4/30/2017

**CS 6375.001 Machine Learning Project Report**

**Gaussian Mixture Models for Keystroke Dynamics**

**Team Members**

**Farheen Shaik-fxs160530**

**Nanditha Valsaraj-nxv160930**

**Sushma Challa-sxc164530**

* **Introduction and problem description:**

Problem Statement:

To successfully authenticate a user based on keystroke dynamics using Gaussian Mixture Models.

Significance of the Problem:

Keystroke dynamics or typing dynamics refers to the automated method of identifying or confirming the identity of an individual based on the manner and the rhythm of typing on a keyboard. **Keystroke dynamics** is a behavioural biometric, this means that the [biometric factor](http://www.biometric-solutions.com/glossary.php?term=Biometric%20factor) is ‘something you do’.

When used in conjunction with traditional methods for authentication keystroke dynamics provide an extra level of security.

Basic Approach:

In order to apply keystroke dynamics, we have obtained data regarding the keystroke biometrics of five users including KeyUp and KeyDown values. The raw measurements used for keystroke dynamics in real time are dwell time and flight time.

* **Dwell time** is the time duration that a key is pressed
* **Flight time** is the time duration in between releasing a key and pressing the next key

The attributes KeyUp and KeyDown can be used to calculate dwell time and flight time. The dwell time for a particular key can be calculated as the difference between the timestamps for KeyUp and KeyDown for that particular key. The flight time can be calculated as the difference between the timestamps for KeyUp of the previous key and KeyDown of the next key. Thus, the attributes can be used as an accurate representation for keystroke dynamics.

We have divided the project into three parts, namely

1. Feature Extraction
2. GMM modelling and training
3. Classification

* **Dataset Description:**

|  |  |
| --- | --- |
| **Data Set Characteristics** | Multivariate |
|  |  |
| **Input data Attribute Characteristics** | 1 Time and 5 Categorical |
|  |  |
| **Associated Tasks** | Clustering |
|  |  |
| **Number of Input Attributes** | 6 |
|  |  |

Since the dataset contains categorical attributes, they must be converted into numerical attributes before they can be used to train.

Below is a list of the six input attributes:

|  |  |  |
| --- | --- | --- |
| Attributes | Type | Description |
| Event\_type | Categorical | Identifies the type of keystroke: either KeyUp or KeyDown |
| Key\_code | Categorical | Identifies the key by identifying the keycode associated with the key(using windows keyboard keycode). |
| Shift | Categorical={“True”,”False”} | Identifies whether the key ‘Shift’ has been pressed |
| Alt | Categorical={“True”,”False”} | Identifies whether the key ‘Alt’ has been pressed |
| Control | Categorical={“True”,”False”} | Identifies whether the key ‘Control’ has been pressed |
| Time | Time | Identifies the time stamp at which the keystroke occurred |

* **Pre-processing techniques:**

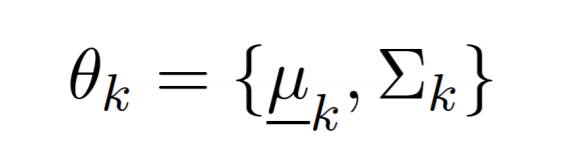
We have decided to use Gaussian Mixture Models to authenticate a user. A lot of feature engineering is required to make the dataset suitable for GMM.

* Feature Extraction:

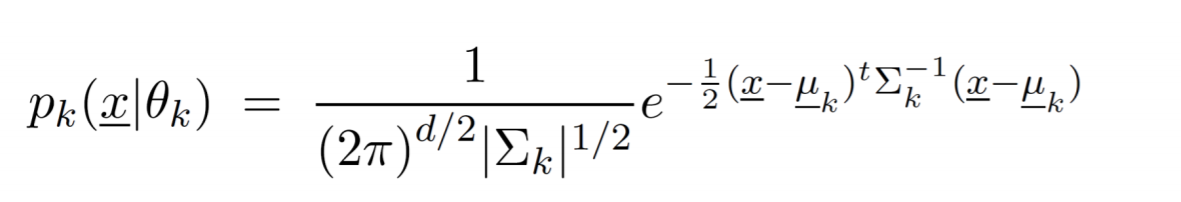
Features play an important role in predictive models. Even a complex model might perform miserably when the dataset contains unnecessary features. We have analysed the dataset and worked on the following:

* Analysis of attributes.
* Accepted login data from five users in .txt format.
* Converting categorical attributes to Boolean attributes.
* Extract data from .txt files and convert them to .csv files.
* Rename the csv files with corresponding names by adding an extra column
* Combine all .csv files into one single file
* **Your proposed solution and methods:**

The main idea is to form clusters using the training data and have each cluster denote the five users. The GMM generates a classifier which in turn is used to test the given user. A GMM is defined by making each “k-th” component a Gaussian density with parameters:



And calculating probability as follows:



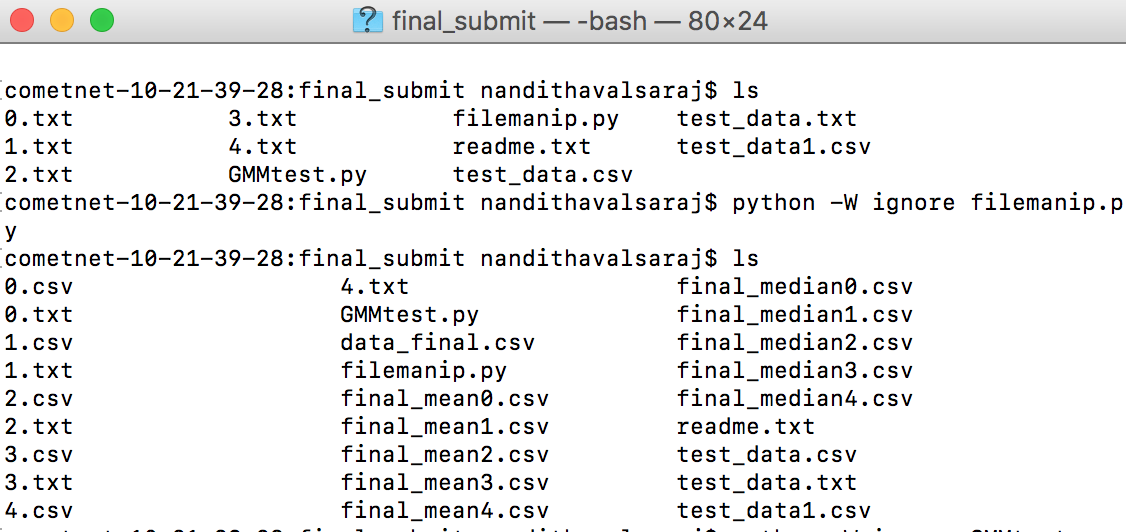
For training data, the class labels are separated from the other attributes. These other attributes are now passed to GMM to form clusters and predict the class label.

For testing data, the GMM uses the classifier to predict the user(class) label.

It also outputs the probability of test data belonging to that label.

* **Experimental results and analysis:**
* Methodology:

The mean and median files of each user’s text data are created after feature extraction. The text data is converted to csv data and the mean and median of each cluster is calculated.

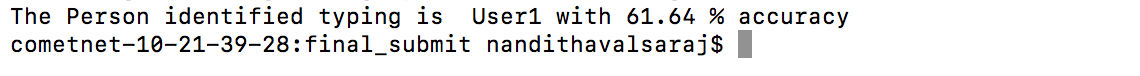


* Results:

We execute the GMMtest.py in order to identify the user.

C:\Users\FARHEEN\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screen Shot 2017-04-30 at 9.54.28 PM.PNG

The algorithm tries to find which cluster the user belongs to and outputs the corresponding label, i.e.,it calculates the probability of the user belonging to each cluster and outputs the label with highest probability. The following output is obtained:



The algorithm identifies the user as User1 with the probability 61.64%

* **Future Work:**

Although the proposed GMM approach is only applied to keystroke dynamics using static text, it can be easily extended to the free text use cases.

The future research should work toward making a large set of keystroke datasets available to the research community, investigating the more challenging problem of keystroke biometrics using free text, developing richer keystroke features, studying context-dependent subword and across-word models, seamlessly integrating language model score, that is, the authorship, into the keystroke dynamic system, and mitigating the effect of different hardware and network delay for remote-access applications.

* **Conclusion:**

We can use may other machine learning algorithms, such as K-means, DBSCAN etc., on such keystroke dynamics data sets.

The key drawback of [DBSCAN](https://en.wikipedia.org/wiki/DBSCAN) is that they expect some kind of density drop to detect cluster borders. With K-means clustering it is difficult to predict the k value.

Out of all the algorithms we had applied, GMM is the best suited for this data set, since each cluster can have unconstrained covariance structure.

* **References:**
* <http://www.cs.columbia.edu/4180/hw/keystroke.pdf>
* Keystroke Dynamics User Authentication Based on Gaussian Mixture Model and Deep Belief Nets- Yunbin Deng and Yu Zhong