Cluster Validation: Improving Stability Measurement

Carlos Amílcar Sánchez Rosa
Department of Information Systems and Applications (ISA)
National Tsing Hua University
Hsinchu City, Taiwan 300
Email: csanchezrhn@gmail.com

Abstract

Several Techniques and algorithms have been implemented and designed for clustering, but since the results comes from describing hidden structures of unlabeled data, the validation of the results is a hard task, many methods to try to solve this problem have been proposed, most of them categorized as Internal and External indexes for measuring similarity and dissimilarity of clusters, cluster results or algorithms, one of the most popular ones is Stability, that it is measured by calculating the expectation of the distance of a perturbed version of the original data, but this method becomes difficult to run since when the data becomes so large size, the performance of the Stability measurements is affected, sometimes requiring several weeks to get a result depending of the nature of the data, in this work we present a variation of the stability algorithm named Critical Stability that focus on the removal of the set composed with different results, that are mostly random generated replacing it by a direct calculation of a modified set improving the performance and precision, for validating this new algorithm we generated Gaussian artificial datasets with known patterns and compare time and precision of both the original method and the new one.

1 Introduction

Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters), Several Techniques and Algorithms have been implemented and designed for this purpose [1].

Cluster Analysis is concept that was first introduced for Anthropology in 1932 by Driver and Kroeber [2] and introduced in Psychology by Zubin in 1938 [3] and Tryon in 1939 [4], famously used by Cattell [5] for trait theory classification in personality psychology and finally widely used in Medicine, Biology, Geography, Data Mining, Pattern Recognition etc.

Cluster Validation have been considered as an unsolved problem since early 80s The validation of clustering structures is the most difficult and frustrating part of cluster analysis. Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage. [6], but still an important part of Cluster Analysis since it helps to avoid finding patterns in noise, Compare Clustering Algorithms, and Compare clusters or sets of them, as an attempt to solve this problem several similarity measurements have been proposed [7] and generally classified as External and Internal Indexes, where external uses foreign information from the dataset to measure metrics while internal uses only the known datset.

Cluster Stability is a concept that have been already studied and applied in several studies and a general algorithm have been already implemented [8], however implementing the Stability Measurement algorithm can be easy but slow to test in huge datasets, in this research we present the Critical Stability that it is an improved version to the generalized method on its performance and precision that helps to measure stability on big datasets in faster times, from now on we will call the generalized stability alorithm just ”Original algorithm”.

2 Related Work

For improving of the original stability measurement algorithm from Ulrike von Luxburg [8], the main line of work related to this research, is the successful implementation of the large variety of methods that has been devised to compute stability scores and use them for model selection, the most popular ones use gene expression data, they focus on the minimization of the instability in a cluster result, which is calculated as an average distance between different cluster results against the original one. Another line of related work is the implementation of the method for bigger datasets, that have been becoming popular in recent years since the big data problems appeared, however these implementations doesnt improve the algorithm for general cases working for specific datasets, and still not so many published until these days.

2.1 Stability Measurement Variations

A popular variation that appears in the literature is the use of similarity using Jaccards coefficient [9], maximizing the similarity between different cluster results, but the algorithm itself remains the same, other approaches focus on changing the method for comparing the different cluster results, one of the most popular ones is dividing the dataset into a training and learning set, where this new results can be compared to a classifier such as LDLA [10] or Nearest Centroid Classifier [11], the original stability measurement focus on the use of k-means to find the most stable k, and further they show that the algorithm can be easily extended to any clustering method, so other related work implements the algorithm with other clustering algorithms such as the EM algorithm (Expectation Maximization Algorithm) [12].

2.2 Cluster Stability in a Large Scale Phylogenetic Analysis

In recent years that Big data is being presented, some studies that try to tackle the performance issue for stability have tried to solve the issue for specific gene datasets and used phylogenetic analysis to determine the stability [13]. Land, Fizzano and Kodner suggests a reliable method for determining relationships between communities using phylogenetic analysis of shotgun sequences for reasonably well characterized communities such as human gut microbiomes, getting improvement in performance for their big dataset, but still a method that only focus on the nature of gene expression data.

This research focus on presenting a variation of the original algorithm named Critical Stability that focus on the removal of the set compound with different results, that are mostly random generated replacing it by a direct calculation of a modified set improving the performance.

3 Methodology

The methodology for this research have been divided into six parts, starting by formally defining cluster stability, showing how to implement the original algorithm, defining the used dataset with known results for proving the algorithm works, analyzing and observing characteristics and behaviors from unstable clusters, presentation of the new variation (Critical Stability) and finally validating the new method showing that we can get the same results but with a better precision and performance.

3.1 Stability Definition

Giving a Dataset $S = \{X_1, X_k, ..., X_n\}$ that as an input to a function f that generates a clustering result $C_k(S_n) = \{C_1, C_2, ..., C_K\}$ where K represents the number of cluster. The instability of a clustering algorithm is defined as the expected distance between two clusterings $C_K(S_n)$ and $C_k(S'_n)$ of size n, this expected distance is summarized in Equation 1.

$$Instab(K, S_n) := E\left(d\left(C_K(S_n), C_K(S'_n)\right)\right) \qquad (1)$$

The expectation is taken with respect to the drawing of the two samples generally a regular dataset and a perturbed version of it. In practice, a large variety of methods has been devised to compute stability scores and use them for model selection. On a very general level they work as shown in Algorithm 1. Since a Distance formula is most general used than a Similarity function, the algorithm gives an Instability value instead of Stability, to get the Stability value the distance Method can be replaced by any similarity function.

Algorithm 1 Instability

```
Input: a set $S$ of data points, a clustering algorithm $A$ that takes the number $k$ of clusters, $k_{max}$ , $b_{max}$
Output: most stable $K$
for $k := 2, ..., k_{max}$ do
    Generate perturbed versions $S_b\left(b = 1, ..., b_{max}\right)$ of the original data set
    for $b := 1, ..., b_{max}$ do
        $C_b = A\left(S_b, k\right)$
    end for
    for $b, b' := 1, ..., b_{max}$ do
        Compute pairwise distances $d\left(C_b, C'_b\right)$
    end for
    $C_b : \widehat{Instab}\left(k, n\right) = \frac{1}{b_{max}^2} \sum_{b,b'=1}^{b_{max}} d\left(C_b, C_{b'}\right)$
end for
Choose the parameter $k$ that gives the best stability, in the simplest case as follows
$K := \underset{k}{argmin} \widehat{Instab}\left(k, n\right)$
```

3.2 Datasets

As presented in previous papers, Gaussians dataset have been widely used for validating models, methods and cluster evaluation techniques before using on real datasets like [14], [15] and [16], for following the previous works we created the same five datasets presented with the same parameters presented by them, and added some other datasets for our own testing, Table 1 presents the description of all the datasets. All datasets have been generated based on a Gaussian distribution defined as Equation 2.

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2 / 2\sigma^2} \quad (2)$$

Name	Clusters	Samples	Features
Uniform1	1	60	600
Gaussian1	1	60	600
Gaussian3	3	60	600
Gaussian3b	3	60000	600
Gaussian4	4	200	2
Gaussian4b	4	1000	2
Gaussian5 $(\lambda = 2)$	5	250	2
Gaussian5 $(\lambda = 3)$	5	250	2
Gaussian5-X$(\lambda = 3)$	5	X	2
Mouse	3	1500	2

Table 1: Generated Gaussian Datasets

Uniform1 and Gaussian1 are two datasets generated in order to evaluate the behavior of the clustering methodology when applied to data known not to contain distinct sub-populations. We considered both the uniform and the Gaussian distributions, as they represent rather different generating processes, and we were interested in examining how the stability would look like.

Gaussian3 is a 3-cluster dataset with 600 features and 60 samples presented in previous papers but we take in consideration the curse of dimensionality [17], and generated a Gaussian3b Dataset with same characteristics but with 600,000 samples.

Gaussian4 and Gaussian5 are a 4-cluster and a 5-cluster dataset respectively, they contain 2 dimension each for being able to visualize in charts as shown in Figure 1, with the same diagonal covariance matrix $\sum = 0.25I$, the clusters where centered at the four corners of a square with side length $\lambda = 2$, while Gaussian5 extra cluster was centered in $(\lambda/2, \lambda/2)$, a second version Gaussian4b with more samples was generated, while Gaussian5 had an extra version with $\lambda = 3$, and this last, had several variations changing the amount of samples, the generated versions where Gaussian5-200 increasing by 200 samples until Gaussian-2600. Figure 2 presents a visual representation of these datasets after running k-means with their corresponding k.

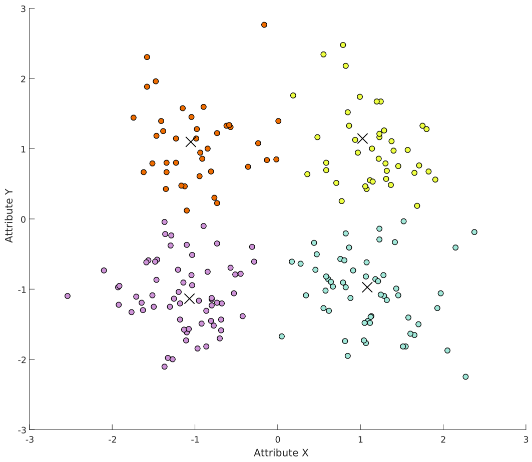


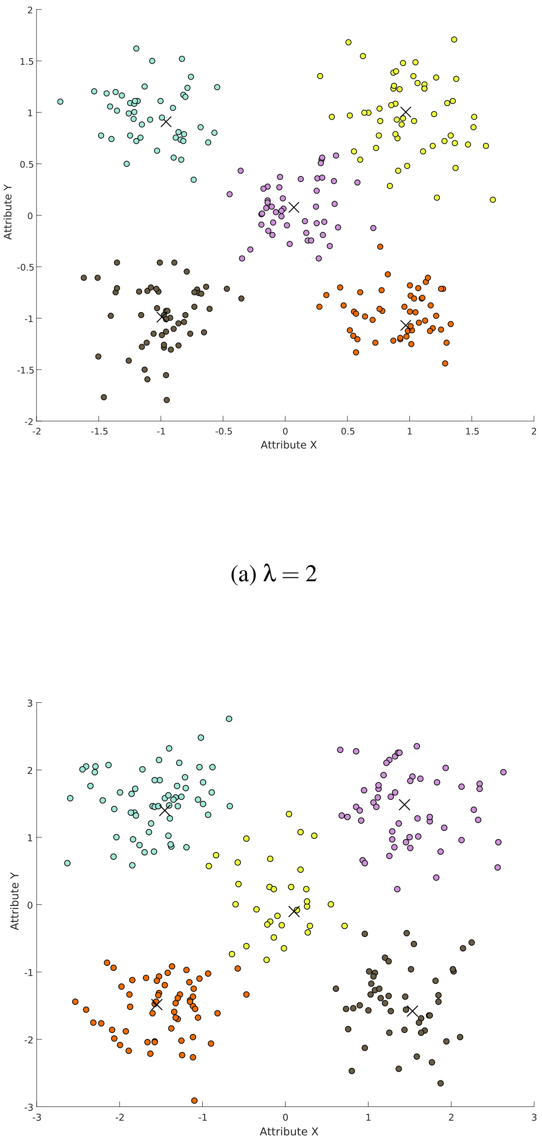
Fig. 1: Gaussian4 visual representation after k-means with $k = 4$

Mouse is a 3-cluster dataset with 3 clusters centered in $(-1,1), (1,1)$ and $(-0,-0.5)$, while the standard deviation for the first 2 clusters was $\sigma = (0.05, 0.05)$ and for y=the last cluster $\sigma = (0.25, 0.25)$ to create the shape of a mouse cartoon face. Figure 3 shows a visual representation of Mouse Data after running dbscan cluster algorithm.

3.3 Algorithm Implementation

For implementing the algorithm in this research, we used MATLAB and strictly followed the generalized algorithm, since the algorithm leave as open the comparison method to use, we used four different method and choose the one who reflects better the observations of the sample data set. The four methods tested in this research are Single-link, Complete-link and Average distance [18] as dissimilarity methods and Jaccards Coefficient as a similarity method, Similarity have been used before for other studies [9], but it changes the results to a Stability Score instead an instability one, the best K is the one with the lowest instability, but changing the measurement to stability, changes the criteria to search for the one with the highest stability, For the dissimilarity Methods Average distance was the one who gives the correct results tested on Gaussian4 Dataset as shown in figure 4.

Jaccards Similarity was used to calculate Stability on Gaussian4 Dataset as shown in Figure 5, with the particularity that $k = 1$ is always the most stable k since the only variation through different bs will just be the position of the center of the cluster, also is interesting to see that the max stable value is approximately 0.9 since the perturbation percentage use was 10%, this also shows that the perturbation percentage used is important for reading the results.



(a) $\lambda = 2$

(b) $\lambda = 3$

Fig. 2: Gaussian5 visual representation after k-means with $k = 5$

Based just on the visual looking of Stability vs Instability, we decided to use Jaccards Similarity for all the experiments since it have been used before by previous works and give a direct Score to Stability based on similarity and no distance, but using the distance as a method should give similar results, but since even in the generalized method, which method to use is open, a further research in its comparison ca be done, and we leave this open for future research.

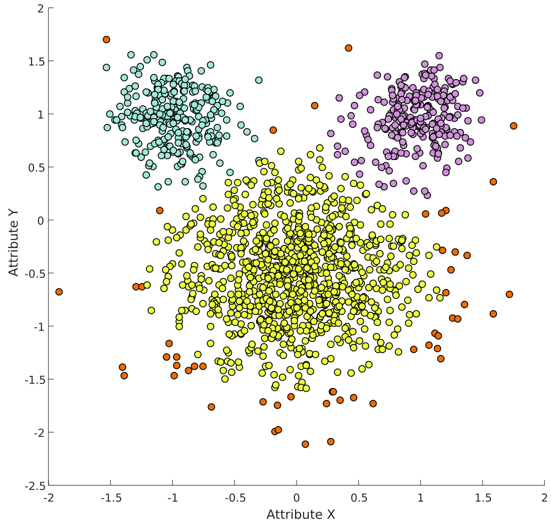


Fig. 3: Mouse data visual representation after dbscan

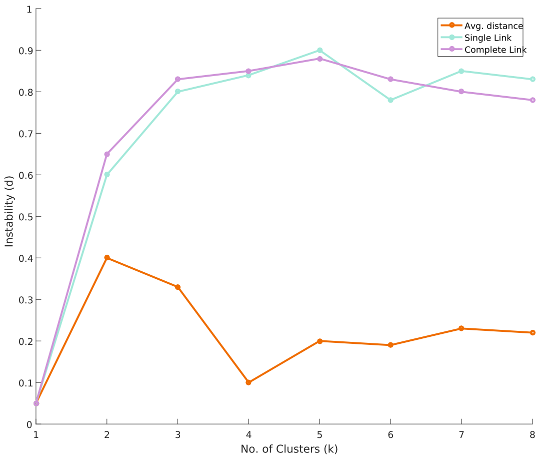


Fig. 4: Single-Link, Complete-Link and Average Distance used to calculate Instability of Gaussian4.

3.4 Unstable Cluster Analysis

An instable cluster is the one that can disappear by merging to other clusters or divide itself into several clusters while new data is added or removed to the dataset, For understanding the characteristics of an unstable cluster Ulrike von Luxburg presents two sample data sets showing how a number of cluster $k = 2$ and $k = 5$ are the most unstable K's for these sample datasets [8], as it can be seen on Figure 6, in the case of $k = 2$, we could have two completely different results, depending on which is the initial points for

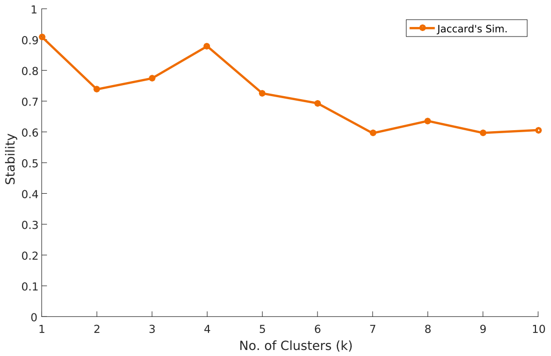


Fig. 5: Jaccard's Similarity used to calculate Stability of Gaussian4.

the k-means algorithm and because of this uncertainty of which of the two possible results we will get, we can see how the instability of this case should be really high, while in the case of $k = 4$ the instability should be really low, since in most cases we will get exactly the four clusters that are visible for human eyes.

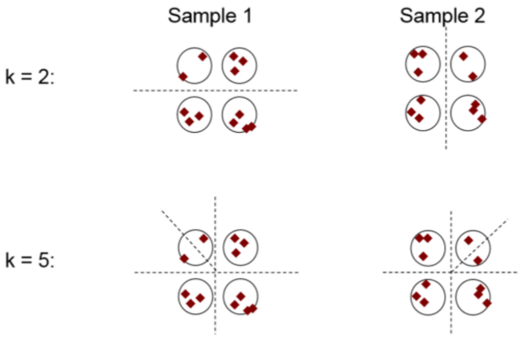


Fig. 6: Most instable k's for a sample data set. [8]

For the case of k = 5 and higher Ks, what will happen is that the instability will become higher since at least one of the four visible clusters will have to be divided to accomplish with a K greater than 4. After several observations on unstable cluster, we were able to identify some specific areas that affect the clusters, these areas were named Critical Area, a visual representation of them are presented in Figure 7, Critical Area B is easy to calculate since it can be defined as the middle point between the 2 clusters, while Areas A and C can be generated by 90 degrees the two cluster center based on the 2 cluster centers.

Since the Critical points Areas for N-dimensions can be complicated to calculate in a formula by just rotating the cluster center in 90 degrees, we present an Algorithm that calculates all the Critical Area center points for N-Dimensions in Algorithm 2.

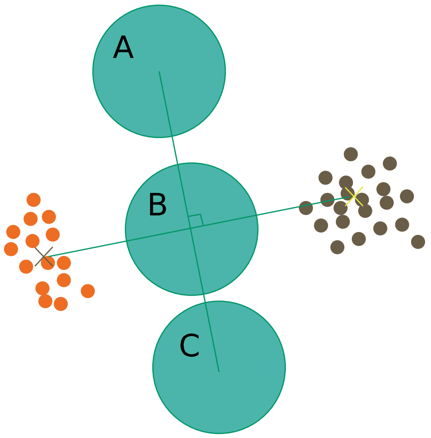


Fig. 7: Critical Areas between 2 Clusters

We can summarize the Critical Points algorithm into Equation 3.

$$CtrPts\,(CP) = midPts\,(CP) \cup PA\,(CP) \cup PC\,(CP) \tag{3}$$

Where CP is a set of pairs for every Cluster Center point with size k and $midPts$ represents a set of all the middle points ($\frac{A+B}{2}$) of every Pair (A,B) from CP , PA and PC are the rotated 90 degree points of every pair in CP on $midPts$ where every dimension of the point can be represented by the recursive equations 4 and 5.

$$PA_i\,(d) = \begin{cases} d = 1, ctxs_i\,(1) - (ctxs_i\,(2) - midPts_i\,(2)) \\ d > 1, midPts_i\,(d) + PA_i\,(d-1) - midPts_i\,(d) \end{cases} \tag{4}$$

$$PC_i\,(d) = \begin{cases} d = 1, midPts_i\,(1) - (ctxs_i\,(2) - midPts_i\,(2)) \\ d > 1, midPts_i\,(d) + PC_i\,(d-1) - midPts_i\,(d) \end{cases} \tag{5}$$

3.5 Critical Stability

Based on the general Equation 1 of Stability measurement and the Unstable cluster analysis we can define S''_n as the only perturbed copy with data extracted from a Gaussian equation setting $\mu = ctrPts$, and reoving the E to only one distance calculation, with this assumptions we can redefine

Algorithm 2 Critical Points

```
Input: a set $S_n$ of data points of size $n$ and dimensionality $N_d$, a set of points $ctxs$ that represents the center of the clusters
Output: a set of points $CtrPts$
$CtrPts = [\emptyset]$
for $i := 1, ..., |\, ctxs \,|$ do
    for $j := i+1, ..., |\, ctxs \,|$ do
        if $i \neq j$ then
            $midPt = \langle \emptyset \rangle$
            for $d := 1, ..., N_d$ do
                $midPt = \langle midPt, (ctxs_i(d) + ctxs_j(d))/2 \rangle$
            end for
            if $j = i+1$ then
                $midPts = [midPts, midPt]$
                $PA = \langle ctxs_i(1) - (ctxs_i(2) - midPt(2)) \rangle$
                $PC = \langle midPt(1) - (ctxs_j(2) - midPt(2)) \rangle$
                for $d := 2, ..., N_d$ do
                    $PA = \langle PA, midPt(d) + PA(d-1) - midPt(d) \rangle$
                    $PC = \langle PC, midPt(d) + PC(d-1) - midPt(d) \rangle$
                end for
                $CtrPts = [CtrPts, PA, PC, midPts]$
            end if
        end if
    end for
end for
```

Algorithm 3 Critical Instability

```
Input: a set $S$ of data points, a clustering algorithm $A$ that takes the number $k$ of clusters, $k_{max}$
Output: most stable $K$
for $k := 2, ..., k_{max}$ do
    $C_s = A(S, k)$
    $C_{s''} = A(S \cup gaussianPerturb(CtrPts(S)), k)$
    Compute pairwise distances $d(C_s, C_{s''})$
end for
Choose the parameter $k$ that gives the best stability, in the simplest case as follows
$K := \underset{k}{argmin} \widehat{Instab}(k, n)$
```

Equation 1 into Equation 6, and finally simplified Algorithm 1 into Algorithm 3.

$$CtrInstab(K, S_n) := d\left(C_K(S_n), C_K\left(S''_n\right)\right) \tag{6}$$

3.6 Complexity Analysis

From Equation 1, the Complexity Analysis of this calculation is equal to the Complexity Analysis of the Expectation Value Formula presented in Equation 7.

$$E(X) = \frac{x_1p_1 + x_2p_2 + ... + x_bp_b}{p_1 + p_2 + ... + p_b} = \frac{x_1p_1 + x_2p_2 + ... + x_bp_b}{1} \tag{7}$$

Where X is a random variable with b possible versions and p_i is the probability for each version i, the sum of the probabilities should give 1, thats why the formula can be simplified. Finally with a simple look is easy to say that the complexity of Equation 7 is $O(|X| \cdot b)$ and for Equation 1, $|X|$ becomes the size of the dataset $|S_n| = n$ giving us a complexity as $O(n \cdot b)$ but we also have to consider that each perturbed version of S_n will be cluster by the Algorithm A, so we need to add this complexity to the final Complexity as it is shown in Equation 8.

$$O(Instab(K, S_n)) = O(b \cdot n \cdot O(A(S_n, K))) \tag{8}$$

In normal Big O notation, it could be consider that b is just a constant that could be removed from the complexity analysis, but in the end is a variable that it's value can't be predicted because it depends on the nature of the data, and since this is the variable that most delays the complexity time of the original algorithm, in the rest of the paper we explain it's importance and how to remove it from the equation.

After giving the mathematical definition of how to calculate the Critical points on Equation 3, we can present the complexity time for it as $O(midPts) + O(PA) + O(PC)$ where all of the complexities are $O(n)$ leaving in the end a total complexity of $O(3 \cdot n) = O(n)$. Knowing the complexity for $ctrPts$ we can easily calculate the entire complexity for Algorithm 3 presented in Equation 9, showing an improvement in the performance since the removal of variable b, that theorically will give an incredible improvement in time, since in most of the cases b is always going to be a high number for improving the precision of the results.

$$O(CtrInstab(K, S_n)) = O(n \cdot O(A(S_n, K))) \tag{9}$$

3.7 Validation

For validating our method, we run the stability algorithm on a dataset and compare the results with the one given by the critical stability algorithm, we used the Gaussian5 dataset with $\lambda = 3$ to perform all our tests, since this is the datasets with more clusters and a bigger sparsity of data, critical stability cant give 100% results, since it still needs to do a small random cluster in the critical areas, and k-means as an algorithm A will not be 100% precise either, and the original method since is mostly random based, it make mandatory the use mean average precision measurement defined by Equation 10.

$$MAP@n = \sum_{i=1}^{N} ap@n_i/N \tag{10}$$

Equation 11 presents how to calculate $ap@n_i$ where $P(k)$ is the precision at a cut-off k in the item list.

$$ap@n_i = \sum_{n}^{k=1} P(k)/min(n,m) \tag{11}$$

For measuring accuracy we simply used the average error of the collected sample results form the algorithm, the equation used for calculating error is presented in Equation 12.

$$Error = (X_i - \mu)/X_i \tag{12}$$

4 Experiment

In this section we present the results of calculating time, precision, and accuracy of the critical Stability measurement and several number of copies b for the Original Method, and divided into 2 sections, were we present the importance of using a big b and the final comparison respectively.

4.1 Importance of a big number of copies b

In this section, we want to present how important is to use a big b, by calculating performance and precision of the original method while increasing the number of copies, for doing this task we set a range of copies stating with only 1 copy, and increasing later to 50 until we reach 950 by 50 copies steps, and for calculating precision and accuracy we used 10 samples per b, we also want to show that the amount of data represents an important role for the success of the method, Figure 8 presents the results for a small disperse dataset Gaussian-200 with $\lambda = 3$.

In the case of Gaussian5-200 we can't see any pattern since the size of the data is small, the precision varies a lot and don't become stable at increasing the number of copies, but the acuracy is always low, the complexity time of the original algorithm is so slow that it requires several weeks to be able to generate this results, Figure 10 presents the real time it requires to calculate the stability of a dataset while increasing the number of copies, the dataset used for this chart was Gaussian5 with $\lambda = 3$.

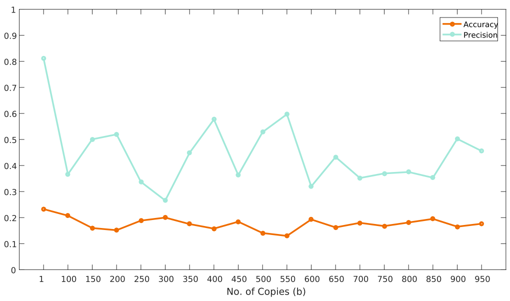


Fig. 8: Precision and Accuracy for Gaussian5-200

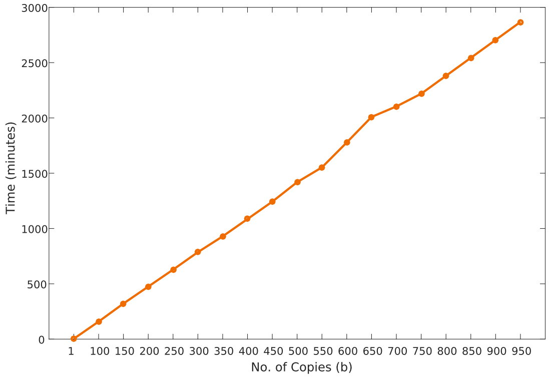


Fig. 9: Time calculations for stability measurement in Gaussian5 with $\lambda = 3$

Trying the same measurements on Gaussian5 with $\lambda = 3$ that contains 250 samples (50 per cluster), we can get different results as it canbe shown on Figure , where is easier to appreciate that after 600 copies, the stability and the accuracy becomes stable, that means that we can get trustful results using a $b > 600$, but still a really big b, where is easy to see that our method is faster since it's complexity time is similar to have $b = 1$ but with a better precision.

4.2 Method Comparison

For comparing this results to our method we choosed 4 different $b's$ from the original method to be compared with the Critical Stability method, the chosen b are $(1, 200, 400, 600, 800)$. The comparison of performances it is shown in Figure 11.

Figure 12 shows with more detail the time for critical stability since the comparison with big $b's$ is too much , and

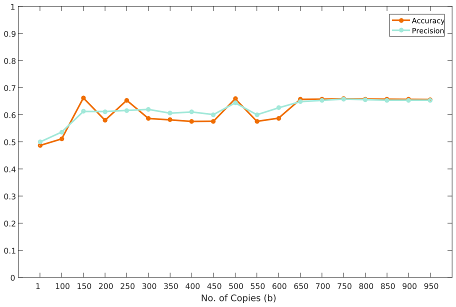


Fig. 10: Precision and Accuracy for Gaussian5 with $\lambda = 3$

to the precision comparison it is shown on Figure 13 using several versions of Gaussian5 with different sample sizes, this comparison is to check if the number of samples affect the precsion of the method, Accuracy was not considered in this research since is still an open problem to know the exact answer of each stability value, just to have an idea we used as an expected value, the esult of the original method with a really high value $b = 10000$.

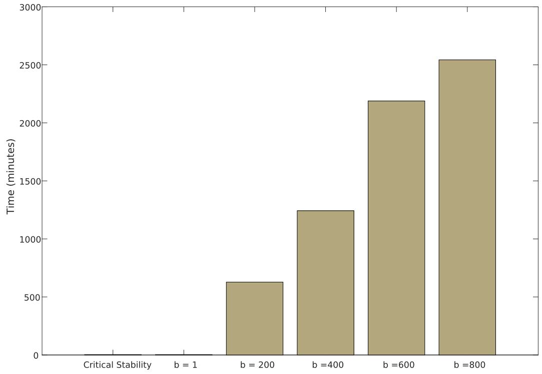


Fig. 11: Time comparison between Critical Stability and original method with several b

It seems that increasing the number of samples in the data increases the precision of all the methods, this can be because the clusters are been strengthen while increasing the data size, and still the Critical Stability precision is the best compared to all these cases.

5 Conclusion

In this research we presented the Critical Stability Algorithm that is an improved version of Stability Measurement Algorithm [8], and we made the time comparison in a dataset showing that the time complexity have been reduce, the real execution time is similar to the Original Method with $b = 1$, but with a precision similar to the best b without affecting performance.

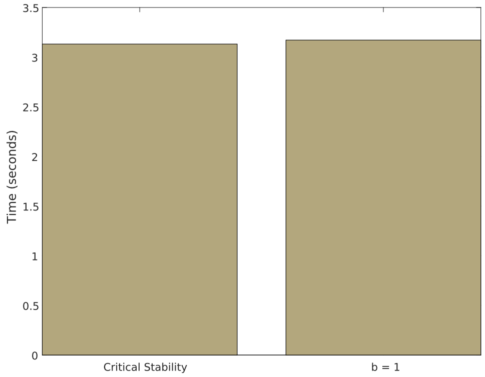


Fig. 12: Time comparison between Critical Stability and original method with $b = 1$

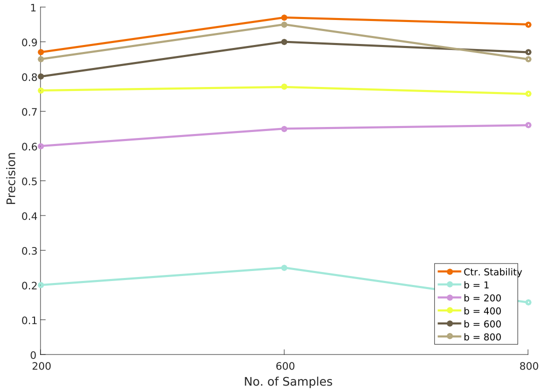


Fig. 13: Precision and Accuracy for Gaussian5 with $\lambda = 3$

References

[1] A., P., 2002. “Data clustering techniques”. In Qualifying Oral Examination Paper.

[2] Driver, H. E., and Kroeber, A. L., 1932. “Quantitative expression of cultural relationships”. In University of California Publications in American Archaeology and Ethnology, pp. 211–256.

[3] Zubin, J., 1938. “A technique for measuring like-mindedness”. In Journal of Abnormal and Social Psychology, pp. 508–516.

[4] Tryon, R. C., 1939. “Cluster analysis: Correlation profile and orthometric (factor) analysis for the isolation of unities in mind and personality”. In Edwards Brothers.

[5] By Cattell, R. B., 1943. “The description of personality: basic traits resolved into clusters”. In The Journal of Abnormal and Social Psychology, pp. 476–506.

[6] Jain, A. K., and Dubes, R. C., 1988. “Algorithms for clustering data”. Prentice-Hall, Inc.

[7] Erndira Rendn, Itzel Abundez, A. A., and Quiroz, E. M., 2011. “Internal versus external cluster validation indexes”. In INTERNATIONAL JOURNAL OF COMPUTERS AND COMMUNICATIONS.

[8] von Luxburg, U., 2010. “Clustering stability: An overview”. In Found. Trends Mach, pp. 235–274.

[9] A, Ben-Hur, A. E., and Guyon, I., 2002. “A stability based method for discovering structure in clustered data”. In Proceedings of the Pacific Symposium on Biocomputing, pp. 6–17.

[10] Dudoit, S., and Fridlyand, J., 2002. “A prediction-based resampling method to estimate the number of clusters in a dataset”. In Genome Biology 3, p. 0036.1 0036.21.

[11] T. Lange, V. Roth, M. L. B., and Buhmann, J. M., 2004. “Stability-based validation of clustering solutions”. In Neural Computation 16, p. 12991323.

[12] Grun”, B., and Leisch, F., 2004. “Bootstrapping finite mixture models”. In Antoch, J. (ed.), COMPSTAT, p. 11151122.

[13] Tyler A. Land, P. F., and Kodner, R. B., 2016. “Measuring cluster stability in a large scale phylogenetic analysis of functional genes in metagenomes using pplacer”. In IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS.

[14] Vinh, N., and Epps., J., 2003. “Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data”. In Machine Learning, pp. 84–91.

[15] Vinh, N., and Epps., J., 2009. “A novel approach for automatic number of clusters detection in microarray data based on consensus clustering”. In the Ninth IEEE International Conference on Bioinformatics and Bioengineering, pp. 84–91.

[16] Iam-on, N., and Garrett, S., 2010. “Linkclue: a matlab package for link-based cluster ensembles”. In Journal of Statistical Software, pp. 84–91.

[17] Bellman, R. E., 1957. “Dynamic programming”. In Princeton University Press.

[18] C. Manning, P. R., and Schtze, H., 2008. “Hierarchical clustering”. In Introduction to Information Retrieval, pp. 377–401.