Machine Intelligence and Expert Systems
EC60091 Autumn '21
Term Project

Group II
17EC34003 Manideep Mamindlapally
17EC32004 Digvijay Anand
17EC32005 Gautam Jha
17EC35036 Debdut Mandal
17EC35023 Shubham Somnath Sahoo

The following is a report of the analysis and observations of the term project for the Machine Intelligence and Expert Systems [EC60091] course at IIT Kharagpur. We sincerely thank the course instructor Prof. Sudipta Mukhopadhyay and the other teaching assistants who have given us this valuable opportunity and guided us throughout the semester.

1 Problem Statement and Objective

1. To collect, acquire and build a dataset of `Keystroke` dynamics data of different laptop users.
2. Extract relevant features from the data and process it by performing a dimensionality reduction using **Kohonen Self Organising Maps** a.k.a SOMs.
3. Use the reduced dimension features to perform an unsupervised classification(clustering) that authenticates laptop users according to their `Keystroke` behaviours.

2 Analysis of the Problem

The motivation for our problem lies in an increasing need for biometric authentication techniques in today's technology usage. Unlike traditional methods like fingerprint sensor, or face id, which use body features that are seemingly getting increasingly forgeable, people's behavioural patterns are difficult to replicate. For a laptop user, his keyboard usage is one such biometric behavioural feature that is worth exploiting.

Please refer the `README.md` *attached in the repository for explanation of the python script.*

2.1 Data Acquisition

As mentioned earlier we focus on collecting `KeyStroke` dynamics data for the purpose of this problem. More specifically we are interested in mathematically quantifying the hold time of pressing each key and latency between pressing consecutive keys.

- We acquired data through a `CONTINUOUS DATA COLLECTION tool` [1] that collects all the key press time stamps while running in the laptop background.
- A total of nine[2] students' data has been acquired over laptop usage of four weeks

[1] Thanks to the TA Bijaylaxmi Das for providing us with this and other resourses.
[2] Thanks to Group III students for providing with some of their data

2.2 Feature Extraction and Data Processing

As per our problem requirements we are interested in the parameters `holdtime` of a key and `latency` in moving from one key to the other. So,

1. First, we extract `holdtime` and `latency` features by iterating through all the `Keylog.txt` files corresponding to each user by visiting corresponding directories.
2. Then, we remove all entries with `holdtime` and `latency` beyond a `threshold` parameter. This is done to get rid of data that was collected when the user was dormant(or away from computer).
3. The data is then split into an evenly sized `splitsize` number of data features each.
4. The alphabet is determined for the data. A feature vector is constructed for each sample by averaging over the feature(holdtime/latency) of that data entry.
5. Then all the vectors are normalised and shuffled uniformly at random.

2.3 SOM dimensionality reduction

The Kohonen Self Organising Map is a dimensionality reduction algorithm. As explained in the original paper by Kohonen [1] and later in [2], the description of an SOM is as follows. For input dimension of m, we define a Kohonen layer of $n \times n$ number of neurons arranged in a 2D Euclidean space. Each neuron (p, q) $\forall p, q \in [n]$ in this Kohonen layer is fully connected to each of the input nodes a $\forall a \in A$ by weights W_{pqa}.

The training of an SOM is governed by hyperparameters: initial learning rate η_0, learning rate decay time τ_η, initial neighbourhood size σ_0, neighbourhood size decay rate τ_σ. For every epoch t of training they are updated to learning rate η and neighbourhood size σ respectively. They both undergo exponential decay.

$$\eta = \eta_0 \exp\left(-\frac{t}{\tau_\eta}\right)$$

$$\sigma = \sigma_0 \exp\left(-\frac{t}{\tau_\sigma}\right)$$

In training, at each step for input vector $(x_1, x_2, ...x_m)$ with counts $(c_1, c_2, ...c_m)$ there are key steps.

1. **Competition :** A neuron with weights closest to the input vector is determined. We call that the winning neuron $(\tilde{p}, \tilde{q})$

$$(\tilde{p}, \tilde{q}) = \arg\min_{p,q} \sum_{a:c_a>0} (x_a - W_{pqa})^2$$

2. **Cooperation :** A neighbourhood region around the winning region is determined and weights would be updated significantly only in that region. This neighbourhood is quantified by Topopogical neighbourhood parameter T_{pq}

$$T_{pq} = \exp(-\frac{(p-\tilde{p})^2 + (q-\tilde{q})^2}{2\sigma^2})$$

3. **Update :** As per the neighbourhood weights obtained, weight parameters are obtained by significant amounts only in the neighbouring regions. The weight connecting neuron (p, q) and input node a is updated as

$$W_{pqa} \leftarrow W_{pqa} + \eta * T_{pq} * (x_a - W_{pqa})$$

2.4 Clustering and Authenticating users

After training an SOM, for the input data we have we find all the winning neurons and then arrange them into clusters using a simple `k-means` algorithm. Accordingly a cluster can be seen to be associated with each user and hence can be authenticated.

2.5 Choice of the parameters and hypermarameters

- Data parameters : `splitsize` $= 400$, `threshold` $= 1sec$
- Model parameters : $n = 15$, $m = 3846$(size of alphabet)
- Hyper parameters : `num_epochs` $= 1000$, $\eta_0 = 1$, $\sigma_0 = 8$, $\tau_\eta = 500$, $\tau_\sigma = 650$.

3 Results and Observations

These are our results

Plots

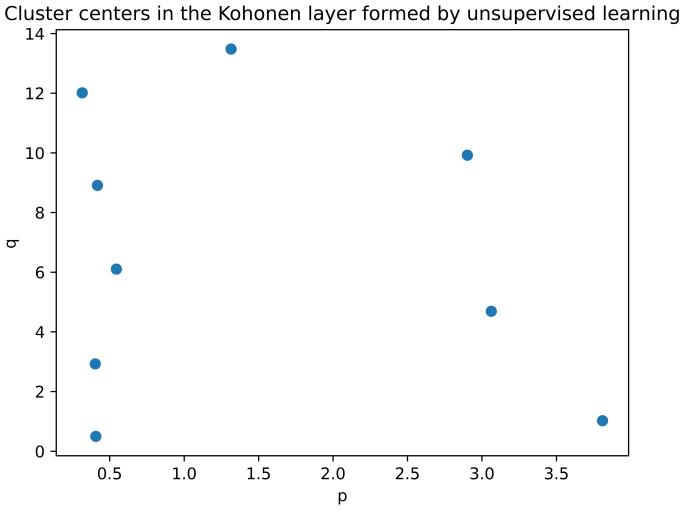

Figure 1: SOM output winning neurons cluster centers in Kohonen layer

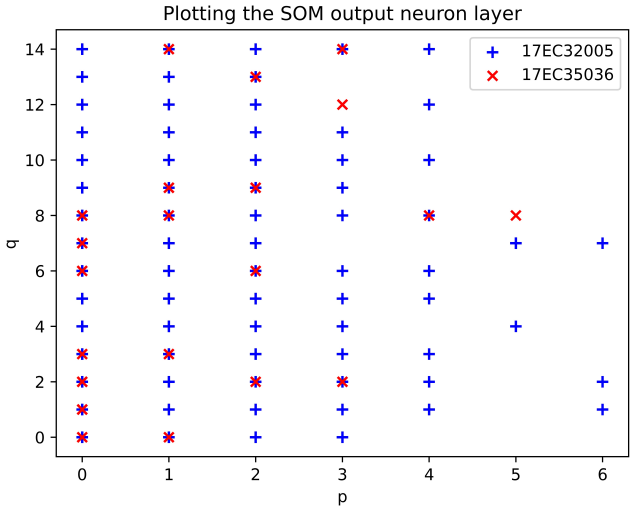

Figure 2(a): SOM output winning neurons for data from the above roll numbers

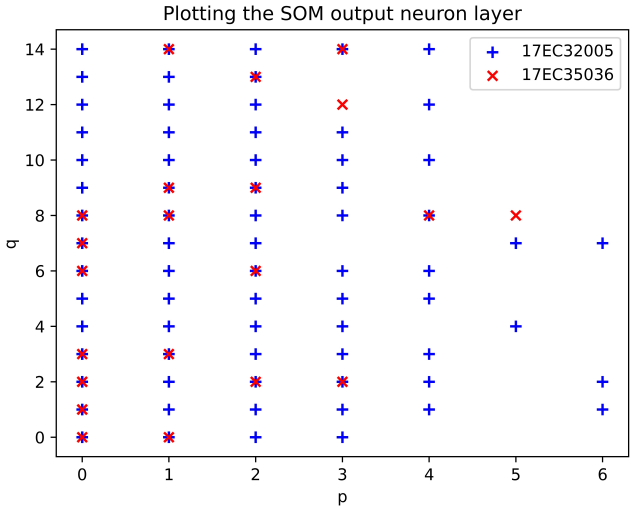

Figure 2(b): SOM output winning neurons for data from the above roll numbers

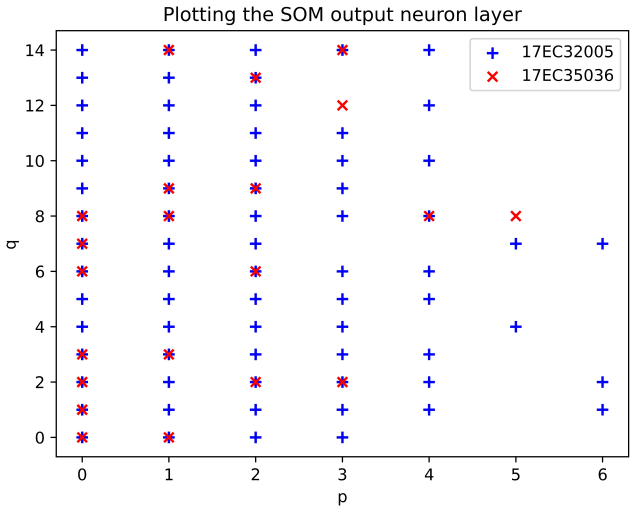

Figure 2(c): SOM output winning neurons for data from the above roll numbers

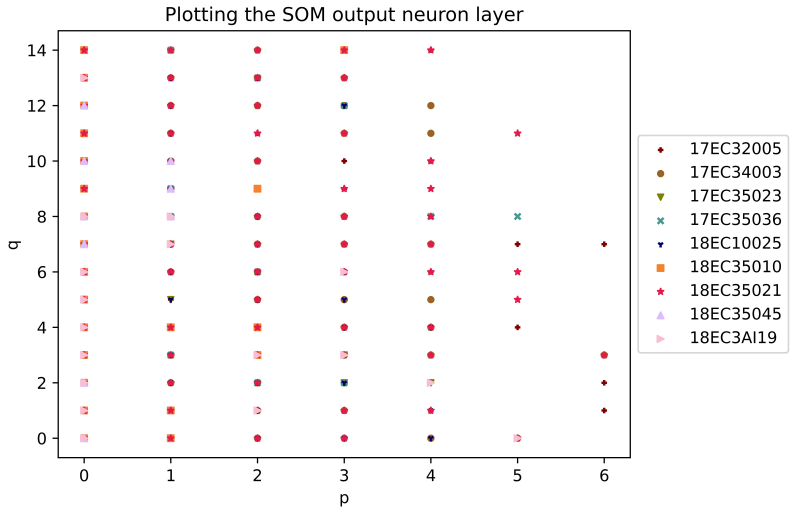

Figure 3: SOM output winning neurons for data from the above roll numbers

Performance metrics

Here is the `accuracies` matrix which shows the number of user data allotted to each class/cluster by the classification problem.

Clusters	c0	c1	c2	c3	c4	c5	c6	c7	c8
17EC32005	46	15	81	26	20	38	46	16	23
17EC34003	57	9	62	25	19	39	36	11	20
17EC35023	2	2	2	0	1	1	4	0	1
17EC35036	6	1	6	0	3	6	3	5	0
18EC10025	28	5	29	11	6	11	18	3	11
18EC35010	34	0	27	6	4	9	16	1	2
18EC35021	58	8	46	22	20	22	29	15	21
18EC35045	2	0	1	1	0	5	1	0	0
18EC3AI19	3	2	4	1	0	3	3	0	2

Here is a corresponding heatmap of the same accuracies normalised by the row.

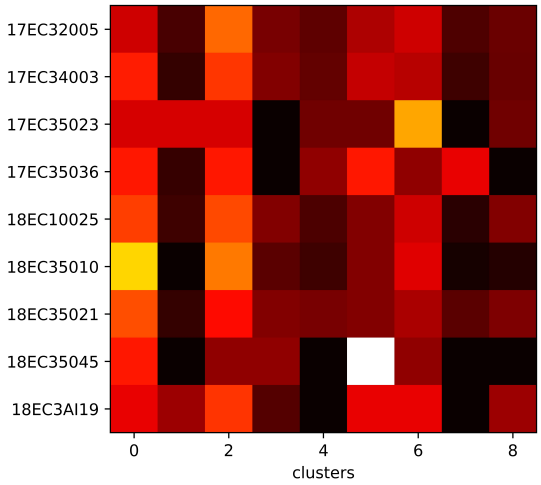

Figure 4: Heatmap of row-normalised accuracy matrix

4 Analysis and Discussions:

- It can be seen that while all the Roll numbers/users couldn't be clubbed into a clear cluster, some of them were. The model is not perfect and it can be attributed to several reasons. One of them being the data not being sufficient in size or diversity. The same data collection over a longer period of time and over students from more different backgrounds should be able to bring more variations to the table. Even from the data plotted in figures 1, we can see or maybe speculate that engineering students more or less have a similar keyboard usage trend. The Kohonen neurons don't seem to have a visible classification boundary.

- While the model is not perfect, it does still cluster some of the users very well. From heatmap in figure 4, we can see that Roll nos '18EC35045', '18EC35010', '17EC35023' can be classified comfortably well. Although the behaviours of other students seems to overlap just too much to be distinguished by the model.

References

[1] Teuvo Kohonen. Self-organizing maps: ophmization approaches. In *Artificial neural networks*, pages 981–990. Elsevier, 1991.

[2] Teuvo Kohonen. Exploration of very large databases by self-organizing maps. In *Proceedings of international conference on neural networks (icnn'97)*, volume 1, pages PL1–PL6. IEEE, 1997.