Contents

1 Clustering properties and quality functions

pyProCT implements functions which goal is to retrieve information from the generated clusterings to evaluate them. Some of these functions retrieve properties from the clustering (“properties”), other functions evaluate quality and form the objective part of the clustering hypothesis. In this section we will define and formalize both of them.

1.1 General definitions

1. D is a set containing all the elements of the dataset with $|D|$ number of elements (number of datum in the dataset e.g conformations, distances, etc).

2. $C = \{C_1, C_2, \dots C_k\}$ is a clustering of k clusters, where $|C|$ is its number of clusters ($|C| = k$ in this case) and $|C_i|$ is the number of elements of cluster i. Also:

$$\forall C_i, C_j \in C, \quad i \neq j \quad C_i \cap C_j \equiv \emptyset \tag{1}$$

3. $D_C \subseteq D$ is the set of clustered elements, which can differ from the initial set if some elements were considered as noise and thus discarded.

$$D_C \quad \equiv \quad \bigcup_{i=1}^{k} C_i, \tag{2}$$

4. $d(a, b)$ is a distance metric (dissimilarity function) applied to an element $a \in D$ and an element $b \in D$.

5. m_i is the medoid of cluster C_i so that:

$$m_i \in C_i \tag{3}$$

$$\forall a, b \neq m_i \quad a, b \in C_i \quad d(a,b) \geq d(a,m) \wedge d(a,b) \geq d(b,m) \tag{4}$$

6. As stated before singleton clustering is a clustering with one cluster that holds all the elements of the dataset. In a trivial clustering, each element of the dataset forms its own clustering.

$$|\{|C_{singleton}\} = 1 \tag{5}$$

$$|\{|C_{trivial}\} = |D_C| \tag{6}$$

1.2 Properties

The properties are functions that allow users to query about simple traits and statistics of the clustering. The information returned by this functions is purely descriptive and, in general, not usable for evaluation purposes. Eight properties have been defined:

Details (String) Returns a string containing information about the type of clustering algorithm and the parameters used to generate it.

NumClusters (Integer) Returns the number of clusters ($|C|$).

MeanClusterSize (Real) Mean number of elements per cluster:

$$mcs(C) = \frac{\sum_{i=1}^{k} |C_i|}{|C|}$$

NumClusteredElems (Integer) Returns the number of elements that were clustered. It can be lower than the initial number of elements if noise was eliminated. Is defined as:

$$nce(C) = \sum_{i=0}^{|C|} |C_i| \equiv |D_C|$$

NoiseLevel (Real) Calculates the ratio of clustered elements over the number of initial elements:

$$nl(C, D) = \frac{\sum_{i=1}^{k} |C_i|}{|D|} \equiv \frac{|D_C|}{|D|}$$

ClustersTo90 (Real) Returns the minimum number of clusters needed to accumulate 90% of the clustered elements.

PercentInTop (Real) Calculates the percentage of clustered elements owned by the biggest cluster.

PercentInTop4 (Real) Calculates the percentage of clustered elements owned by the four larger clusters.

1.3 Quality functions

Sometimes referred to as Clustering Validity Indices (CVI), are functions which aim is to tell at which degree clusterings are an artificial partition or reflect the real inner structure of the dataset (assuming that it exists). Quality functions must use only internal information of the clustering, that is, not to be based on a solution that is considered correct.

The following are some initial definitions that will help us to carachterize them:

Whitin cluster distance The sum of all distances inside a cluster.

$$wd(C_i) = \sum_{a,b \in C_i} d(a,b)$$

Between cluster distance Sum of the pairwise distances of elements pertaining to different clusters.

$$bd(C_i, C_j) = \sum_{a \in C_i} \sum_{b \in C_j} d(a,b)$$

Average distance Average of all pairwise distances of the elements inside a cluster.

$$avg(C_i) = \frac{wd(C_i)}{|C_i|}$$

Standard deviation of distance The standard deviation of all pairwise distances of the elements inside a cluster

$$stdev(C_i) = \sqrt{\frac{1}{|C_i|} \sum_{a \in C_i} d^2(a, m_i)}$$

1.3.1 Cohesion

Is a measure of cluster compactness. The cohesion factor measures the inner similarity of a cluster. Its value for one cluster is calculated by summing up the distance of all elements belonging to that cluster. Its value for a clustering can be defined as the sum of its clusters partial cohesions weighted by the inverse of the cluster size. Due to the use of dissimilarity metrics, the interpretation of cohesion can be misleading: the smaller its value, the more compact the clusters are.

$$Ch(C) = \sum_{C_i \in C} \frac{1}{C_i} wd(C_i)$$

A cohesion value of 0 can be obtained with the singleton clustering. It reaches its maximum value in a trivial clustering ($|D_C|^{-1} wd(D_C)$).

1.3.2 MirrorCohesion / CythonMirrorCohesion

Is a more intuitive redefinition of Cohesion. An increment of cluster compactness increases the index value:

$$Mch(C) = 1 - \frac{Ch(C)}{|D_C|^{-1} wd(D_C)}$$

Its value ranges between 0 and 1.

1.3.3 Separation

Measures how distinct a cluster is from other clusters (how isolated a cluster is from the others). In practice, it calculates the sum of distances weighted by its cohesion:

$$Sep(C) = \sum_{\substack{i=1 \\ j>i}}^{k} \frac{bd(C_i, C_j)}{Ch(C_i)}$$

When its value increases, cluster separation increases. Its value ranges from 0 (trivial clustering) to infinity (sigleton clustering, which implies $Ch(C_i) \equiv 0$.

1.3.4 MinimumMeanSeparation / CythonMinimumMeanSeparation

For each pair of clusters, the mean distance of its elements is calculated. Then a random sample of the element pairs with distances lower than the mean are selected. The mean of this distances will be the value of the metric for that cluster, and the average for all clusters will be the value for a clustering. Maximizing minimum mean separation maximizes separation.

1.3.5 Compactness

Compares the standard deviation (std. dev.) of the clustered dataset (std. dev. of its clusters) with the std. dev. of the whole dataset. Note that the std. dev. calculation function is defined using the medoid of the cluster instead of the mean point.

$$Cmp(C) = \frac{1}{|C|} \sum_{i=1}^{|C|} \frac{stdev(C_i)}{stdev(D_C)}$$

Maximizing the value minimizes compactness.

1.3.6 Gaussian Separation

Is a prototype-based separation index where distances are attenuated using an exponential. This intends to produce the same behaviour that some exponential kernels used to calculate the adjacency matrix in graph-like representations of the dataset: diminish long range distances and sharpen subgraphs contours.

$$Gsep = \frac{1}{|C|(|C|-1)} \sum_{\substack{i,j=1,\dots,|C| \\ j\neq i}} e^{\frac{-d^2(m_i,m_j)}{2\sigma^2}}$$

Maximizing the value maximizes separation.

1.3.7 Davies-Bouldin

A prototype-based measure that compares compactness (represented by the average of intra-cluster distances) and separation (distance between the prototypes).

$$Db(C) = \frac{1}{|C|} \sum_{i=1}^{|C|} max^{j\in 1,\dots,|C|}_{j\neq i} \left(\frac{avg(C_i)+avg(C_j)}{d(m_i,m_j)} \right)$$

Measures compactness and separation. The smaller the value, the better overall quality the clustering has.

1.3.8 Dunn

Ratio of the minimum inter-cluster distance and the maximum intra-cluster distance.

$$mind(C_i) = min_{r,t\in C_i} d(r,t)$$

$$maxd(C_i,C_j) = max_{\substack{r\in C_i \\ t\in C_j}} d(r,t)$$

$$Dunn(C) = \frac{min_{C_i\in C}(mind(C_i))}{max_{\substack{C_i,C_j\in C \\ i\neq j}}(maxd(C_i,C_j))}$$

Dunn index evaluates compactness and separation simultaneously. Quality clusterings should have high values for this function.

1.3.9 Calinski-Harabasz

Another variation of the intra- and inter-cluster distances ratio calculation.

$$A_k(C) = \frac{1}{|D_C| - |C|} \sum_{i=1}^{|C|} (|C_i| - 1)(avg(D_C) - avg(C_i))$$

$$CH(C) = \frac{avg(D_C) + \frac{|D_C| - |C|}{|C| - 1} A_k}{|D_C| - A_k(C)}$$

It also measures compactness and separation.The higher the value, the better quality the clustering has.

1.3.10 Silhouette / CythonSilhouette

Useful when distances are on a ratio scale (for instance euclidean distance), allowing to measure compactness and separation all together by calculating the pairwise difference of inter and intracluster distances. The Silhouette index for a single element of a cluster can be calculated as:

$$S(e) = \begin{cases} \frac{b(e) - a(e)}{max(a(e), b(e))} & \text{if } |C_i| > 1 \\ 0 & \text{if } |C_i| \leq 1 \end{cases}$$

Where $a(e)$ is the average inner dissimilarity of its cluster and $b(e)$ is the outer dissimilarity of this element with the other clusters, calculated as follows:

$$e \in C_e$$

$$a(e) = \sum_{\substack{t \in C_e \\ t \neq e}} d(e, t)$$

$$b(e) = \sum_{\substack{\forall t \in C_i \\ t \neq e \\ C_f \neq C_e}} d(e, t)$$

Cluster and clustering values for this index can be calculated as the cluster average and global average of their per-element values. Its value ranges from -1 (worst quality) to 1 (best quality).

1.3.11 PCAanalysis

Calculates the axes of variance and gives an estimation of the amount of variance in each axis for each cluster. The final value of this index will be the average value of the variance of the major variance axis for each cluster. As with other compactness measures it depends on the size of the clusters. The higher the value is, the less compact the clusters are. Measures compactness (by means of variance).

1.3.12 Graph cut indices

The distance relationships between elements of the clustering can be viewed as a similarity graph were each vertex is an element and edges are weighted by their distances. In general, small values for this functions mean good quality of the partition (almost all are a sum of adjacency weights). Clustering can then be seen as a graph partitioning problem where one objective graph cut function is optimized. First, we will need to define some helper functions. In this case, distances are the edge values in the adjacency matrix:

$$deg(a) = \sum_{b \in |D_C|} d(a,b)$$

$$vol(A) = \sum_{a \in A} deg(a)$$

$$W(C_1, C_2) = \sum_{a \in C_1, b \in C_2} d(a,b)$$

$$cut(C) = \frac{1}{2} \sum_{C_i \in C} W(C_i, \overline{C}_i)$$

1.3.13 Ncut

$$Ncut(C) = \sum_{i=1}^{k} \frac{cut(C)}{vol(C_i)}$$

Is as a separation measure.

1.3.14 MinMaxCut

$$MinMaxCut(C) = \sum_{i=1}^{k} \frac{cut(C_i, C)}{W(C_i, C_i)}$$

1.3.15 RatioCut

$$RatioCut(C) = \sum_{i=1}^{k} \frac{cut(C_i, C)}{|C_i|}$$

Is a separation measure.