Comparison and Evaluation of Clustering Algorithms
K-Means, Agglomerative and DBSCAN

Haluk Erhan
McMaster University CCE BDA Program

1. Introduction

Clustering is the task of partitioning the dataset into groups. The goal is to split up the data in such a way that points within a single cluster are very similar and points in different clusters are different. One of the most important challenges in clustering tasks is defining optimum number of clusters which leads stable clustering algorithm. Stability is being widely used in practical applications as tuning hyper parameters of clustering algorithms like the number of clusters or various stopping criteria. (Rakhlin & Caponnetto, 2006)

The goal of **K-Means** algorithm is to find groups in the data, with the number of groups represented by the variable *K* which works iteratively to assign each data point to one of K groups based on the features that are provided. **Agglomerative** clustering refers to a collection of clustering algorithms that all build upon the same principles: the algorithm starts by declaring each point its own cluster, and then merges the two most similar clusters until some stopping criterion is satisfied. (Xu & Wunsch, 2005) The main benefits of **DBSCAN** (density based spatial clustering of applications with noise) are that it does not require the user to set the number of clusters a priori, it can capture clusters of complex shapes, and it can identify points that are not part of any cluster. (Alpaydin, 2010)

In this study, K-Means, Agglomerative and DBSCAN clustering algorithms will be compared in terms of Silhouette Score, Adjusted Rand Score, Normalized Mutual Info and Clustering time elapsed. In addition, decomposition methods like Principal Component Analysis (PCA) and Non-negative Matrix Factorization (NMF) are intended to be discussed.

Since, it is aimed to compare clustering algorithms the most versatile, easy and resourceful dataset Iris Date Set will be used at first stage. As a second stage blobs and moon synthetic data will be produced and compared. As final stage hand written dataset will be used for comparisons.

2. Problem description

Clustering algorithms are designed to explore an inherent natural structure of the data objects, where objects in the same cluster are as similar as possible and objects in different clusters are as dissimilar as possible. The equivalence classes induced by the clusters provide generalising over the data objects and their features. Clustering methods are applied in many domains, such as medical research, psychology, economics and pattern recognition.

In typical uses of clustering the goal is to determine all the following:

- The number of clusters
- The absolute and relative positions of the clusters
- The size of the clusters
- The shape of the clusters
- The density of the clusters (Kriegel, Kröger, Sander, & Zimek, 2011)

Clustering algorithms are considered under unsupervised learning techniques, which means there is no known output, neither any training algorithm. This shows a major challenge in unsupervised learning as evaluating whether the algorithm learned something useful. Since there is no label information, it can't be easily defined if the solution reached to the right output or not.

The choice of a suitable clustering algorithm and of a suitable measure for the evaluation depends on the clustering objects and the clustering task. In this study K-Means, Agglomerative and DBSCAN algorithms will be discussed.

a. K-Means Clustering

K-means clustering is one of the simplest and most commonly used clustering algorithms which is aimed to find the cluster centers that are representative of certain regions. The algorithm alternates between two steps, assigning each data point to the closest center, and then assigning that data center as the mean of the points. Then it finishes when the assignments of distances to cluster no longer changes.

One of the drawbacks of k-means is that it relies on a random initialization, which means the outcome of the algorithm depends on a random seed. (scikitlearn runs the algorithm 10 times with 10 different random initializations, and returns the best result).

Further downsides of k-means are the relatively restrictive assumptions made on the shape of clusters, and the requirement to specify the number of clusters you are looking for (which might not be known in a real-world application).

b. Agglomerative Clustering

Agglomerative clustering refers to a collection of clustering algorithms that all build upon the same principles: the algorithm starts by declaring each point its own cluster, and then merges the **two most similar clusters** until some stopping criterion is satisfied.

Agglomerative clustering produces what is known as a **hierarchical clustering**. The clustering proceeds iteratively, and every point makes a journey from being a single point cluster to belonging to some final cluster.

c. DBSCAN

DBSCAN (which stands for "densitybased spatial clustering of applications with noise"). The **main benefits of DBSCAN** are that it does not require the user to set the number of clusters a priori, it can capture clusters of complex shapes, and it can identify points that are not part of any cluster. DBSCAN is somewhat slower than agglomerative clustering and k-means, but still scales to relatively **large datasets**.

DBSCAN works by identifying points that are in "crowded" regions of the feature space, where many data points are close together. These regions are referred to as dense regions in feature space. The idea behind DBSCAN is that **clusters form dense regions of data, separated by regions that are relatively empty**.

d. Silhouette Score

The **silhouette** value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The **silhouette** ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

e. Adjusted Rand Score

The Rand Index computes a similarity measure between two clustering by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clustering.

f. Normalized Mutual Info

Normalized Mutual Information (NMI) is a normalization of the Mutual Information (MI) score to scale the results between 0 (no mutual information) and 1 (perfect correlation). Perfect labeling are both homogeneous and complete, hence have score 1.0. (Guido, 2017)

3. Solution description

In order to compare the clustering algorithms first the most versatile, easy and resourceful dataset Iris Date Set is used. Then after two different synthetic dataset compared. Finally hand written digits dataset is used and compared.

For the coding simplicity a definition is developed which allows to input datasets, algorithm and hyper parameters. Also, in order to understand how the clustering captured by the algorithm visualization techniques are used.

4. Results

a. Iris Dataset

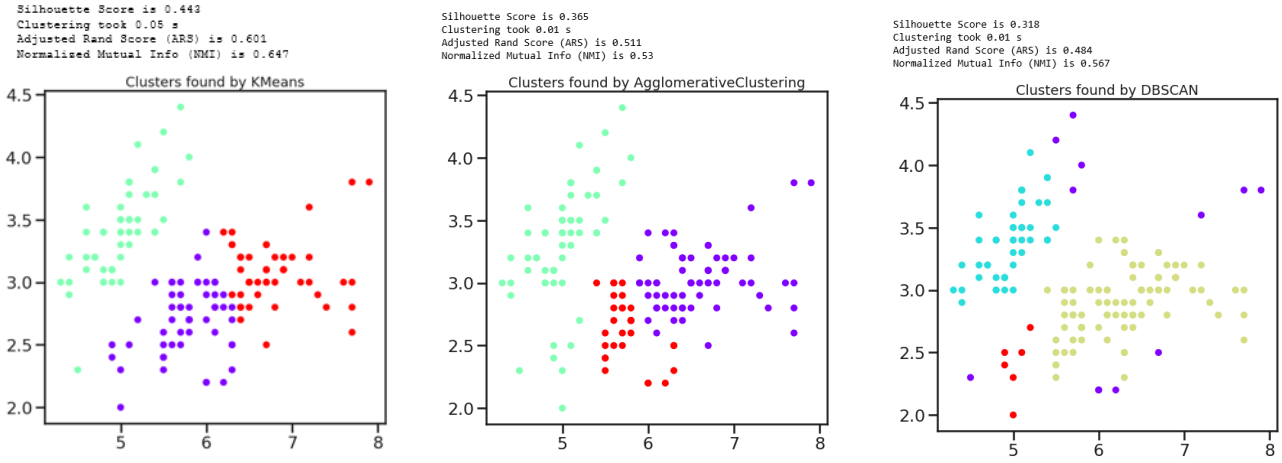

Figure 1 Iris Dataset Clustering Comparisons

The clustering parameters of Iris dataset is intuitive, so the cluster number is set to 4. Kmeans algorithm gives the highest score in term of Silhouette, ARS and NMI.

In Agglomerative clustering the linkage criteria determine the metric used for the merge strategy. For predefined clustered datasets, ward linkage gives the optimum clustering as seen in figure 1. In this solution **Ward** criteria is used which minimizes the sum of squared differences within all clusters.

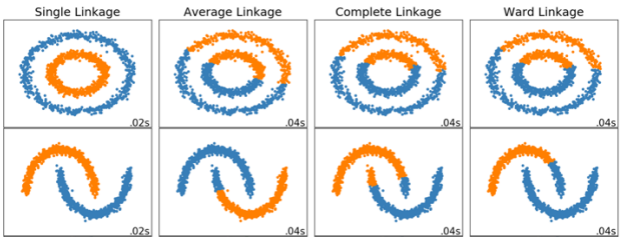

Figure 2 Linkage Criteria of Agglomerative Clustering

Maximum or **complete linkage** minimizes the maximum distance between observations of pairs of clusters. **Average linkage** minimizes the average of the distances between all

observations of pairs of clusters. **Single linkage** minimizes the distance between the closest observations of pairs of clusters. (scikit-learn.org, n.d.)

DBSCAN is not efficient enough as the geometry of data is not not-flat and there is no uneven cluster size. (scikit-learn.org, n.d.)

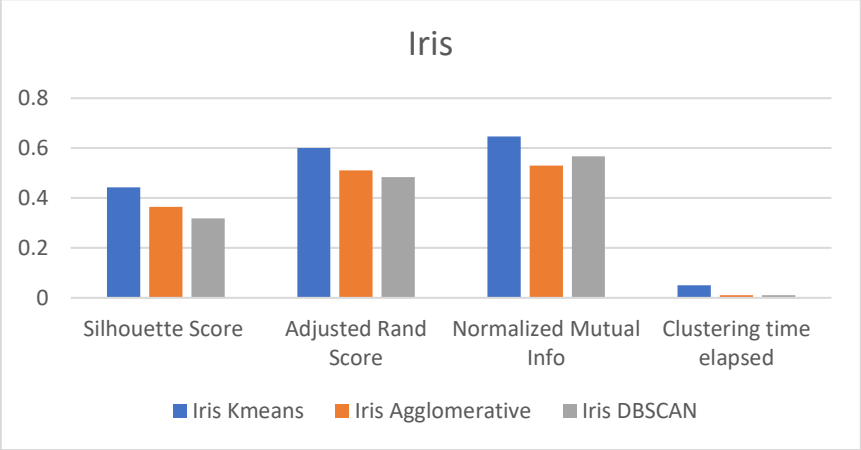

Figure 3 Algorithm Comparisons in Iris Dataset

b. Synthetic Blob Dataset

In order to understand the how the algorithms are working in different shaped dataset a isotropic Gaussian blobs are generated. Number of centers are chosen as 6, and also the cluster size is verified with elbow method as seen in figure 4.

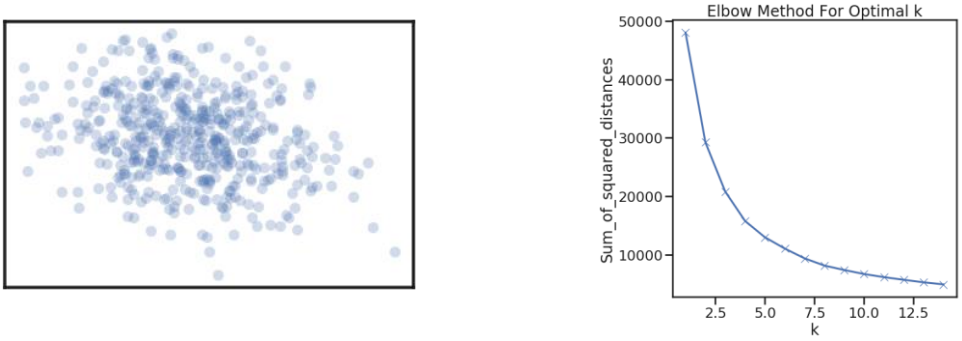

Figure 4 Gaussian Blobs and Elbow Illustration

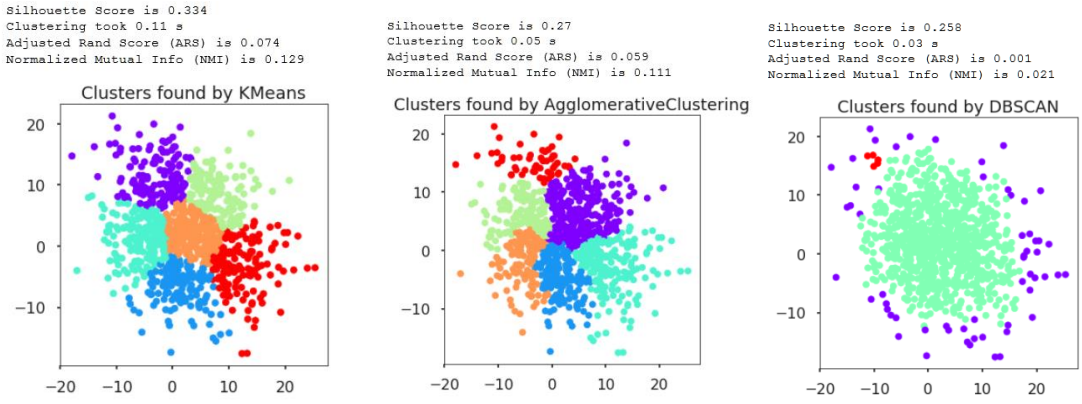

Figure 5 Comparison of Algorithms in Blob Dataset

As seen in figure 5, cluster segregations are represented well in both Kmeans and Agglomerative algorithms. The results could be improved by tweaking the parameters for each clustering strategy. Since the idea behind DBSCAN is to try and ignore the data points that are more spread out and focus on the dense parts, which should be the central cores of the clusters, it can be seen figure 5 that algorithm disregarded the center points.

PCA and tSNE

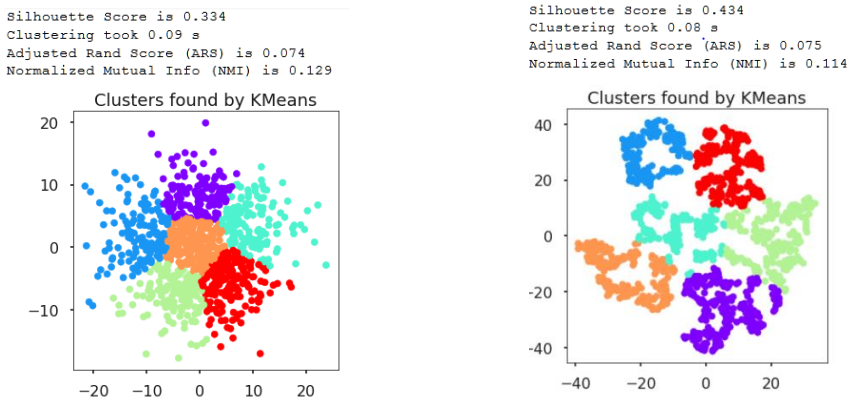

Figure 6 Kmeans Algorithm on Blob Dataset after PCA and tSNE Application

PCA which is linear dimensionality reduction using Singular Value Decomposition of the data, projects to a lower dimensional space. But in this case since the original data and the dimensionally reduced data are same, K means algorithm gives the same results in terms of scores.

tSNE (t-distributed Stochastic Neighbor Embedding) is a tool to visualize high-dimensional data. The hyperparameter “perplexity” has been put in various values between 5 to 40, but no significant difference was appeared because tSNE is very insensitive to this parameter.

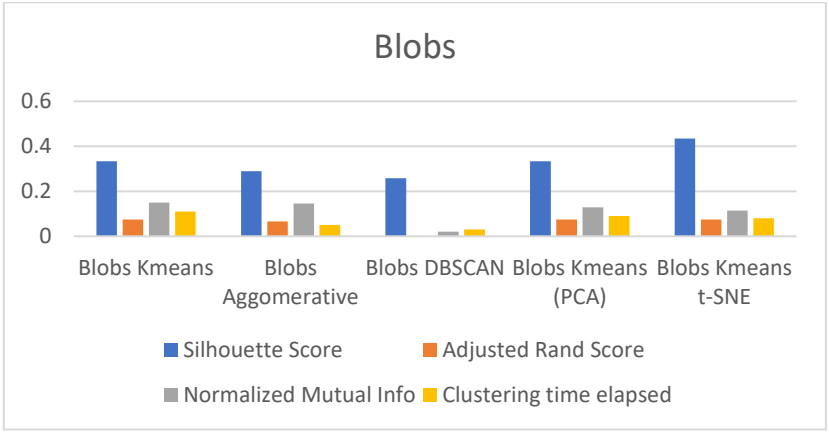

Figure 7 Algorithm Comparison in Blob Dataset

Non-negative matrix factorization which is also used for dimensionally reduction techniques, is capable to produce region or part-based representations of objects and images. As in PCA, it tries to write each data point as a weighted sum of some components. But whereas the components and the coefficients must be nonnegative; that is, we want both the components and the coefficients to be greater than or equal to zero. So this could not be used in blob dataset where negative values are available.

c. Two Moon Synthetic Dataset

In order to investigate the behaviour of algorithms depending on the dataset type, moon dataset which is very useful for visualize the clusters.

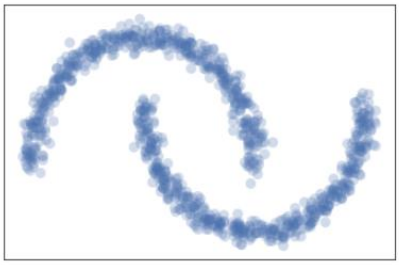

Figure 8 Two Moon dataset

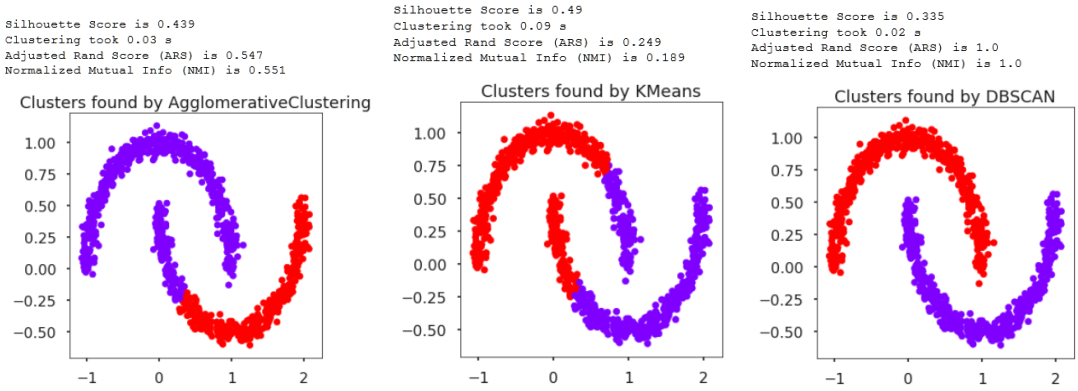

Figure 9 Comparison of Algorithms in Moon Dataset

In this dataset labels are clear and clustering should be mapped on two separate shape. As seen in figure 9, The DBSCAN algorithm views clusters as areas of high density separated by areas of low density. And, the scores indicates they 2 labeled cluster is captured very clearly.

In DBSCAN, there are two parameters to the algorithm, **min_samples** and **eps**, which define formally what we mean when we say *dense*. Higher **min_samples** or lower eps indicate higher density necessary to form a cluster. (scikit-learn.org, n.d.)

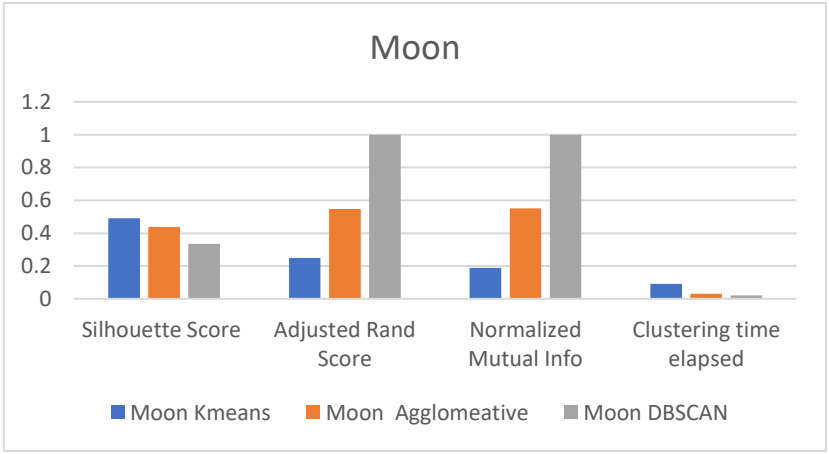

Figure 10 Algorithm Comparison in Moon Dataset

d. Hand Written Digits Dataset

This dataset is made up of 1797 8x8 images. Each image, like the one shown below, is of a hand-written digit. The observation's feature values are presented as a vector.

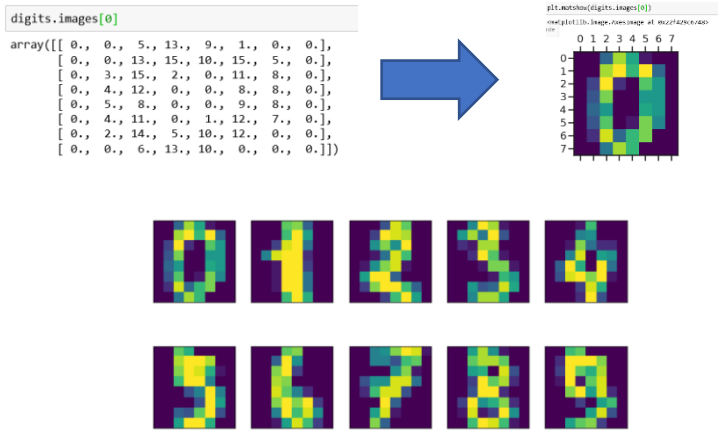

Figure 11 Digits Dataset (from sklearn datasets)

In order to reduce the dimensions, at first PCA was applied, then tSNE respectively.

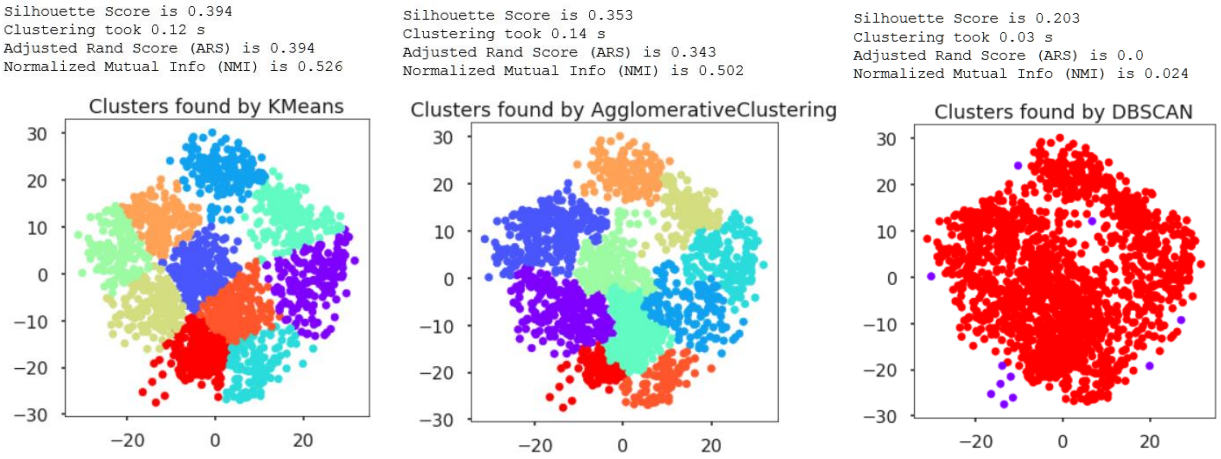

Figure 12 Algorithm Comparisons of Digit Dataset after PCA application

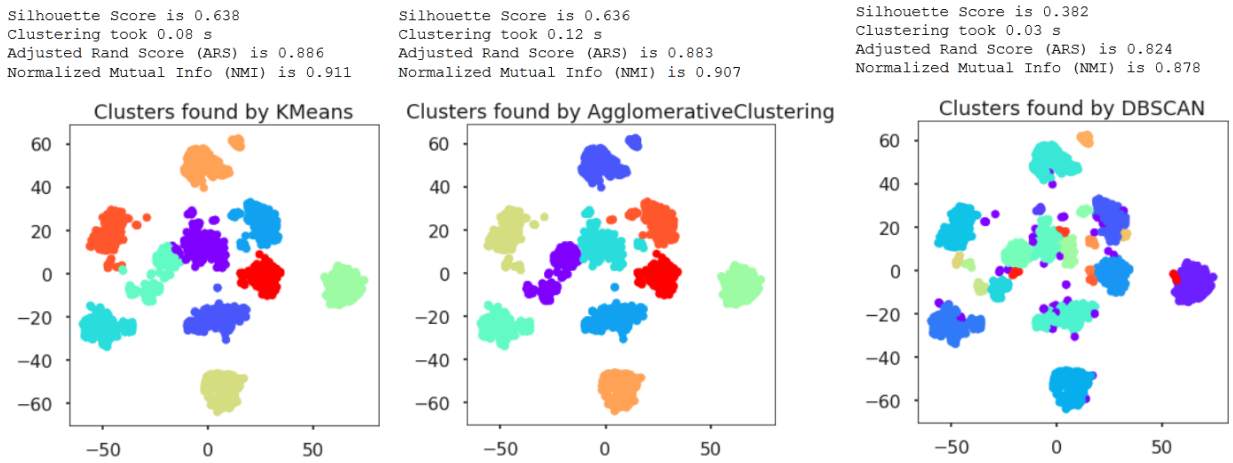

Figure 13 Algorithm Comparisons of Digit Dataset after tSNE application

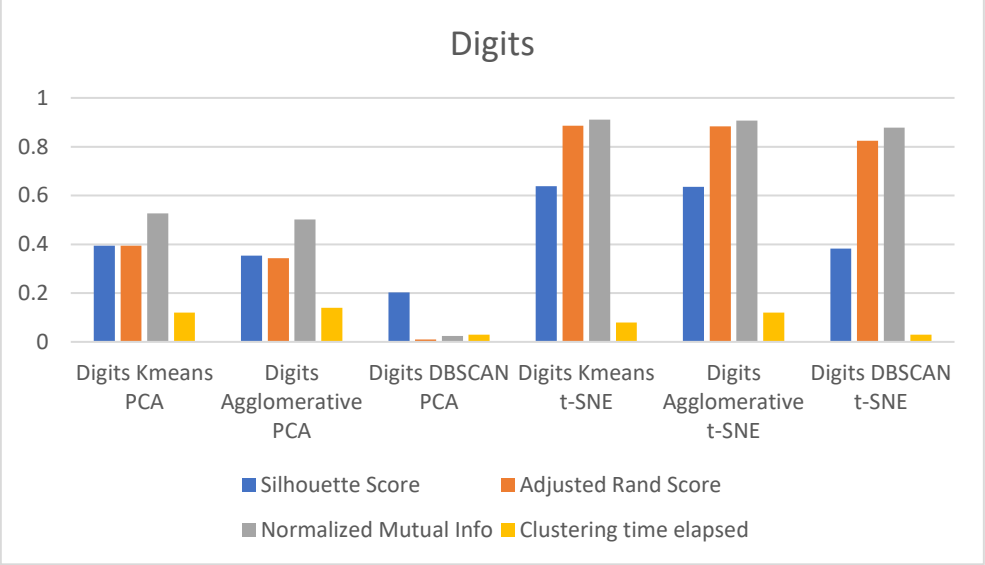

Figure 14 Comparison of Algorithms on Digits Dataset

As seen in evaluation parameters, ARS and NMI has great values very close to 1 because both requires knowledge of the ground truth classes. So, this example shows a representation of clustering evaluation values.

5. Analysis

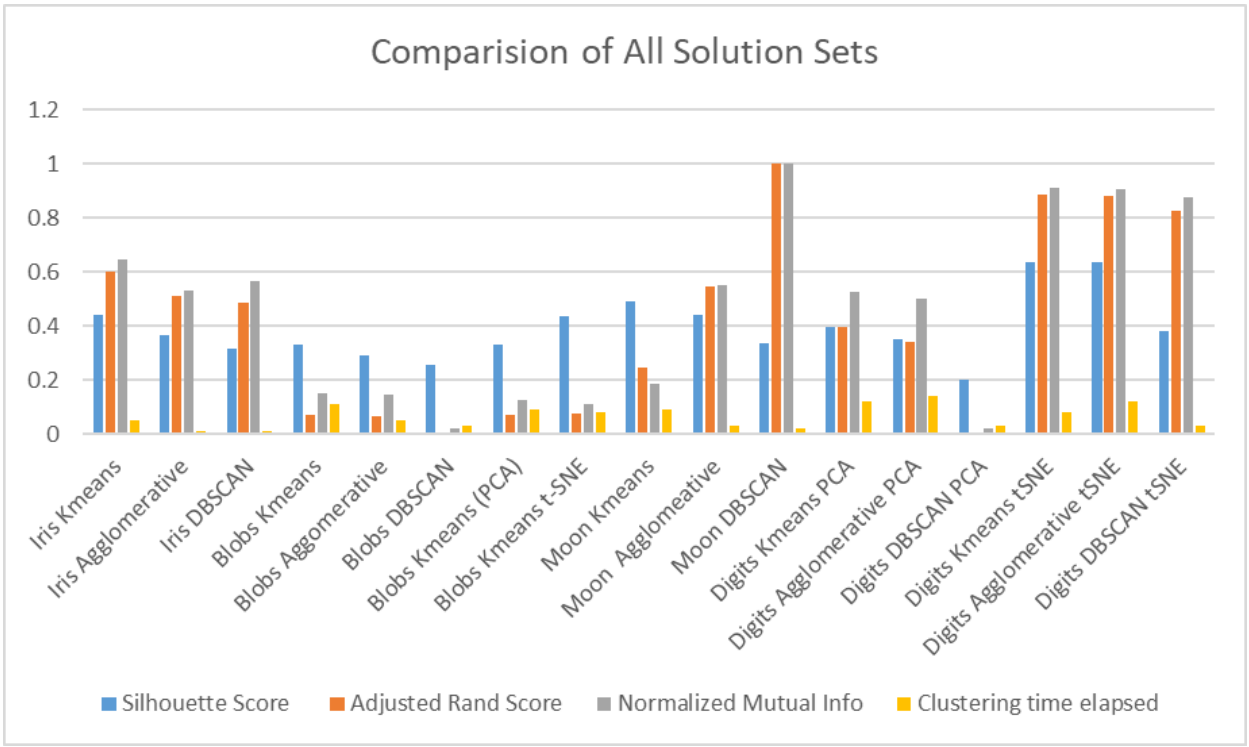

Figure 15 Comparison of All Solution Sets

In terms of Dataset type, Figure 15 shows that Digits dataset is the most clusterable set. Whereas moons dataset which can be considered as a good example of density differentiated shapes, is very suitable for DBSCAN algorithms.

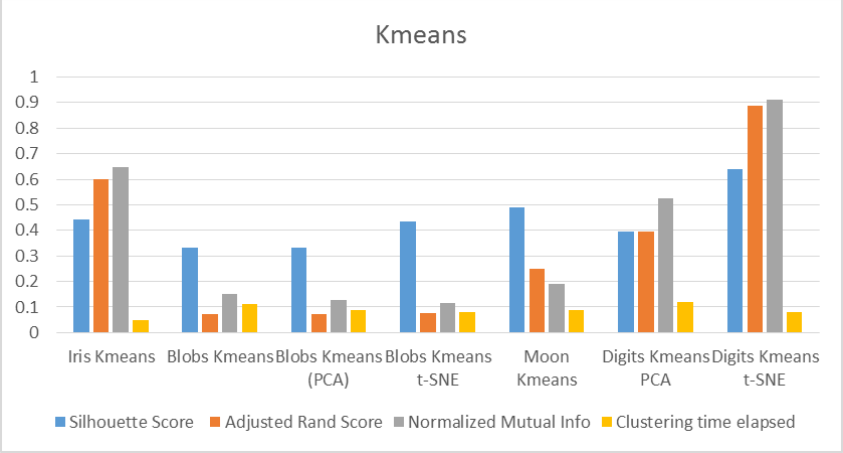

Figure 16 Kmeans Comparison on All Datasets

Considering the logic of Kmeans algorithm, it can be said that Kmeans, throws points into clusters whether they belong or not. Although we had a good intuition about the number of clusters Blobs data did not give desired scores. Even if you know the "right" number of clusters for a given dataset, Kmeans might not always be able to recover them. Kmeans can only capture relatively simple shapes. Kmeans also assumes that all clusters have the same diameter in some sense; it always draws the boundary between clusters to be exactly in the middle between the cluster centers.

But it resulted well scores especially in Iris and Digits datasets whose clusters can be considered as intuitive. This comparison also shows that dimensionally reduction to original dimension do not change the results obtained from the algorithms.

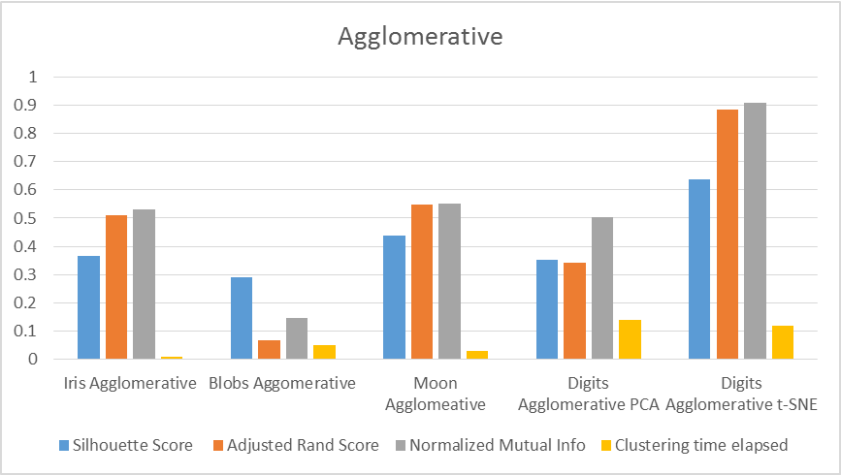

Figure 17 Agglomerative Comparison on All Datasets

Agglomerative algorithm is all applied with Ward linkage criteria in order to grab the collated compared parameters. This resulted better scores then Kmeans as the scores shows in figure. But if linkage criteria were changed depending on the dataset then better results would be obtained. For example, single linkage criteria would give close to 1 for the two-moon dataset.

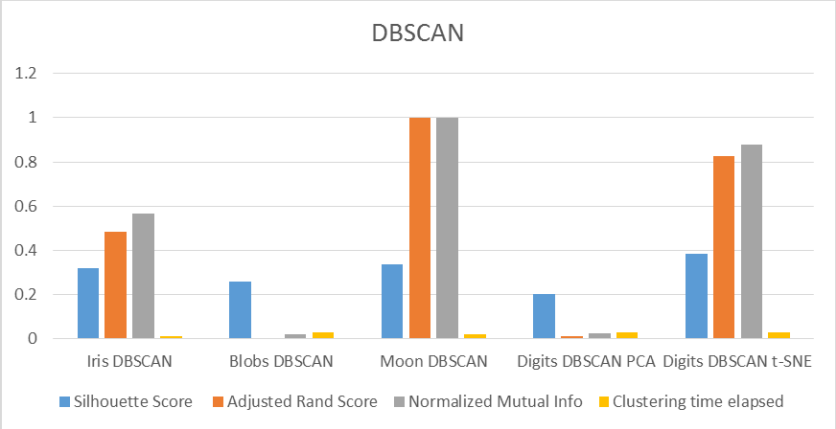

Figure 18 Comparison of Algorithms with All Solution Datasets

Since the DBSCAN clusters as areas of high density separated by areas of low density, it gace good results in density-dominant datasets. Digits dataset represents a good example for characteristic for DBSCAN algorithm. Because the clusters density gradient increased after tSNE application, the DBSAN algorithm gave better results than PCA applied dataset.

6- Conclusion

This study can be considered as a theory exercise and litterateur survey for whom they are new to machine learning and data science. In order to compare the algorithms, the code architecture should be very organized and detailed. So, this study was a good opportunity to develop the coding skills.

Although the datasets chosen are not realistic, it is understood that applying and evaluating clustering is a highly qualitative procedure, and often most helpful in the exploratory phase of data analysis. And also, it supports the definition theory presents a systematic way of understanding events.

In this study three clustering algorithms: Kmeans, Agglomerative and DBSCAN clusterings are compared. Each of the algorithms has somewhat different strengths. Kmeans and Agglomerative can be viewed as a decomposition method, where each data point is represented by the cluster center. DBSCAN works through the densified points and help to determine the number of clusters. DBSCAN sometimes produces clusters of very differing size, which can be a strength or a weakness. Agglomerative clustering can provide a whole hierarchy of possible partitions of the data, which can be easily inspected via dendrograms.

All clustering algorithms have parameters which needs to be adjusted respectively. The real problem is how pick settings for those parameters. So, if there is no enough information about data, in other words domain knowledge, it can be hard to determine what value or setting a parameter should have. This means parameters need to be intuitive enough that you can hopefully set them without having to know a lot about the data.

6. References

Alpaydin, E. (2010). *Introduction to Machine Learning.* The MIT Press. Retrieved 11 9, 2018, from https://mitpress.mit.edu/books/introduction-machine-learning

Guido, A. C. (2017). *Introduction to Machine Learning with Python.* USA: O'Reilly.

Kriegel, H.-P., Kröger, P., Sander, J., & Zimek, A. (2011). Density-based Clustering. *WIREs Data Mining and Knowledge Discovery, 1*(3), 231–240. Retrieved 11 29, 2018, from http://wires.wiley.com/WileyCDA/WiresArticle/wisId-WIDM30.html

Rakhlin, A., & Caponnetto, A. (2006). *Stability of K-Means Clustering*. Retrieved 11 9, 2018, from http://www-stat.wharton.upenn.edu/~rakhlin/papers/stability_clustering.pdf

scikit-learn.org.(n.d.).Retrieved from https://scikit-learn.org/stable/modules/clustering.html#overview-of-clustering-methods

Steinbach, M., Karypis, G., & Kumar, V. (2000). *A Comparison of Document Clustering Techniques*. Retrieved 11 9, 2018, from http://cs.fit.edu/~pkc/classes/ml-internet/papers/steinbach00tr.pdf

Xu, R., & Wunsch, D. C. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks, 16*(3), 645-678. Retrieved 11 9, 2018, from http://scholarsmine.mst.edu/cgi/viewcontent.cgi?article=1763&context=ele_comeng_facwork