**Programming Assignment #1**

**Introduction**

With the rapid increase of the information that is handed nowadays, it is not uncommon to see huge datasets for all different types of applications, for instance, speech recognition, market studies, etc. Clustering techniques come as an aid to obtain the most information out of this extensive gathered data, by separating data into meaningful groups that can be analyzed more easily. The purpose of the following assignment is to experiment with different clustering algorithms, showing the cases where they perform the best and the worst; all while also being compared to each other with different kinds of metrics.

**Objectives**

The main objectives of this assignment are the following:

* Implement some of the most known clustering algorithms discussed in the lectures.
* Experiment with different parameters of the algorithms to see different results.
* Compare the results of own implementations with those made by popular clustering libraries.

**Review of the methods used**

Formally speaking, clustering algorithms are the ones that separate every instance of the data into groups called clusters, in which every instance is as similar as possible from other instances in their cluster, is as different as possible from instances in other clusters, and the final separation of clusters in based on a clear and meaningful measurement criterion.

* Datasets

To test the algorithms’ performance on a variety of scenarios, very different datasets were used during the experiments shown in the next section. The main datasets used can be seen in Figure 1. Each figure is composed by 2,000 datapoints.

Scatter chart

Description automatically generated

Figure From left to right, top to bottom: (a)Noisy circles (b)Noisy moons (c)Blobs with varied variances (d)Anisotropicly distributed blobs (e)Equally sized blobs (f)Unclustered data

* Clustering algorithms used
  + K-Means

K-Means algorithm clusters data by separating samples into groups where every sample belongs exactly to one group. The algorithm minimizes does the separation by trying to minimize the inertia of the group. This measure is defined as the sum-of-squares criterion where:

Text

Description automatically generated with low confidence

K-Means defines the representative values of the clusters as the mean of all points in the cluster, so the value is not necessarily a data point.

Although this algorithm is one of the most popular clustering methods, it has some drawbacks:

* The inertia measurement assumes that clusters are convex, making the creation of irregular shapes difficult.
* The number of clusters, K, needs to be specified.
* It is sensible to the initialization values.
* It is sensible to outliers.
  + - K-Medoids

K-Medoids is an extension of the original K-Means algorithm. This process replaces the mean of the cluster with its median, so the medoid of a cluster is one of its samples that minimizes its total dissimilarity to the other samples in the same cluster.

This algorithm is useful for relational data, where there is no information about individual data, and it is impossible to compute the mean. Compared to means, medoids are less sensitive to outliers, but their calculations are harder.

* + Mixture decomposition

These algorithms estimate a probability distribution as the lineal combination of a set of parametric components. This parametric probability distribution be any function, but the most common one is the Gaussian function.

* + - Gaussian mixture model

The use of Gaussian functions in mixture models implies that the approach to clustering differs from K-means in the sense that it is a flexible and probabilistic approach to modelling the data, producing soft assignments to each cluster. Every final distribution has a weight associated with every data point. The Gaussian mixtures models are formed by leveraging another popular algorithm, the Expectation Maximization (EM) algorithm.

The EM algorithm consists of two steps, the E-Step and the M-Step. The E-Step computes the expectation with the current guessed parameters, while the M-Step finds a new set of parameters that maximize the expectation.

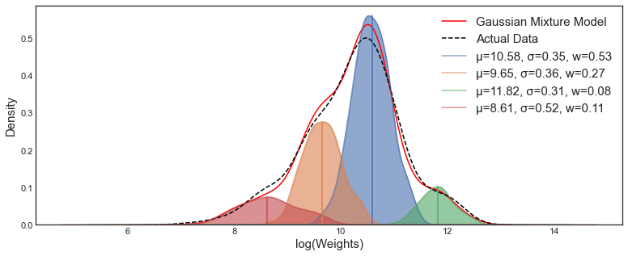


Figure Example of GMM formed by four different Gaussian functions

* + Hierarchical clustering

Hierarchical clustering is a group of algorithms that produce a hierarchy of possible clusterings, and they produce this by continuously merging or splitting clusters. The final hierarchy is usually represented as a dendogram. The root of this dendogram is the unique cluster that contains all the samples. These kind of clustering techniques are common for relational data.

The two main types of hierarchical clustering are: agglomerative clustering and divisive clustering algorithms. The first one iteratively merges clusters, while the latter iteratively splits them. A linkage criterion is set during the agglomerative clustering algorithms, and it determines the metric used for the merging. There are four popular linkages (single link, complete link, average linkage, and ward linkage), but only two are experimented with in this assignment:

* Ward minimizes the sum of squared differences within all clusters.
* Single linkage minimizes the distance between the closest observations of pairs of clusters.

Single linkage often gives the most irregular shapes, while ward linkage gives the most regular sizes. However, single linkage can be computed very efficiently and so, it is very useful on large datasets.

* + - BIRCH

This algorithm was designed for datasets that do not fit on memory. The special thing about the algorithm is that it builds a tree called Clustering Feature Tree (CFT) for the data. Essentially, the process compresses the data into Clustering Feature nodes that hold only the necessary information for clustering, which prevents the need to have everything in memory. The information included features the following elements:

* Number of samples in subcluster
* Linear sum of all samples
* Squared L2 norm of all samples
* Centroids
* Squared norm of the centroids
  + Density-based

This kind of algorithms view clusters as areas of high density that are separated by areas of low density. They are good at finding clusters of any shape, opposed to k-means for example. In this assignment, the DBSCAN and OPTICS algorithms are further explored.

* + - DBSCAN

The main contribution of DBSCAN is the creation of core points. These are datapoints with sufficiently high local density defined by a radius . Then, the algorithm defines a cluster as a set of core points, and a set of non-core points close enough to a core point. Moreover, DBSCAN identifies outliers and marks them as noise, so by definition, every core point will belong to a cluster, but a non-core point will not necessarily form a part of a cluster, depending on the defined radius.

Some challenges of the DBSCAN algorithm include the efficiency issues, the uncertainty of the best , and the fact that it does not deal very well to clusters that have different densities.

* + - OPTICS

To deal with clusters that have multiple density regions, an improvement to DBSCAN was proposed in OPTICS. This algorithm creates an ordering of all data points, not directly creating clusters, but making it easy to create them from the ordering. The two main concepts of the technique are the core distance, defined as the smallest for a point to be considered a core point; and the reachability distance of a point from a point , defined as . The main contribution of OPTICS is the relaxed handle of the radius , as the value can: 1) be set after the ordering has been created, and 2) be a value range.

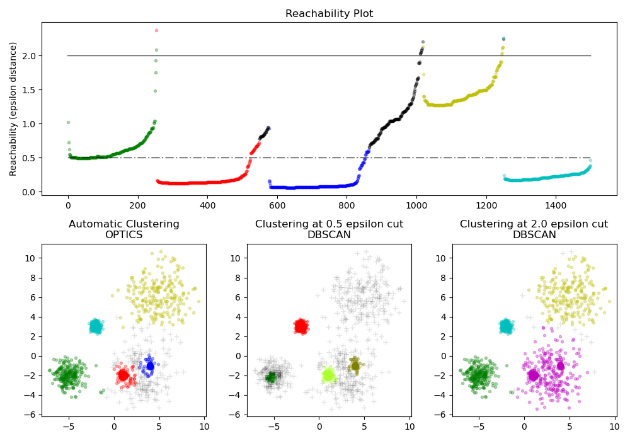


Figure Comparison of DBSCAN and OPTICS

* + Mode-Seeking
    - Mean-shift

This mode-seeking algorithm implements a process resembling a gradient descent process, as it selects a datapoint and makes its way to a denser area of the data. Every local maxima of density, denominated a mode, becomes a cluster at the end of the algorithm. If there are some similar modes, they join during the final post-processing phase.

The algorithm has one hyperparameter, the bandwidth, which defines the size of the region that is going to be taken into consideration during the search for the mode/centroid.

* Metrics

Evaluating the performance of a clustering algorithm is often a tricky process, as a “good” result highly depends on the meaning and purpose of the data, so it is not always as easy as counting the number of errors or the precision as in a supervised learning algorithm. And so, a good evaluation can be defined as a good representation of the actual cluster structure of the data. There are two main types of metrics, internal and external metrics.

* + Internal metrics: These rely only on the clustering results, with no external information, to determine whether the performance of the algorithm is good or not. These metrics are usually used to compare different parameters of the same algorithm, such as the K value in K-means, as the comparisons between different algorithms may not be ideal. The internal metrics used in this assignment are the Davies-Bouldin and Silhouette Index.
    - Davies-Bouldin: The minimum score is zero, with lower values indicating better clustering.
    - Silhouette index: The score is constrained between -1 for incorrect clustering and +1 for highly dense clustering. If the score is around zero, it indicates that there are overlapping clusters.
  + External metrics: These rely on external information, such as cluster labels to give a measurement on the performance of the algorithm. These metrics help choose between different various algorithms to determine which one fits the task-at-hand the most. The external metrics used in this assignment are the Rand index and Normalized Mutual Information.
    - Rand Index: A perfect labeling has a score of 1.0. Bad scores are not necessarily close to 0, but they are low.
    - Normalized Mutual Information: A perfect labeling has a score of 1.0. Bad scores can even have negative scores.

**Explanation of the experiments done**

The nine different algorithms were tested in all six datasets at once. The result of the clustering algorithms can be seen in Figure 4 and the correct results can be seen in Figure 5. The results also show the execution time of every algorithm on the datasets. Right away, comparing both figures, by some qualitative analysis, we can conclude that the best algorithms are the DBSCAN and OPTICS. However, we can see that OPTICS is the slowest algorithm, so just by the information given by these results, we would choose DBSCAN for these types of clustering problems.

A picture containing text

Description automatically generated

Figure Clustering results side by side for different algorithms. Processing time included.

Scatter chart

Description automatically generated with medium confidence

Figure Correct clustering labels for datasets.

In the first dataset, we can also notice that the single link agglomerative clustering is the only other algorithm that can correctly separate the first dataset, however, it has a lot of problems in other datasets, and we can conclude that the algorithm works only when there is not any sort of connection between clusters, like in datasets 1, 2 and 4.

Moreover, we can also notice that all algorithms can successfully detect the three clusters in dataset 5, because they are clearly defined and not touching each other at all.

In the last dataset, there is not a clustering structure in the data, but still, we can observe the algorithms trying to create clusters with the given information. The only algorithms that we can observe did not create unnecessary clusters were again the DBSCAN and OPTICS. From the last dataset we can also sort of see the intuition behind the search for clusters during different algorithms, such as the clear split at the half for the mean-shift algorithm, as both halves have the same density of points.

However, because qualitative results are sometimes hard to agree upon, quantitative metrics were calculated for the results. The results of four different metrics for every dataset are shown on Table 1 to Table 5.

Table Scores for dataset 1 (Noisy circles)



In table 1, we can see that the algorithms that had the best qualitative performance have the worst internal metric scores. However, this makes sense, as these two internal metrics measure the shapes of clusters, so they tend to have higher scores for non-convex, such as the ones generated correctly in this case by Single, DBSCAN and OPTICS. Moreover, these metrics are not meant to be used to compare different algorithms, but different parameters of the algorithm. To demonstrate different internal metrics interpretations, the results in Figure 6 show the K-Means algorithm evaluated with different K values. We would be able to rely on this metrics to define the K value, however, we can see that the metrics give very different results, and it would be up to us to interpret their meaning and usefulness in each specific situation. Yet, we can see some interesting behaviors such as the DB-Index choosing K=6 for the third and last datasets, as this value is the one that creates the most symmetrical clusters for the given data. Overall, for this set of results the silhouette index is the one that chooses the best K parameter, but this may not be true for other parameters in other algorithms.

Chart, scatter chart

Description automatically generated

Figure K-means with different K value. The Silhouette score is on top of the DB-Index in the bottom right of each result. The best results for each dataset for both metrics are marked with a red box.

Table Scores for dataset 2 (Noisy moons)



Table Scores for dataset 3 (Blobs with varied variance)



Table Scores for dataset 4 (Anisotropicly distributed blobs)



Table Scores for dataset 5 (Equally sized blobs)



What we can fully conclude by looking at the tables is that in most cases, the RAND and NMI indexes agree with the qualitative analysis of clusters, as they give their maximum scores to DBSCAN and OPTICS several times, for the first, second and fifth dataset. However, for dataset 3 the Ward algorithm has the highest scores, and that also reflects in the qualitative analysis, as it correctly classifies points that we Also noting that the fifth dataset is the easiest and every algorithm gets the maximum or same score, as they all achieve a perfect clustering.

**Own Implementations**

The Gaussian Mixture Model and DBSCAN were both implemented, and the results were compared with the methods provided by the sklearn library that is used throughout the rest of this assignment.

* DBSCAN

The results of both implementations can be seen in Figure 7. The epsilon value and minimum points in a core point’s vicinity parameters are the same for both methods (0.1 and 5 respectively). Both results are practically the same, except for some minor differences on the edges of the clusters, possibly due to the order of calculations in the methods, but overall both clustering results are what we would expect.

Chart, scatter chart

Description automatically generated

Figure DBSCAN implementation comparison

* GMM

**Discussions**

As always, the results for clustering are very hard to objectively describe, as the result’s quality will highly depend on what the data means to us. However, with the assistance and good interpretation of internal and external metrics, we can aid our decision for the best hyperparameters and algorithms for different situations.

On the other hand, even if judging a result is often tricky, sometimes when the data only has 2 dimensions, we can also qualitatively see which algorithm fits data the best, and realize that some algorithms are clearly meant for different situations, so there are situations where even if the K-Means cannot create non-convex shapes, it may still be the algorithm we choose due to its simplicity and results in that situation.

**Main References**

* Xu, D., Tian, Y. A Comprehensive Survey of Clustering Algorithms. Ann. Data. Sci. 2, 165–193 (2015). <https://doi.org/10.1007/s40745-015-0040-1>
* Wong, K. C. (2015, November). A short survey on data clustering algorithms. In 2015 Second international conference on soft computing and machine intelligence (ISCMI) (pp. 64-68). IEEE.
* Siwei Causevic, Implement Expectation-Maximization Algorithm(EM) in Python from Scratch, 2020. https://towardsdatascience.com/implement-expectation-maximization-em-algorithm-in-python-from-scratch-f1278d1b9137 (Accessed May 2022).
* Daniel Foley, Gaussian Mixture Modelling (GMM), 2019. https://towardsdatascience.com/gaussian-mixture-modelling-gmm-833c88587c7f (Accessed May 2022).
* Moosa Ali, DBSCAN Clustering Algorithm Implementation from scratch, 2021. https://github.com/Moosa-Ali/DBscan-Clustering-Implementation/blob/main/DBSCAN%20implementation.ipynb

**Code**