Personal Identification through Keystroke Dynamics

Stock Sawasdee (paphop@stanford.edu)

CS 229 Final Project

Problem

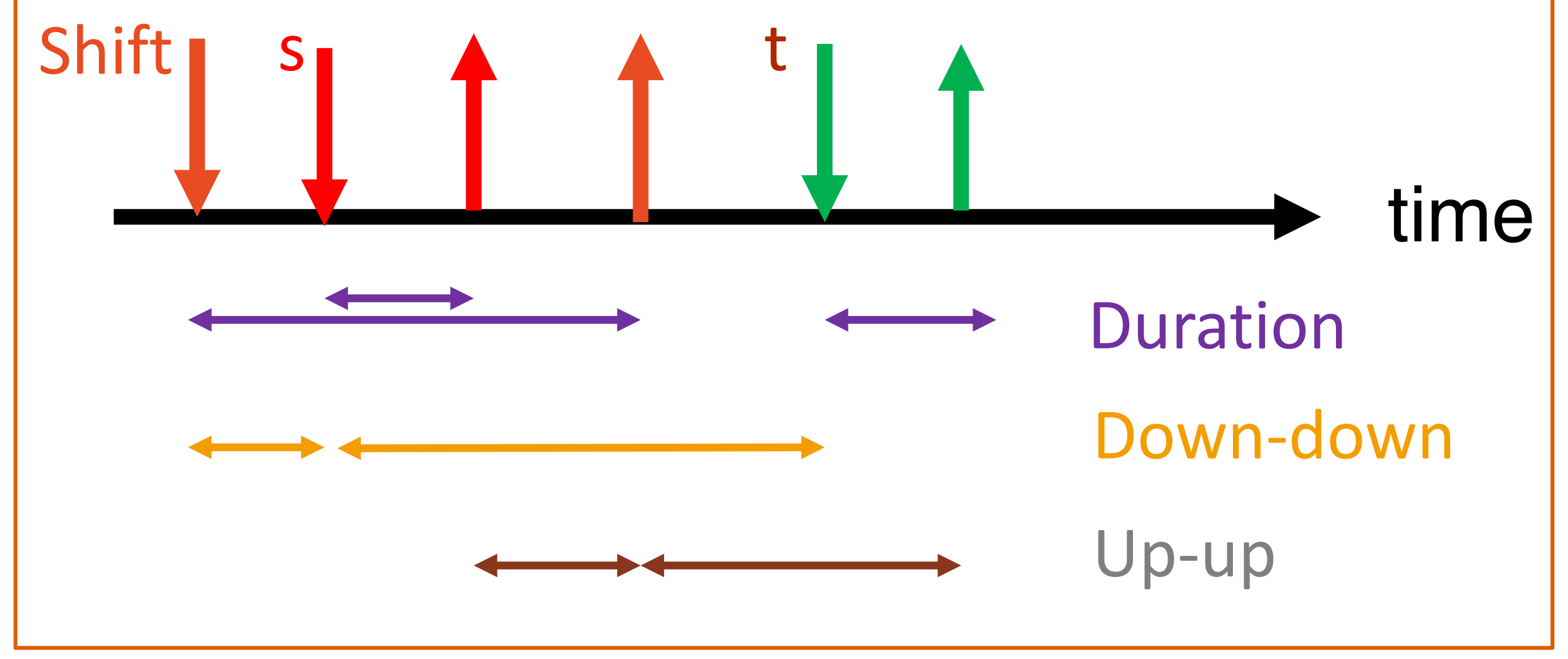
Identification of a user is essential for a computer to authenticate some action or detect any suspicious activities. In this project, I tried to experiment how keystroke dynamics can be used as an identification tool, using machine learning. Features such as the amount of time we press a key and the delay between each key should vary from person to person.

Data

I collected the data by having 5 subjects type a word, 'Stanford', 200 times each, and record the timestamps when each key is pressed and released. That will include a total of 9 keys (18 events), 'Shift', 's', 't', 'a', 'n', 'f', 'o', 'r', 'd', and 'Enter'. Each data is labeled by the subject ID (1 - 5).

Features

Each data consists of 20 timestamps, 10 for key-up and 10 for key-down. I extracted key durations (10 features), down-down key latencies (9 features), and up-up key latencies (9 features). Notice that latencies can be negative if the keys are not in order.

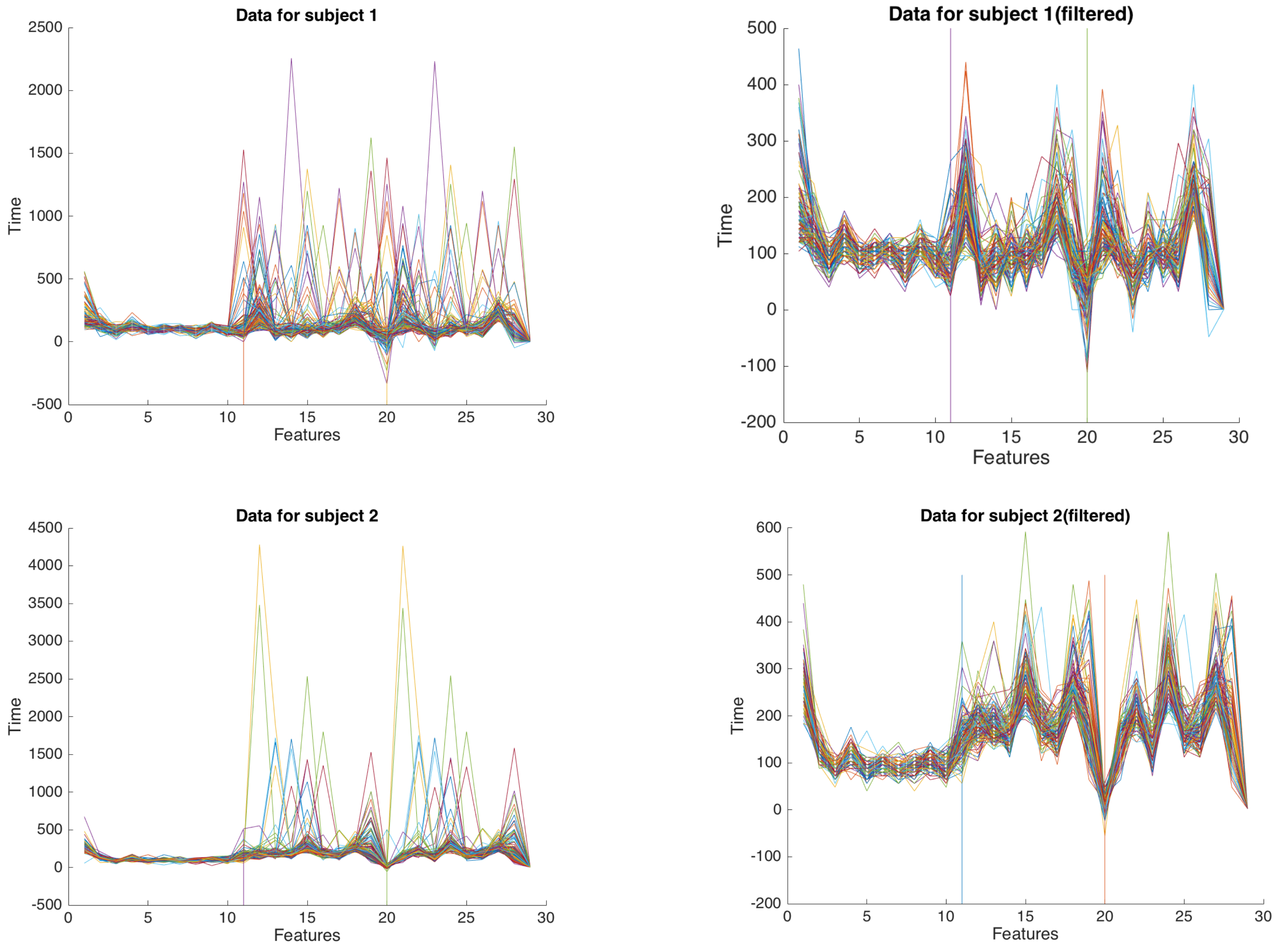


Models

The algorithms I used are k-nearest neighbors and softmax regression. I also filter out the data point that has too high latency, mostly when the subject typed incorrectly and needed to edit. When filtered, the number of samples reduce from 998 to 633.

kNN: Choose y based on maximum number of k nearest points to x with the same corresponding y

Softmax: $J(\theta) = -\left[\sum_{i=1}^{m} \sum_{k=1}^{K} 1\{y^{(i)} = k\} \log \frac{\exp(\theta^{(k)T} x^{(i)})}{\sum_{j=1}^{K} \exp(\theta^{(j)T} x^{(i)})}\right]$



Future

Although including data with incorrect typing decreases the efficiency of the model, the information about the location where a person frequently type wrong and what incorrect key is pressed should be useful as a feature. We should also collect the same data from the same person within a long period of time to see whether the data are consistent.

Results

Model	Training (80%) error	Test (20%) error
1-NN w/o filter	N/A	0.1800
3-NN w/o filter	0.1090	0.1650
5-NN w/o filter	0.1441	0.2250
Softmax w/o filter	0.3133	0.3133
1-NN w/ filter	N/A	0.0551
3-NN w/ filter	0.0217	0.0315
5-NN w/ filter	0.0356	0.0472
Softmax w/ filter	0.1937	0.1417

Discussion

We can see from the results that when the data include incorrectly typed data point, the data is skewed and the error increases a lot. Knn works much better than softmax because the data has a large spread and a lot of overlap between classes, so the linear classifier like softmax might not work very well.

It's interesting to see that the test error for softmax on the filtered data is smaller than the test data. That is probably because the training set has more 'bad data'.

Reference

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