MATH178 Midterm Progress Report

Alberto Garcia and Matthew LeMay

June 7th 2019

1 Introduction

The paper by Sitová et al. describes features of Hand Movement, Orientation, and Grasp (HMOG) extracted from cell phone data, and proposed these features for continuous authentication of users. Based on these features, we were curious to see if we could classify user's based on how stable their grip was. This could potentially be used to identify unstable grips and improve the physical safety of phones. Classifying grips by stability could also improve authentication performance: if a person has multiple situations in which they commonly use their phone with different motion characteristics, classification would allow authentication with separate sets of motion features for each situation.

We were interested in using the same features used in their experiments in order classify different taps based on stability. Rather than pulling out distinctive characteristics of individual users, we were interested in the characteristics of tap events themselves. Our plan for the midterm was to extract these stability features from tap events and then use a clustering algorithm to classify them. After doing some research we decided to use scikit's sklearn.cluster module to use the K-Means method of clustering. While this may not be the best method for clustering the points since the algorithm uses distances between points instead of a distance metric more specific to the situation, we figured this would be a good starting point for classifying our tap events.

2 Data set and extraction

We used the data set provided by the authors of the HMOG paper. This data set was collected from 100 users of smartphones, who each participated in 24 sessions. In each session, they were sitting or walking, and either reading text, reading a map, or writing text. During these activities, sensor data from the accelerometer, gyroscope, and magnetometer data was recorded, along with the time stamps of each time they touched the screen.

For each touch event, and for data on the x,y,z axes of the accelerometer, gyroscope, and magnetometer, the authors extracted 5 features related to grasp resistance, and the 3 features related to grasp stability. We used the same features as they did, so a brief explanation of what these features are and how they are computed will be helpful.

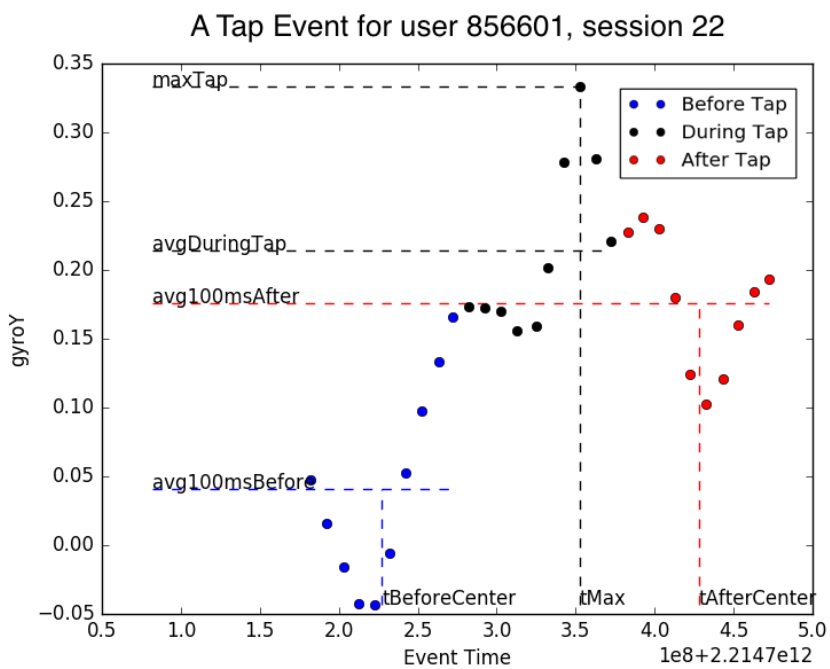

Figure 1: Gyroscope y-axis readings for a single tap event, with key values annotated

Figure 1 shows a time series of gyroscope y-axis readings for one such tap event. Let Y represent the value of the sensor reading plotted on the vertical axis. The time readings are divided into readings from the 100ms window before the tap (shown in blue), readings during the tap (shown in black), and readings from the 100ms window after the tap (shown in red). We first compute the average of the sensor readings in each of these windows, denoted by avg100msBefore, avgDuringTap, and avg100msAfter, respectively, and the maximum reading achieved during the tap, denoted by maxTap. Then the following are the 5 resistance features:

1. Mean of Y during tap (or simply avgDuringTap).
2. Standard deviation of Y during tap.
3. Difference in Y readings before and after, computed by
$$\text{avg100msAfter} - \text{avg100msBefore}.$$
4. Net change in Y readings caused by the tap, computed by
$$\text{avgDuringTap} - \text{avg100msBefore}.$$
5. Maximum change in Y readings caused by the tap, computed by
$$\text{maxTap} - \text{avg100msBefore}.$$

We then compute the stability features, which are based on time durations. We first find the time value at the center of the before and after windows, denoted by tBeforeCenter and tAfterCenter, and the time value during the tap at which the maximum reading is achieved, denoted by tMax. Then the following are the 3 stability features:

1. Restoration time: The duration of time to regain stability after a tap event. To find this, we must find the time tMin at which stability is regained. If $Y_1, Y_2, ..., Y_n$ are the readings during the window 100ms after

the tap event, and $t_1, t_2, ..., t_n$ are the corresponding time values, then min is defined as the value of i at which

$$\frac{\sum_{j=i}^{n}(|Y_j - \text{avg100msBefore}|)}{n - i + 1}$$

achieves its minimum, and $\text{tMin} = t_{min}$. Then the feature is computed as the difference betwen t_{min} and the final time value during the tap.

2. The normalized time duration for the sensor value to change from its value before the tap to its value after, computed by

$$\frac{\text{tAfterCenter} - \text{tBeforeCenter}}{\text{avg100msAfter} - \text{avg100msBefore}}.$$

3. The normalized time duration for the sensor valus to change from its maximum value during the tap to its value after, computed by

$$\frac{\text{tAfterCenter} - \text{tMax}}{\text{avg100msAfter} - \text{maxTap}}.$$

We computed these 8 features on 3 axes (x, y, and z) for both the accelerometer and the gyroscope data. Although in the original HMOG paper the authors also used magnetometer data, they found that "combining magnetometer features with features from accelerometer and gyroscope did not improve performance" (Sitová 2016, 6). This could be because magnetic features vary more in response to different locations (possibly with differing amounts of metal) than they do in response to characteristics of motion. Consequently we decided to ignore the magnetometer data, focusing instead on the accelerometer and gyroscope data. Therefore for each tap we computed a total of $8 \times 3 \times 2$ or 48 features.

3 Data analysis

In order to analyze the data, we took a user and split their 24 csv files, which contained the features of all tap events for a given session, and split them into two groups, ensuring that there was an equal amount of activities, such as map+walking or reading+sitting, in both groups. Then we constructed two DataFrames using the two groups of csv files to create our training data and testing data.

Before using K-Means we pre-processed the data. First we replaced any missing values or inf values in the train set and test set with the mean column values. Then we converted the train set and test set into numpy arrays in order to scale the feature values to values between 0 and 1 to improve the accuracy of K-Means. Finally we computed the K-Means Clustering on the training data and predicted the closest cluster that each tap event in the testing data belonged to.

To visualize the predictions, we plotted the tap events based on the following two features: the accelerometer restoration time along the z-axis and the gyroscope restoration time along the z-axis. The two colors indicate which cluster the tap event was grouped with.

In Figures 2 - 4 we can see that the tap events with higher restoration times for both the accelerometer and the gyroscope are colored purple and those with a lower restoration time for both sensors are colored yellow. By using K-Means we were able to successfully classify tap events based on how stable they were. The yellow tap events are more stable than the purple tap events since they have a shorter restoration time. Although we had some incorrect classifications in Figure 2, Figure 1 and Figure 3 show that overall K-Means was a good method for classifying the tap events.

This is a good starting point for analyzing a user's grip and determining if they need to change their grip stability which would allow the user to improve the physical safety of their phone.

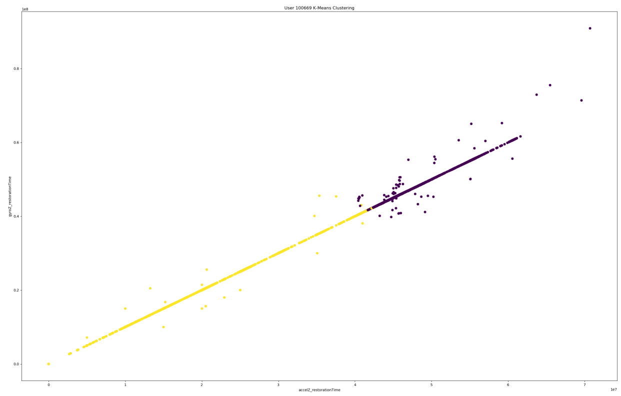

Figure 2: Graph showing accelerometer restoration time vs gyroscope restoration time along z-axis to visualize K-Means Clustering for User 100669

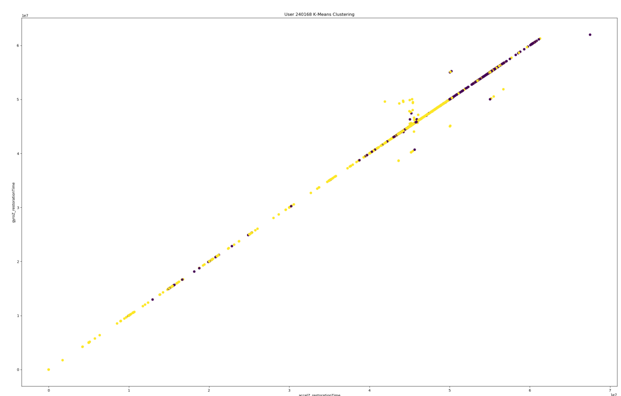

Figure 3: Graph showing accelerometer restoration time vs gyroscope restoration time along z-axis to visualize K-Means Clustering for User 240168

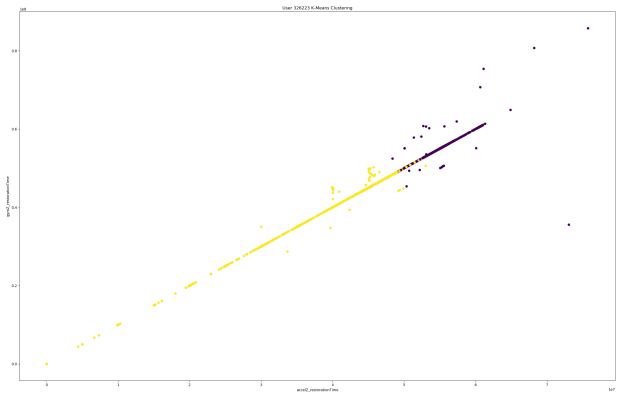

Figure 4: Graph showing accelerometer restoration time vs gyroscope restoration time along z-axis to visualize K-Means Clustering for User 326223

4 Future Work

As we continue work on this project, there are several things we believe we should work further on. Firstly we should attempt to simplify the large number of features we have into a smaller number of key features, through Principal Component Analysis or something equivalent to it. As can be seen in figures 2 - 4, many of our features, such as restoration time along different axes, are strongly correlated, implying that our 48 features can be reduced to a more fundamental set of features.

Our analysis used a k-means algorithm which assumes a Euclidean distance metric between different features. It would be worthwhile to investigate whether the feature vectors we constructed might live on a lower-dimensional subspace, and therefore if some non-Euclidean distance metric might be more appropriate. If so, we could perform k-means using a more effective metric, and get a clustering of the data that is more physically meaningful. We could also investigate results other clustering methods besides K-means.

Most importantly, we should try to use the clusters we have constructed to make testable predictions, so we can evaluate their effectiveness. For example, we could use our existing data to determine how accurately our clusters can predict whether a user is writing, reading text, or reading a map. Ultimately we would also like to give the different clusters a name with real-world significance, like "one-handed grip" or "two-handed grip".

5 References

Z. Sitová, J. Šeděnka, Q. Yang, G. Peng, G. Zhou, P. Gasti, and K. Balagani. "HMOG: New Behavioral Biometric Features for Continuous Authentication of Smartphone Users". *Information Forensics and Security, IEEE Transactions on*, PP(99)1-1, 2016.