Report 2: Detect activities

Mohammad Eftekhari Pour, s307774

ICT for Health attended in A.Y. 2022/2023

January15th 2023

1 introduction

In fitness, people would like to measure automatically the number of calories lost during activities that they do. The purpose of this lab is to detect the activity (one among 19). 3 sensors per position are placed in 5 parts of 8 subjects' bodies: torso, right arm, right leg, left arm, and left leg. The frequency of these sensors is 25 Hz from 3-axis accelerometers, gyroscopes, and magnetometers, 3 sensors per position, each measuring 3 values on the x-y-z coordinates, a total of 45 values for each sample. Each subject practiced the activity for 5 minutes and slices/files of 5 seconds were created. I use 19 activities and 14 sensors in this lab, these are Activities that 8 subjects have done.

1- sitting	6 - descending stairs	11- walking on a treadmill.4km/h.15deg	16- cycling on an exercise bike in vertical positions
2- standing	7- standing in an elevator still	12- running on a treadmill.8 km/h	17- rowing
3- lying on back	8- moving around in an elevator	13- exercising on a stepper	18- jumping
4- lying on right side	9- walking in a parking lot	14- exercising on a cross trainer	19- playing basketball
5- ascending stairs	10- walking on a treadmill.4 km/h.flat	15- cycling on an exercise bike in horizontal positions	

Table 1 - Activities

2 K-means algorithm

The K-means algorithm is a method for clustering data points into a specified number K of clusters. It works by iteratively assigning each data point to the cluster whose centroid is closest to it, and then recalculating the centroid of each cluster based on the data points assigned to it. The algorithm continues until the assignments of data points to clusters no longer change.

The stages of the K-means algorithm:

1. $intially\ vectors\ x_k^{(0)}, k = 1, \ldots, K\ are\ randomly\ generated. set\ i = 0$
2. At the i-th step, for k∈[1, K], all the points y(n) which are closer to x (i) k than to the other points $x_h^{(i)}$ are given to cluster k (assignment step or maximization step):

   $$C_k\,(i) = \left\{y(n), n = 1, \ldots, N: \left\|y(n) - x_k^{(i)}\right\| \leq \left\|y(n) - x_h^{(i)}\right\|, \forall h \neq k\right\}. \qquad (1)$$

3. Evaluate the mean value $\boldsymbol{m}_k^{(i)}$ of the points y(n) that have been associated with $x_k^{(i)}$ (update step or expectation step):

   $$\boldsymbol{m}_k(\mathrm{i}) = \frac{1}{|C_k(i)|} \sum\nolimits_{y(n)\in c_k(i)} y(n). \qquad (2)$$

   Where $|C_k(i)|$is the cardinality of $C_k(i)$ (i.e. number of points in the set).

4. Define$x_k^{(i+1)} = \boldsymbol{m}_k^{(i)}$, we set i: = i+1, we go back to step 2 until the convergence of the algorithm.

In stage 1, $x_k^{(0)}$ is the number of points in step zero. $x_1^{(0)}, x_2^{(0)}, \ldots, x_k^{(0)}$. $y(n)$ in formula (1) is the set of points each of rows in our matrix and it is a vector. The meaning of the convergence in the 4 stage is, if the point associated to the given cluster do not change from iteration (i) to iteration $(i + 1)$.

3 centroid and standard deviation

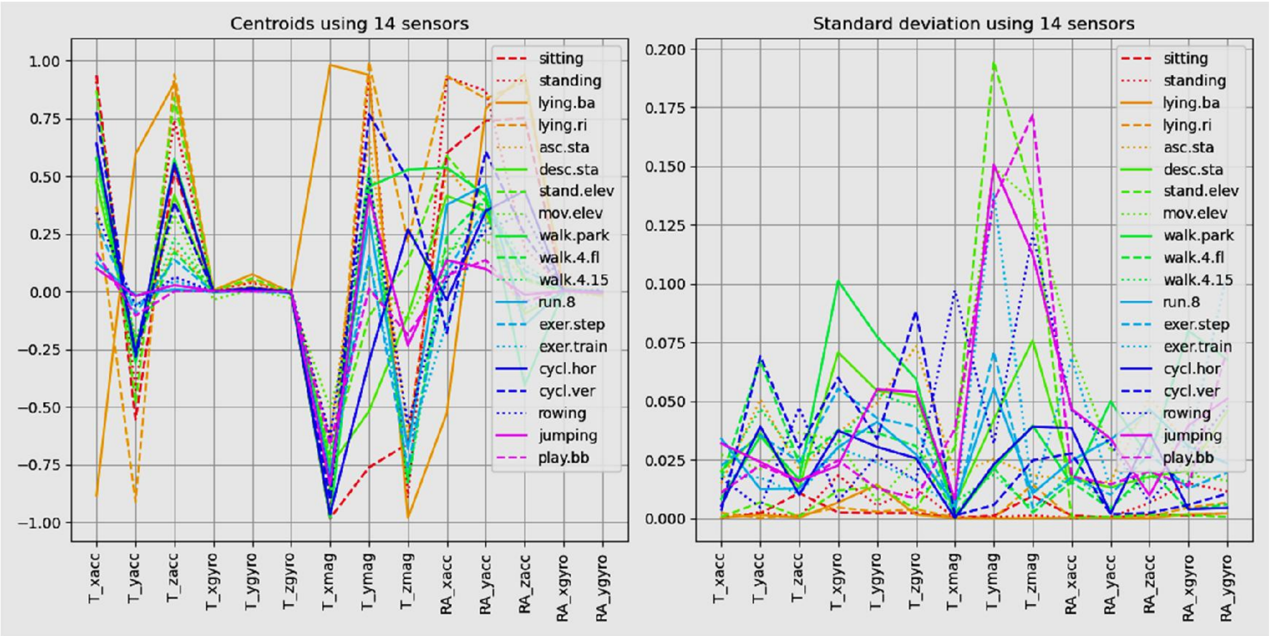

Figure 1 - centroid and standard deviation of 14 sensors

the standard deviation can use be to measure the spread or variability of the data, with a smaller standard deviation indicating that the data points are closer to the mean and a larger standard deviation indicating that the data points are more dispersed. In figure (1), we can see that the standard deviation of 14 sensors is between 0.0 to 0.2, and it represents that our data closer to the mean.

4 minimum centroid distance, mean distance from points to centroid

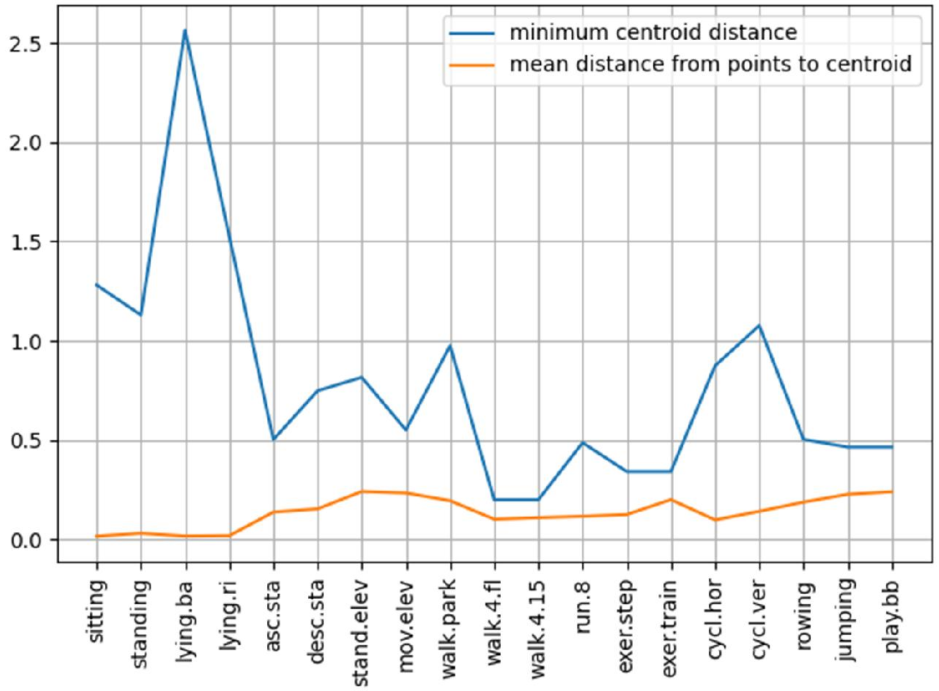

Figure 2 – distance

We have two model clustering desired case and undesired case. The desired case occurs when clusters are far apart and points are close to their centroids. On the other hand, the undesired case occurs when points are very spread and distant from centroids. As we can see in figure (2) the minimum centroid distance (the blue line) is above of the mean distance from points to centroid (the orange line), it means that we are in a desired case. To achieve this at fist I standardized the data and then I changed the inputs, I use all 19 activities and 14 sensors.

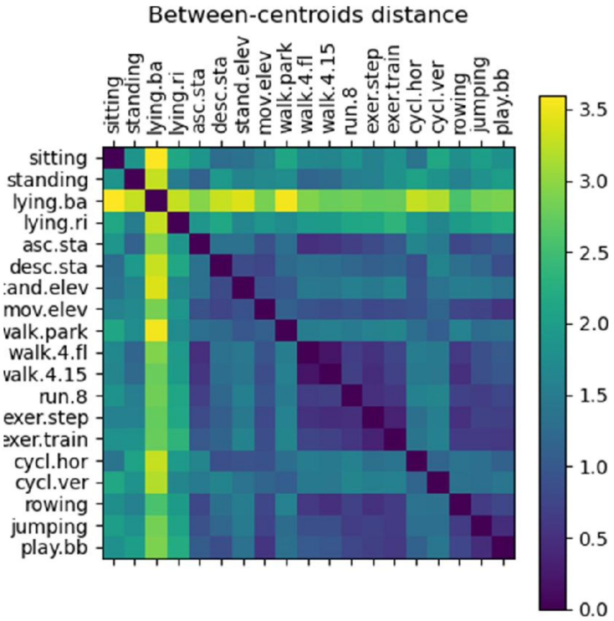

Figure 3 – correlation

5 training and test slice

In this data set, each subject practiced the activity for 5 minutes and slices/files of 5 seconds were created (each slice/file contains 125 rows and 45 columns). The number of total slices are 60. So I separate them in two part equally, 30 slices are for data training and the other 30 slices are for test data. So my training and test dataset have (30×125×19) rows.

6 Mapping and remap

The k-means algorithm allocates the index of the cluster randomly, that means for instance activity 1 corresponds to cluster 10. To detect this first we use the mapping function, the input of the mapping function is the output of the k-means algorithm and is divided by the actions (19 activities). This function detects how k-means assigned the index to the cluster and the remap function remaps indexes to activities.

7 The accuracy and confusion matrix

In this lab we use a clustering algorithm k-means to solve a classification problem. The confusion matrix is used to define a performance of classification algorithm, each row of the matrix represents the instances in a predicted class while each column represents the instances in an activities in here (or vice versa). Figure 4 represent the confusion matrix of our model, in the right side we can see confusion matrix for the training data set and for the test dataset in the left side. I use the *confusion_matrix()* function from the Scikit-Learn library.

Accuracy is a simple and straight measure to calculate the quality of an algorithm. For this I use *accuracy_score()* in the Scikit-Learn library to calculate the accuracy, and I got this number.

Accuracy on test set:

0.7894736842105263

Accuracy on train set:

0.7894736842105263

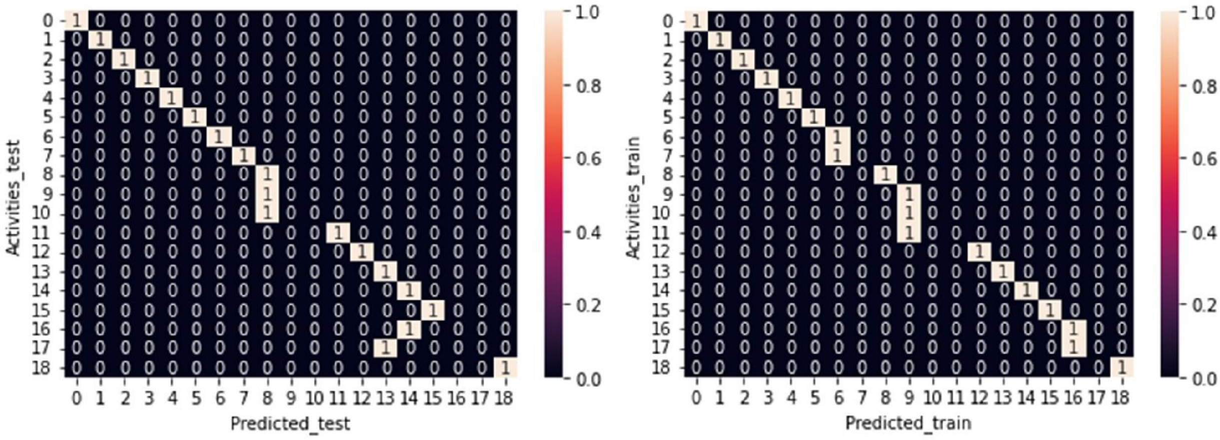

Figure 4 – confusion matrix(test and train dataset)

References

https://archive.ics.uci.edu/ml/datasets/Daily+and+Sports+Activities