When the data is filtered, the number of samples is reduced to 633.

The data are shuffled and divided 80% into training set and 20% into test set.

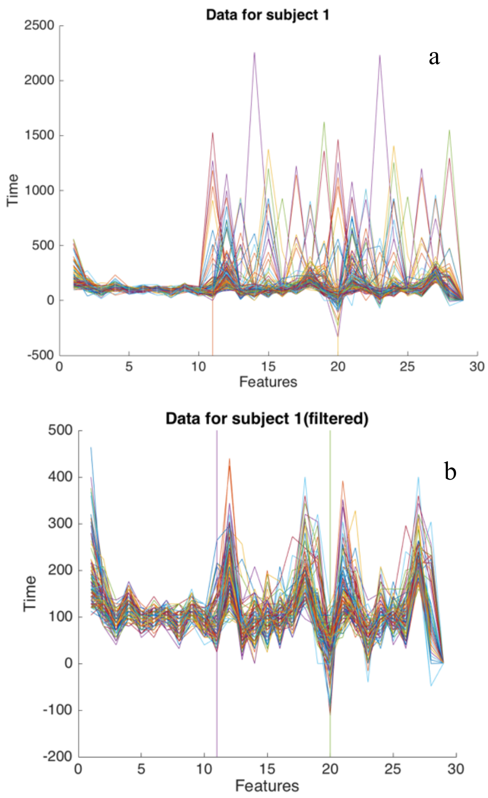

Figure 1: Data for subject 1 before (a) and after (b) mistakes are removed. The features are in the following order: key durations, down-down latencies, and up-up latencies.

Methods

I used two algorithms in this project, k-nearest neighbors and softmax regression.

1. k-Nearest Neighbors (kNN)

This algorithm uses training set as a model. Given a test data point, the algorithm will find k points in the training set that are closest to the data point. Then it looks at the majority vote for label and return the label with maximum vote as an estimate.

2. Softmax Regression

Softmax regression is like an extended version of logistic regression. Instead of sigmoid function, softmax regression uses softmax function, which is defined by Equation (1).

$$\sigma(z)_j = \frac{\exp(\theta^{(k)T} x^{(i)})}{\sum_{j=1}^{K} \exp(\theta^{(j)T} x^{(i)})} \qquad (1)$$

The estimate will be the class (user) that has the largest sigmoid function. Therefore, the loss function that needs to be minimized can be written as Equation (2).

$$J(\theta) = \text{-}[\sum_{i=1}^{m} \sum_{k=1}^{K} 1\{y^{(i)} = k\} \log \frac{\exp(\theta^{(k)T} x^{(i)})}{\sum_{j=1}^{K} \exp(\theta^{(j)T} x^{(i)})}] \qquad (2)$$

I used gradient descent as an optimization scheme.

Results

The metrics that I used for this experiment are accuracy and confusion matrices. From Table 1, we can see that k-nearest neighbors works better than softmax regression in general. The number of neighbors, k, is optimized at k=3. The reason that softmax regression does not perform well is because the data have very large spread and the clusters are close together, so for a linear algorithm like soft max regression it is more likely that the data is misinterpret. On the other hand, k-nearest neighbors works better because when a user types a word once, there is a very high possibility that they will type with a very similar pattern for another time even though there are many times that they type very differently.

When the mistakes are removed from the data, the accuracy improved in both k-nearest neighbors and softmax regression. This suggests that the incorrectly typed data skew the dataset.

If we look at the confusion matrices (Table 2), we can see that there are not many data points for user 5 when the data is filtered. Because user 5 made a lot of typing mistakes, the results in a non-filtered dataset are therefore less accurate when predicting user 5.

Table 1: Accuracy

Model	Training (%)	Test (%)
1-NN w/o filter	N/A	82
3-NN w/o filter	89.1	83.5
5-NN w/o filter	85.59	77.5
Softmax w/o filter	68.8	68.5
1-NN w/ filter	N/A	94.49
3-NN w/ filter	97.83	96.85
5-NN w/ filter	96.44	95.28
Softmax w/ filter	80.63	85.83

Table 2: Confusion matrices (A=Actual, P=Predicted) on test dataset

a. 1-NN without filter

	P:1	P:2	P:3	P:4	P:5
A:1	39	1	2	2	4
A:2		30			
A:3		1	36	4	2
A:4		4	3	33	4
A:5	2	2	5		26

b. 3-NN without filter

	P:1	P:2	P:3	P:4	P:5
A:1	38	1	3	2	4
A:2		30			
A:3	1	1	38	2	1
A:4	3	2	4	34	1
A:5	2	2	3	1	27

c. 5-NN without filter

	P:1	P:2	P:3	P:4	P:5
A:1	36		2	3	7
A:2		29			1
A:3	1	2	35	3	2
A:4	2	3	4	33	2
A:5	2	3	3	5	22

d. Softmax without filter

	P:1	P:2	P:3	P:4	P:5
A:1	34		4	2	
A:2	1	31	2	11	2
A:3	2	3	26	5	1
A:4		5		25	1
A:5	2	8	7	7	21

e. 1-NN with filter

	P:1	P:2	P:3	P:4	P:5
A:1	24		1		1
A:2		26			
A:3			29		
A:4		3		34	
A:5			1	1	7

f. 3-NN with filter

	P:1	P:2	P:3	P:4	P:5
A:1	25		1		
A:2		26			
A:3			29		
A:4		1		36	
A:5			1	1	7

g. 5-NN with filter

	P:1	P:2	P:3	P:4	P:5
A:1	24		2		
A:2		26			
A:3			29		
A:4		2		35	
A:5			1	1	7

h. Softmax with filter

	P:1	P:2	P:3	P:4	P:5
A:1	25		1		
A:2		26			
A:3			26	3	
A:4		5		32	
A:5	1	3	1	4	

Conclusions

I collected the keystroke data from 5 users typing a word 'Stanford' and applied k-nearest neighbors and softmax regression to classify the data. The k-nearest neighbors works better than softmax regression because the clusters of data have large spread that overlap each other a lot.

In the future, there should be a study on whether mistakes can be used as a supplementary tool to improve accuracy, instead of being included directly in the model, especially for users that have a high mistake rate and the correctly typed data is scarce.

In this project, we only study the classification of different people. However, for the method to work as an identification, consistency is important. There should be another study on the same person whether they can produce the similar keystrokes throughout a long-term period.

References

[1] Checco, John C. “Keystroke Dynamics And Corporate Security.” *Checco Services, Inc.*, 2003, www.checco.com/about/john.checco/publications/2003_Keystroke_Biometrics_Intro.pdf

[2] Joyce, Rick, and Gopal Gupta. “Identity Authentication Based on Keystroke Latencies.”*Communications of the ACM*, vol. 33, no. 2, Jan. 1990, pp. 168–176., doi:10.1145/75577.75582.

[3] Monrose, Fabian, and Aviel D. Rubin. “Keystroke Dynamics as a Biometric for Authentication.” *Future Generation Computer Systems*, vol. 16, no. 4, 2000, pp. 351–359., doi:10.1016/s0167-739x(99)00059-x.

[4] Roy S., Roy U., Sinha D.D. (2016) Comparative Study of Various Features-Mining- Based Classifiers in Different Keystroke Dynamics Datasets. In: Satapathy S., Das S. (eds) *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 1*. Smart Innovation, Systems and Technologies, vol 50. Springer, Cham