Clustering Final Project

Author:
Maria Jose Rueda Montes

Supervisor:
Kerdoncuff Tanguy
Habrard Amaury

***Index Terms*—Clustering Techniques, Clustering applications, Data preprocessing, Unsupervised algorithm.**

1 Introduction

Clustering is a data analysis technique which is responsible for the unsupervised classification of patterns to group the data in different groups (clusters), depending on an established measure of similarity. Data belonging to the same cluster has the same label. In this way, the data with the same label has a greater degree of proximity to each other than to the rest of the data. Therefore, we will have a high clustering quality when the intra-class similarity of the data is high, while the inter-class similarity must be low. The measure of the similarity that excludes or includes the data to a certain cluster, and the way in which the data is associated, defines the different clustering algorithms.

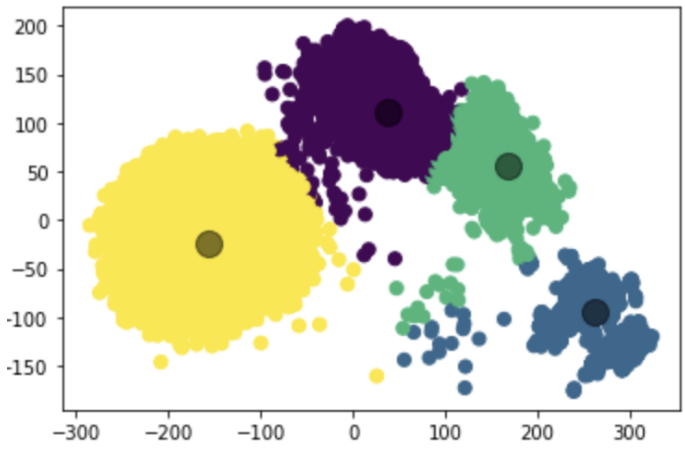


Figure 1: K-Means clustering example. Each group of data is inside a cluster.

There are two different categories dedicated to data analysis, unsupervised and supervised classification. In supervised classification, the collection of training data (which are used to learn the descriptions of classes) are associated by their respective

labels. With this information, new unlabeled data provided is labeled. On the other hand, clustering belongs to unsupervised classification, where for this method the data labels are not available. In this way, the data is grouped into clusters where each cluster is associated with a label, and the information to group the data in clusters comes only from the information that can be extracted from the data, not from external labels, unlike supervised classification [1].

In general, clustering is widely used in statistical data analysis. Using clustering we can organize and synthesize data, identify the degree of similarity and study the interrelationships between data points, identify outliers, and extract information to generate hypotheses [2].

Within clustering we can find two categories, hard clustering and soft clustering. These two ways of grouping data differ in the number of clusters that can be associated with each data point. In hard clustering method each data point belong to a single cluster, while for soft clustering a data point can belong simultaneously to more than one cluster [2].

In this paper we are going to present clustering, some of its most important properties and applications. It is intended to address the issue of clustering in an explanatory and illustrative way. For this, the Motion-Capture-Based Hand Gesture Recognition database is used to which different clustering algorithms, K-means, Agglomerative, DBSCAN and Mean-Shift, implemented in Python. On the other hand, we are going to study the performance after adding some labels to the data, in order to deal with the differences with a supervised algorithm. Finally, we conclude with the discussion of the topic covered and some future perspectives.

2 Clustering Algorithms

Since clustering is used for different purposes, there are more than 100 clustering algorithms that adapt to the needs of each task [3]. In this section we explain 3 of the most important categories into which these algorithms are divided, whose main difference comes from the methodology used to assign a data point within a cluster: partitional algorithms, hierarchical clustering, and density based clustering. Within each category, the examples of algorithms that we have implemented for clustering will be explained.

2.1 Partitional Clustering

Partitional algorithms are iterative methods in which a single partition of the data is obtained. For this, it must be respected that each cluster must have at least one data point, and that each data point must belong to at least one cluster [2]. One of the advantages of partitional algorithms is that they involve less computational cost and are therefore useful for large databases [1]. However, these algorithms have the disadvantage that they are know-order algorithms, where the cluster number k has to be previously set. As can be guessed, the disadvantage is the consequence that choosing a different number of clusters produces different results. In this way the data is divided into the number of clusters used as input parameter according to a set of rules [4]. Besides partitioning algorithms are clustering algorithms working by function optimization. The algorithm find a partition of k clusters that optimizes

the chosen partitioning criterion, either globally or locally.

In this category we can find the algorithms: K-Means, K-Medoids, K-Modes, PAM, CLARANAS, CLARA, FCM, FCMdC, Fanny, etc [2].

2.1.1 K-means

K-means algorithm, also known as Lloyd's algorithm, is very popular in clustering, it consists of dividing a set of n samples X into k clusters C, of equal variance by minimizing the total distance between data points and their assigned cluster's centroid [4]. In this way, k-means selects the centers that minimize the following criterion [5].

$$\sum_{i=0}^{n} \min_{\mu_j \in C} (\|x_i - \mu_j\|^2), \tag{1}$$

where μ_j is the mean of the samples in each cluster C.

K-means algorithms is made up of a series of fundamental steps. First, the centers of each cluster must be chosen, as well as their number k. In the next step, each data point is included in the closest center. As third step, the average of the distances between all the samples of each cluster with its corresponding center is calculated, and the position of said center is updated based on this average. These last two steps are repeated in a loop until the centers do not change position, or move less than a set threshold [5].

This algorithm scales well on long databases, but has several drawbacks [5]. K-means obtains poor results for data groups that are not convex and is unable to handle noisy data and outliers. On the other hand, the criterion used is not a normalized metric, this implies that in high-dimensional spaces the Euclidean distances appear to be greater than they should. The use of PCA in the data to reduce its dimensionality, before applying k-means, prevents this last problem [5].

2.2 Hierarchical Clustering

There are two different approaches of hierarchical clustering, agglomerative and divisive. In the agglomerative type, each data point begins as part of it own cluster, therefore it begins with the same number of clusters as data points. In the following steps, the clusters will merge until they form a single cluster that includes all the data points. On the contrary, in divisive algorithms, the algorithm begins with a cluster that contains all the data points, and this one is divided until each point belongs to a different cluster [4]. The hierarchical algorithm process is represented by a denogram. This denogram shows the sequence of clusterings which are produced [1].

Within the different hierarchical clustering algorithms, there are two widely used variants, single-link and complete-link algorithms. For single-link the distance between clusters to be merged is calculated taking into account the minimum distance between the samples in two clusters. In the complete-link algorithm, the distance between two clusters is calculated as the maximum of the distances between the data points that belong to two clusters. Either in single-link or complete-link algorithm, the

clusters that meet the distance criterion are merged into a new cluster according to the minimum distance between them. As general characteristics, complete-link generates more compact clusters, while single-link produces elongated or straggly clusters, although it is more versatile compared to the complete-link algorithm [1].

In this category we can find the algorithms: BIRCH, CURE, ROCK, Chameleon, Echidna, Diana, Agnes, etc [2].

2.2.1 Agglomerative Single-Link Clustering Algorithm

In the first step of the Agglomerative single-link algorithm, each data point is included in an independent cluster, to later create an ascending list with the distances between each pair of clusters [1]. To calculate the distance between clusters we can use different metrics, that we can set as parameter, such as Euclidean distance, Squared Euclidean distance, Manhattan distance, Maximum distance or Mahalanobis distance [3]. In the next step, the list of distances formed in the first step is used to construct the denogram in which the most similar cluster pairs are connected, until all the clusters appear converge in an unic cluster in the upper part of the denogram. In this way, the distance between two clusters can be interpreted in the denogram by the height at which two clusters merge [3]. The denogram obtained as output can be cut at a dissimilarity level, where we can find different number of clusters depending on the chosen level [1].

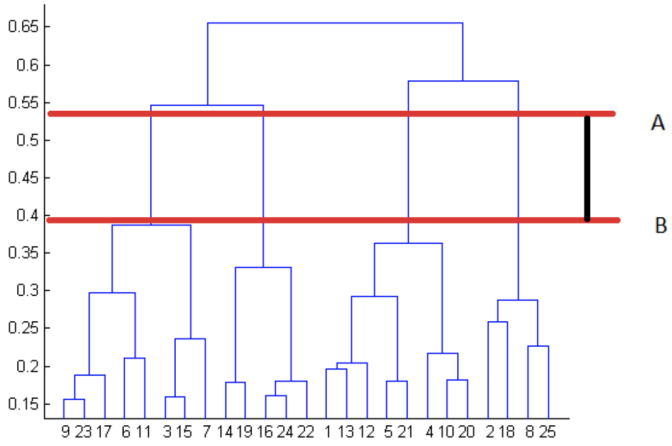


Figure 2: How to choose the number of clusters based on the dendrogram. In this case is 4 the optimal number of clusters. AB represents the maximal vertical distance.

Source: [6]

Since we can find different number of clusters depending on the height at which the denogram is cut, it is difficult to interpret which is the best cluster number for a given data. For example, one of the ways to know which is the best number of clusters is by choosing the maximum height in the denogram at which no new cluster is formed, and draw a horizontal line that counts the number of clusters (Figure 2) [3].

On the other hand, agglomeraive algorithm has more computational cost than partitional algorithms (Section 2.1), so dealing with big data will take considerably more time. However, if we want to use an agglomerative algorithm, it is preferable

to choose single-link before complete-link in order to decrease the space complexity. Additionally, single-link works well with non-isotropic clusters such as well-separated, chain-like, and concentric clusters [1].

2.3 Density Based Clustering

Density Based algorithms use the density distribution of the data points in the data space as a criterion for cluster formation. Continuous low density data points are considered outliers or noise data and their function in this method is to delimit the separation between clusters [7]. In this way, data points as outliers are not included in the clusters, which is one of the advantages of this method. Density based algorithm selects random data points and analyses the density of its neighbours. This process is repeated for all the data, at the points where the density has not been checked. Finally, a cluster will be generated when the density of the points in a particular distance is sufficient, otherwise it will be classified as an outlier point [2].

In density based methods it is not necessary to enter the number of clusters, in addition, the densities of each cluster, or the variance that exists within each cluster, are not assumed. Thus, we can obtain clusters with different shapes, not necessarily convex [7].

In this category we can find the algorithms: DBSCAN, OPTICS, DBCLASD, DENCLUE, Spectral Method, Subtractive method, Mean Shift, etc [2].

2.3.1 DBSCAN Algorithm

One of the most important components of the DBSCAN algorithm is the concept of core samples. Core samples are the data points found in high-density areas. In this way the clusters are made up of core samples and their neighbors [5]. Given a distance threshold ε and a density threshold $minPts$, the density of a data point is given by the number of neighboring points k within the radius ε. A point is considered a core point when the density of the data point is greater than the density of the threshold $minPts$. Thus, two points are connected when the distance between them is less than ε and they will be density connected when they connect to core points that are density connected to each other, that is, they will form a cluster [7]. Data points that are non-core samples and are more than ε away from a core sample, are classified as outliers by the algorithm.

We can deduce the parameter $minPts$ must be greater for large or noisy databases, since this parameter mainly controls the tolerance to noise. Regarding the parameter ε, it controls the local neighbors of each data point, if it is chosen too small it will classify the points as noise, while if it is chosen too large it will only generate a cluster. In general, a higher value for the $minPts$ parameter, and a lower value for ε, will indicate that higher density is needed to form a cluster [5].

2.3.2 Mean-Shift Algorithm

Mean-Shift is a non-parametric method which assigns as centroids the regions of highest density in the d-dimensional feature space. On each data point a gradient

ascent algorithm is applied on the local estimated density until it reaches the convergence, thus updating the centroid candidates [8]. The centroid candidate x_i is uptaded for the interation t as following

$$x_i^{t+1} = v(x_i^t), \tag{2}$$

where v is the mean shift vector calculated for each centroid candidate and $N(x_i)$ are the neighboring data points of x_i. In this way, after calculating the mean shif vector v, to update the centroids the algorithm use the following equation [5]

$$v(x_i) = \frac{\sum_{x_j \in N(x_i)} K(x_j - x_i) x_j}{\sum_{x_j \in N(x_i)} K(x_j - x_i)}, \tag{3}$$

where K determine the weighted mean of the density in the correspondent window. The centroids are updated to be the mean of the neighboring data points [5].

In this algorithm the shape of the distribution is not assumed and the number of clusters does not have to be specified. However, the bandwidth parameter that determines the size of the region in which the algorithm will move is unspecified which implies a limitation [8].

2.4 Applications

The algorithms explained in the previous section have been successfully applied in a large number of fields to analyze data.

K-means algorithm is applied on: market analysis; web search; medical image analysis, such as recognition of tumors; drawing patterns in social media; improving the data transfer speed and its security; or in bank frauds by broadcasting warning messages. Hierarchical clustering is more frequent for: education field; image segmentation by using hierarchy in segments; behavioral malware clustering; reducing energy consumption in sensor networks; web search engines; and obtaining patterns in social media. DBSCAN is recognized in the fields of: network traffic management; profit maximization and clientele expansion; finding a strategic location for ATMs installation; or in anti money laundering regulatory systems. By last, Mean-Shift algorithm is widely used in: blood oxygen level dependent functional MRI activation detection; image processing techniques; and in Medical image analysis field. [2]

3 Methodology

3.1 Data Set

3.1.1 Data Set Description

The clustering task will be performed on the Posture database provided by Andrew Gardner [9]. This dataset is made up of five hand postures from 12 different users, which is collected using a glove with fifteen markers attached. We can find some missing values due to the resolution of the capture volume and self-occlusion derived

from the different postures of the hand. We also find strange markers and that vary constantly given the way in which the data is captured [9].

The data format is CSV file, where the header shows the name of the attributes. Each value in the file corresponds to a frame collected by the capture system. Missing values are represented by the '?' sign, and in the first row are recorded only zero values.

The attributes (columns in the CSV file) that are part of this database will be described below:

- Class: this column is composed of numbers from 1 to 5, each representing the class ID of each posture:

  1: Fist (with thumb out),

  2: Stop (hand flat),

  3: Point1 (point with pointer finger),

  4: Point2 (point with pointer and middle fingers),

  5: Grab (fingers curled as if to grab).

- User: the ID of the user from whom the data was collected.
- X0, X1,..., Xi, Y0, Y1,..., Yi, Z0, Z1,..., Zi: x-coordinate, y-coordinate and z-coordinate of the i-th unlabeled marker positions.

3.1.2 Data Set Processing

By clustering hand postures we can obtain interesting information about gestural behaviours which are essential in our daily life. The most reasonable is that we find 5 clusters more differentiated from each other, so that each hand posture is within a cluster. Another way of grouping the data would be in 14 clusters, where each cluster corresponds to one of the people from whom the data has been collected. Although this number of clusters would be the most logical, it must be taken into account that we can obtain different amounts of clusters, for example 4 clusters in which one of them represents two similar positions. We cluster the data regardless of labels.

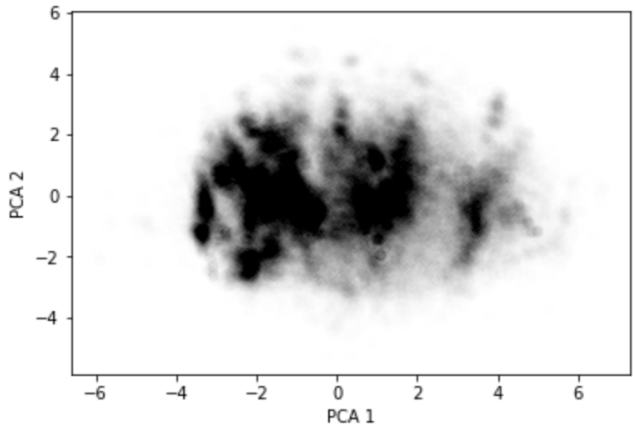


Figure 3: Representing two principal components (PCA 1 and PCA 2) we can see how two groups are formed, so at least we will be able to obtain two clusters. Although this data is going to be complicated to cluster since these two groups of data are not very different from each other.

First and before starting the clustering, we must process the data. To do this, we start by making an overview of the data, in which, as explained in the previous section (3.1.1 Data Set Description), there is a row of zeros and missing values, which is the first thing we are going to solve. After eliminating the row of zeros, and visualizing the number of missing values for each feature, we observed that the last 12 columns contain a large number of missing values, so we eliminated them. With the missing values of the rest of the features, we can choose between two methods: eliminate these rows, or replace them with a new value. Since the amount of missing values is very different between features, we choose the second method. This lack of values and how to deal with them is critical before clustering. The database has a very disproportionate quantity of missing values and also the data has large dimensions, so the method chosen to replace the data will completely change the final clustering result. After analyzing various ways to replace the values, such as eliminating the rows of missing values, performing imputation using mean or median values, imputation using zero or a constant value, or other more complex methods such as k-NN, MICE, Datawing, etc. Taking into account that we do not have a GPU, that an appropriate method has to be chosen given the complexity of the database, and the results we get have to make sense, we chose the k-NN method. k-NN or rather k nearest neighbors, specify the values of the new data points that will be replaced by the missing values using feature similarity [10].

Since we have a large number of features, to reduce the complexity of the data in each algorithm, and improve the processing speed, we implement PCA with the aim of reducing dimensionality. To know how many components to choose, we can plot the percentage of variance that each of the components represents, in this way we choose the components that best represent the data, which are the ones with a greater amount of variance [11]. Furthermore, PCA will solve the problem that occurs in high-dimensional spaces when applying euclidean distance, since these results tend to inflate [5].

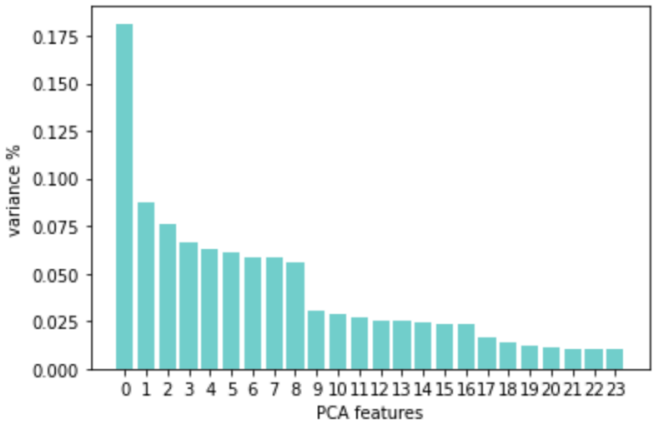


Figure 4: Variance percentage of each PCA componen.

The Figure 4 shows the first main component describes most of the information since it contains the highest percentage of variance, with an important difference with respect to the other components, for this reason it is sufficient if we choose the first three principal components to describe the data.

Before doing dimensionality reduction (PCA), we perform feature scaling through standardization. This process is important to avoid the model becoming biased to the higher magnitude variables [12]. On the other hand, it is also useful before performing PCA since this algorithm works with the components that maximize the variance, thus the variables with a greater standard deviation have more weight for the calculation of the axis, and therefore the change in these variables will be considered of greater importance. In this way, the ideal is that all the variables have the same weight, so with this objective we carry out feature scaling [13].

3.2 Evaluation Metrics

1) Homogeneity, Completeness and V-measure: these metrics take into account the labels of the samples and are based on conditional entropy analysis. The value score ranges are from 0 to 1, with one being the best mark. Homogeneity, h, evaluates that in each cluster there are data points that belong to only one class. Completeness, c, evaluates that data points belonging to one class are grouped within the same cluster. Finally V-measure, v, is the harmonic mean between Homogeneity and Completeness metrics [5].

$$h = 1 - \frac{H(C|K)}{H(C)}, \tag{4}$$

where $H(C|K)$ is the conditional entropy of the classes given the cluster, and $H(C)$ is the entropy of the classes.

$$c = 1 - \frac{H(K|C)}{H(K)}, \tag{5}$$

where $H(K|C)$ is the conditional entropy of clusters given the class, and $H(K)$ is

the entropy of the clusters.

$$v = 2 \cdot \frac{h \cdot c}{h + c}. \tag{6}$$

2) Fowlkes-Mallows scores: this metric needs the labels of the dataset and its score is given by the geometric mean of the pairwise precision and recall, from 0 to 1, where 1 is the best possible mark. Values close to 1 means that there is a good similarity between two clusters [5].

$$FMI = \frac{TP}{\sqrt{(TP + FP)(TP + FN)}}, \tag{7}$$

where TP is the total positive number of data points, FP false positives, and FN the number of false negative.

3) Silhouette Coefficient: data labels are not required in this metric. The score ranges from -1 to 1, with 1 being the best possible value indicating dense and well-separated clusters. Values close to 0 show cluster overlapping [5].

$$s = \frac{b - a}{max(a, b)}, \tag{8}$$

where a is the mean distance between a data point and the rest of the data in the same class, and b describes the distance between a data point and the rest of the data in the next nearest cluster.

3.3 Results of clustering methods

1) K-means: before running the algorithm we have to know the optimal number of clusters that we choose as input. With this objective we plot the change of inertia, i.e., the sum of the squared distances to the nearest cluster center [11]. Fig3 shows that the inertia values are relevant until reaching 4 clusters, so we keep this quantity as a parameter of the number of clusters and discard the others.

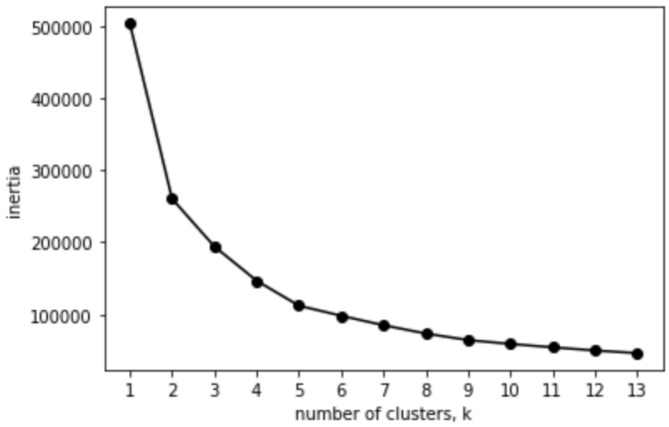


Figure 5: This plot shows how inertia varies depending on the number of clusters.

The results after clustering are shown in Table 1, whose outcome is not bad considering that the data is very complex and disorganized. We get these good marks because K-means scales well in large databases and achieves good results for convex databases like ours. Figure 6 shows that each cluster is distributed in an equitable way.

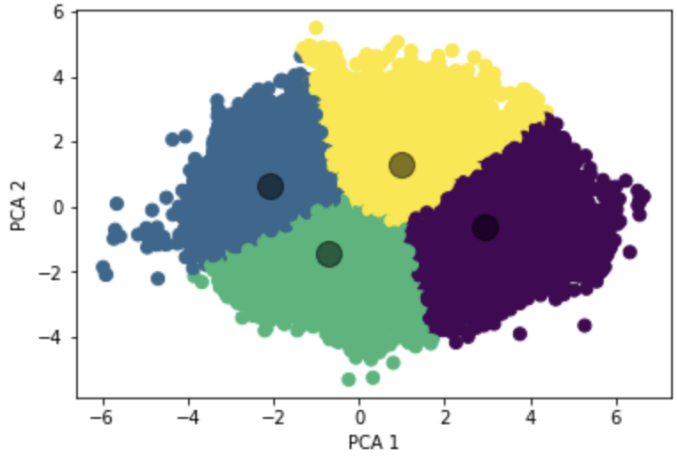
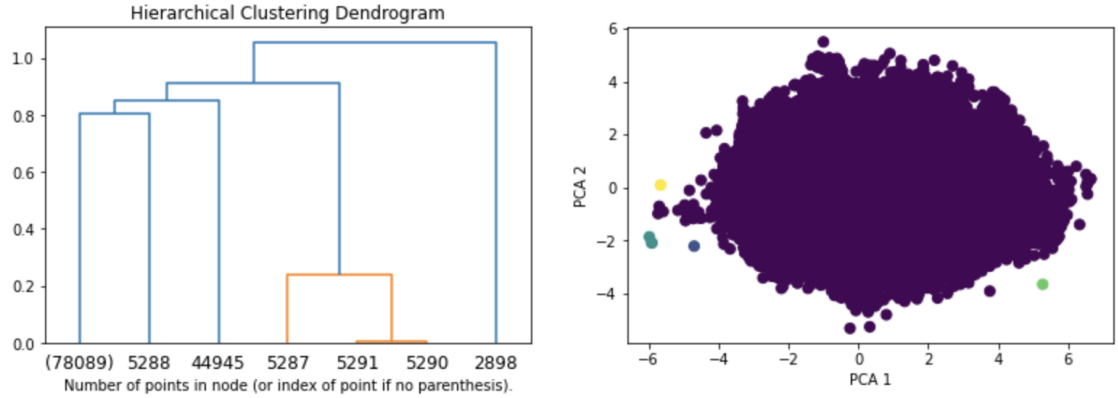


Figure 6: K-means clustering result. The black dots indicate the centers of each cluster.

2) Agglomerative Single-Link Clustering: we choose single-link approach because it decreases computational cost and we are dealing with big data, additionally because single-link works well with non-isotropic clusters. On the other hand, to know the number of clusters and choosing this parameter, we analyze the dendrogram, as explained in Section 2.2.1. After cutting the dendrogram by the appropriate dissimilarity level, we obtain 5 as the number of clusters. However, from the dendrogram and clustering plot (Figure 7) we can see how there is a cluster composed by most of the points, and the remaining four contain a single data point. These four clusters correspond with noise data points.



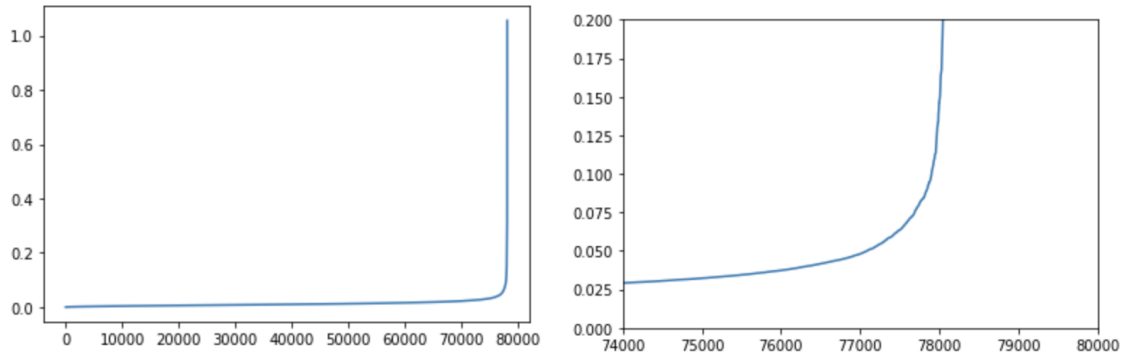
(A) (B)

Figure 7: (A) is the hierarchical clustering dendrogram using agglomerative single-link algorithm. (B) shows the clustering result after applying the algorithm with 5 as the number of clusters.

Table 1 shows very bad results. These results are expected since single linkage is the

worst method that we could choose and the simplest. Alternatively, in possession of GPU more complex strategies could be tried, such as Ward, Complete linkage or Average linkage.

3) DBSCAN: as explained in Section 2.3.1, we have to select $minPts$ and ε as input parameters in DBSCAN algorithm. In the first place, to select $minPts$, we use the criterion of [14] in which we choose as a value twice the amount corresponding to the number of features that exist in our dataset, i.e, 2×24. On the other hand, to calculate ε based on the value of $minPts$, we compute the average distance between each point and its k ($minPts$) nearest neighbors. Thus, the value of ε corresponds to the elbow that appears in the plot (Figure 8).



(A) (B)

Figure 8: (A) is the average of k-distances. (B) is the zoom of k-distance plot. This plot shows how the optimal value for the parameter ε is 0.075.

Although after executing the algorithm we obtain a total of 48 clusters, in Figure 9 we can see how there are two main clusters which are surrounded by other small clusters. We also found a large number of noise points, but this is expected as all the data is agglomerated.

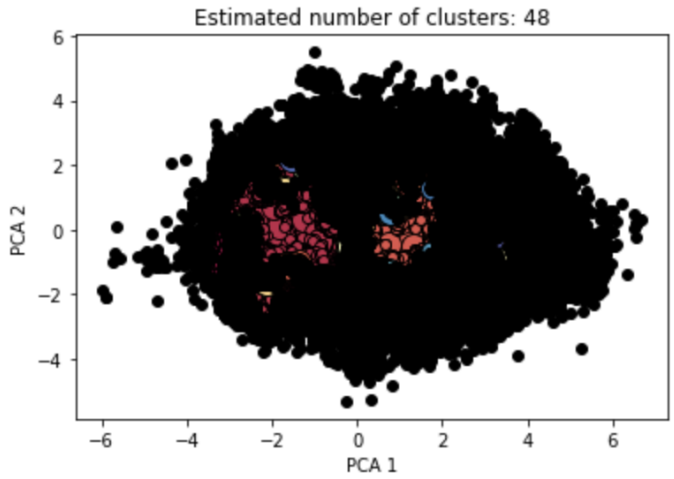


Figure 9: DBSCAN Clustering result. The black points correspond with the noise data points.

4) Mean-Shift: this algorithm depends on the bandwidth parameter. This parameter

specifies the region across which to search for high-density points in the data set. To set this parameter we use the *estimate_bandwith*() function, which estimates the bandwith based on a nearest-neighbour analysis [5].

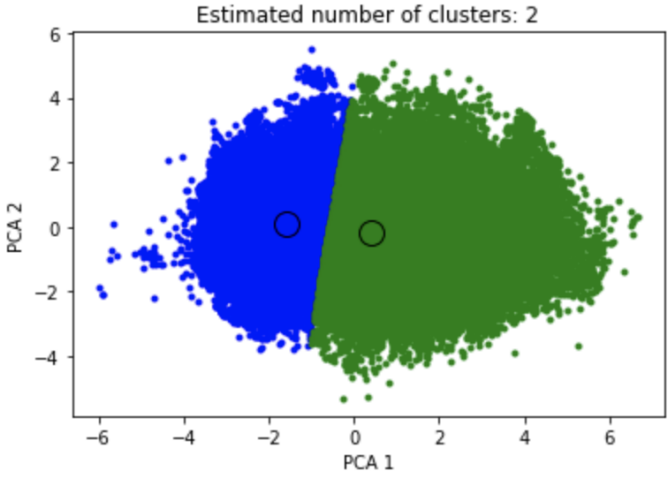


Figure 10: Mean-Shift Clustering.

As Figure 10 shows, with Mean-Shift algorithm we obtain two clusters. These results are similar to those obtained with DBSCAN, but clearer in this case. Furthermore, the results agree with Figure 3 in which we can also distinguish these two clusters found with Mean-shift.

	Homogeneity	Completeness	V-measure	FMI	Silhouette
K-means	0.226	0.265	0.244	0.389	0.353
Agglomerative Single-Link	0.000	0.140	0.000	0.448	0.210
DBSCAN	0.169	0.161	0.165	0.303	-0.555
Mean-Shift	0.157	0.368	0.220	0.449	0.400

Table 1: Quantitative comparison between clustering methods. The best mark in each metric appears in RED, and the second best mark in GREEN.

Table 1 shows the values obtained in each clustering method for the different metrics used. K-means and Mean-Shift obtain the best scores. K-means divides the dataset into four clusters, a number close to the number of classes in our dataset, where each class corresponds to a gestural posture. These results could be analysed as three postures that are distinct from each other, while the remaining two postures are similar between them, forming a single cluster. On the other hand, Mean-Shift properly identifies the two high density points, as can also be seen in Figure 9 with DBSCAN and in Figure 3. In this case the database forms two clusters, which can be related to two groups of people performing the postures in a similar way.

4 Experiments

As introduced in Section 1, the main difference between unsupervised and supervised techniques lies in the use of the labels of the data to run the algorithms. To study the difference between these two techniques, we use the same database but in this

case with a supervised learning algorithm, Support Vector Machine (SVM), with which we use some labels for training. To do this, we start by dividing the dataset into two sets, the train set that we use in SVM training and contains the labels of each data point, and the test set in which the data is passed without the labels, and the algorithm is responsible for predicting them based on what is learned in the training stage.

After measuring the accuracy we obtain a value of 0.938, this is very good result. The supervised technique, for this database, gives better results compared to the unsupervised technique, although it must be taken into account that both SVM and Clustering have two different purposes.

5 Conclusion

In this project we have treated the Posture database provided by Andrew Gardner with some of the most popular clustering algorithms, K-means, Agglomerative Hierarchical Clustering, DBSCAN and Mean-Shift. These algorithms have been exposed and studied both theoretically and in practice with the mentioned database, in order to understand in detail how they work. Some of the techniques for choosing the appropriate parameters within each algorithm have also been explained.

Before applying clustering, the database has been processed, which has proved to be a complex task as it is very large and has many missing values. The k-NN technique was applied to solve the latter problem. The results after clustering show that the best algorithms for this database would be K-means and Mean-Shift, although for example in agglomerative single-link technique it would be convenient to use a different strategy to single-link, which could not be implemented on this occasion due to lack of GPU.

On the other hand, we also study the data with a supervised algorithm, SVM, with which we obtain a good classification for this database.

Bibliography

[1] A. K. Jain, M. N. Murty and P. J. Flynn, 'Data clustering: A review', *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[2] A. Ghosal, A. Nandy, A. K. Das, S. Goswami and M. Panday, 'A short review on different clustering techniques and their applications', *Emerging technology in modelling and graphics*, pp. 69–83, 2020.

[3] A. Vidhya. (). 'Clustering: Types of clustering: Clustering applications', [Online]. Available: https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/ (visited on 25/05/2021).

[4] J. P. Mouton, M. Ferreira and A. S. Helberg, 'A comparison of clustering algorithms for automatic modulation classification', *Expert Systems with Applications*, vol. 151, p. 113 317, 2020.

[5] scikit. (). '2.3. clustering', [Online]. Available: https://scikit-learn.org/stable/modules/clustering.html#overview-of-clustering-methods (visited on 25/05/2021).

[6] A. Vidhya. (). 'Clustering: Types of clustering: Clustering applications', [Online]. Available: https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/ (visited on 25/05/2021).

[7] R. J. Campello, P. Kröger, J. Sander and A. Zimek, 'Density-based clustering', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 2, e1343, 2020.

[8] K. G. Derpanis, 'Mean shift clustering', *Lecture Notes*, p. 32, 2005.

[9] A. Gardner, 'Datasets for motion-capture-based hand gesture recognition', 2017.

[10] Medium. (). '6 different ways to compensate for missing data (data imputation with examples)', [Online]. Available: https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779 (visited on 25/05/2021).

[11] ——, (). 'Principal component analysis and k-means clustering to visualize a high dimensional', [Online]. Available: https://medium.com/@dmitriy.kavyazin/principal-component-analysis-and-k-means-clustering-to-visualize-a-high-dimensional-dataset-577b2a7a5fe2 (visited on 25/05/2021).

[12] A. Vidhya. (). 'Hierarchical clustering: Hierarchical clustering python', [Online]. Available: https://www.analyticsvidhya.com/blog/2019/05/beginners-guide-hierarchical-clustering/ (visited on 25/05/2021).

[13] scikit. (). 'Importance of feature scaling', [Online]. Available: https://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html (visited on 25/05/2021).

[14] Medium. (). 'Dbscan parameter estimation using python', [Online]. Available: https://medium.com/@tarammullin/dbscan-parameter-estimation-ff8330e3a3bd (visited on 25/05/2021).