Clustering validation indexes

Levy G. da S. Galvão[1] e José A. F. Costa[2]

***Abstract*— Multidimensional clustering data sets interpose a difficult task to determine the ideal number of clusters, even when using data dimensionality reduction techniques. In this context, clustering validation indexes (CVIs) plays an important role to find the ideal number of clusters given a data set. Considering each CVI particularity and machine learning clustering model, its performance may change when quantifying the number of clusters. So the objective of this work is to evaluate multiples CVIs applied to different data sets that were clustered by different algorithms.**

I. INTRODUCTION

A clustering algorithm is used when there is a data set unlabeled, thus presenting as an unsupervised classification method that aims to discover its real partition, assuming that this data set has a clustering tendency. [2], [3].

But when considering clusters a major hyper parameter to feed the model is the number of clusters, be it a direct value of clusters as in K-Means algorithm [4], or indirectly as in DBSCAN or HDBSCAN algorithm via proximity metrics [5], [6]. Considering high dimensional data, even when applying dimensionality reduction techniques, the visual aid to determine the numbers of clusters might fail [2]. In this case some clustering validation indexes (CVIs) must to be evoke to establish a criteria for the number of clusters to be selected.

The CVIs are extremely important to define the ideal number of clusters for a given data set. Considering that there are multiples of them, it is up to the operator evaluate and decide which metric is more viable to its analysis.

In this context, this work aims to compare some CVIs applied in multiples data sets clustered with different algorithms, thus serving as a base methodology to evaluate clustering algorithms in the production environment.

II. METHODS

This section is responsible to present a quick summary about: the CVIs used to parse the clustering algorithms; the clustering algorithms used in this analysis; and the data sets that are used in the clustering;

A. Cluster validation indexes

The description of the CVIs are limited to a summary about its main characteristics like presented in [7], thus avoiding any mathematical or historical explanation as such.

1) Calinski and Harabasz index (CH): This index measures the amount of clusters presents in a dataset by dispersion metrics between and inside clusters. The ideal number of groups is the one that maximizes this index.

2) Dunn index: This index take into account the relation between the distance of the two closest points in different groups with the distance of the furthest points inside the same group. Thus a high value for this index meaning that different clusters are far from each other, but the points inside a cluster are close together. This index do not deal well with noise

3) Davies-Bouldin index (DB): This index is a function of the relation between sum of the internal group dispersion and the separation between groups. Low values for this index means that the groups are compacts with clusters centers well separated.

4) Pakhiraa-Bandyopadhyayb-Maulik index (PBM): This index is based in a compactation and separation of groups. As the compactation is measured by the sum of the distances between each point of a cluster and its centroid and the separation is measured by the maximum distance between centroids of a given group. The optimal numbers of clusters to use in the data is given by the maximum value of this index.

5) Density-Based Clustering Validation (DBCV): DBCV uses an approach to compute the density within a cluster and the density between clusters, thus allowing to Here, we implement DBCV which can validate clustering assignments on non-globular, arbitrarily shaped clusters (such as the example above). In essence, DBCV computes two values:

The density within a cluster The density between clusters

High density within a cluster, and low density between clusters indicates good clustering assignments.

B. Clustering algorithms

This subsection is reserved to a quick description of the clusterings algorithms used thus helping to emphasize the importance of the CVIs usage. [8], [9]

1) K-Means Clustering (K-Means): This clustering algorithm starts by defining K centroids based in the K hyper parameter fed in random points. Therefore tries to assimilate each point in the data set to one of the closest clusters. Then when all points are assigned to the clusters based in a centroid, each centroid is recomputed to represent the center of the cluster. Those steps are repeated until all centroids do not change.

The downgrade of this algorithm is that the total number of clusters must to be defined from the beggining. Data dimensionality reduction algorithms and clustering validation

*This work was not supported by any organization

[1] Electrical engineering undergraduate, Universidade Federal do Rio Grande do Norte, Brazil.

[2] Electrical engineering Ph.D, Universidade Federal do Rio Grande do Norte, Brazil.

indexes are essential to evaluate the ideal numbers of clusters to instantiate the K-means model.

2) Density-Based Spatial Clustering of Applications with Noise (DBSCAN): Different from K-means that the number of clusters must to be defined, DBSCAN does not have this hyper parameter. It works locating regions of high density of points that are separated by regions of low density of points. This algorithm use couple hyper parameters to track the density of the clusters areas, one of them is the ϵ value that indicates a radius which the maximum distance to a point to be considered inside a cluster starting from its neighbor. Other hyper parameters is the minum number of points grouped together to be considered a different cluster. Thus with this approach DBSCAN is capable of defining the number of clusters.

The downgrade of this approach is that if the hyper parameters are not well set, i.e. a higher value of ϵ means that less clusters will be considered and low values of ϵ with consider lots of clusters, the performance of the final clustering might be compromised.

Also outliers that are no close to the main clusters can not be attribute to them and an undefined label is assigned to them.

In this work the ϵ value will variate but the minimum of points is considered based in the data set shape, as two times the number of features.

3) Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN): HDBSCAN is an improved DBSCAN algorithm. Unlike the latter that uses a static epsilon, this one performs its algorithm over varying epsilon values and integrates the result to find a clustering that gives the best stability over epsilon. This allows a better performance since the there is no need to set the epsilon hyper parameter.

C. Data sets

All data sets used are from the UCI machine learning repository [10].

1) Iris: This data set contains 3 classes of 50 instances each and its label refering to a type of iris plant and 4 attributes. In a matter of fact, when the labels are dropped, this data set with a clustering algorithm should present 3 different clusters, as considering independence between classes. When applying PCA to reduce the data set to two dimensions the figure 1 can be obtained, showing three clusters but two sharing proximity.

2) Wine: This data set contains information about 13 feature attributes that links to 3 different classes of wine. Same for the Iris, this one should have 3 clusters. When applying PCA to reduce the data set to two dimensions the figure 2 can be obtained, showing three clusters totally sharing proximity.

3) Synthetic Control Data: This data set contains 600 examples of control charts synthetically generated and separated in 6 different classes of control charts with 100 charts for each class. Same for the Iris and Wine, this one should have 6 clusters. When applying PCA to reduce the data set to two dimensions the figure 3 can be obtained, showing six clusters but each pair sharing proximity since the curves in the data set are similar.

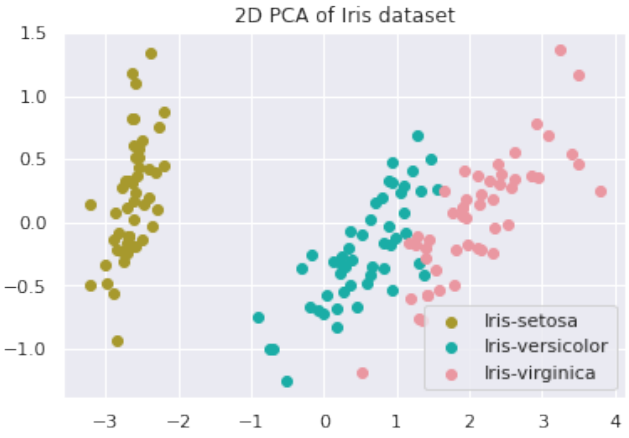

Fig. 1. Two dimensional Iris data set obtained with PCA.

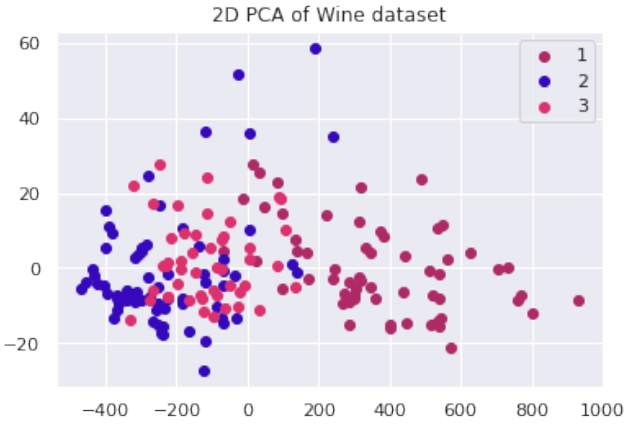

Fig. 2. Two dimensional Wine data set obtained with PCA.

III. Experimental Setup

The experimental setup set place entirely in the Google Colab environment with a Jupyter notebook running Python code resposible to load data sets, install third party libraries and to execute the analysis.

To execute the clustering algorithms and CVIs, third parties open source libraries were used and a wrapper class was designed to support easy integration between different APIs.

As a disclaimer, the index values displayed in the plots of this works are all normalized within its maximum and minimum value to a easy view, once the absolute values of these metrics do not hold any particular information, but only its minimum and maximum point, therefore the curve tendency.

In this work the CVIs will be evaluated within a range of number of clusters only form the K-Means algorithm, since as will be seen below, it is the only one that the number

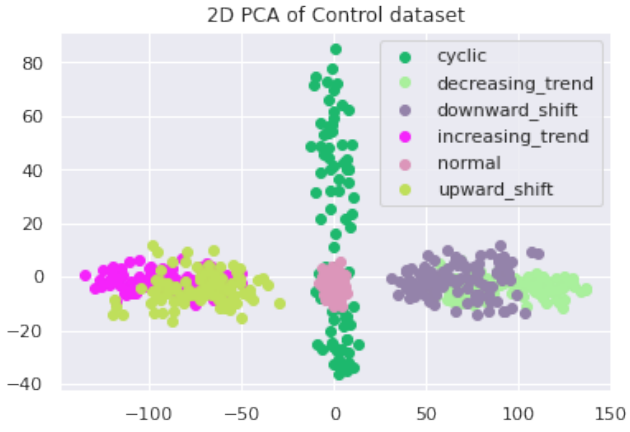

Fig. 3. Two dimensional Control data set obtained with PCA.

of clusters can be set. The range of choice is from 2 to 10 clusters.

The remaining clustering algorithms will be evaluated apart without a curve showing all CVIs variations, since their number of clusters are not a easy parameter to control. In this case K-Means will be invoked with the number of clusters defined by the original labels of the data sets and compared with DBSCAN and HDBSCAN for each data set.

Before testing the CVIs, the clustering algorithms are tested in toy data sets to evaluate its performance, i.e. blobs, moons and circles.

Noise in DBSCAN and HDBSCAN will be assigned to a new cluster composed of noise only.

According to the CVIs theory, the table I was built to track which the desired behavior for each CVI.

TABLE I

CVIs DESIRED BEHAVIOR.

CVI	Value for best number of clusters
CH	Maximum
Dunn	Maximum
DB	Minimum
PBM	Maximum
DBCV	Maximum

IV. DISCUSSION

A. Comparison between clustering algorithms in toy data sets

A demo of how each clustering algoirhtms works can be seen using toy datasets. There were used 10th dimensional blobs with 3 main clusters, double moons and double circles. The results can be seen in the figures 4, 5, 6 using Principal Component Analysis (PCA) to reduce the data for 2D.

In blobs K-means were used with 3 clusters, DBSCAN with $\epsilon = 0.5$ and HDBSCAN with a minimum number of clusters of 3. For the moon and circle data set those values were: 2, 0.1 and 2; respectively.

Regarding figure 4, the blobs are well clustered with each algorithm, despite not labeled in the same order (which is not a problem).

Fig. 4. Clustering algorithms applied to blobs toy data set.

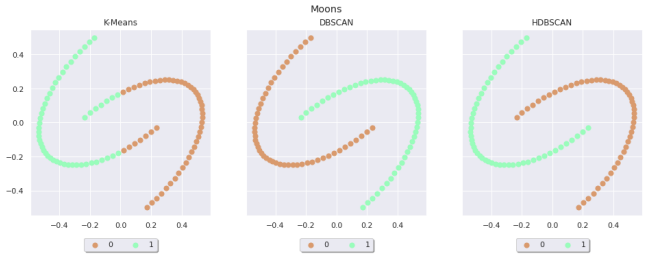

Fig. 5. Clustering algorithms applied to moons toy data set.

Figure 5 shows that K-Means had difficulties in the moon data set, since some points in each moon border mixed with each defined cluster. But in this case DBSCAN and HDBSCAN did not had much trouble to clusterize. The only problem was the the ϵ value was hardly sintonized to represent the results shown.

In the figure 6 the DBSCAN shown the worst value, despite synthonizing its ϵ value did not worked as intended. The same problem from the previous dataset remains to K-Means and HDBSCAN again showed the best result.

B. Comparison between clustering algorithms in real data sets

In this subsection the results shown regards a single traning for each clustering algorithm in the Iris, Wine and Control data set, thus showing the comparison between the CVIs for each clustering algorithm. The figures 7, 8 and 9 show the results of the simulation.

In the Iris data set the proximity of 2 labels resulted in HDBSCAN confusing the two clusters, but with good separation between the furthest clusters. K-Means with 3 as the number of clusters showed results more similar to the PCA plot in 1. The DBSCAN had the worst result with lots of noise.

In the Wine data set the K-Means with 3 clusters showed high similarity with the PCA plot in 2. In this case HDBSCAN had a worst performance with 2 main clusters and lot more noise. DBSCAN were not capable to separate the data in clusters even with hyper parameter tuning.

Same for the Control data set regarding DBSCAN and HDBSCAN, but in this case the last had less noise, but were capable of tracing only 2 clusters and DBSCAN were not capacble of clustering. K-Means with 3 clusters set had good results and was capable of separating the right and left blob.

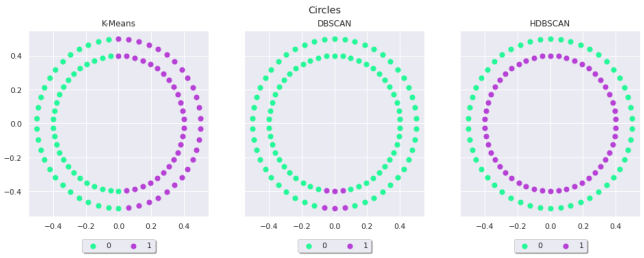

Fig. 6. Clustering algorithms applied to circles toy data set.

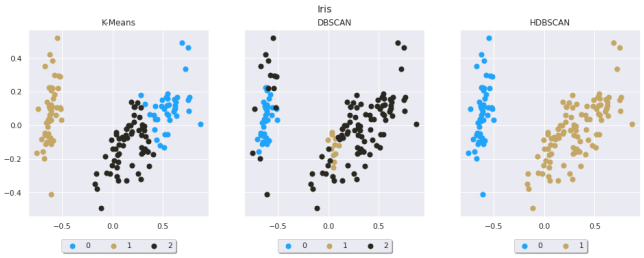

Fig. 7. Clustering algorithms applied to Iris.

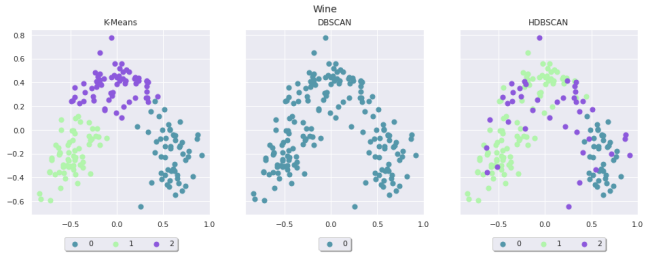

Fig. 8. Clustering algorithms applied to Wine.

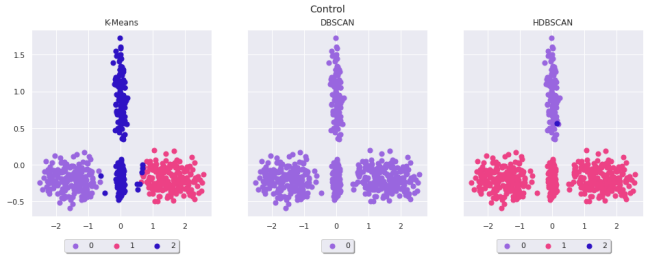

Fig. 9. Clustering algorithms applied to Control.

The tables II, III and IV compare the results for each CVI for each data set were the green values are the best results for the clustering algorithm used. In the Wine and Control data the DBSCAN algorithm were not capable to set more then 2 clusters thus not returning any CVI.

TABLE II
CVI OF DIFFERENT CLUSTERS ALGORITHMS APPLIED TO IRIS DATA SET.

Algorithm	CH	Dunn	DB	PBM	DBCV
K-Means	358.5	0.069	0.76	0.945	-0.211
DBSCAN	0.89	0.05	3.8	0.006	-0.77
HDBSCAN	353.36	0.358	0.48	0.82	0.48

TABLE III
CVI OF DIFFERENT CLUSTERS ALGORITHMS APPLIED TO WINE DATA SET.

Algorithm	CH	Dunn	DB	PBM	DBCV
K-Means	83.37	0.189	1.305	0.264	-0.45
HDBSCAN	33.99	0.126	2.828	0.14	-0.21

Looking the tables, each CVI showed a preference to a particular clustering algorithm, whether K-Means or HDBSCAN – since DBSCAN were not good in any case. But overall the indexes points K-Means as a better clustering algorithm for those cases.

C. Defining the number of clusters

Using the K-Means algorithm the exact number of clusters could be assigned, so when applying the algorithm for all data sets and evaluating the CVIs values the plots in figures 10, 11 and 12 could be traced.

In the case of the Iris data set, the DB index find its minimum in 2, close to the ideal number of clusters i.e. 3, but since the PCA plot for this data set in figure 1 show proximity between two clusters, this result is acceptable. The others indexes that are evaluated in the maximum value also finds its peak close to 2, except the PBM that extends to 3. The main detail is that the Dunn index gets really low right after 2.

In the case of the Wine data set since the swarm in this data set from the PCA plot 2, the ideal number of clusters of 3 can be confused with 2 or more, thus the indexes corresponding as such, i.e. with Dunn and DBCV with peak in 4, CH and PBM with peak in 2 or 3 and DB with valley in 3, thus DB representing the best index for this data set.

In the case of the Control data set, since the original disposition of the data accordingly to the PCA plot 3 is very confusing, i.e. with the 6 original labels with tendencies to 3 or 4 clusters, the CVI curves are also disperse. Looking for each CVI, CH and PBM are close together such in the previous case and show a peak close to 2-3, that is an acceptable value considering the figure 3. DB showed its minimum value at 2, that is also an acceptable result. In the furthest side DBCV and Dunn showed the worst results with an ascending peak that did not stopped in 10 and tends to increase, thus not representing a good metric for this data set.

V. CONCLUSIONS

The cluster valiadtion index present itself as a good metric to choose the ideal number of clusters to apply in a given data set. These metrics gives a degree of freedom to help academics and professionals in the clustering area to define more precisely and solid analysis based in long used metrics.

REFERENCES

[1] Abdoli, Javad, Ming Jia, and Jianglei Ma. "Filtered OFDM: A new waveform for future wireless systems." 2015 IEEE 16th International

TABLE IV

CVI OF DIFFERENT CLUSTERS ALGORITHMS APPLIED TO CONTROL DATA SET.

Algorithm	CH	Dunn	DB	PBM	DBCV
K-Means	388.29	0.249	1.368	2.27	-0.2
HDBSCAN	23.95	0.21	1.89	0.58	NaN

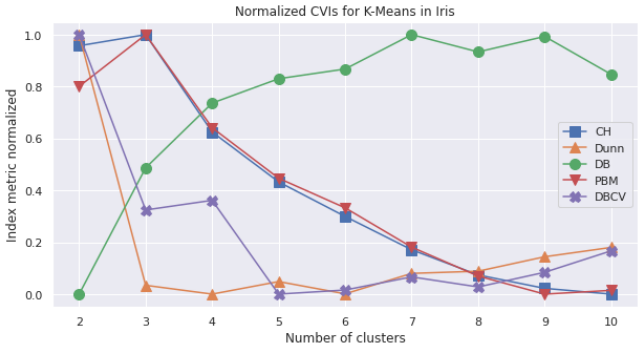

Fig. 10. Normalized CVI curves for K-Means applied to Iris data set.

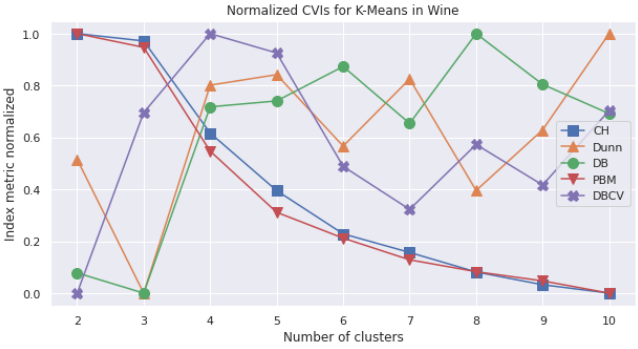

Fig. 11. Normalized CVI curves for K-Means applied to Wine data set.

Workshop on Signal Processing Advances in Wireless Communications (SPAWC). IEEE, 2015.

[2] Halkidi, Maria, and Michalis Vazirgiannis. "Clustering validity assessment using multi representatives." Proceedings of the Hellenic Conference on Artificial Intelligence, SETN. 2002.

[3] Pakhira, Malay K., Sanghamitra Bandyopadhyay, and Ujjwal Maulik. "Validity index for crisp and fuzzy clusters." Pattern recognition 37.3 (2004): 487-501.

[4] Krishna, K., and M. Narasimha Murty. "Genetic K-means algorithm." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 29.3 (1999): 433-439.

[5] Khan, Kamran, et al. "DBSCAN: Past, present and future." The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014). IEEE, 2014.

[6] McInnes, Leland, John Healy, and Steve Astels. "hdbscan: Hierarchical density based clustering." J. Open Source Softw. 2.11 (2017): 205.

[7] Pasa, Leandro Antonio. "Contribuição ao estudo de fusão de mapas auto organizáveis de Kohonen com ponderação por meio de índices de validação de agrupamentos." (2016).

[8] Géron, Aurélien. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. " O'Reilly Media, Inc.", 2019.

[9] Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. Introduction to data mining. Pearson Education India, 2016.

[10] Asuncion, Arthur, and David Newman. "UCI machine learning repository." (2007).

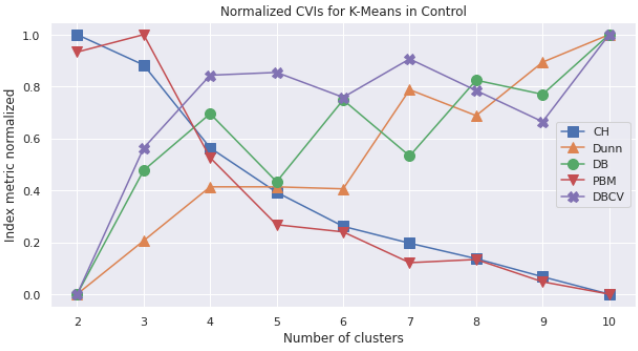

Fig. 12. Normalized CVI curves for K-Means applied to Control data set.