Temporal sequence alignment for clustering longitudinal clinical data

Nuno Miguel Canhoto da Silva

Instituto Superior Técnico, Universidade de Lisboa, Portugal
nuno.da.silva@tecnico.ulisboa.pt

Abstract. Clustering is concerned with finding patterns in the data. A data set is given to an algorithm as input and groups of similar items are the output. The objects in a data set are gathered in groups (clusters), and the criteria to follow when computing such clusters is that objects in the same cluster should be more similar between each other than to items in different clusters. Our main concern in this study is the evaluation of the produced clusters, what measures can be used too validate them and in what situations should we use a specific measure. Here we present the three main criteria for validation (external, internal, relative) that can be followed and whose application can greatly improves the choice of a clustering structure. Although, we can find a lot of information in the literature about validation indices, there is not a lot of available programs to test them. For this purpose a Python library will be developed. With the help of this library, we expect to see an improvements on the results presented previously by Kishan in it's paper (AliClu), from where we will use a synthetic data set too present our results, as well as an analysis of the Reuma.pt data set, from the *Sociedade Portuguesa de Reumatologia*.

Keywords: clustering · internal validation · external validation · relative validation · python library · AliClu

Table of Contents

1 Introduction 3
2 Kishan's thesis 5
3 Related Work 5
3.1 External Clustering Validation Measures 6
3.2 Internal Clustering Validation Measures 8
3.3 Relative Clustering Validation Measures 9
4 Solution Proposal 13
5 Preliminary results 14
6 Work Schedule 17
7 Conclusions 18

1 Introduction

Clustering is a way to find relations in an underlying data set. It's goal is to find objects or items that are similar between each other and then assign them into groups (clusters). With the use of clustering methods we can get one or more possible partitions of our data, where each partition can be seen as a gathering of items of the same type or, if we are looking for classification, one partition would correspond to a label [3].

We can find numerous clustering methods and they are commonly divided into two big groups, those that are hierarchical and the partitional based ones, the main difference between them being that the former produces a nested series of partitions, while the later produces only one partition [2]. From these two approaches, come up other branches (see Fig. 1).

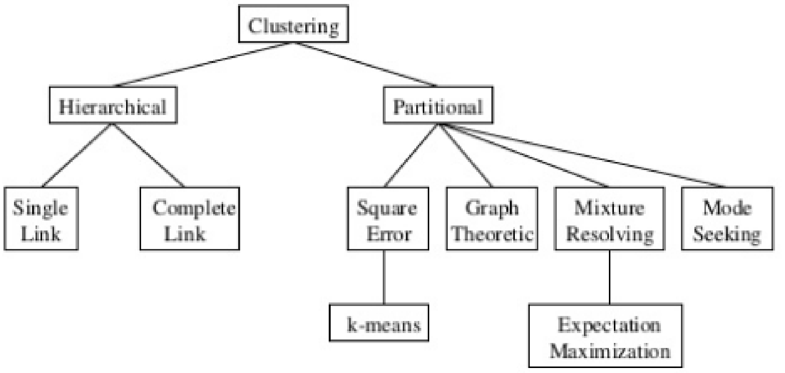


Fig. 1. Taxonomy of clustering approaches [3]

Agglomerative clustering follows a bottom-up approach. We start with a single object that will be merged with other objects to form clusters that will became larger until all objects are assigned to a cluster or a termination condition is met. To merge objects we need to have a method that tells us their similarity, if they are very similar they are merged. This method uses distance between objects to evaluate their similarity and several can be used.

Single link method for hierarchical clustering takes the distance between two clusters as the minimum distance between all pairs of patterns:

$$D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2) \tag{1}$$

The **complete link** takes the maximum distance of all pairwise distances between patterns:

$$D(c_1, c_2) = \max_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2) \tag{2}$$

The **average link** is the average distance between any member of a cluster to any member of another cluster:

$$D(c_1, c_2) = D(\frac{1}{\|c_1\|} \frac{1}{\|c_2\|} \sum_{x \in c_1} \sum_{x \in c_2} D(x_1, x_2) \tag{3}$$

Centroid link measures the distance between the centroid of one cluster and the centroid of another cluster:

$$D(c_1, c_2) = D((\frac{1}{\|c_1\|} \sum_{x \in c_1} x), (\frac{1}{\|c_2\|} \sum_{x \in c_2} x)) \tag{4}$$

Ward's method measures how much the sum squares will increase if we merge a cluster with another.

$$D(c_1, c_2) = \sum_{x \in c_1} (x - r_1)^2 + \sum_{x \in c_2} (x - r_2)^2 - \sum_{x \in c_1 2} (x - r_1 2)^2 \tag{5}$$

,

where r_1, r_2 are the centroid of cluster 1 and cluster 2 , respectively.

Since clustering is an unsupervised learning task, there is not information about what is to be accomplished or what relations to be found [4]. Therefore, one issue that arises is the evaluation of the clustering results. The clustering algorithm produces partitions where the elements present are close to each other, but we need to have a way of knowing if it made the right choice of clusters.

Well defined clusters follow two criteria: *i)* The individual clusters should be compact; ii) well separated from the external clusters.

One way to evaluate the results is through visual aid. When we have 2D data, it usually works well to look at the clustering results and check if the criteria of compactness and separation is met. However, if the data is multi-dimensional it is not possible to visualise the clusters, and so we require tools that will replace that visual aid.

To accomplish this validation we can make use of the metrics that have been suggested through out the years and are available in the literature [1], [4], [5]

This work has the objective of providing an extended list of metrics to validate clusters, as well as to continue and improve the contribution in [6] for a temporal sequence alignment algorithm to find clusters in medical data, that lacks in term of validating the clusters obtained.

We will go deeper in the works of [6] in section 2 and in section 3 we give a more thorough analysis of the related work on clustering validation measures, precisely, on three methods to evaluate quantitatively the clustering results, external validation, internal validation and relative validation. In section 4 a solution proposal e given as well as some preliminary results. Section 5 contains a work schedule that will help in the production of the solution. Finally, we will share some conclusion on the research and work done so far.

2 Kishan's thesis

Given the increasing availability of electronic medical records (EMRs) with longitudinal data, this data can be used for making clinical decisions so that physicians can choose personalized treatments for the patients. The work done sets out with the goal of finding an efficient way to cluster patients based on their temporal information from medical appointments. It is proposed the application of the the Temporal Needlman and Wunsch algorithm [8], a modified version of the original Needleman and Wunsch [7], to align discrete sequences with the transition time information between symbols. This symbols may correspond to a patient's therapy, overall health status, or any other discrete state. The obtained TNW pairwise scores are then used to perform hierarchical clustering. To find the best number of clusters and assess their stability, a resampling technique is applied After obtaining the clusters, several validation methods will be computed, such as Rand, Adjusted Rand, Fowlkes and Mallows, Jaccard, Wallace and Adjusted Wallace.

The algorithm was applied for the analysis of the rheumatoid arthritis EMRs obtained from the Portuguese database of rheumatologic patient visits (Reuma.pt). Like the previous work, identifying patterns and common features between patients in the Reuma.pt data set will be our final goal.

The AliClu is a very promising tool to analyse longitudinal patient data and unravel patterns that exist in clinical outcomes. But the validation of the resulting clusters being a very important step in identifying the patterns is not getting enough focus. Only external validation measures are considered and there was not a study done to see if they are the correct ones for the data we are dealing with. Therefore, we propose to go deeper and analyse thoroughly several more validation measures, concluding on the ones that are best suited for the data in Reuma.pt and achieve better results than before.

3 Related Work

On this section, our focus will be on discussing validation indices. We can find in the literature three main methods that will give us a measure of the quality of our clustering results [9], [19],[24].
The first type is based on comparison of partitions, the partitions to be compared consist of the one generated by the clustering algorithm and a given partition of the data (or a subset of the data). The approached is called *external validation.* A second approach is called *internal validation.* And is based on calculating properties of the resulting clusters, such as compactness, separation and roundness. It gets its name from the fact that it does not require any more information about the data. The third way is called *relative validation.* It's based on comparisons of partitions generated by the same algorithm with different parameters, or different subsets of the data. It does not require any more information except

the data itself.
In the sequel we will describe the fundamentals for each of the three cluster validity approaches and list the several indices that will be part of our work.

3.1 External Clustering Validation Measures

This approache idea is to test whether the clustering results have any structure or are they just random. To reach a conclusion, we use statistical tests. Based on the external criteria, we can evaluate the resulting clustering C, by comparing it to an independent partition of the data set, P. Considering this partitions $C = \{C_1, ..., C_k\}$ and $P = \{P_1, ..., P_l\}$ we run through the data set and for each pair we consider this terms:

- **SS**: if both points belong to the same cluster of the clustering structure C and to the same group of partition P.
- **SD**: if points belong to the same cluster of the clustering structure C and to different groups of partition P.
- **DS**: if points belong to different clusters of the clustering structure C and to the same group of partition P.
- **DD**: if points belong to different clusters of the clustering structure C and to different groups of partition P.

Assigning **a, b, c, d** to, respectively, **SS, SD, DS, DD** we then **M** $= a + b + c + d$, which is the maximum number of pairs in the data set (it means $M = N(N\text{-}1)/2$ where N is the total number of points in the data set). From these terms, we can now define the following indices:

- **Rand [10]** :
$$R = (a{+}d)/M \tag{6}$$
It has been shown that Rand index is highly dependent upon the number of clusters [29].
- **Jaccard [9]**:
$$J = a/(a{+}b{+}c), \tag{7}$$
It is very similar to Rand index, however it disregards the pairs of elements that are in different clusters for both clusterings.
- **Fowlkes and Mallows [12]**:
$$FM = a/\sqrt{m_1 m_2} = \sqrt{\frac{a}{a+b}.\frac{a}{a+c}}, \tag{8}$$
This measure has the undesirable property that for small numbers of clusters, the value is very high, even for independant clusterings.

  where $m_1 = a/(a{+}b), m_2 = a/(a{+}c)$. The above indices take values between 0 and 1, and higher values have been proven to indicate greater similarity between C and P, which implies a better clustering structure.

- **Huberts [11]**:

$$\Gamma = (1/M) \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} X(i,j)Y(i,j), \tag{9}$$

  where *X(i,j)* and *Y(i,j)* are the *(i,j)* element of matrices X and Y, respectively, and *X(i,j)* is set to 1 if x_i and x_j belong to the same cluster in C and 0 otherwise, *Y(i,j)* equal to 1 if x_i and x_j belong to the same group in P and 0 otherwise.
- **Huberts Normalized [11]**: The normalized Hubert's statistic Γ' can be defined as:

$$\Gamma' = \frac{[(1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(X(i,j)-\mu_x)(Y(i,j)-\mu_y)]}{\sigma_x\sigma_y}, \tag{10}$$

  where $\mu_x, \mu_y, \sigma_x, \sigma_y$ are the respective means and variances of X, Y matrices. The index takes values in the interval of -1 and 1.
  Both Hubert's statistic depend on the identification of a "knee" in the plot of values, and this can sometimes be very subjective.

- **F-Measure [13]**:

$$F = \frac{2a}{2a+b+c} \tag{11}$$

- **Variation of Information [14]**:

  This measure is different from the ones listed so far because it's not based on counting pairs. Instead, we use entropy and mutual information, both presented in detail in [15].

$$VI = [\mathcal{H}(C) - \mathcal{I}(C,P)] + [\mathcal{H}(P) - \mathcal{I}(C,P)], \tag{12}$$

  where $\mathcal{H}(C), \mathcal{H}(P)$ is the entropy of clustering C and entropy of partition P, respectively, and $\mathcal{I}(C,P)$ is the mutual information between C and P.

- **Minkowski score [1]**: The Minkowski score measures the difference between the clustering results C and a reference clustering (true clusters). And the difference is computed by counting the disagreements of the pairs of data objects in two partitions, it's given by:

$$MS = \sqrt{b+c+2a}/\sqrt{c}, \tag{13}$$

  the score ranges between 0 and $+\infty$.

- **Van Dongen [16], [17]**: The measure is based on maximum intersections of clusters. To find intersections between C and P we need to make use of a contingency table $M = (m_{ij})$ whose ij-th entry values equals to the number of elements in the intersection of the clusters C_i and P_i.
  Van Dongen measures is then given by:

$$VD = 2M - \sum_{i=1}^{k} \max_{j} m_{ij} - \sum_{j=1}^{l} \max_{i} m_{ij} \tag{14}$$

3.2 Internal Clustering Validation Measures

By using this approach our goal will be to test if the clustering structure produced by the clustering algorithm fits the data or not. Therefore, we use only information inherent in the data set. There is two cases when too apply internal criteria for cluster validation:

1. Validating hierarchy of clustering schemes

   Hierarchical clustering produces a dendrogram that we can represent by it's cophenetic matrix, P_c. A good approach to evaluate a hierarchical clustering algorithm result is is by measuring the degreee of agreement between the cophenetic matrix, P_c, and the proximity matrix, P, of the data set. This measure is know as **Cophenetic Correlation Coefficient** [18], [9]:

$$CPCC = \frac{(1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d_{ij}c_{ij} - \mu_p\mu_c}{\sqrt{((1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d_{ij}^2 - \mu_p^2)((1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} c_{ij}^2 - \mu_c^2)}}, \tag{15}$$

   where μ_p, μ_c are the mean values of matrices P and P_c, respectively. Moreover, the values d_{ij}, c_{ij}, correspond to the (i,j) elements of P and P_c, respectively.

2. Validating a single clustering scheme

   The goal here is to find the degree of agreement between a given clustering scheme C, with n_c clusters and the proximity matrix P. The defined index for this approach is Hubert's Γ statistic (or normalized Γ statistic). One more matrix is used on the computations:

$$Y(i,j) = \begin{cases} 1, & \text{if } x_i \text{ and } x_j \text{ belong to different clusters} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

   Then the Γ or (normalized Γ) is applied to P and Y. For each partition of the data set, X_i, generated by bootstrapping we compute the proximity matrix, P_i. Then we apply to each of them, the clustering algorithm used for C. We compute Y_i and Γ_i. Finally we compare all values for the different n_c and choose the appropriate one.

3.3 Relative Clustering Validation Measures

So far, we have looked at indices that use statistical tests to validate the clustering structure resulted, but due to the use of tools like bootstrapping or Monte Carlo methodology, these approaches tend to demand high computational resources. So called Relative Criteria measures [9], don't involve statistical testing, instead the fundamental idea is choosing the best clustering that fits the data, among the clustering produced by a clustering algorithm when it's parameter's vary.

There two cases we must consider:

- **The clustering algorithm doesn't contain the number of clusters, n_c, as parameter**.
  The choice of the "best" parameter for this type of algorithms is based on the assumption that if the data set *possesses a clustering structure, this structure is captured for a "wide" range of values of the parameters in the algorithm.* Based on this, we run the algorithm for a wide range of values of its parameters and we choose the widest range for which n_c remains constant. Then we choose the values of the parameters as the values that correspond to the middle of this range..

- **The clustering algorithm contains n_c as a parameter**.
  For this case, the different procedure is followed. We first select a suitable performance index. We run the algorithm for values of n_c between a minimum and a maximum, that are set a priori. For each value of n_c we run the algorithm r times, using different sets of values for each n_c, and we plot the best values the index gives for each n_c. Finally, we seek in the plot the point where there is a significant "knee", which is an indicator of good clustering. The absence of a "knee" indicates no clustering structure.

- **Modified Hubert $\varGamma$ [9], [11]**: With this measure we look too evaluate the difference between clusters by counting the disagreements of pairs of data objects in two partitions.

$$\varGamma = (1/M) \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} P(i,j).Q(i,j), \tag{17}$$

  where P is the proximity matrix and Q is an NxN matrix whose (i,j) element is equal to the distance between the representative points (v_{ci}, v_{cj}) of the clusters where the objects x_i and x_j belong.

- **Dunn index [20], [21]**: Dunn's uses the minimum pairwise distance between objects in different clusters to measure the intercluster separation and the maximum diameter among all clusters to measure the intracluster compactness. The index is defined for a given number of clusters:

$$D_{n_c} = \min_{i=1,\ldots,n_c} \{ \min_{j=i+1,..,n_c} (\frac{d(c_i, c_j)}{\max_{k=1,\ldots,n_c} diam(c_k))} \}, \tag{18}$$

where $d(c_i, c_j)$ is the dissimilarity function between two clusters c_i and c_j defined as:

$$d(c_i, c_j) = \min_{x \in c_i, y \in c_j} d(x, y)$$

, and $diam(c)$ is the diameter of a cluster, which may be considered as a measure of clusters' dispersion. The diameter of a cluster C can be defined as follows:

$$diam(C) = \max_{x,y \in C} d(x, y)$$

- **Davies-Bouldin index[22]**: This measure computes the average distances between each pair of clusters, therefore a lower value of this measure indicates a high dissimilarity between clusters, which is desirable.
  The similarity measure is

  $$R_{ij} = (s_i + s_j)/d_{ij} \tag{19}$$

  , where s_i, s_j are the measures of dispersion of clusters C_i and C_j, respectively, and d_{ij} the dissimilarity measure between the two clusters.
  Then the index is defined as:

  $$DB_{n_c} = \frac{1}{n_c} \sum_{i=1}^{n_c} \max_{i=1,\ldots,n_c, i \neq j} R_{ij}, \tag{20}$$

- **SD validity index[23]**:
  It's based on the concepts of average scattering, which indicates the compactness between clusters, given by equation 21 and the total separation of clusters, which indicates the separation between the items of a cluster, given by equation 22.

  $$Scat(n_c) = \frac{1}{n_c} \sum_{i=1}^{n_c} \frac{\|\sigma(v_i)\|}{\|\sigma(X)\|}, \tag{21}$$

  where v_i is the center of cluster i, X is a data set and $\sigma(v_i), \sigma(X)$ are the variance of cluster i and variance of a data set X, respectively.

  $$Dis(n_c) = \frac{D_{max}}{D_{min}} \sum_{k=1}^{n_c} (\sum_{z=1}^{n_c} \|v_k - v_z\|)^{-1}, \tag{22}$$

  where $D_{max} = max(\|v_i - v_j\| \, \forall_{i,j} \in 1, 2, 3, ..., n_c$ is the maximum distance between cluster centers. $D_{min} = min(\|v_i - v_j\| \, \forall_{i,j} \in 1, 2, 3, ..., n_c$ is the minimum distance between cluster centers.
  Now we can define the index based on equations 21 and 22:

  $$SD(n_c) = \alpha Scat(n_c) + Dis(n_c), \tag{23}$$

  where α is a weighting factor equal to $Dis(max_{n_c})$. The n_c that minimizes the index can be considered the optimal value for the number of clusters in the data set.

- **S_Dbw validity index [4]**:

  Like SD index, it evaluates both criteria of "good clustering" (i.e., compactness and separation):

  $$S\_Dbw(n_c) = Scat(n_c) + Dens_bw(n_c), \tag{24}$$

  where the first term is equal to what we previous saw in equation 21 and the new term that evaluates the inter-cluster density is given by:

  $$Dens\_bw(n_c) = \frac{1}{n_c.(n_c-1)} \sum_{i=1}^{n_c} (\sum_{j=1, i \neq j}^{n_c} \frac{density(u_{ij})}{max\{density(v_i), density(v_j)\}}), \tag{25}$$

  where v_i, v_j are, respectively, the center of clusters c_i, c_j and u_{ij} the middle point of the line segment defined by the clusters' centers v_i, v_j. The term density(u) is defined in equation 25:

  $$density(u) = \sum_{l=1}^{n_{ij}} f(x_l, u), \tag{26}$$

  where n_{ij} = number of tuples that belong to clusters c_i and c_j. The function f(x,u) represents the number of points in the neighbourhood of u. We define this neighbourhood as a hyper-sphere with center u and radius the average standard deviation of the clusters, stdev:

  $$f(x,u) = \begin{cases} 0, & \text{if d(x,u)} > \text{stdev} \\ 1, & \text{otherwise} \end{cases} \tag{27}$$

- **CVNN index [25]**: The Clustering Validation index based on Nearest Neighbors(CVNN), is a validation measure based on the notion of nearest neighbours, and complements some of the already existing measures.
  Like other measures, is generally based on 1) inter-cluster separation and 2) intra-cluster compactness.
  In 1), the idea is that, if an object is located in the center of a cluster and is surrounded by objects in the same cluster, then it is well separated from other clusters and thus contributes little to the inter-cluster separation. If an object is located at the edge of a cluster, it is surrounded mostly by objects in other clusters, thus contributes a lot to the inter-cluster separation. The measurement is given by equation 28 and a low value indicates a better inter-cluster separation.
  To measure 2) we compute the average pairwise distance between objects in the same cluster. A low value for 29 indicates a better intra-cluster compactness.

$$Sep(n_c, k) = \max_{i=1,2,\ldots,n_c}(\frac{1}{n_i}\sum_{j=1}^{n_i}\frac{q_j}{k}), \tag{28}$$

where n_c is the number of clusters, k is the number of nearest neighbours, n_i is the number of objects in ith cluster C_i, and q_j is the number of nearest neighbours of the jth object in C_i that are not in C_i.

$$Com(n_c) = \sum_i [\frac{2}{n_i \cdot (n_i - 1)} \sum_{x,y \in C_i} d(x, y), \tag{29}$$

the notation for n_c, n_i is defined in equation 28. We define x and y as two different objects in cluster C_i, and $d(x, y)$ is the distance between x and y. Based on the above equations we can define the CVNN index:

$$CVNN(n_c, k) = Sep_{norm}(n_c, k) + Com_{norm}(n_c, k), \tag{30}$$

where

$$Sep_{norm}(n_c, k) = \frac{Sep(n_c, k)}{(max_{n_{c_{min}}} \leq n_c \leq n_{c_{max}} Sep(n_c, k))}, \tag{31}$$

and

$$Com_{norm}(n_c) = \frac{Com(n_c)}{(max_{n_{c_{min}}} \leq n_c \leq n_{c_{max}} Com(n_c))}, \tag{32}$$

A normalization the two terms since they should have the same range. A lower value of CVNN indicates a better clustering result.

4 Solution Proposal

In order to get better results with the AliClu algorithm and also to explore the capabilities of all the clustering validation indices we propose ourselves to create a python library containing the 15 indices listed in section 3.

All indices in the extensive list presented will be analysed with different data set (including the Reuma.pt), yet too be chosen/generated.

In this library will be possible to apply the indices to outputs of the hierarchical agglomerative clustering algorithm only, by varying all combinations of number of clusters and distance measures. We make this choice because this way we can extend the code available in [6], where only hierarchical clustering is applied.

Distance measures available are Single link, Complete link, Average link, Centroid link and Ward's method [26], [27], [28].

With this library, the user can also simultaneously select more than one index and number of clusters in a single function call.

The library will be available to user at https://pypi.org/.

5 Preliminary results

With the goal of understanding if using new indices to validate the resulting clustering structures with AliClu will lead too the same conclusions or not, three of indices described in section 3 were implemented, extending the code available in [6].

We chose too evaluate the performances of CVNN, S_Dbw and Van Dongen (VD) measures. The reason for CVNN and S_Dbw was clearly because both achieved good results in [1], also in AliClu there was only external indices tested and it would be intersting too see what other types of measures could give us. As for VD, we chose it because it gave good results in the literature compared too others and also too compare to the others external measures already implemented.

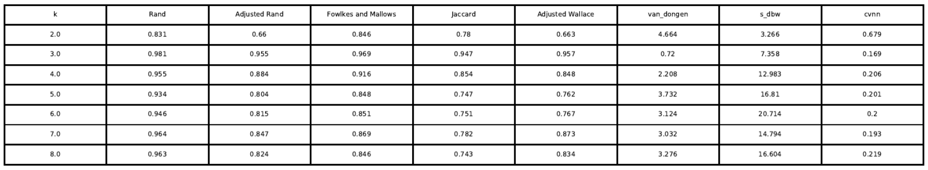
It's important too clarify that although this three measures gave good results in previous studies, it doesn't necessarily mean they will perform the same way in ours, it always depends on the clustering structure we have as input.

The results presented were achieved by running AliClu on the synthetic data set presented in [6]. This synthetic data set was built to provide a proof of concept in a controlled scenario where we know the correct number of clusters a priori.

The data set consists of temporal sequences generated by continuous-time Markov chains (CTMC), that simply provides a sequence of states and holding times in those states. If a process starts at a state i then it will stay for a random amount of time, say T_1, where T_1 is a continuous random variable. Then at time T_1, the process jumps to a new state j where will stay for a random amount of time T_2 , and so on. The random variables T_1 , T_2 ,... have exponential distribution and the probability of going from state i to state j is given by p_{ij} (probabilities from the discrete-time Markov chain). Hence, CTMC is the perfect tool for generating the temporal sequences like the ones in Reuma.pt. A more deep explanation of CTMC can be found in [6].

The experiment consists in running the algorithm for each distance metric (ward, single, complete, average and centroid) and minimum number of clusters equal too two and maximum number of clusters equal too eight, and show the measures of eight clustering indices for the chosen configuration (number of clusters and gap penalty). The indices Rand, Adjusted Rand, Fowlkes and Mallows, Jaccard and Adjusted Wallace were already evaluated by the algorithm.

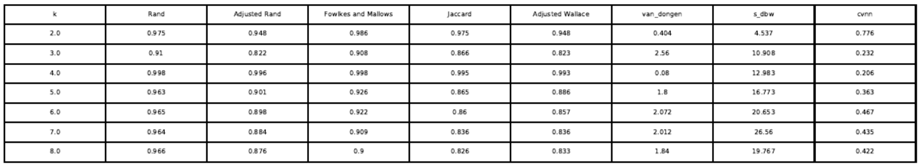
In this experiment, the correct number of clusters is three.



k	Rand	Adjusted Rand	Fowlkes and Mallows	Jaccard	Adjusted Wallace	van_dongen	s_dbw	cvnn
2.0	0.831	0.66	0.846	0.78	0.663	4.664	3.266	0.679
3.0	0.981	0.955	0.969	0.947	0.957	0.72	7.358	0.169
4.0	0.955	0.884	0.916	0.854	0.848	2.208	12.983	0.206
5.0	0.934	0.804	0.848	0.747	0.762	3.732	16.81	0.201
6.0	0.946	0.815	0.851	0.751	0.767	3.124	20.714	0.2
7.0	0.964	0.847	0.869	0.782	0.873	3.032	14.794	0.193
8.0	0.963	0.824	0.846	0.743	0.834	3.276	16.604	0.219

Fig. 2. Average values for ward distance and gap penalty equal too 0.10. Number of clusters chosen = 3

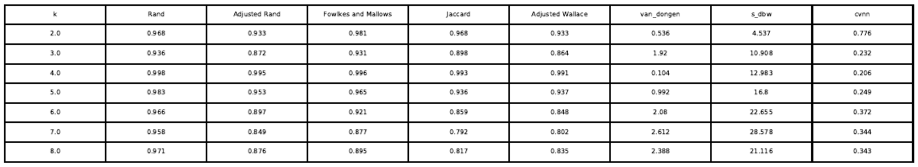
In Fig. 2, we can see than CVNN points to the correct number of cluster, equal too three, while VD and S_dbw do not, both point to a two cluster structure.



k	Rand	Adjusted Rand	Fowlkes and Mallows	Jaccard	Adjusted Wallace	van_dongen	s_dbw	cvnn
2.0	0.975	0.948	0.986	0.975	0.948	0.404	4.537	0.776
3.0	0.91	0.822	0.908	0.866	0.823	2.56	10.908	0.232
4.0	0.998	0.996	0.998	0.995	0.993	0.08	12.983	0.206
5.0	0.963	0.901	0.926	0.865	0.886	1.8	16.773	0.363
6.0	0.965	0.898	0.922	0.86	0.857	2.072	20.653	0.467
7.0	0.964	0.884	0.909	0.836	0.836	2.012	26.56	0.435
8.0	0.966	0.876	0.9	0.826	0.833	1.84	19.767	0.422

Fig. 3. Average values for single link distance and gap penalty equal too 0.20. Number of clusters chosen = 4

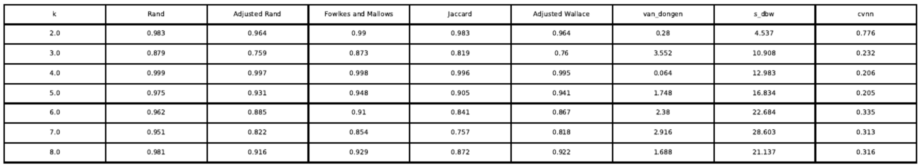
For single link distance, VD indicates correctly the number of clusters, while S_dbw and CVNN guess incorrectly, number of clusters equal too two and four, respectively.



k	Rand	Adjusted Rand	Fowlkes and Mallows	Jaccard	Adjusted Wallace	van_dongen	s_dbw	cvnn
2.0	0.968	0.933	0.981	0.968	0.933	0.536	4.537	0.776
3.0	0.936	0.872	0.931	0.898	0.864	1.92	10.908	0.232
4.0	0.998	0.995	0.996	0.993	0.991	0.104	12.983	0.206
5.0	0.983	0.953	0.965	0.936	0.937	0.992	16.8	0.249
6.0	0.966	0.897	0.921	0.859	0.848	2.08	22.655	0.372
7.0	0.958	0.849	0.877	0.792	0.802	2.612	28.578	0.344
8.0	0.971	0.876	0.895	0.817	0.835	2.388	21.116	0.343

Fig. 4. Average values for complete link distance and gap penalty equal too 0.30. Number of clusters chosen = 4

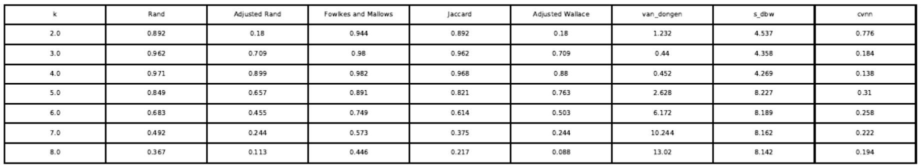
For complete linkage, all three indices guess incorrectly, VD being equal too eight, S_dbw equal too two and CVNN equal too four.



k	Rand	Adjusted Rand	Fowlkes and Mallows	Jaccard	Adjusted Wallace	van_dongen	s_dbw	cvnn
2.0	0.983	0.964	0.99	0.983	0.964	0.28	4.537	0.776
3.0	0.879	0.759	0.873	0.819	0.76	3.552	10.908	0.232
4.0	0.999	0.997	0.998	0.996	0.995	0.064	12.983	0.206
5.0	0.975	0.931	0.948	0.905	0.941	1.748	16.834	0.205
6.0	0.962	0.885	0.91	0.841	0.867	2.38	22.684	0.335
7.0	0.951	0.822	0.854	0.757	0.818	2.916	28.603	0.313
8.0	0.981	0.916	0.929	0.872	0.922	1.688	21.137	0.316

Fig. 5. Average values for average link distance and gap penalty equal too 0.70. Number of clusters chosen = 4

For average linkage, VD is indicating the current number of cluster, three, and the two other indices give the wrong answer, S_dbw equal too two and CVNN equal too five.



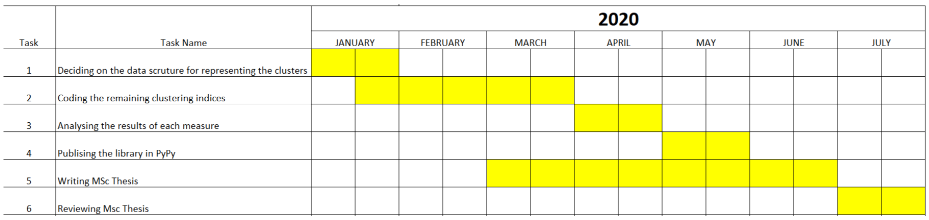
k	Rand	Adjusted Rand	Fowlkes and Mallows	Jaccard	Adjusted Wallace	van_dongen	s_dbw	cvnn
2.0	0.892	0.18	0.944	0.892	0.18	1.232	4.537	0.776
3.0	0.962	0.709	0.98	0.962	0.709	0.44	4.358	0.184
4.0	0.971	0.899	0.982	0.968	0.88	0.452	4.269	0.138
5.0	0.849	0.657	0.891	0.821	0.763	2.628	8.227	0.31
6.0	0.683	0.455	0.749	0.614	0.503	6.172	8.189	0.258
7.0	0.492	0.244	0.573	0.375	0.244	10.244	8.162	0.222
8.0	0.367	0.113	0.446	0.217	0.088	13.02	8.142	0.194

Fig. 6. Average values for centroid link distance and gap penalty equal too -0.10. Number of clusters chosen = 4

Finally, for the centroid linkage, all indices point to a incorrect answer. VD is equal too seven, S_dbw and CVNN are both equal too four.

More research as to be made in order too understand the reasons why the indices are giving incorrect answers.

6 Work Schedule



		2020													
Task	Task Name	JANUARY		FEBRUARY		MARCH		APRIL		MAY		JUNE		JULY	
1	Deciding on the data scruture for representing the clusters	🟨	🟨												
2	Coding the remaining clustering indices		🟨	🟨	🟨	🟨	🟨								
3	Analysing the results of each measure							🟨	🟨						
4	Publising the library in PyPy									🟨	🟨				
5	Writing MSc Thesis					🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨		
6	Reviewing Msc Thesis													🟨	🟨

Fig. 7. Work Planning Gantt Chart

7 Conclusions

This work starts of with the goal of achieving better results from clustering methods and to give clustering users an easier time when experimenting in the field.

At this point, we are looking forward too tackle the challenges that will arise from the development of the solution. So far, three indices were implemented and the preliminary results show us that they generally are not identifying the correct number of clusters.

Relevant studies were surveyed, setting for us a starting point to work from. The proposed solution aims to give users all information they need to select the best clustering configuration for their data.

References

1. Charu C. Aggarwal, Chandan K. Reddy: Data Clustering: Algorithms and Applications (pp. 571–605) (2013)
2. A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. P. Patel, A. Tiwari, M. J. Er, W. Ding, and C.-T. Lin: A review of clustering techniques and developments. Neurocomputing, 267:664–681, Dec. 2017. https://doi.org/10.1016/j.neucom.2017.06.053
3. A.K. Jain, M.N. Murty, P.J. Flynn: Data Clustering: A Review. ACM Computing Surveys, Vol. 31, No. 3. (1999).
4. Maria Halki, Michalis Vazirgian: Clustering Validity Assessment: Finding the optimal partitioning of a data set. Proceedings - IEEE International Conference on Data Mining, ICDM, 187-194. (2001). https://doi.org/10.1109/ICDM.2001.989517
5. Hämäläinen, Joonas and Jauhiainen, Susanne and Kärkkäinen, Tommi: Comparison of Internal Clustering Validation Indices for Prototype-Based Clustering. Algorithms. (2017). https://doi.org/10.3390/a10030105
6. Rama, K., Canhão, H., Carvalho, A.M. et al.: AliClu - Temporal sequence alignment for clustering longitudinal clinical data. BMC Med Inform Decis Mak 19, 289 (2019). https://doi.org/10.1186/s12911-019-1013-7
7. Needleman, Saul B., Wunsch, Christian D. : A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology. Volume 48. 443-453. (1970). https://doi.org/10.1016/0022-2836(70)90057-4
8. Syed, Haider; Das, Amar K.: Temporal Needleman-Wunsch. (2015). https://doi.org/10.1109/DSAA.2015.7344785
9. Sergios Theodoridis, Konstantinos Koutroumbas: Pattern recognition. Academic Press. (pp. 591-628). (2003).
10. William M. Rand: Objective Criteria for the Evaluation of Clustering Methods. Journal of the American Statistical Association. (pp. 846-850). (1971). https://doi.org/10.1080/01621459.1971.10482356
11. L. Hubert and P. Arabie: Comparing partitions. Journal of Classification. 2(1):193–218. (1985) https://doi.org/10.1007/BF01908075
12. E. B. Fowlkes and C. Mallows: A method for comparing two hierachical clusterings. Journal of The American Statistical Association. 78:553–569. (1983).
13. E. Achtert, S. Goldhofer, H. Kriegel, E. Schubert and A. Zimek: Evaluation of Clusterings – Metrics and Visual Support. 2012 IEEE 28th International Conference on Data Engineering. pp. 1285-1288. (2012) https://doi.org/10.1109/ICDE.2012.128
14. Marina Meilă: Comparing clusterings—an information based distance. Journal of Multivariate Analysis. pp. 873-895. (2007) https://doi.org/https://doi.org/10.1016/j.jmva.2006.11.013
15. T.M. Cover, J.A. Thomas: Elements of Information Theory. Wiley. (1991)
16. Stijn van Dongen: Performance criteria for graph clustering and Markov cluster experiments. (2000)
17. Wagner, Silke and Wagner, Dorothea: Comparing Clusterings - An Overview. Technical Report 2006-04. (2007)
18. R. R. Sokal and F. J. Rohlf: The comparison of dendrograms by objective methods. Taxon. pp. 33–40. (1962)
19. Halkidi, Maria and Batistakis, Yannis and Vazirgiannis, Michalis: Cluster Validity Methods: Part I. SIGMOD Record. (2002) https://doi.org/10.1145/565117.565124
20. Halkidi, Maria and Batistakis, Yannis and Vazirgiannis, Michalis: Clustering Validity Checking Methods: Part II. ACM SIGMOD Record. (2002) https://doi.org/10.1145/601858.601862

21. Dunn, J. C.: Well-Separated Clusters and Optimal Fuzzy Partitions. Journal of Cybernetics. (1974) https://doi.org/10.1080/01969727408546059
22. Davies, David L.; Bouldin, Donald W.: A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence. (1979) https://doi.org/10.1109/tpami.1979.4766909
23. Halkidi, Maria and Vazirgiannis, Michalis and Batistakis, Yannis: Quality Scheme Assessment in the Clustering Process. LNCS (LNAI). pp. 265-276 (2000) https://doi.org/10.1007/3-540-45372-5_26
24. Anil K. Jain, Richard C. Dubes: Algorithms for clustering data. Prentice Hall College Div. (1988)
25. Y. Liu, Z. Li, H. Xiong, X. Gao, J. Wu, S. Wu: Understanding and Enhancement of Internal Clustering Validation Measures. PIEEE Transactions on Cybernetics. Vol. 43. pp. 982-994. (2013) https://doi.org/10.1109/TSMCB.2012.2220543
26. JH. Ward: Hierarchical Grouping to Optimize an Objective Function. Journal of the American Statistical Association. pp. 236–244. (1963).
27. Sokal R, Michener C: A Statistical Method for Evaluating Systematic Relationships. University of Kansas Science Bulletin. pp. 1409–1438. (1958).
28. J. Han, M. Kamber, and J. Pei: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers. Chapter 10. Third edition. (2011).
29. Morey, L. C., Agresti, A.: An Adjustment to the Rand Statistic for Chance Agreement. The Classification Society Bulletin 5, 9-10.