CHAPTER 17 Clustering Validation

There exist many different clustering methods, depending on the type of clusters sought and on the inherent data characteristics. Given the diversity of clustering algorithms and their parameters it is important to develop objective approaches to assess clustering results. Cluster validation and assessment encompasses three main tasks: *clustering evaluation* seeks to assess the goodness or quality of the clustering, *clustering stability* seeks to understand the sensitivity of the clustering result to various algorithmic parameters, for example, the number of clusters, and *clustering tendency* assesses the suitability of applying clustering in the first place, that is, whether the data has any inherent grouping structure. There are a number of validity measures and statistics that have been proposed for each of the aforementioned tasks, which can be divided into three main types:

External: External validation measures employ criteria that are not inherent to the dataset. This can be in form of prior or expert-specified knowledge about the clusters, for example, class labels for each point.

Internal: Internal validation measures employ criteria that are derived from the data itself. For instance, we can use intracluster and intercluster distances to obtain measures of cluster compactness (e.g., how similar are the points in the same cluster?) and separation (e.g., how far apart are the points in different clusters?).

Relative: Relative validation measures aim to directly compare different clusterings, usually those obtained via different parameter settings for the same algorithm.

In this chapter we study some of the main techniques for clustering validation and assessment spanning all three types of measures.

17.1 EXTERNAL MEASURES

As the name implies, external measures assume that the correct or ground-truth clustering is known *a priori*. The true cluster labels play the role of external information that is used to evaluate a given clustering. In general, we would not know the correct clustering; however, external measures can serve as way to test and validate different methods. For instance, classification datasets that specify the class for each point can be used to evaluate the quality of a clustering. Likewise, synthetic datasets with known cluster structure can be created to evaluate various clustering algorithms by quantifying the extent to which they can recover the known groupings.

Let $\mathbf{D} = \{\mathbf{x}_i\}_{i=1}^n$ be a dataset consisting of n points in a d-dimensional space, partitioned into k clusters. Let $y_i \in \{1, 2, \ldots, k\}$ denote the ground-truth cluster membership or label information for each point. The ground-truth clustering is given as $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$, where the cluster T_j consists of all the points with label j, i.e., $T_j = \{\mathbf{x}_i \in \mathbf{D} | y_i = j\}$. Also, let $\mathcal{C} = \{C_1, \ldots, C_r\}$ denote a clustering of the same dataset into r clusters, obtained via some clustering algorithm, and let $\hat{y}_i \in \{1, 2, \ldots, r\}$ denote the cluster label for $\mathbf{x}_i$. For clarity, henceforth, we will refer to $\mathcal{T}$ as the ground-truth *partitioning*, and to each T_i as a *partition*. We will call $\mathcal{C}$ a clustering, with each C_i referred to as a cluster. Because the ground truth is assumed to be known, typically clustering methods will be run with the correct number of clusters, that is, with $r = k$. However, to keep the discussion more general, we allow r to be different from k.

External evaluation measures try capture the extent to which points from the same partition appear in the same cluster, and the extent to which points from different partitions are grouped in different clusters. There is usually a trade-off between these two goals, which is either explicitly captured by a measure or is implicit in its computation. All of the external measures rely on the $r \times k$ *contingency table* $\mathbf{N}$ that is induced by a clustering $\mathcal{C}$ and the ground-truth partitioning $\mathcal{T}$, defined as follows

$$\mathbf{N}(i, j) = n_{ij} = \left|C_i \cap T_j\right|$$

In other words, the count n_{ij} denotes the number of points that are common to cluster C_i and ground-truth partition T_j. Further, for clarity, let $n_i = |C_i|$ denote the number of points in cluster C_i, and let $m_j = |T_j|$ denote the number of points in partition T_j. The contingency table can be computed from $\mathcal{T}$ and $\mathcal{C}$ in $O(n)$ time by examining the partition and cluster labels, y_i and $\hat{y}_i$, for each point $\mathbf{x}_i \in \mathbf{D}$ and incrementing the corresponding count $n_{y_i \hat{y}_i}$.

17.1.1 Matching Based Measures

Purity

Purity quantifies the extent to which a cluster C_i contains entities from only one partition. In other words, it measures how "pure" each cluster is. The purity of cluster C_i is defined as

$$purity_i = \frac{1}{n_i} \max_{j=1}^{k} \{n_{ij}\}$$

The purity of clustering $\mathcal{C}$ is defined as the weighted sum of the clusterwise purity values:

$$purity = \sum_{i=1}^{r} \frac{n_i}{n} purity_i = \frac{1}{n} \sum_{i=1}^{r} \max_{j=1}^{k} \{n_{ij}\}$$

where the ratio $\frac{n_i}{n}$ denotes the fraction of points in cluster C_i. The larger the purity of $\mathcal{C}$, the better the agreement with the groundtruth. The maximum value of purity is 1, when each cluster comprises points from only one partition. When $r = k$, a purity value of 1 indicates a perfect clustering, with a one-to-one correspondence between the clusters and partitions. However, purity can be 1 even for $r > k$, when each of the clusters is a subset of a ground-truth partition. When $r < k$, purity can never be 1, because at least one cluster must contain points from more than one partition.

Maximum Matching

The maximum matching measure selects the mapping between clusters and partitions, such that the sum of the number of common points (n_{ij}) is maximized, provided that only one cluster can match with a given partition. This is unlike purity, where two different clusters may share the same majority partition.

Formally, we treat the contingency table as a complete weighted bipartite graph $G = (V, E)$, where each partition and cluster is a node, that is, $V = \mathcal{C} \cup \mathcal{T}$, and there exists an edge $(C_i, T_j) \in E$, with weight $w(C_i, T_j) = n_{ij}$, for all $C_i \in \mathcal{C}$ and $T_j \in \mathcal{T}$. A *matching* M in G is a subset of E, such that the edges in M are pairwise nonadjacent, that is, they do not have a common vertex. The maximum matching measure is defined as the *maximum weight matching* in G:

$$match = \arg\max_M \left\{ \frac{w(M)}{n} \right\}$$

where the weight of a matching M is simply the sum of all the edge weights in M, given as $w(M) = \sum_{e \in M} w(e)$. The maximum matching can be computed in time $O(|V|^2 \cdot |E|) = O((r+k)^2 rk)$, which is equivalent to $O(k^4)$ if $r = O(k)$.

F-Measure

Given cluster C_i, let j_i denote the partition that contains the maximum number of points from C_i, that is, $j_i = \max_{j=1}^{k}\{n_{ij}\}$. The *precision* of a cluster C_i is the same as its purity:

$$prec_i = \frac{1}{n_i} \max_{j=1}^{k} \left\{ n_{ij} \right\} = \frac{n_{ij_i}}{n_i}$$

It measures the fraction of points in C_i from the majority partition T_{j_i}.

The *recall* of cluster C_i is defined as

$$recall_i = \frac{n_{ij_i}}{|T_{j_i}|} = \frac{n_{ij_i}}{m_{j_i}}$$

where $m_{j_i} = |T_{j_i}|$. It measures the fraction of point in partition T_{j_i} shared in common with cluster C_i.

The F-measure is the harmonic mean of the precision and recall values for each cluster. The F-measure for cluster C_i is therefore given as

$$F_i = \frac{2}{\frac{1}{prec_i} + \frac{1}{recall_i}} = \frac{2 \cdot prec_i \cdot recall_i}{prec_i + recall_i} = \frac{2\, n_{ij_i}}{n_i + m_{j_i}} \tag{17.1}$$

The F-measure for the clustering $\mathcal{C}$ is the mean of clusterwise F-measure values:

$$F = \frac{1}{r}\sum_{i=1}^{r} F_i$$

F-measure thus tries to balance the precision and recall values across all the clusters. For a perfect clustering, when $r = k$, the maximum value of the F-measure is 1.

Example 17.1. Figure 17.1 shows two different clusterings obtained via the K-means algorithm on the Iris dataset, using the first two principal components as the two dimensions. Here $n = 150$, and $k = 3$. Visual inspection confirms that Figure 17.1a is a better clustering than that in Figure 17.1b. We now examine how the different contingency table based measures can be used to evaluate these two clusterings.

Consider the clustering in Figure 17.1a. The three clusters are illustrated with different symbols; the gray points are in the correct partition, whereas the white ones are wrongly clustered compared to the ground-truth Iris types. For instance, C_3 mainly corresponds to partition T_3 (`Iris-virginica`), but it has three points (the white triangles) from T_2. The complete contingency table is as follows:

	`iris-setosa`	`iris-versicolor`	`iris-virginica`	
	T_1	T_2	T_3	n_i
C_1(squares)	0	47	14	61
C_2(circles)	50	0	0	50
C_3(triangles)	0	3	36	39
m_j	50	50	50	$n = 100$

To compute purity, we first note for each cluster the partition with the maximum overlap. We have the correspondence (C_1, T_2), (C_2, T_1), and (C_3, T_3). Thus, purity is given as

$$purity = \frac{1}{150}(47+50+36) = \frac{133}{150} = 0.887$$

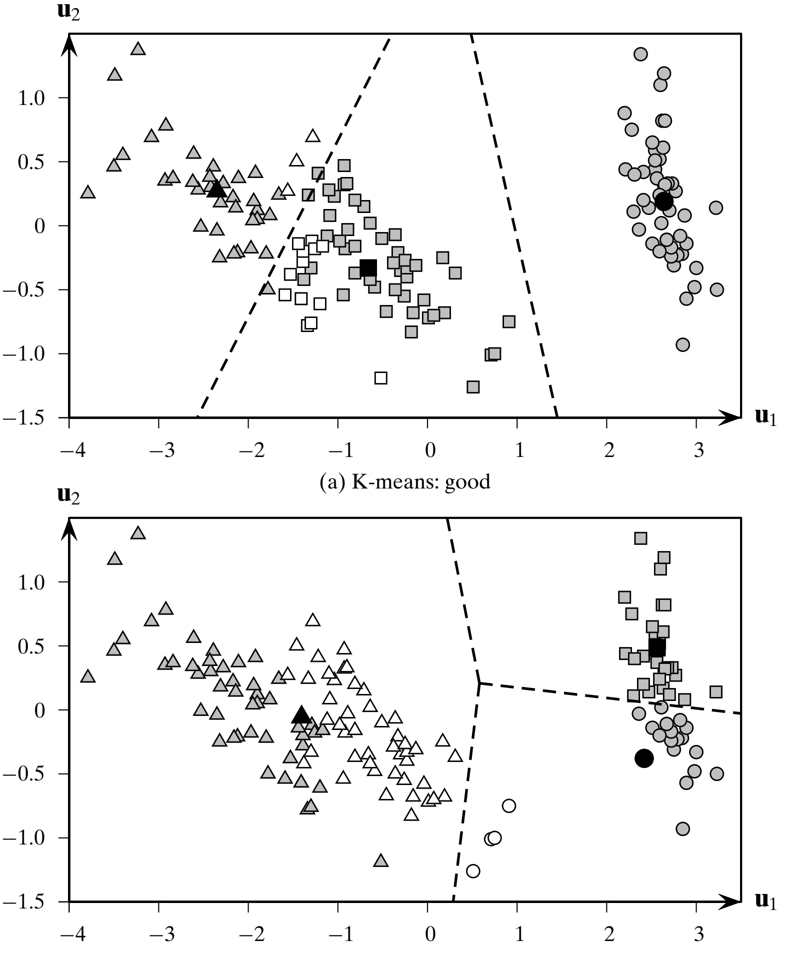
For this contingency table, the maximum matching measure gives the same result, as the correspondence above is in fact a maximum weight matching. Thus, $match = 0.887$.

The cluster C_1 contains $n_1 = 47 + 14 = 61$ points, whereas its corresponding partition T_2 contains $m_2 = 47 + 3 = 50$ points. Thus, the precision and recall for C_1 are given as

$$prec_1 = \tfrac{47}{61} = 0.77$$
$$recall_1 = \tfrac{47}{50} = 0.94$$

The F-measure for C_1 is therefore

$$F_1 = \frac{2 \cdot 0.77 \cdot 0.94}{0.77 + 0.94} = \tfrac{1.45}{1.71} = 0.85$$



(a) K-means: good

(b) K-means: bad

Figure 17.1. K-means: Iris principal components dataset.

We can also directly compute F_1 using Eq. (17.1)

$$F_1 = \frac{2 \cdot n_{12}}{n_1 + m_2} = \frac{2 \cdot 47}{61 + 50} = \frac{94}{111} = 0.85$$

Likewise, we obtain $F_2 = 1.0$ and $F_3 = 0.81$. Thus, the F-measure value for the clustering is given as

$$F = \frac{1}{3}(F_1 + F_2 + F_3) = \frac{2.66}{3} = 0.88$$

For the clustering in Figure 17.1b, we have the following contingency table:

	`iris-setosa`	`iris-versicolor`	`iris-virginica`	
	T_1	T_2	T_3	n_i
C_1	30	0	0	30
C_2	20	4	0	24
C_3	0	46	50	96
m_j	50	50	50	$n = 150$

For the purity measure, the partition with which each cluster shares the most points is given as (C_1, T_1), (C_2, T_1), and (C_3, T_3). Thus, the purity value for this clustering is

$$purity = \frac{1}{150}(30+20+50) = \frac{100}{150} = 0.67$$

We can see that both C_1 and C_2 choose partition T_1 as the maximum overlapping partition. However, the maximum weight matching is different; it yields the correspondence (C_1, T_1), (C_2, T_2), and (C_3, T_3), and thus

$$match = \frac{1}{150}(30+4+50) = \frac{84}{150} = 0.56$$

The table below compares the different contingency based measures for the two clusterings shown in Figure 17.1.

	purity	*match*	*F*
(a) Good	0.887	0.887	0.885
(b) Bad	0.667	0.560	0.658

As expected, the good clustering in Figure 17.1a has higher scores for the purity, maximum matching, and F-measure.

17.1.2 Entropy-based Measures

Conditional Entropy

The entropy of a clustering $\mathcal{C}$ is defined as

$$H(\mathcal{C}) = -\sum_{i=1}^{r} p_{C_i} \log p_{C_i}$$

where $p_{C_i} = \frac{n_i}{n}$ is the probability of cluster C_i. Likewise, the entropy of the partitioning $\mathcal{T}$ is defined as

$$H(\mathcal{T}) = -\sum_{j=1}^{k} p_{T_j} \log p_{T_j}$$

where $p_{T_j} = \frac{m_j}{n}$ is the probability of partition T_j.

The cluster-specific entropy of $\mathcal{T}$, that is, the conditional entropy of $\mathcal{T}$ with respect to cluster C_i is defined as

$$H(\mathcal{T}|C_i) = -\sum_{j=1}^{k} \left(\frac{n_{ij}}{n_i}\right) \log \left(\frac{n_{ij}}{n_i}\right)$$

The conditional entropy of $\mathcal{T}$ given clustering $\mathcal{C}$ is then defined as the weighted sum:

$$\begin{aligned} H(\mathcal{T}|\mathcal{C}) &= \sum_{i=1}^{r} \frac{n_i}{n} H(\mathcal{T}|C_i) = -\sum_{i=1}^{r}\sum_{j=1}^{k} \frac{n_{ij}}{n} \log \left(\frac{n_{ij}}{n_i}\right) \\ &= -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij} \log \left(\frac{p_{ij}}{p_{C_i}}\right) \qquad (17.2) \end{aligned}$$

where $p_{ij} = \frac{n_{ij}}{n}$ is the probability that a point in cluster i also belongs to partition j. The more a cluster's members are split into different partitions, the higher the conditional entropy. For a perfect clustering, the conditional entropy value is zero, whereas the worst possible conditional entropy value is $\log k$. Further, expanding Eq. (17.2), we can see that

$$\begin{aligned}
H(\mathcal{T}|\mathcal{C}) &= -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij}\left(\log p_{ij} - \log p_{C_i}\right) \\
&= -\left(\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij}\log p_{ij}\right) + \sum_{i=1}^{r}\left(\log p_{C_i}\sum_{j=1}^{k} p_{ij}\right) \\
&= -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij}\log p_{ij} + \sum_{i=1}^{r} p_{C_i}\log p_{C_i} \\
&= H(\mathcal{C},\mathcal{T}) - H(\mathcal{C}) \qquad (17.3)
\end{aligned}$$

where $H(\mathcal{C},\mathcal{T}) = -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij}\log p_{ij}$ is the joint entropy of $\mathcal{C}$ and $\mathcal{T}$. The conditional entropy $H(\mathcal{T}|\mathcal{C})$ thus measures the remaining entropy of $\mathcal{T}$ given the clustering $\mathcal{C}$. In particular, $H(\mathcal{T}|\mathcal{C}) = 0$ if and only if $\mathcal{T}$ is completely determined by $\mathcal{C}$, corresponding to the ideal clustering. On the other hand, if $\mathcal{C}$ and $\mathcal{T}$ are independent of each other, then $H(\mathcal{T}|\mathcal{C}) = H(\mathcal{T})$, which means that $\mathcal{C}$ provides no information about $\mathcal{T}$.

Normalized Mutual Information

The *mutual information* tries to quantify the amount of shared information between the clustering $\mathcal{C}$ and partitioning $\mathcal{T}$, and it is defined as

$$I(\mathcal{C},\mathcal{T}) = \sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij}\log\left(\frac{p_{ij}}{p_{C_i}\cdot p_{T_j}}\right) \qquad (17.4)$$

It measures the dependence between the observed joint probability p_{ij} of $\mathcal{C}$ and $\mathcal{T}$, and the expected joint probability $p_{C_i}\cdot p_{T_j}$ under the independence assumption. When $\mathcal{C}$ and $\mathcal{T}$ are independent then $p_{ij} = p_{C_i}\cdot p_{T_j}$, and thus $I(\mathcal{C},\mathcal{T}) = 0$. However, there is no upper bound on the mutual information.

Expanding Eq. (17.4) we observe that $I(\mathcal{C},\mathcal{T}) = H(\mathcal{C}) + H(\mathcal{T}) - H(\mathcal{C},\mathcal{T})$. Using Eq. (17.3), we obtain the two equivalent expressions:

$$I(\mathcal{C},\mathcal{T}) = H(\mathcal{T}) - H(\mathcal{T}|\mathcal{C})$$
$$I(\mathcal{C},\mathcal{T}) = H(\mathcal{C}) - H(\mathcal{C}|\mathcal{T})$$

Finally, because $H(\mathcal{C}|\mathcal{T}) \geq 0$ and $H(\mathcal{T}|\mathcal{C}) \geq 0$, we have the inequalities $I(\mathcal{C},\mathcal{T}) \leq H(\mathcal{C})$ and $I(\mathcal{C},\mathcal{T}) \leq H(\mathcal{T})$. We can obtain a normalized version of mutual information by considering the ratios $I(\mathcal{C},\mathcal{T})/H(\mathcal{C})$ and $I(\mathcal{C},\mathcal{T})/H(\mathcal{T})$, both of which can be at most one. The *normalized mutual information* (NMI) is defined as the geometric mean of these two ratios:

$$NMI(\mathcal{C},\mathcal{T}) = \sqrt{\frac{I(\mathcal{C},\mathcal{T})}{H(\mathcal{C})}\cdot\frac{I(\mathcal{C},\mathcal{T})}{H(\mathcal{T})}} = \frac{I(\mathcal{C},\mathcal{T})}{\sqrt{H(\mathcal{C})\cdot H(\mathcal{T})}}$$

The NMI value lies in the range $[0,1]$. Values close to 1 indicate a good clustering.

Variation of Information

This criterion is based on the mutual information between the clustering $\mathcal{C}$ and the ground-truth partitioning $\mathcal{T}$, and their entropy; it is defined as

$$\begin{aligned} VI(\mathcal{C},\mathcal{T}) &= (H(\mathcal{T}) - I(\mathcal{C},\mathcal{T})) + (H(\mathcal{C}) - I(\mathcal{C},\mathcal{T})) \\ &= H(\mathcal{T}) + H(\mathcal{C}) - 2I(\mathcal{C},\mathcal{T}) \end{aligned} \tag{17.5}$$

Variation of information (VI) is zero only when $\mathcal{C}$ and $\mathcal{T}$ are identical. Thus, the lower the VI value the better the clustering $\mathcal{C}$.

Using the equivalence $I(\mathcal{C},\mathcal{T}) = H(\mathcal{T}) - H(\mathcal{T}|\mathcal{C}) = H(\mathcal{C}) - H(\mathcal{C}|\mathcal{T})$, we can also express Eq. (17.5) as

$$VI(\mathcal{C},\mathcal{T}) = H(\mathcal{T}|\mathcal{C}) + H(\mathcal{C}|\mathcal{T})$$

Finally, noting that $H(\mathcal{T}|\mathcal{C}) = H(\mathcal{T},\mathcal{C}) - H(\mathcal{C})$, another expression for VI is given as

$$VI(\mathcal{C},\mathcal{T}) = 2H(\mathcal{T},\mathcal{C}) - H(\mathcal{T}) - H(\mathcal{C})$$

Example 17.2. We continue with Example 1, which compares the two clusterings shown in Figure 17.1. For the entropy-based measures, we use base 2 for the logarithms; the formulas are valid for any base as such.

For the clustering in Figure 17.1a, we have the following contingency table:

	iris-setosa	iris-versicolor	iris-virginica	
	T_1	T_2	T_3	n_i
C_1	0	47	14	61
C_2	50	0	0	50
C_3	0	3	36	39
m_j	50	50	50	$n = 100$

Consider the conditional entropy for cluster C_1:

$$\begin{aligned} H(\mathcal{T}|C_1) &= -\frac{0}{61}\log_2\left(\frac{0}{61}\right) - \frac{47}{61}\log_2\left(\frac{47}{61}\right) - \frac{14}{61}\log_2\left(\frac{14}{61}\right) \\ &= -0 - 0.77\log_2(0.77) - 0.23\log_2(0.23) = 0.29 + 0.49 = 0.78 \end{aligned}$$

In a similar manner, we obtain $H(\mathcal{T}|C_2) = 0$ and $H(\mathcal{T}|C_3) = 0.39$. The conditional entropy for the clustering $\mathcal{C}$ is then given as

$$H(\mathcal{T}|\mathcal{C}) = \frac{61}{150}\cdot 0.78 + \frac{50}{150}\cdot 0 + \frac{39}{150}\cdot 0.39 = 0.32 + 0 + 0.10 = 0.42$$

To compute the normalized mutual information, note that

$$H(\mathcal{T}) = -3\left(\frac{50}{150}\log_2\left(\frac{50}{150}\right)\right) = 1.585$$

$$H(\mathcal{C}) = -\left(\frac{61}{150}\log_2\left(\frac{61}{150}\right) + \frac{50}{150}\log_2\left(\frac{50}{150}\right) + \frac{39}{150}\log_2\left(\frac{39}{150}\right)\right)$$

$$= 0.528 + 0.528 + 0.505 = 1.561$$

$$I(\mathcal{C},\mathcal{T}) = \frac{47}{150}\log_2\left(\frac{47\cdot 150}{61\cdot 50}\right) + \frac{14}{150}\log_2\left(\frac{14\cdot 150}{61\cdot 50}\right) + \frac{50}{150}\log_2\left(\frac{50\cdot 150}{50\cdot 50}\right)$$

$$+\frac{3}{150}\left(\log_2\frac{3\cdot 150}{39\cdot 50}\right) + \frac{36}{150}\log_2\left(\frac{36\cdot 150}{39\cdot 50}\right)$$

$$= 0.379 - 0.05 + 0.528 - 0.042 + 0.353 = 1.167$$

Thus, the NMI and VI values are

$$NMI(\mathcal{C},\mathcal{T}) = \frac{I(\mathcal{C},\mathcal{T})}{\sqrt{H(\mathcal{T})\cdot H(\mathcal{C})}} = \frac{1.167}{\sqrt{1.585 \times 1.561}} = 0.742$$

$$VI(\mathcal{C},\mathcal{T}) = H(\mathcal{T}) + H(\mathcal{C}) - 2I(\mathcal{C},\mathcal{T}) = 1.585 + 1.561 - 2\cdot 1.167 = 0.812$$

We can likewise compute these measures for the other clustering in Figure 17.1b, whose contingency table is shown in Example 1.

The table below compares the entropy based measures for the two clusterings shown in Figure 17.1.

	$H(\mathcal{T}\|\mathcal{C})$	NMI	VI
(a) Good	0.418	0.742	0.812
(b) Bad	0.743	0.587	1.200

As expected, the good clustering in Figure 17.1a has a higher score for normalized mutual information, and lower scores for conditional entropy and variation of information.

17.1.3 Pairwise Measures

Given clustering $\mathcal{C}$ and ground-truth partitioning $\mathcal{T}$, the pairwise measures utilize the partition and cluster label information over all pairs of points. Let $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{D}$ be any two points, with $i \neq j$. Let y_i denote the true partition label and let $\hat{y}_i$ denote the cluster label for point $\mathbf{x}_i$. If both $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same cluster, that is, $\hat{y}_i = \hat{y}_j$, we call it a *positive* event, and if they do not belong to the same cluster, that is, $\hat{y}_i \neq \hat{y}_j$, we call that a *negative* event. Depending on whether there is agreement between the cluster labels and partition labels, there are four possibilities to consider:

- *True Positives:* $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same partition in $\mathcal{T}$, and they are also in the same cluster in $\mathcal{C}$. This is a true positive pair because the positive event, $\hat{y}_i = \hat{y}_j$, corresponds to the ground truth, $y_i = y_j$. The number of true positive pairs is given as

$$TP = \left|\{(\mathbf{x}_i, \mathbf{x}_j) :\ y_i = y_j \text{ and } \hat{y}_i = \hat{y}_j\}\right|$$

- *False Negatives:* $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same partition in $\mathcal{T}$, but they do not belong to the same cluster in $\mathcal{C}$. That is, the negative event, $\hat{y}_i \neq \hat{y}_j$, does not correspond to the truth, $y_i = y_j$. This pair is thus a false negative, and the number of all false negative pairs is given as

$$FN = \left|\{(\mathbf{x}_i, \mathbf{x}_j):\ y_i = y_j \text{ and } \hat{y}_i \neq \hat{y}_j\}\right|$$

- *False Positives:* $\mathbf{x}_i$ and $\mathbf{x}_j$ do not belong to the same partition in $\mathcal{T}$, but they do belong to the same cluster in $\mathcal{C}$. This pair is a false positive because the positive event, $\hat{y}_i = \hat{y}_j$, is actually false, that is, it does not agree with the ground-truth partitioning, which indicates that $y_i \neq y_j$. The number of false positive pairs is given as

$$FP = \left|\{(\mathbf{x}_i, \mathbf{x}_j):\ y_i \neq y_j \text{ and } \hat{y}_i = \hat{y}_j\}\right|$$

- *True Negatives:* $\mathbf{x}_i$ and $\mathbf{x}_j$ neither belong to the same partition in $\mathcal{T}$, nor do they belong to the same cluster in $\mathcal{C}$. This pair is thus a true negative, that is, $\hat{y}_i \neq \hat{y}_j$ and $y_i \neq y_j$. The number of such true negative pairs is given as

$$TN = \left|\{(\mathbf{x}_i, \mathbf{x}_j):\ y_i \neq y_j \text{ and } \hat{y}_i \neq \hat{y}_j\}\right|$$

Because there are $N = \binom{n}{2} = \frac{n(n-1)}{2}$ pairs of points, we have the following identity:

$$N = TP + FN + FP + TN \tag{17.6}$$

A naive computation of the preceding four cases requires $O(n^2)$ time. However, they can be computed more efficiently using the contingency table $\mathbf{N} = \{n_{ij}\}$, with $1 \leq i \leq r$ and $1 \leq j \leq k$. The number of true positives is given as

$$\begin{aligned} TP &= \sum_{i=1}^{r}\sum_{j=1}^{k}\binom{n_{ij}}{2} = \sum_{i=1}^{r}\sum_{j=1}^{k}\frac{n_{ij}(n_{ij}-1)}{2} = \frac{1}{2}\left(\sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^2 - \sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}\right) \\ &= \frac{1}{2}\left(\left(\sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^2\right) - n\right) \end{aligned} \tag{17.7}$$

This follows from the fact that each pair of points among the n_{ij} share the same cluster label (i) and the same partition label (j). The last step follows from the fact that the sum of all the entries in the contingency table must add to n, that is, $\sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij} = n$.

To compute the total number of false negatives, we remove the number of true positives from the number of pairs that belong to the same partition. Because two points $\mathbf{x}_i$ and $\mathbf{x}_j$ that belong to the same partition have $y_i = y_j$, if we remove the true positives, that is, pairs with $\hat{y}_i = \hat{y}_j$, we are left with pairs for whom $\hat{y}_i \neq \hat{y}_j$, that is, the false negatives. We thus have

$$\begin{aligned} FN &= \sum_{j=1}^{k}\binom{m_j}{2} - TP = \frac{1}{2}\left(\sum_{j=1}^{k} m_j^2 - \sum_{j=1}^{k} m_j - \sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^2 + n\right) \\ &= \frac{1}{2}\left(\sum_{j=1}^{k} m_j^2 - \sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^2\right) \end{aligned} \tag{17.8}$$

The last step follows from the fact that $\sum_{j=1}^{k} m_j = n$.

The number of false positives can be obtained in a similar manner by subtracting the number of true positives from the number of point pairs that are in the same cluster:

$$FP = \sum_{i=1}^{r} \binom{n_i}{2} - TP = \frac{1}{2}\left(\sum_{i=1}^{r} n_i^2 - \sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^2\right) \tag{17.9}$$

Finally, the number of true negatives can be obtained via Eq. (17.6) as follows:

$$TN = N - (TP + FN + FP) = \frac{1}{2}\left(n^2 - \sum_{i=1}^{r} n_i^2 - \sum_{j=1}^{k} m_j^2 + \sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^2\right) \tag{17.10}$$

Each of the four values can be computed in $O(rk)$ time. Because the contingency table can be obtained in linear time, the total time to compute the four values is $O(n + rk)$, which is much better than the naive $O(n^2)$ bound. We next consider pairwise assessment measures based on these four values.

Jaccard Coefficient

The Jaccard Coefficient measures the fraction of true positive point pairs, but after ignoring the true negatives. It is defined as follows:

$$Jaccard = \frac{TP}{TP + FN + FP} \tag{17.11}$$

For a perfect clustering $\mathcal{C}$ (i.e., total agreement with the partitioning $\mathcal{T}$), the Jaccard Coefficient has value 1, as in that case there are no false positives or false negatives. The Jaccard coefficient is asymmetric in terms of the true positives and negatives because it ignores the true negatives. In other words, it emphasizes the similarity in terms of the point pairs that belong together in both the clustering and ground-truth partitioning, but it discounts the point pairs that do not belong together.

Rand Statistic

The Rand statistic measures the fraction of true positives and true negatives over all point pairs; it is defined as

$$Rand = \frac{TP + TN}{N} \tag{17.12}$$

The Rand statistic, which is symmetric, measures the fraction of point pairs where both $\mathcal{C}$ and $\mathcal{T}$ agree. A prefect clustering has a value of 1 for the statistic.

Fowlkes-Mallows Measure

Define the overall *pairwise precision* and *pairwise recall* values for a clustering $\mathcal{C}$, as follows:

$$prec = \frac{TP}{TP + FP} \qquad\qquad recall = \frac{TP}{TP + FN}$$

Precision measures the fraction of true or correctly clustered point pairs compared to all the point pairs in the same cluster. On the other hand, recall measures the fraction of correctly labeled points pairs compared to all the point pairs in the same partition.

The Fowlkes–Mallows (FM) measure is defined as the geometric mean of the pairwise precision and recall

$$FM = \sqrt{prec \cdot recall} = \frac{TP}{\sqrt{(TP+FN)(TP+FP)}} \tag{17.13}$$

The FM measure is also asymmetric in terms of the true positives and negatives because it ignores the true negatives. Its highest value is also 1, achieved when there are no false positives or negatives.

Example 17.3. Let us continue with Example 1. Consider again the contingency table for the clustering in Figure 17.1a:

$$\left(\begin{array}{c|ccc} & \textbf{iris-setosa} & \textbf{iris-versicolor} & \textbf{iris-virginica} \\ & T_1 & T_2 & T_3 \\ \hline C_1 & 0 & 47 & 14 \\ C_2 & 50 & 0 & 0 \\ C_3 & 0 & 3 & 36 \end{array}\right)$$

Using Eq. (17.7), we can obtain the number of true positives as follows:

$$\begin{aligned} TP &= \binom{47}{2} + \binom{14}{2} + \binom{50}{2} + \binom{3}{2} + \binom{36}{2} \\ &= 1081 + 91 + 1225 + 3 + 630 = 3030 \end{aligned}$$

Using Eqs. (17.8), (17.9), and (17.10), we obtain

$$FN = 645 \qquad FP = 766 \qquad TN = 6734$$

Note that there are a total of $N = \binom{150}{2} = 11175$ point pairs.

We can now compute the different pairwise measures for clustering evaluation. The Jaccard coefficient [Eq. (17.11)], Rand statistic [Eq. (17.12)], and Fowlkes–Mallows measure [Eq. (17.13)], are given as

$$\begin{aligned} Jaccard &= \frac{3030}{3030+645+766} = \frac{3030}{4441} = 0.68 \\ Rand &= \frac{3030+6734}{11175} = \frac{9764}{11175} = 0.87 \\ FM &= \frac{3030}{\sqrt{3675 \cdot 3796}} = \frac{3030}{3735} = 0.81 \end{aligned}$$

Using the contingency table for the clustering in Figure 17.1b from Example 1, we obtain

$$TP = 2891 \qquad FN = 784 \qquad FP = 2380 \qquad TN = 5120$$

The table below compares the different contingency based measures on the two clusterings in Figure 17.1.

	Jaccard	*Rand*	*FM*
(a) Good	0.682	0.873	0.811
(b) Bad	0.477	0.717	0.657

As expected, the clustering in Figure 17.1a has higher scores for all three measures.

17.1.4 Correlation Measures

Let $\mathbf{X}$ and $\mathbf{Y}$ be two symmetric $n \times n$ matrices, and let $N = \binom{n}{2}$. Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ denote the vectors obtained by linearizing the upper triangular elements (excluding the main diagonal) of $\mathbf{X}$ and $\mathbf{Y}$ (e.g., in a row-wise manner), respectively. Let μ_X denote the element-wise mean of $\mathbf{x}$, given as

$$\mu_X = \frac{1}{N} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{X}(i,j) = \frac{1}{N} \mathbf{x}^T \mathbf{x}$$

and let $\mathbf{z}_x$ denote the centered $\mathbf{x}$ vector, defined as

$$\mathbf{z}_x = \mathbf{x} - \mathbf{1} \cdot \mu_X$$

where $\mathbf{1} \in \mathbb{R}^N$ is the vector of all ones. Likewise, let μ_Y be the element-wise mean of $\mathbf{y}$, and $\mathbf{z}_y$ the centered $\mathbf{y}$ vector.

The Hubert statistic is defined as the averaged element-wise product between $\mathbf{X}$ and $\mathbf{Y}$

$$\Gamma = \frac{1}{N} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{X}(i,j) \cdot \mathbf{Y}(i,j) = \frac{1}{N} \mathbf{x}^T \mathbf{y} \tag{17.14}$$

The normalized Hubert statistic is defined as the element-wise correlation between $\mathbf{X}$ and $\mathbf{Y}$

$$\Gamma_n = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \big(\mathbf{X}(i,j) - \mu_X\big)\big(\cdot \mathbf{Y}(i,j) - \mu_Y\big)}{\sqrt{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \big(\mathbf{X}(i,j) - \mu_X\big)^2 \quad \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \big(\mathbf{Y}[i] - \mu_Y\big)^2}} = \frac{\sigma_{XY}}{\sqrt{\sigma_X^2 \sigma_Y^2}}$$

where σ_X^2 and σ_Y^2 are the variances, and σ_{XY} the covariance, for the vectors $\mathbf{x}$ and $\mathbf{y}$, defined as

$$\sigma_X^2 = \frac{1}{N} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \big(\mathbf{X}(i,j) - \mu_X\big)^2 = \frac{1}{N} \mathbf{z}_x^T \mathbf{z}_x = \frac{1}{N} \|\mathbf{z}_x\|^2$$

$$\sigma_Y^2 = \frac{1}{N} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \big(\mathbf{Y}(i,j) - \mu_Y\big)^2 = \frac{1}{N} \mathbf{z}_y^T \mathbf{z}_y = \frac{1}{N} \|\mathbf{z}_y\|^2$$

$$\sigma_{XY} = \frac{1}{N} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \big(\mathbf{X}(i,j) - \mu_X\big)\big(\mathbf{Y}(i,j) - \mu_Y\big) = \frac{1}{N} \mathbf{z}_x^T \mathbf{z}_y$$

Thus, the normalized Hubert statistic can be rewritten as

$$\Gamma_n = \frac{\mathbf{z}_x^T \mathbf{z}_y}{\|\mathbf{z}_x\| \cdot \|\mathbf{z}_y\|} = \cos\theta \tag{17.15}$$

where θ is the angle between the two centered vectors $\mathbf{z}_x$ and $\mathbf{z}_y$. It follows immediately that Γ_n ranges from -1 to $+1$.

When $\mathbf{X}$ and $\mathbf{Y}$ are arbitrary $n \times n$ matrices the above expressions can be easily modified to range over all the n^2 elements of the two matrices. The (normalized) Hubert statistic can be used as an external evaluation measure, with appropriately defined matrices $\mathbf{X}$ and $\mathbf{Y}$, as described next.

Discretized Hubert Statistic

Let $\mathbf{T}$ and $\mathbf{C}$ be the $n \times n$ matrices defined as

$$\mathbf{T}(i,j) = \begin{cases} 1 & \text{if } y_i = y_j, i \neq j \\ 0 & \text{otherwise} \end{cases} \qquad \mathbf{C}(i,j) = \begin{cases} 1 & \text{if } \hat{y}_i = \hat{y}_j, i \neq j \\ 0 & \text{otherwise} \end{cases}$$

Also, let $\mathbf{t}, \mathbf{c} \in \mathbb{R}^N$ denote the N-dimensional vectors comprising the upper triangular elements (excluding the diagonal) of $\mathbf{T}$ and $\mathbf{C}$, respectively, where $N = \binom{n}{2}$ denotes the number of distinct point pairs. Finally, let $\mathbf{z}_t$ and $\mathbf{z}_c$ denote the centered $\mathbf{t}$ and $\mathbf{c}$ vectors.

The discretized Hubert statistic is computed via Eq. (17.14), by setting $\mathbf{x} = \mathbf{t}$ and $\mathbf{y} = \mathbf{c}$:

$$\Gamma = \frac{1}{N}\mathbf{t}^T\mathbf{c} = \frac{TP}{N} \tag{17.16}$$

Because the ith element of $\mathbf{t}$ is 1 only when the ith pair of points belongs to the same partition, and, likewise, the ith element of $\mathbf{c}$ is 1 only when the ith pair of points also belongs to the same cluster, the dot product $\mathbf{t}^T\mathbf{c}$ is simply the number of true positives, and thus the Γ value is equivalent to the fraction of all pairs that are true positives. It follows that the higher the agreement between the ground-truth partitioning $\mathcal{T}$ and clustering $\mathcal{C}$, the higher the Γ value.

Normalized Discretized Hubert Statistic

The normalized version of the discretized Hubert statistic is simply the correlation between $\mathbf{t}$ and $\mathbf{c}$ [Eq. (17.15)]:

$$\Gamma_n = \frac{\mathbf{z}_t^T\mathbf{z}_c}{\|\mathbf{z}_t\| \cdot \|\mathbf{z}_c\|} = \cos\theta \tag{17.17}$$

Note that $\mu_T = \frac{1}{N}\mathbf{t}^T\mathbf{t}$ is the fraction of point pairs that belong to the same partition, that is, with $y_i = y_j$, regardless of whether $\hat{y}_i$ matches $\hat{y}_j$ or not. Thus, we have

$$\mu_T = \frac{\mathbf{t}^T\mathbf{t}}{N} = \frac{TP + FN}{N}$$

Similarly, $\mu_C = \frac{1}{N}\mathbf{c}^T\mathbf{c}$ is the fraction of point pairs that belong to the same cluster, that is, with $\hat{y}_i = \hat{y}_j$, regardless of whether y_i matches y_j or not, so that

$$\mu_C = \frac{\mathbf{c}^T\mathbf{c}}{N} = \frac{TP + FP}{N}$$

Substituting these into the numerator in Eq. (17.17), we get

$$\begin{aligned}
\mathbf{z}_t^T\mathbf{z}_c &= (\mathbf{t} - \mathbf{1}\cdot\mu_T)^T(\mathbf{c} - \mathbf{1}\cdot\mu_C) \\
&= \mathbf{t}^T\mathbf{c} - \mu_C\mathbf{t}^T\mathbf{1} - \mu_T\mathbf{c}^T\mathbf{1} + \mathbf{1}^T\mathbf{1}\mu_T\mu_C \\
&= \mathbf{t}^T\mathbf{c} - N\mu_C\mu_T - N\mu_T\mu_C + N\mu_T\mu_C \\
&= \mathbf{t}^T\mathbf{c} - N\mu_T\mu_C \\
&= TP - N\mu_T\mu_C
\end{aligned} \tag{17.18}$$

where $\mathbf{1} \in \mathbb{R}^N$ is the vector of all 1's. We also made use of identities $\mathbf{t}^T\mathbf{1} = \mathbf{t}^T\mathbf{t}$ and $\mathbf{c}^T\mathbf{1} = \mathbf{c}^T\mathbf{c}$. Likewise, we can derive

$$\|\mathbf{z}_t\|^2 = \mathbf{z}_t^T\mathbf{z}_t = \mathbf{t}^T\mathbf{t} - N\mu_T^2 = N\mu_T - N\mu_T^2 = N\mu_T(1-\mu_T) \tag{17.19}$$

$$\|\mathbf{z}_c\|^2 = \mathbf{z}_c^T\mathbf{z}_c = \mathbf{c}^T\mathbf{c} - N\mu_C^2 = N\mu_C - N\mu_C^2 = N\mu_C(1-\mu_C) \tag{17.20}$$

Plugging Eqs. (17.18), (17.19), and (17.20) into Eq. (17.17) the normalized, discretized Hubert statistic can be written as

$$\Gamma_n = \frac{\frac{TP}{N} - \mu_T\mu_C}{\sqrt{\mu_T\mu_C(1-\mu_T)(1-\mu_C)}} \tag{17.21}$$

because $\mu_T = \frac{TP+FN}{N}$ and $\mu_C = \frac{TP+FP}{N}$, the normalized Γ_n statistic can be computed using only the TP, FN, and FP values. The maximum value of $\Gamma_n = +1$ is obtained when there are no false positives or negatives, that is, when $FN = FP = 0$. The minimum value of $\Gamma_n = -1$ is when there are no true positives and negatives, that is, when $TP = TN = 0$.

Example 17.4. Continuing Example 17.3, for the good clustering in Figure 17.1a, we have

$$TP = 3030 \qquad FN = 645 \qquad FP = 766 \qquad TN = 6734$$

From these values, we obtain

$$\mu_T = \frac{TP+FN}{N} = \frac{3675}{11175} = 0.33$$

$$\mu_C = \frac{TP+FP}{N} = \frac{3796}{11175} = 0.34$$

Using Eqs. (17.16) and (17.21) the Hubert statistic values are

$$\Gamma = \frac{3030}{11175} = 0.271$$

$$\Gamma_n = \frac{0.27 - 0.33\cdot 0.34}{\sqrt{0.33\cdot 0.34\cdot(1-0.33)\cdot(1-0.34)}} = \frac{0.159}{0.222} = 0.717$$

Likewise, for the bad clustering in Figure 17.1b, we have

$$TP = 2891 \qquad FN = 784 \qquad FP = 2380 \qquad TN = 5120$$

and the values for the discretized Hubert statistic are given as

$$\Gamma = 0.258 \qquad\qquad \Gamma_n = 0.442$$

We observe that the good clustering has higher values, though the normalized statistic is more discerning than the unnormalized version, that is, the good clustering has a much higher value of Γ_n than the bad clustering, whereas the difference in Γ for the two clusterings is not that high.

17.2 INTERNAL MEASURES

Internal evaluation measures do not have recourse to the ground-truth partitioning, which is the typical scenario when clustering a dataset. To evaluate the quality of the clustering, internal measures therefore have to utilize notions of intracluster similarity or compactness, contrasted with notions of intercluster separation, with usually a trade-off in maximizing these two aims. The internal measures are based on the $n \times n$ *distance matrix*, also called the *proximity matrix*, of all pairwise distances among the n points:

$$\mathbf{W} = \Big\{\delta(\mathbf{x}_i, \mathbf{x}_j)\Big\}_{i,j=1}^{n} \tag{17.22}$$

where

$$\delta(\mathbf{x}_i, \mathbf{x}_j) = \big\|\mathbf{x}_i - \mathbf{x}_j\big\|_2$$

is the Euclidean distance between $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{D}$, although other distance metrics can also be used. Because $\mathbf{W}$ is symmetric and $\delta(\mathbf{x}_i, \mathbf{x}_i) = 0$, usually only the upper triangular elements of $\mathbf{W}$ (excluding the diagonal) are used in the internal measures.

The proximity matrix $\mathbf{W}$ can also be considered as the adjacency matrix of the weighted complete graph G over the n points, that is, with nodes $V = \{\mathbf{x}_i \mid \mathbf{x}_i \in \mathbf{D}\}$, edges $E = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \mathbf{x}_i, \mathbf{x}_j \in \mathbf{D}\}$, and edge weights $w_{ij} = \mathbf{W}(i, j)$ for all $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{D}$. There is thus a close connection between the internal evaluation measures and the graph clustering objectives we examined in Chapter 16.

For internal measures, we assume that we do not have access to a ground-truth partitioning. Instead, we assume that we are given a clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ comprising $r = k$ clusters, with cluster C_i containing $n_i = |C_i|$ points. Let $\hat{y}_i \in \{1, 2, \ldots, k\}$ denote the cluster label for point $\mathbf{x}_i$. The clustering $\mathcal{C}$ can be considered as a k-way cut in G because $C_i \neq \emptyset$ for all i, $C_i \cap C_j = \emptyset$ for all i, j, and $\bigcup_i C_i = V$. Given any subsets $S, R \subset V$, define $W(S, R)$ as the sum of the weights on all edges with one vertex in S and the other in R, given as

$$W(S, R) = \sum_{\mathbf{x}_i \in S} \sum_{\mathbf{x}_j \in R} w_{ij}$$

Also, given $S \subseteq V$, we denote by $\overline{S}$ the complementary set of vertices, that is, $\overline{S} = V - S$.

The internal measures are based on various functions over the intracluster and intercluster weights. In particular, note that the sum of all the intracluster weights over all clusters is given as

$$W_{in} = \frac{1}{2}\sum_{i=1}^{k} W(C_i, C_i) \tag{17.23}$$

We divide by 2 because each edge within C_i is counted twice in the summation given by $W(C_i, C_i)$. Also note that the sum of all intercluster weights is given as

$$W_{out} = \frac{1}{2}\sum_{i=1}^{k} W(C_i, \overline{C_i}) = \sum_{i=1}^{k-1}\sum_{j>i} W(C_i, C_j) \tag{17.24}$$

Here too we divide by 2 because each edge is counted twice in the summation across clusters. The number of distinct intracluster edges, denoted N_{in}, and intercluster edges, denoted N_{out}, are given as

$$N_{in} = \sum_{i=1}^{k}\binom{n_i}{2} = \frac{1}{2}\sum_{i=1}^{k} n_i(n_i - 1)$$

$$N_{out} = \sum_{i=1}^{k-1}\sum_{j=i+1}^{k} n_i \cdot n_j = \frac{1}{2}\sum_{i=1}^{k}\sum_{\substack{j=1\\ j\neq i}}^{k} n_i \cdot n_j$$

Note that the total number of distinct pairs of points N satisfies the identity

$$N = N_{in} + N_{out} = \binom{n}{2} = \frac{1}{2}n(n-1)$$

Example 17.5. Figure 17.2 shows the graphs corresponding to the two K-means clusterings shown in Figure 17.1. Here, each vertex corresponds to a point $\mathbf{x}_i \in \mathbf{D}$, and an edge $(\mathbf{x}_i, \mathbf{x}_j)$ exists between each pair of points. However, only the intracluster edges are shown (with intercluster edges omitted) to avoid clutter. Because internal measures do not have access to a ground truth labeling, the goodness of a clustering is measured based on intracluster and intercluster statistics.

BetaCV Measure

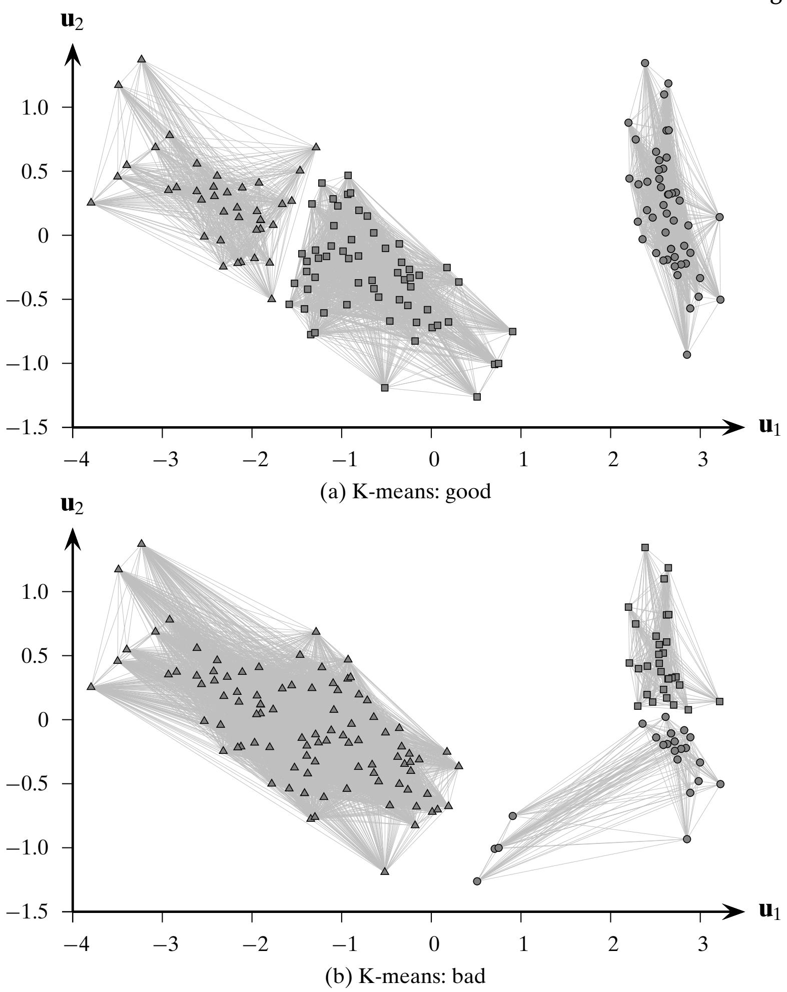
The BetaCV measure is the ratio of the mean intracluster distance to the mean intercluster distance:

$$BetaCV = \frac{W_{in}/N_{in}}{W_{out}/N_{out}} = \frac{N_{out}}{N_{in}} \cdot \frac{W_{in}}{W_{out}} = \frac{N_{out}}{N_{in}} \frac{\sum_{i=1}^{k} W(C_i, C_i)}{\sum_{i=1}^{k} W(C_i, \overline{C_i})}$$

The smaller the BetaCV ratio, the better the clustering, as it indicates that intracluster distances are on average smaller than intercluster distances.

C-index

Let $W_{\min}(N_{in})$ be the sum of the smallest N_{in} distances in the proximity matrix $\mathbf{W}$, where N_{in} is the total number of intracluster edges, or point pairs. Let $W_{\max}(N_{in})$ be the sum of the largest N_{in} distances in $\mathbf{W}$.



(a) K-means: good

(b) K-means: bad

Figure 17.2. Clusterings as graphs: Iris.

The C-index measures to what extent the clustering puts together the N_{in} points that are the closest across the k clusters. It is defined as

$$Cindex = \frac{W_{in} - W_{\min}(N_{in})}{W_{\max}(N_{in}) - W_{\min}(N_{in})}$$

where W_{in} is the sum of all the intracluster distances [Eq. (17.23)]. The C-index lies in the range $[0, 1]$. The smaller the C-index, the better the clustering, as it indicates more compact clusters with relatively smaller distances within clusters rather than between clusters.

Normalized Cut Measure

The normalized cut objective [Eq. (16.17)] for graph clustering can also be used as an internal clustering evaluation measure:

$$NC = \sum_{i=1}^{k} \frac{W(C_i, \overline{C_i})}{vol(C_i)} = \sum_{i=1}^{k} \frac{W(C_i, \overline{C_i})}{W(C_i, V)}$$

where $vol(C_i) = W(C_i, V)$ is the volume of cluster C_i, that is, the total weights on edges with at least one end in the cluster. However, because we are using the proximity or distance matrix $\mathbf{W}$, instead of the affinity or similarity matrix $\mathbf{A}$, the higher the normalized cut value the better.

To see this, we make use of the observation that $W(C_i, V) = W(C_i, C_i) + W(C_i, \overline{C_i})$, so that

$$NC = \sum_{i=1}^{k} \frac{W(C_i, \overline{C_i})}{W(C_i, C_i) + W(C_i, \overline{C_i})} = \sum_{i=1}^{k} \frac{1}{\dfrac{W(C_i, C_i)}{W(C_i, \overline{C_i})} + 1}$$

We can see that NC is maximized when the ratios $\dfrac{W(C_i, C_i)}{W(C_i, \overline{C_i})}$ (across the k clusters) are as small as possible, which happens when the intracluster distances are much smaller compared to intercluster distances, that is, when the clustering is good. The maximum possible value of NC is k.

Modularity

The modularity objective for graph clustering [Eq. (16.26)] can also be used as an internal measure:

$$Q = \sum_{i=1}^{k} \left(\frac{W(C_i, C_i)}{W(V, V)} - \left(\frac{W(C_i, V)}{W(V, V)} \right)^2 \right)$$

where

$$\begin{aligned} W(V, V) &= \sum_{i=1}^{k} W(C_i, V) \\ &= \sum_{i=1}^{k} W(C_i, C_i) + \sum_{i=1}^{k} W(C_i, \overline{C_i}) \\ &= 2(W_{in} + W_{out}) \end{aligned}$$

The last step follows from Eqs. (17.23) and (17.24). Modularity measures the difference between the observed and expected fraction of weights on edges within the clusters. Since we are using the distance matrix, the smaller the modularity measure the better the clustering, which indicates that the intracluster distances are lower than expected.

Dunn Index

The Dunn index is defined as the ratio between the minimum distance between point pairs from different clusters and the maximum distance between point pairs from the same cluster. More formally, we have

$$Dunn = \frac{W_{out}^{\min}}{W_{in}^{\max}}$$

where $W_{out}^{\min}$ is the minimum intercluster distance:

$$W_{out}^{\min} = \min_{i, j > i} \left\{ w_{ab} | \mathbf{x}_a \in C_i, \mathbf{x}_b \in C_j \right\}$$

and $W_{in}^{\max}$ is the maximum intracluster distance:

$$W_{in}^{\max} = \max_i \left\{ w_{ab} | \mathbf{x}_a, \mathbf{x}_b \in C_i \right\}$$

The larger the Dunn index the better the clustering because it means even the closest distance between points in different clusters is much larger than the farthest distance between points in the same cluster. However, the Dunn index may be insensitive because the minimum intercluster and maximum intracluster distances do not capture all the information about a clustering.

Davies–Bouldin Index

Let μ_i denote the cluster mean, given as

$$\mu_i = \frac{1}{n_i} \sum_{\mathbf{x}_j \in C_i} \mathbf{x}_j \tag{17.25}$$

Further, let σ_{μ_i} denote the dispersion or spread of the points around the cluster mean, given as

$$\sigma_{\mu_i} = \sqrt{\frac{\sum_{\mathbf{x}_j \in C_i} \delta(\mathbf{x}_j, \mu_i)^2}{n_i}} = \sqrt{var(C_i)}$$

where $var(C_i)$ is the total variance [Eq. (1.4)] of cluster C_i.

The Davies–Bouldin measure for a pair of clusters C_i and C_j is defined as the ratio

$$DB_{ij} = \frac{\sigma_{\mu_i} + \sigma_{\mu_j}}{\delta(\mu_i, \mu_j)}$$

DB_{ij} measures how compact the clusters are compared to the distance between the cluster means. The Davies–Bouldin index is then defined as

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \{DB_{ij}\}$$

That is, for each cluster C_i, we pick the cluster C_j that yields the largest DB_{ij} ratio. The smaller the DB value the better the clustering, as it means that the clusters are well separated (i.e., the distance between cluster means is large), and each cluster is well represented by its mean (i.e., has a small spread).

Silhouette Coefficient

The silhouette coefficient is a measure of both cohesion and separation of clusters, and is based on the difference between the average distance to points in the closest cluster and to points in the same cluster. For each point $\mathbf{x}_i$ we calculate its silhouette coefficient s_i as

$$s_i = \frac{\mu_{out}^{\min}(\mathbf{x}_i) - \mu_{in}(\mathbf{x}_i)}{\max\left\{\mu_{out}^{\min}(\mathbf{x}_i), \mu_{in}(\mathbf{x}_i)\right\}} \tag{17.26}$$

where $\mu_{in}(\mathbf{x}_i)$ is the mean distance from $\mathbf{x}_i$ to points in its own cluster $\hat{y}_i$:

$$\mu_{in}(\mathbf{x}_i) = \frac{\sum_{\mathbf{x}_j \in C_{\hat{y}_i}, j \neq i} \delta(\mathbf{x}_i, \mathbf{x}_j)}{n_{\hat{y}_i} - 1}$$

and $\mu_{out}^{\min}(\mathbf{x}_i)$ is the mean of the distances from $\mathbf{x}_i$ to points in the closest cluster:

$$\mu_{out}^{\min}(\mathbf{x}_i) = \min_{j \neq \hat{y}_i} \left\{ \frac{\sum_{\mathbf{y} \in C_j} \delta(\mathbf{x}_i, \mathbf{y})}{n_j} \right\}$$

The s_i value of a point lies in the interval $[-1, +1]$. A value close to $+1$ indicates that $\mathbf{x}_i$ is much closer to points in its own cluster and is far from other clusters. A value close to zero indicates that $\mathbf{x}_i$ is close to the boundary between two clusters. Finally, a value close to -1 indicates that $\mathbf{x}_i$ is much closer to another cluster than its own cluster, and therefore, the point may be mis-clustered.

The silhouette coefficient is defined as the mean s_i value across all the points:

$$SC = \frac{1}{n} \sum_{i=1}^{n} s_i \tag{17.27}$$

A value close to $+1$ indicates a good clustering.

Hubert Statistic

The Hubert Γ statistic [Eq. (17.14)], and its normalized version Γ_n [Eq. (17.15)], can both be used as internal evaluation measures by letting $\mathbf{X} = \mathbf{W}$ be the pairwise distance matrix, and by defining $\mathbf{Y}$ as the matrix of distances between the cluster means:

$$\mathbf{Y} = \left\{ \delta(\mu_{\hat{y}_i}, \mu_{\hat{y}_j}) \right\}_{i,j=1}^{n} \tag{17.28}$$

Because both $\mathbf{W}$ and $\mathbf{Y}$ are symmetric, both Γ and Γ_n are computed over their upper triangular elements.

Example 17.6. Consider the two clusterings for the Iris principal components dataset shown in Figure 17.1, along with their corresponding graph representations in Figure 17.2. Let us evaluate these two clusterings using internal measures.

The good clustering shown in Figure 17.1a and Figure 17.2a has clusters with the following sizes:

$$n_1 = 61 \qquad n_2 = 50 \qquad n_3 = 39$$

Thus, the number of intracluster and intercluster edges (i.e., point pairs) is given as

$$N_{in} = \binom{61}{2} + \binom{50}{2} + \binom{31}{2} = 1830 + 1225 + 741 = 3796$$

$$N_{out} = 61 \cdot 50 + 61 \cdot 39 + 50 \cdot 39 = 3050 + 2379 + 1950 = 7379$$

In total there are $N = N_{in} + N_{out} = 3796 + 7379 = 11175$ distinct point pairs.

The weights on edges within each cluster $W(C_i, C_i)$, and those from a cluster to another $W(C_i, C_j)$, are as given in the intercluster weight matrix

$$\left(\begin{array}{c|ccc} W & C_1 & C_2 & C_3 \\ \hline C_1 & 3265.69 & 10402.30 & 4418.62 \\ C_2 & 10402.30 & 1523.10 & 9792.45 \\ C_3 & 4418.62 & 9792.45 & 1252.36 \end{array}\right) \tag{17.29}$$

Thus, the sum of all the intracluster and intercluster edge weights is

$$W_{in} = \frac{1}{2}(3265.69 + 1523.10 + 1252.36) = 3020.57$$

$$W_{out} = (10402.30 + 4418.62 + 9792.45) = 24613.37$$

The BetaCV measure can then be computed as

$$BetaCV = \frac{N_{out} \cdot W_{in}}{N_{in} \cdot W_{out}} = \frac{7379 \times 3020.57}{3796 \times 24613.37} = 0.239$$

For the C-index, we first compute the sum of the N_{in} smallest and largest pair-wise distances, given as

$$W_{\min}(N_{in}) = 2535.96 \qquad W_{\max}(N_{in}) = 16889.57$$

Thus, C-index is given as

$$Cindex = \frac{W_{in} - W_{\min}(N_{in})}{W_{\max}(N_{in}) - W_{\min}(N_{in})} = \frac{3020.57 - 2535.96}{16889.57 - 2535.96} = \frac{484.61}{14535.61} = 0.0338$$

For the normalized cut and modularity measures, we compute $W(C_i, \overline{C_i})$, $W(C_i, V) = \sum_{j=1}^{k} W(C_i, C_j)$ and $W(V, V) = \sum_{i=1}^{k} W(C_i, V)$, using the intercluster weight matrix [Eq. (17.29)]:

$$W(C_1, \overline{C_1}) = 10402.30 + 4418.62 = 14820.91$$

$$W(C_2, \overline{C_2}) = 10402.30 + 9792.45 = 20194.75$$

$$W(C_3, \overline{C_3}) = 4418.62 + 9792.45 = 14211.07$$

$$W(C_1, V) = 3265.69 + W(C_1, \overline{C_1}) = 18086.61$$

$$W(C_2, V) = 1523.10 + W(C_2, \overline{C_2}) = 21717.85$$

$$W(C_3, V) = 1252.36 + W(C_3, \overline{C_3}) = 15463.43$$

$$W(V, V) = W(C_1, V) + W(C_2, V) + W(C_3, V) = 55267.89$$

The normalized cut and modularity values are given as

$$NC = \frac{14820.91}{18086.61} + \frac{20194.75}{21717.85} + \frac{14211.07}{15463.43} = 0.819 + 0.93 + 0.919 = 2.67$$

$$\begin{aligned} Q &= \left(\frac{3265.69}{55267.89} - \left(\frac{18086.61}{55267.89}\right)^2\right) + \left(\frac{1523.10}{55267.89} - \left(\frac{21717.85}{55267.89}\right)^2\right) \\ &\quad + \left(\frac{1252.36}{55267.89} - \left(\frac{15463.43}{55267.89}\right)^2\right) \\ &= -0.048 - 0.1269 - 0.0556 = -0.2305 \end{aligned}$$

The Dunn index can be computed from the minimum and maximum distances between pairs of points from two clusters C_i and C_j, computed as follows:

$$\left(\begin{array}{c|ccc} W^{\min} & C_1 & C_2 & C_3 \\ \hline C_1 & 0 & 1.62 & 0.198 \\ C_2 & 1.62 & 0 & 3.49 \\ C_3 & 0.198 & 3.49 & 0 \end{array}\right) \qquad \left(\begin{array}{c|ccc} W^{\max} & C_1 & C_2 & C_3 \\ \hline C_1 & 2.50 & 4.85 & 4.81 \\ C_2 & 4.85 & 2.33 & 7.06 \\ C_3 & 4.81 & 7.06 & 2.55 \end{array}\right)$$

The Dunn index value for the clustering is given as

$$Dunn = \frac{W_{out}^{\min}}{W_{in}^{\max}} = \frac{0.198}{2.55} = 0.078$$

To compute the Davies–Bouldin index, we compute the cluster mean and dispersion values:

$$\mu_1 = \begin{pmatrix} -0.664 \\ -0.33 \end{pmatrix} \qquad \mu_2 = \begin{pmatrix} 2.64 \\ 0.19 \end{pmatrix} \qquad \mu_3 = \begin{pmatrix} -2.35 \\ 0.27 \end{pmatrix}$$

$$\sigma_{\mu_1} = 0.723 \qquad \sigma_{\mu_2} = 0.512 \qquad \sigma_{\mu_3} = 0.695$$

and the DB_{ij} values for pairs of clusters:

$$\left(\begin{array}{c|ccc} DB_{ij} & C_1 & C_2 & C_3 \\ \hline C_1 & - & 0.369 & 0.794 \\ C_2 & 0.369 & - & 0.242 \\ C_3 & 0.794 & 0.242 & - \end{array}\right)$$

For example, $DB_{12} = \frac{\sigma_{\mu_1}+\sigma_{\mu_2}}{\delta(\mu_1,\mu_2)} = \frac{1.235}{3.346} = 0.369$. Finally, the DB index is given as

$$DB = \frac{1}{3}(0.794 + 0.369 + 0.794) = 0.652$$

The silhouette coefficient [Eq. (17.26)] for a chosen point, say $\mathbf{x}_1$, is given as

$$s_1 = \frac{1.902 - 0.701}{\max\{1.902, 0.701\}} = \frac{1.201}{1.902} = 0.632$$

The average value across all points is $SC = 0.598$

The Hubert statistic can be computed by taking the dot product over the upper triangular elements of the proximity matrix $\mathbf{W}$ [Eq. (17.22)] and the $n \times n$ matrix of distances among cluster means $\mathbf{Y}$ [Eq. (17.28)], and then dividing by the number of distinct point pairs N:

$$\Gamma = \frac{\mathbf{w}^T\mathbf{y}}{N} = \frac{91545.85}{11175} = 8.19$$

where $\mathbf{w}, \mathbf{y} \in \mathbb{R}^N$ are vectors comprising the upper triangular elements of $\mathbf{W}$ and $\mathbf{Y}$. The normalized Hubert statistic can be obtained as the correlation between $\mathbf{w}$ and $\mathbf{y}$ [Eq. (17.15)]:

$$\Gamma_n = \frac{\mathbf{z}_w^T\mathbf{z}_y}{\|\mathbf{x}_w\| \cdot \|\mathbf{z}_y\|} = 0.918$$

where $\mathbf{z}_w, \mathbf{z}_y$ are the centered vectors corresponding to $\mathbf{w}$ and $\mathbf{y}$, respectively.

The following table summarizes the various internal measure values for the good and bad clusterings shown in Figure 17.1 and Figure 17.2.

	Lower better				Higher better				
	BetaCV	*Cindex*	*Q*	*DB*	*NC*	*Dunn*	*SC*	Γ	Γ_n
(a) Good	0.24	0.034	−0.23	0.65	2.67	0.08	0.60	8.19	0.92
(b) Bad	0.33	0.08	−0.20	1.11	2.56	0.03	0.55	7.32	0.83

Despite the fact that these internal measures do not have access to the ground-truth partitioning, we can observe that the good clustering has higher values for normalized cut, Dunn, silhouette coefficient, and the Hubert statistics, and lower values for BetaCV, C-index, modularity, and Davies–Bouldin measures. These measures are thus capable of discerning good versus bad clusterings of the data.

17.3 RELATIVE MEASURES

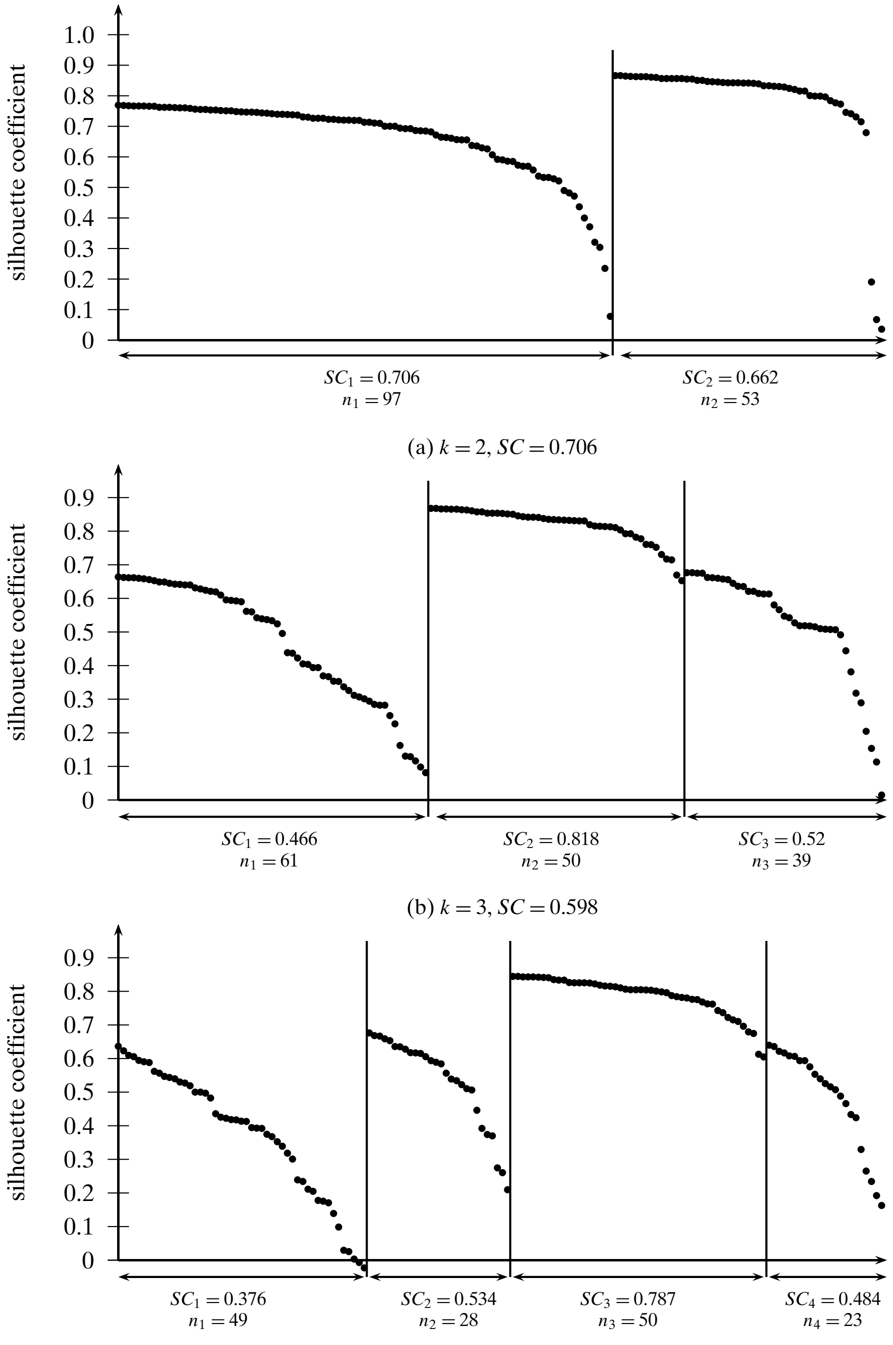
Relative measures are used to compare different clusterings obtained by varying different parameters for the same algorithm, for example, to choose the number of clusters k.

Silhouette Coefficient

The silhouette coefficient [Eq. (17.26)] for each point s_j, and the average SC value [Eq. (17.27)], can be used to estimate the number of clusters in the data. The approach consists of plotting the s_j values in descending order for each cluster, and to note the overall SC value for a particular value of k, as well as clusterwise SC values:

$$SC_i = \frac{1}{n_i} \sum_{\mathbf{x}_j \in C_i} s_j$$

We can then pick the value k that yields the best clustering, with many points having high s_j values within each cluster, as well as high values for SC and SC_i $(1 \le i \le k)$.



(a) $k = 2, SC = 0.706$

(b) $k = 3, SC = 0.598$

(c) $k = 4, SC = 0.559$

Figure 17.3. Iris K-means: silhouette coefficient plot.

Example 17.7. Figure 17.3 shows the silhouette coefficient plot for the best clustering results for the K-means algorithm on the Iris principal components dataset for three different values of k, namely $k = 2, 3, 4$. The silhouette coefficient values s_i for points within each cluster are plotted in decreasing order. The overall average (SC) and clusterwise averages (SC_i, for $1 \leq i \leq k$) are also shown, along with the cluster sizes.

Figure 17.3a shows that $k = 2$ has the highest average silhouette coefficient, $SC = 0.706$. It shows two well separated clusters. The points in cluster C_1 start out with high s_i values, which gradually drop as we get to border points. The second cluster C_2 is even better separated, since it has a higher silhouette coefficient and the pointwise scores are all high, except for the last three points, suggesting that almost all the points are well clustered.

The silhouette plot in Figure 17.3b, with $k = 3$, corresponds to the "good" clustering shown in Figure 17.1a. We can see that cluster C_1 from Figure 17.3a has been split into two clusters for $k = 3$, namely C_1 and C_3. Both of these have many bordering points, whereas C_2 is well separated with high silhouette coefficients across all points.

Finally, the silhouette plot for $k = 4$ is shown in Figure 17.3c. Here C_3 is the well separated cluster, corresponding to C_2 above, and the remaining clusters are essentially subclusters of C_1 for $k = 2$ (Figure 17.3a). Cluster C_1 also has two points with negative s_i values, indicating that they are probably misclustered.

Because $k = 2$ yields the highest silhouette coefficient, and the two clusters are essentially well separated, in the absence of prior knowledge, we would choose $k = 2$ as the best number of clusters for this dataset.

Calinski–Harabasz Index

Given the dataset $\mathbf{D} = \{\mathbf{x}_i\}_{i=1}^n$, the scatter matrix for $\mathbf{D}$ is given as

$$\mathbf{S} = n\boldsymbol{\Sigma} = \sum_{j=1}^{n} \left(\mathbf{x}_j - \boldsymbol{\mu}\right)\left(\mathbf{x}_j - \boldsymbol{\mu}\right)^T$$

where $\boldsymbol{\mu} = \frac{1}{n}\sum_{j=1}^{n} \mathbf{x}_j$ is the mean and $\boldsymbol{\Sigma}$ is the covariance matrix. The scatter matrix can be decomposed into two matrices $\mathbf{S} = \mathbf{S}_W + \mathbf{S}_B$, where $\mathbf{S}_W$ is the within-cluster scatter matrix and $\mathbf{S}_B$ is the between-cluster scatter matrix, given as

$$\mathbf{S}_W = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in C_i} \left(\mathbf{x}_j - \boldsymbol{\mu}_i\right)\left(\mathbf{x}_j - \boldsymbol{\mu}_i\right)^T$$

$$\mathbf{S}_B = \sum_{i=1}^{k} n_i \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}\right)\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}\right)^T$$

where $\boldsymbol{\mu}_i = \frac{1}{n_i}\sum_{\mathbf{x}_j \in C_i} \mathbf{x}_j$ is the mean for cluster C_i.

The Calinski–Harabasz (CH) variance ratio criterion for a given value of k is defined as follows:

$$CH(k) = \frac{tr(\mathbf{S}_B)/(k-1)}{tr(\mathbf{S}_W)/(n-k)} = \frac{n-k}{k-1} \cdot \frac{tr(\mathbf{S}_B)}{tr(\mathbf{S}_W)}$$

where $tr(\mathbf{S}_W)$ and $tr(\mathbf{S}_B)$ are the traces (the sum of the diagonal elements) of the within-cluster and between-cluster scatter matrices.

For a good value of k, we expect the within-cluster scatter to be smaller relative to the between-cluster scatter, which should result in a higher $CH(k)$ value. On the other

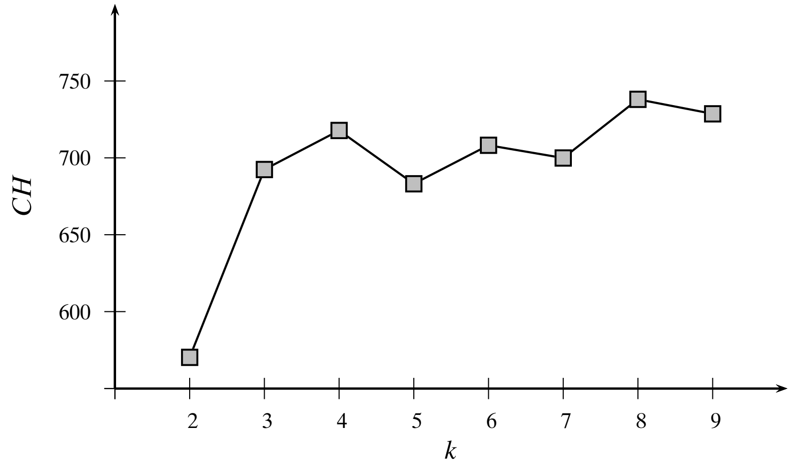


Figure 17.4. Calinski–Harabasz variance ratio criterion.

hand, we do not desire a very large value of k; thus the term $\frac{n-k}{k-1}$ penalizes larger values of k. We could choose a value of k that maximizes $CH(k)$. Alternatively, we can plot the CH values and look for a large increase in the value followed by little or no gain. For instance, we can choose the value $k > 3$ that minimizes the term

$$\Delta(k) = \Big(CH(k+1) - CH(k)\Big) - \Big(CH(k) - CH(k-1)\Big)$$

The intuition is that we want to find the value of k for which $CH(k)$ is much higher than $CH(k-1)$ and there is only a little improvement or a decrease in the $CH(k+1)$ value.

Example 17.8. Figure 17.4 shows the CH ratio for various values of k on the Iris principal components dataset, using the K-means algorithm, with the best results chosen from 200 runs.

For $k = 3$, the within-cluster and between-cluster scatter matrices are given as

$$\mathbf{S}_W = \begin{pmatrix} 39.14 & -13.62 \\ -13.62 & 24.73 \end{pmatrix} \qquad \mathbf{S}_B = \begin{pmatrix} 590.36 & 13.62 \\ 13.62 & 11.36 \end{pmatrix}$$

Thus, we have

$$CH(3) = \frac{(150-3)}{(3-1)} \cdot \frac{(590.36+11.36)}{(39.14+24.73)} = (147/2) \cdot \frac{601.72}{63.87} = 73.5 \cdot 9.42 = 692.4$$

The successive $CH(k)$ and $\Delta(k)$ values are as follows:

k	2	3	4	5	6	7	8	9
$CH(k)$	570.25	692.40	717.79	683.14	708.26	700.17	738.05	728.63
$\Delta(k)$	–	−96.78	−60.03	59.78	−33.22	45.97	−47.30	–

If we choose the first large peak before a decrease we would choose $k = 4$. However, $\Delta(k)$ suggests $k = 3$ as the best (lowest) value, representing the "knee-of-the-curve". One limitation of the $\Delta(k)$ criteria is that values less than $k = 3$ cannot be evaluated, since $\Delta(2)$ depends on $CH(1)$, which is not defined.

Gap Statistic

The gap statistic compares the sum of intracluster weights W_{in} [Eq. (17.23)] for different values of k with their expected values assuming no apparent clustering structure, which forms the null hypothesis.

Let $\mathcal{C}_k$ be the clustering obtained for a specified value of k, using a chosen clustering algorithm. Let $W_{in}^k(\mathbf{D})$ denote the sum of intracluster weights (over all clusters) for $\mathcal{C}_k$ on the input dataset $\mathbf{D}$. We would like to compute the probability of the observed W_{in}^k value under the null hypothesis that the points are randomly placed in the same data space as $\mathbf{D}$. Unfortunately, the sampling distribution of W_{in} is not known. Further, it depends on the number of clusters k, the number of points n, and other characteristics of $\mathbf{D}$.

To obtain an empirical distribution for W_{in}, we resort to Monte Carlo simulations of the sampling process. That is, we generate t random samples comprising n randomly distributed points within the same d-dimensional data space as the input dataset $\mathbf{D}$. That is, for each dimension of $\mathbf{D}$, say X_j, we compute its range $[\min(X_j), \max(X_j)]$ and generate values for the n points (for the jth dimension) uniformly at random within the given range. Let $\mathbf{R}_i \in \mathbb{R}^{n\times d}$, $1 \le i \le t$ denote the ith sample. Let $W_{in}^k(\mathbf{R}_i)$ denote the sum of intracluster weights for a given clustering of $\mathbf{R}_i$ into k clusters. From each sample dataset $\mathbf{R}_i$, we generate clusterings for different values of k using the same algorithm and record the intracluster values $W_{in}^k(\mathbf{R}_i)$. Let $\mu_W(k)$ and $\sigma_W(k)$ denote the mean and standard deviation of these intracluster weights for each value of k, given as

$$\mu_W(k) = \frac{1}{t}\sum_{i=1}^{t} \log W_{in}^k(\mathbf{R}_i)$$

$$\sigma_W(k) = \sqrt{\frac{1}{t}\sum_{i=1}^{t}\Big(\log W_{in}^k(\mathbf{R}_i) - \mu_W(k)\Big)^2}$$

where we use the logarithm of the W_{in} values, as they can be quite large.

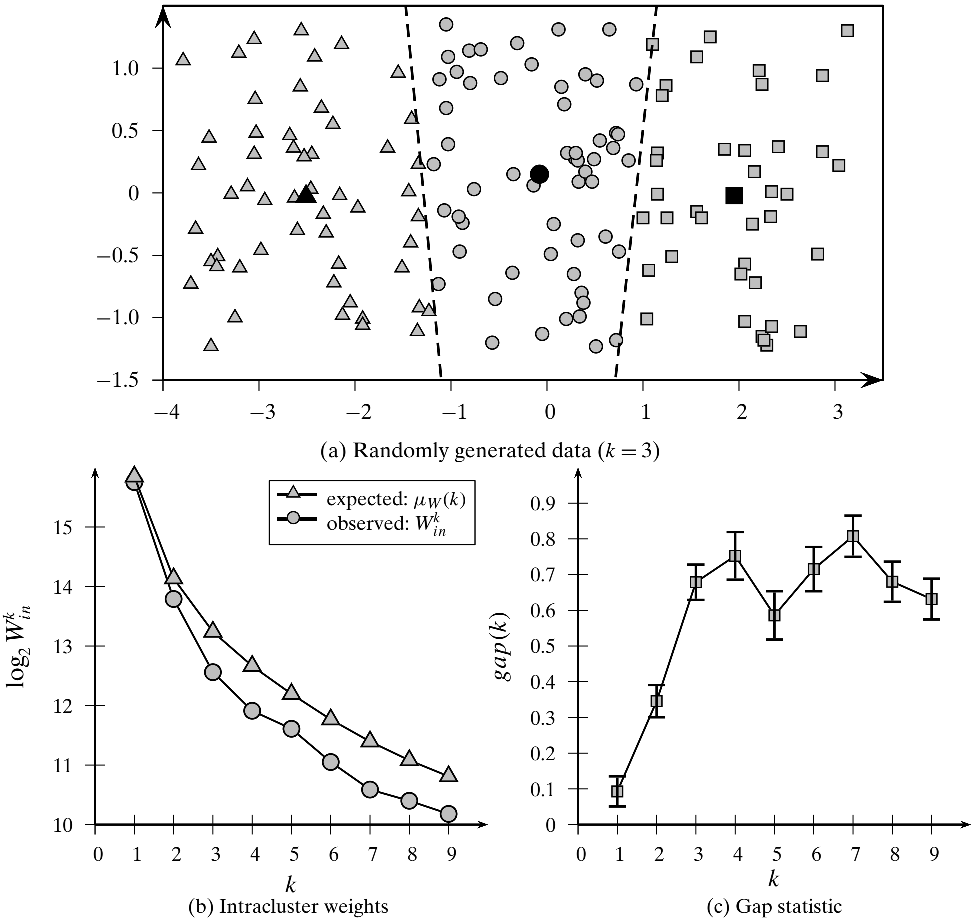
The *gap statistic* for a given k is then defined as

$$gap(k) = \mu_W(k) - \log W_{in}^k(\mathbf{D})$$

It measures the deviation of the observed W_{in}^k value from its expected value under the null hypothesis. We can select the value of k that yields the largest gap statistic because that indicates a clustering structure far away from the uniform distribution of points. A more robust approach is to choose k as follows:

$$k^* = \arg\min_k \Big\{ gap(k) \ge gap(k+1) - \sigma_W(k+1) \Big\}$$

That is, we select the least value of k such that the gap statistic is within one standard deviation of the gap at $k+1$.



(a) Randomly generated data ($k = 3$)

(b) Intracluster weights

(c) Gap statistic

Figure 17.5. Gap statistic. (a) Randomly generated data. (b) Intracluster weights for different k. (c) Gap statistic as a function of k.

Example 17.9. To compute the gap statistic we have to generate t random samples of n points drawn from the same data space as the Iris principal components dataset. A random sample of $n = 150$ points is shown in Figure 17.5a, which does not have any apparent cluster structure. However, when we run K-means on this dataset it will output some clustering, an example of which is also shown, with $k = 3$. From this clustering, we can compute the $\log_2 W_{in}^k(\mathbf{R}_i)$ value; we use base 2 for all logarithms.

For Monte Carlo sampling, we generate $t = 200$ such random datasets, and compute the mean or expected intracluster weight $\mu_W(k)$ under the null hypothesis, for each value of k. Figure 17.5b shows the expected intracluster weights for different values of k. It also shows the observed value of $\log_2 W_{in}^k$ computed from the K-means clustering of the Iris principal components dataset. For the Iris dataset, and each of the uniform random samples, we run K-means 100 times and select the best possible clustering, from which the $W_{in}^k(\mathbf{R}_i)$ values are computed. We can see that the observed $W_{in}^k(\mathbf{D})$ values are smaller than the expected values $\mu_W(k)$.

Table 17.1. Gap statistic values as a function of k

k	$gap(k)$	$\sigma_W(k)$	$gap(k)-\sigma_W(k)$
1	0.093	0.0456	0.047
2	0.346	0.0486	0.297
3	0.679	0.0529	0.626
4	0.753	0.0701	0.682
5	0.586	0.0711	0.515
6	0.715	0.0654	0.650
7	0.808	0.0611	0.746
8	0.680	0.0597	0.620
9	0.632	0.0606	0.571

From these values, we then compute the gap statistic $gap(k)$ for different values of k, which are plotted in Figure 17.5c. Table 17.1 lists the gap statistic and standard deviation values. The optimal value for the number of clusters is $k=4$ because

$$gap(4)=0.753>gap(5)-\sigma_W(5)=0.515$$

However, if we had relaxed the gap test to be within two standard deviations, then the optimal value would have been $k=3$ because

$$gap(3)=0.679>gap(4)-2\sigma_W(4)=0.753-2\cdot 0.0701=0.613$$

Essentially, there is still some subjectivity in selecting the right number of clusters, but the gap statistic plot can help in this task.

17.3.1 Cluster Stability

The main idea behind cluster stability is that the clusterings obtained from several datasets sampled from the same underlying distribution as $\mathbf{D}$ should be similar or "stable." The cluster stability approach can be used to find good parameter values for a given clustering algorithm; we will focus on the task of finding a good value for k, the correct number of clusters.

The joint probability distribution for $\mathbf{D}$ is typically unknown. Therefore, to sample a dataset from the same distribution we can try a variety of methods, including random perturbations, subsampling, or bootstrap resampling. Let us consider the bootstrapping approach; we generate t samples of size n by sampling from $\mathbf{D}$ with replacement, which allows the same point to be chosen possibly multiple times, and thus each sample $\mathbf{D}_i$ will be different. Next, for each sample $\mathbf{D}_i$ we run the same clustering algorithm with different cluster values k ranging from 2 to $k^{\max}$.

Let $\mathcal{C}_k(\mathbf{D}_i)$ denote the clustering obtained from sample $\mathbf{D}_i$, for a given value of k. Next, the method compares the distance between all pairs of clusterings $\mathcal{C}_k(\mathbf{D}_i)$ and $\mathcal{C}_k(\mathbf{D}_j)$ via some distance function. Several of the external cluster evaluation measures can be used as distance measures, by setting, for example, $\mathcal{C}=\mathcal{C}_k(\mathbf{D}_i)$ and $\mathcal{T}=\mathcal{C}_k(\mathbf{D}_j)$, or vice versa. From these values we compute the expected pairwise distance for each value of k. Finally, the value k^* that exhibits the least deviation between the clusterings

ALGORITHM 17.1. Clustering Stability Algorithm for Choosing k

```
ClusteringStability ($A, t, k^{\max}, \mathbf{D}$):
$n \leftarrow |\mathbf{D}|$
// Generate $t$ samples
for $i = 1, 2, \dots, t$ do
    $\mathbf{D}_i \leftarrow$ sample $n$ points from $\mathbf{D}$ with replacement
// Generate clusterings for different values of $k$
for $i = 1, 2, \dots, t$ do
    for $k = 2, 3, \dots, k^{\max}$ do
        $\mathcal{C}_k(\mathbf{D}_i) \leftarrow$ cluster $\mathbf{D}_i$ into $k$ clusters using algorithm $A$
// Compute mean difference between clusterings for each $k$
foreach pair $\mathbf{D}_i, \mathbf{D}_j$ with $j > i$ do
    $\mathbf{D}_{ij} \leftarrow \mathbf{D}_i \cap \mathbf{D}_j$ // create common dataset using Eq. (17.30)
    for $k = 2, 3, \dots, k^{\max}$ do
        $d_{ij}(k) \leftarrow d\big(\mathcal{C}_k(\mathbf{D}_i), \mathcal{C}_k(\mathbf{D}_j), \mathbf{D}_{ij}\big)$ // distance between
            clusterings
for $k = 2, 3, \dots, k^{\max}$ do
    $\mu_d(k) \leftarrow \frac{2}{t(t-1)} \sum_{i=1}^{t} \sum_{j>i} d_{ij}(k)$ // expected pairwise distance
// Choose best $k$
$k^* \leftarrow \arg\min_k \{\mu_d(k)\}$
```

obtained from the resampled datasets is the best choice for k because it exhibits the most stability.

There is, however, one complication when evaluating the distance between a pair of clusterings $\mathcal{C}_k(\mathbf{D}_i)$ and $\mathcal{C}_k(\mathbf{D}_j)$, namely that the underlying datasets $\mathbf{D}_i$ and $\mathbf{D}_j$ are different. That is, the set of points being clustered is different because each sample $\mathbf{D}_i$ is different. Before computing the distance between the two clusterings, we have to restrict the clusterings only to the points common to both $\mathbf{D}_i$ and $\mathbf{D}_j$, denoted as $\mathbf{D}_{ij}$. Because sampling with replacement allows multiple instances of the same point, we also have to account for this when creating $\mathbf{D}_{ij}$. For each point $\mathbf{x}_a$ in the input dataset $\mathbf{D}$, let m_i^a and m_j^a denote the number of occurrences of $\mathbf{x}_a$ in $\mathbf{D}_i$ and $\mathbf{D}_j$, respectively. Define

$$\mathbf{D}_{ij} = \mathbf{D}_i \cap \mathbf{D}_j = \Big\{ m^a \text{ instances of } \mathbf{x}_a \mid \mathbf{x}_a \in \mathbf{D}, m^a = \min\{m_i^a, m_j^a\} \Big\} \tag{17.30}$$

That is, the common dataset $\mathbf{D}_{ij}$ is created by selecting the minimum number of instances of the point $\mathbf{x}_a$ in $\mathbf{D}_i$ or $\mathbf{D}_j$.

Algorithm 17.1 shows the pseudo-code for the clustering stability method for choosing the best k value. It takes as input the clustering algorithm A, the number of samples t, the maximum number of clusters $k^{\max}$, and the input dataset $\mathbf{D}$.

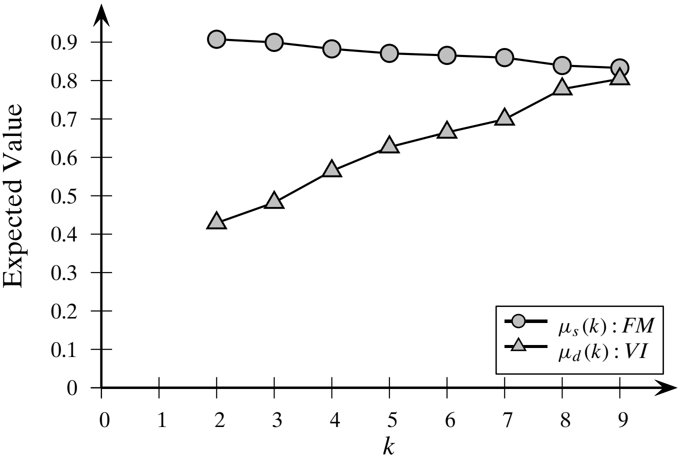


Figure 17.6. Clustering stability: Iris dataset.

It first generates the t bootstrap samples and clusters them using algorithm A. Next, it computes the distance between the clusterings for each pair of datasets $\mathbf{D}_i$ and $\mathbf{D}_j$, for each value of k. Finally, the method computes the expected pairwise distance $\mu_d(k)$ in line 12. We assume that the clustering distance function d is symmetric. If d is not symmetric, then the expected difference should be computed over all ordered pairs, that is, $\mu_d(k) = \frac{1}{t(t-1)} \sum_{i=1}^{r} \sum_{j \neq i} d_{ij}(k)$.

Instead of a distance function d, we can also evaluate clustering stability via a similarity measure, in which case, after computing the average similarity between pairs of clusterings for a given k, we can choose the best value k^* as the one that maximizes the expected similarity $\mu_s(k)$. In general, those external measures that yield lower values for better agreement between $\mathcal{C}_k(\mathbf{D}_i)$ and $\mathcal{C}_k(\mathbf{D}_j)$ can be used as distance functions, whereas those that yield higher values for better agreement can be used as similarity functions. Examples of distance functions include normalized mutual information, variation of information, and conditional entropy (which is asymmetric). Examples of similarity functions include Jaccard, Fowlkes–Mallows, Hubert Γ statistic, and so on.

Example 17.10. We study the clustering stability for the Iris principal components dataset, with $n = 150$, using the K-means algorithm. We use $t = 500$ bootstrap samples. For each dataset $\mathbf{D}_i$, and each value of k, we run K-means with 100 initial starting configurations, and select the best clustering.

For the distance function, we used the variation of information [Eq. (17.5)] between each pair of clusterings. We also used the Fowlkes–Mallows measure [Eq. (17.13)] as an example of a similarity measure. The expected values of the pairwise distance $\mu_d(k)$ for the VI measure, and the pairwise similarity $\mu_s(k)$ for the FM measure are plotted in Figure 17.6. Both the measures indicate that $k = 2$ is the best value, as for the VI measure this leads to the least expected distance between pairs of clusterings, and for the FM measure this choice leads to the most expected similarity between clusterings.

17.3.2 Clustering Tendency

Clustering tendency or clusterability aims to determine whether the dataset $\mathbf{D}$ has any meaningful groups to begin with. This is usually a hard task given the different definitions of what it means to be a cluster, for example, partitional, hierarchical, density-based, graph-based and so on. Even if we fix the cluster type, it is still a hard task to define the appropriate null model (e.g., the one without any clustering structure) for a given dataset $\mathbf{D}$. Furthermore, if we do determine that the data is clusterable, then we are still faced with the question of how many clusters there are. Nevertheless, it is still worthwhile to assess the clusterability of a dataset; we look at some approaches to answer the question whether the data is clusterable or not.

Spatial Histogram

One simple approach is to contrast the d-dimensional spatial histogram of the input dataset $\mathbf{D}$ with the histogram from samples generated randomly in the same data space. Let $X_1, X_2, \ldots, X_d$ denote the d dimensions. Given b, the number of bins for each dimension, we divide each dimension X_j into b equi-width bins, and simply count how many points lie in each of the b^d d-dimensional cells. From this spatial histogram, we can obtain the empirical joint probability mass function (EPMF) for the dataset $\mathbf{D}$, which is an approximation of the unknown joint probability density function. The EPMF is given as

$$f(\mathbf{i}) = P(\mathbf{x}_j \in \text{cell } \mathbf{i}) = \frac{\left|\{\mathbf{x}_j \in \text{cell } \mathbf{i}\}\right|}{n}$$

where $\mathbf{i} = (i_1, i_2, \ldots, i_d)$ denotes a cell index, with i_j denoting the bin index along dimension X_j.

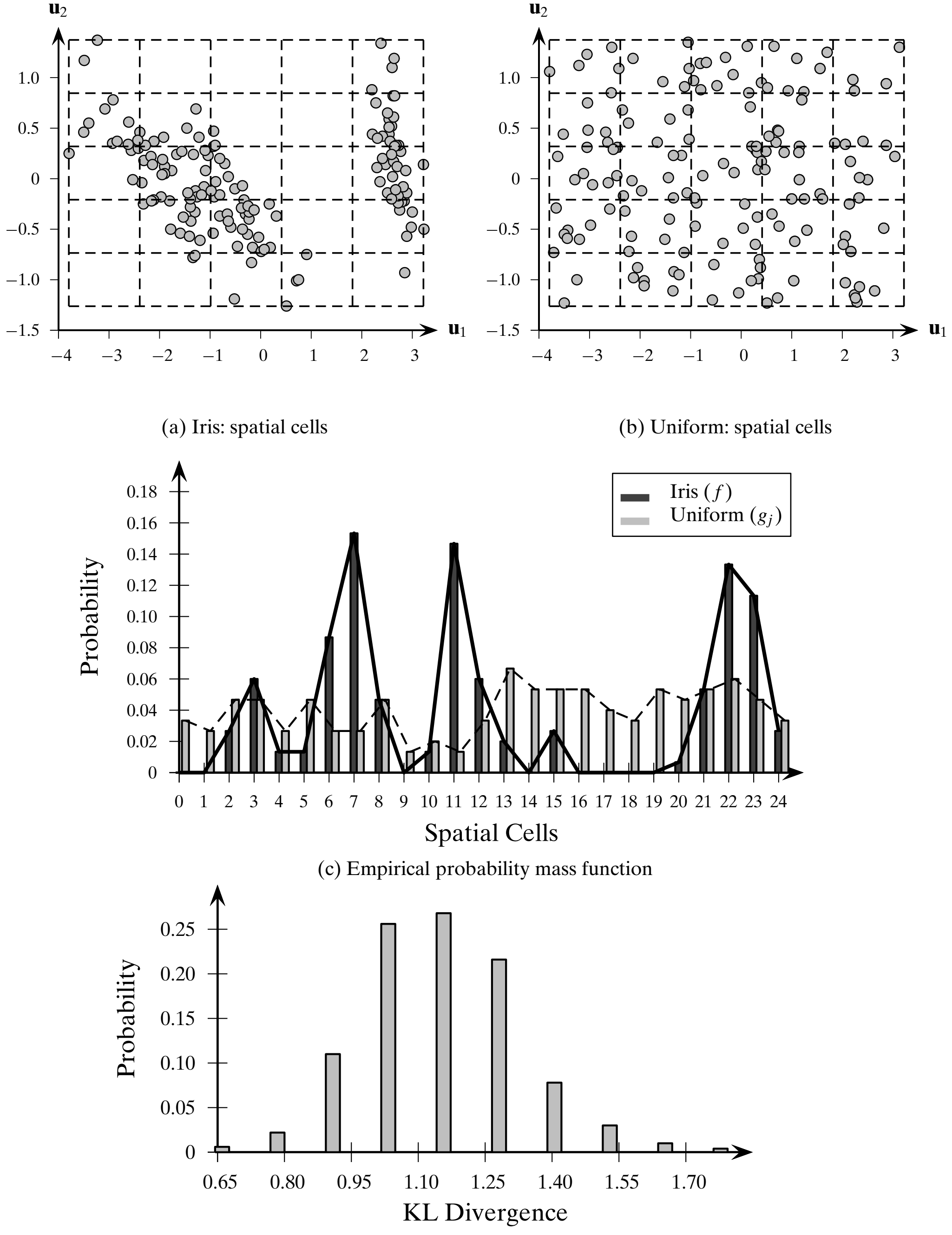
Next, we generate t random samples, each comprising n points within the same d-dimensional space as the input dataset $\mathbf{D}$. That is, for each dimension X_j, we compute its range $[\min(X_j), \max(X_j)]$, and generate values uniformly at random within the given range. Let $\mathbf{R}_j$ denote the jth such random sample. We can then compute the corresponding EPMF $g_j(\mathbf{i})$ for each $\mathbf{R}_j$, $1 \le j \le t$.

Finally, we can compute how much the distribution f differs from g_j (for $j = 1, \ldots, t$), using the Kullback–Leibler (KL) divergence from f to g_j, defined as

$$KL(f|g_j) = \sum_{\mathbf{i}} f(\mathbf{i}) \log\left(\frac{f(\mathbf{i})}{g_j(\mathbf{i})}\right) \tag{17.31}$$

The KL divergence is zero only when f and g_j are the same distributions. Using these divergence values, we can compute how much the dataset $\mathbf{D}$ differs from a random dataset.

The main limitation of this approach is that as dimensionality increases, the number of cells (b^d) increases exponentially, and with a fixed sample size n, most of the cells will be empty, or will have only one point, making it hard to estimate the divergence. The method is also sensitive to the choice of parameter b. Instead of histograms, and the corresponding EPMF, we can also use density estimation methods (see Section 15.2) to determine the joint probability density function (PDF) for the



(a) Iris: spatial cells

(b) Uniform: spatial cells

(c) Empirical probability mass function

(d) KL-divergence distribution

Figure 17.7. Iris dataset: spatial histogram.

dataset **D**, and see how it differs from the PDF for the random datasets. However, the curse of dimensionality also causes problems for density estimation.

Example 17.11. Figure 17.7c shows the empirical joint probability mass function for the Iris principal components dataset that has $n = 150$ points in $d = 2$ dimensions. It also shows the EPMF for one of the datasets generated uniformly at random in the same data space. Both EPMFs were computed using $b = 5$ bins in each dimension, for a total of 25 spatial cells. The spatial grids/cells for the Iris dataset **D**, and the random sample **R**, are shown in Figures 17.7a and 17.7b, respectively. The cells are numbered starting from 0, from bottom to top, and then left to right. Thus, the bottom left cell is 0, top left is 4, bottom right is 19, and top right is 24. These indices are used along the x-axis in the EPMF plot in Figure 17.7c.

We generated $t = 500$ random samples from the null distribution, and computed the KL divergence from f to g_j for each $1 \leq j \leq t$ (using logarithm with base 2). The distribution of the KL values is plotted in Figure 17.7d. The mean KL value was $\mu_{KL} = 1.17$, with a standard deviation of $\sigma_{KL} = 0.18$, indicating that the Iris data is indeed far from the randomly generated data, and thus is clusterable.

Distance Distribution

Instead of trying to estimate the density, another approach to determine clusterability is to compare the pairwise point distances from **D**, with those from the randomly generated samples $\mathbf{R}_i$ from the null distribution. That is, we create the EPMF from the proximity matrix **W** for **D** [Eq. (17.22)] by binning the distances into b bins:

$$f(i) = P(w_{pq} \in \text{ bin } i \mid \mathbf{x}_p, \mathbf{x}_q \in \mathbf{D}, p < q) = \frac{\left|\{w_{pq} \in \text{ bin } i\}\right|}{n(n-1)/2}$$

Likewise, for each of the samples $\mathbf{R}_j$, we can determine the EPMF for the pairwise distances, denoted g_j. Finally, we can compute the KL divergences between f and g_j using Eq. (17.31). The expected divergence indicates the extent to which **D** differs from the null (random) distribution.

Example 17.12. Figure 17.8a shows the distance distribution for the Iris principal components dataset **D** and the random sample $\mathbf{R}_j$ from Figure 17.7b. The distance distribution is obtained by binning the edge weights between all pairs of points using $b = 25$ bins.

We then compute the KL divergence from **D** to each $\mathbf{R}_j$, over $t = 500$ samples. The distribution of the KL divergences (using logarithm with base 2) is shown in Figure 17.8b. The mean divergence is $\mu_{KL} = 0.18$, with standard deviation $\sigma_{KL} = 0.017$. Even though the Iris dataset has a good clustering tendency, the KL divergence is not very large. We conclude that, at least for the Iris dataset, the distance distribution is not as discriminative as the spatial histogram approach for clusterability analysis.

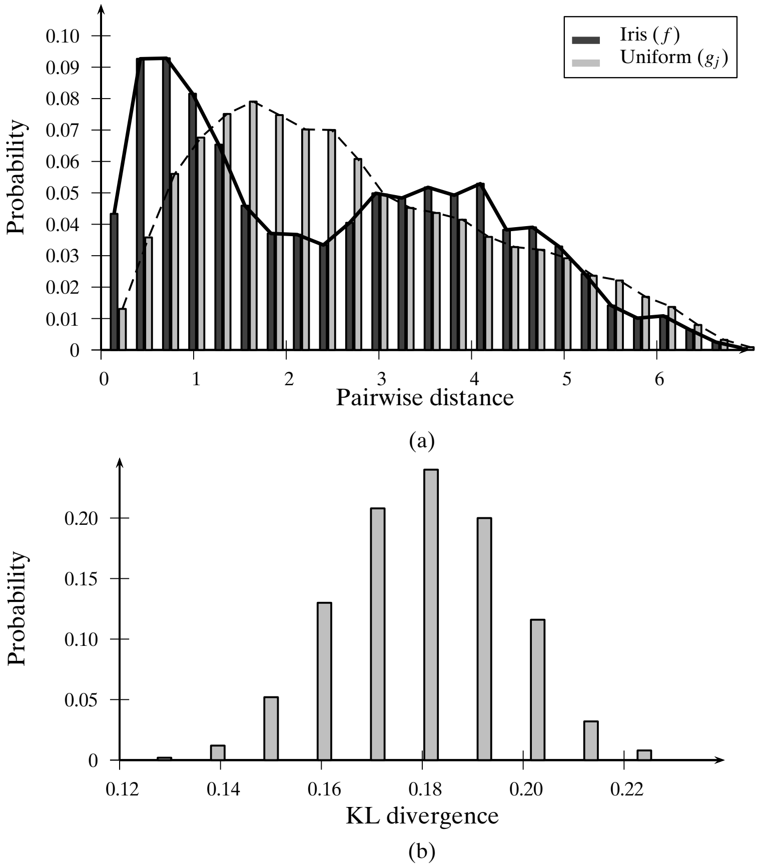


Figure 17.8. Iris dataset: distance distribution.

Hopkins Statistic

The Hopkins statistic is a sparse sampling test for spatial randomness. Given a dataset $\mathbf{D}$ comprising n points, we generate t random subsamples $\mathbf{R}_i$ of m points each, where $m \ll n$. These samples are drawn from the same data space as $\mathbf{D}$, generated uniformly at random along each dimension. Further, we also generate t subsamples of m points directly from $\mathbf{D}$, using sampling without replacement. Let $\mathbf{D}_i$ denote the ith direct subsample. Next, we compute the minimum distance between each point $\mathbf{x}_j \in \mathbf{D}_i$ and points in $\mathbf{D}$

$$\delta_{\min}(\mathbf{x}_j) = \min_{\mathbf{x}_i \in \mathbf{D}, \mathbf{x}_i \neq \mathbf{x}_j} \Big\{ \delta(\mathbf{x}_j, \mathbf{x}_i) \Big\}$$

Likewise, we compute the minimum distance $\delta_{\min}(\mathbf{y}_j)$ between a point $\mathbf{y}_j \in \mathbf{R}_i$ and points in $\mathbf{D}$.

The Hopkins statistic (in d dimensions) for the ith pair of samples $\mathbf{R}_i$ and $\mathbf{D}_i$ is then defined as

$$HS_i = \frac{\sum_{\mathbf{y}_j \in \mathbf{R}_i} \big(\delta_{\min}(\mathbf{y}_j)\big)^d}{\sum_{\mathbf{y}_j \in \mathbf{R}_i} \big(\delta_{\min}(\mathbf{y}_j)\big)^d + \sum_{\mathbf{x}_j \in \mathbf{D}_i} \big(\delta_{\min}(\mathbf{x}_j)\big)^d}$$

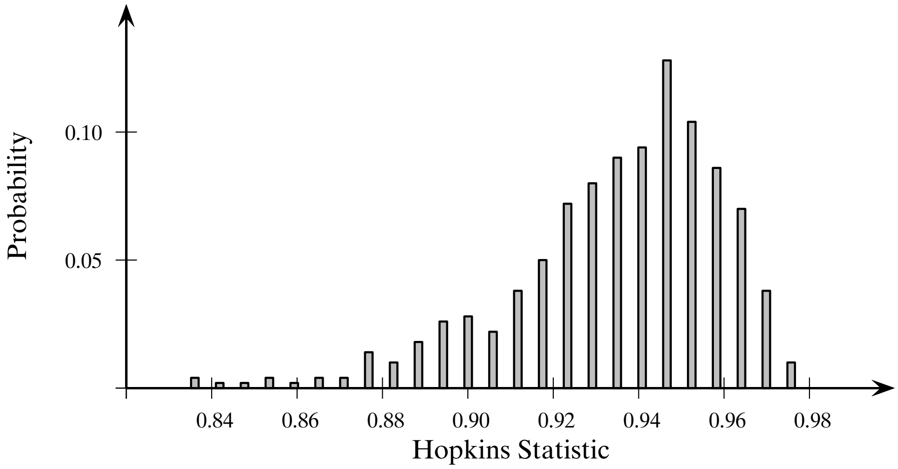


Figure 17.9. Iris dataset: Hopkins statistic distribution.

This statistic compares the nearest-neighbor distribution of randomly generated points to the same distribution for random subsets of points from **D**. If the data is well clustered we expect $\delta_{\min}(\mathbf{x}_j)$ values to be smaller compared to the $\delta_{\min}(\mathbf{y}_j)$ values, and in this case HS_i tends to 1. If both nearest-neighbor distances are similar, then HS_i takes on values close to 0.5, which indicates that the data is essentially random, and there is no apparent clustering. Finally, if $\delta_{\min}(\mathbf{x}_j)$ values are larger compared to $\delta_{\min}(\mathbf{y}_j)$ values, then HS_i tends to 0, and it indicates point repulsion, with no clustering. From the t different values of HS_i we may then compute the mean and variance of the statistic to determine whether **D** is clusterable or not.

Example 17.13. Figure 17.9 plots the distribution of the Hopkins statistic values over $t = 500$ pairs of samples: $\mathbf{R}_j$ generated uniformly at random, and $\mathbf{D}_j$ subsampled from the input dataset **D**. The subsample size was set as $m = 30$, using 20% of the points in **D**, that is, the Iris principal components dataset, which has $n = 150$ points in $d = 2$ dimensions. The mean of the Hopkins statistic is $\mu_{HS} = 0.935$, with a standard deviation of $\sigma_{HS} = 0.025$. Given the high value of the statistic, we conclude that the Iris dataset has a good clustering tendency.

17.4 FURTHER READING

For an excellent introduction to clustering validation see Jain and Dubes (1988); the book describes many of the external, internal, and relative measures discussed in this chapter, including clustering tendency. Other good reviews appear in Halkidi, Batistakis, and Vazirgiannis (2001) and Theodoridis and Koutroumbas (2008). For recent work on formal properties for comparing clusterings via external measures see Amigó et al. (2009) and Meilă (2007). For the silhouette plot see Rousseeuw (1987), and for gap statistic see Tibshirani, Walther, and Hastie (2001). For an overview of cluster stability methods see Luxburg (2009). A recent review of clusterability appears in Ackerman and Ben-David (2009). Overall reviews of clustering methods appear in Xu and Wunsch (2005) and Jain, Murty, and Flynn (1999). See Kriegel, Kröger, and Zimek (2009) for a review of subspace clustering methods.

Ackerman, M. and Ben-David, S. (2009). "Clusterability: A theoretical study." *In Proceedings of 12th International Conference on Artificial Intelligence and Statistics.*

Amigó, E., Gonzalo, J., Artiles, J., and Verdejo, F. (2009). "A comparison of extrinsic clustering evaluation metrics based on formal constraints." *Information Retrieval*, 12 (4): 461–486.

Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2001). "On clustering validation techniques." *Journal of Intelligent Information Systems*, 17 (2–3): 107–145.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Upper Saddle River, NJ: Prentice-Hall.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). "Data clustering: A review." *ACM Computing Surveys*, 31 (3): 264–323.

Kriegel, H.-P., Kröger, P., and Zimek, A. (2009). "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering." *ACM Transactions on Knowledge Discovery from Data*, 3 (1): 1.

Luxburg, U. von (2009). "Clustering stability: An overview." *Foundations and Trends in Machine Learning*, 2 (3): 235–274.

Meilă, M. (2007). "Comparing clusterings – an information based distance." *Journal of Multivariate Analysis*, 98 (5): 873–895.

Rousseeuw, P. J. (1987). "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis." *Journal of Computational and Applied Mathematics*, 20: 53–65.

Theodoridis, S. and Koutroumbas, K. (2008). *Pattern Recognition,* 4th ed. San Diego: Academic Press.

Tibshirani, R., Walther, G., and Hastie, T. (2001). "Estimating the number of clusters in a dataset via the gap statistic." *Journal of the Royal Statistical Society B*, 63: 411–423.

Xu, R. and Wunsch, D. (2005). "Survey of clustering algorithms." *IEEE Transactions on Neural Networks*, 16 (3): 645–678.

17.5 EXERCISES

Q1. Prove that the maximum value of the entropy measure in Eq. (17.2) is $\log k$.

Q2. Show that if $\mathcal{C}$ and $\mathcal{T}$ are independent of each other then $H(\mathcal{T}|\mathcal{C}) = H(\mathcal{T})$, and further that $H(\mathcal{C},\mathcal{T}) = H(\mathcal{C}) + H(\mathcal{T})$.

Q3. Show that $H(\mathcal{T}|\mathcal{C}) = 0$ if and only if $\mathcal{T}$ is completely determined by $\mathcal{C}$.

Q4. Show that $I(\mathcal{C},\mathcal{T}) = H(\mathcal{C}) + H(\mathcal{T}) - H(\mathcal{T},\mathcal{C})$.

Q5. Show that the variation of information is 0 only when $\mathcal{C}$ and $\mathcal{T}$ are identical.

Q6. Prove that the maximum value of the normalized discretized Hubert statistic in Eq. (17.21) is obtained when $FN = FP = 0$, and the minimum value is obtained when $TP = TN = 0$.

Q7. Show that the Fowlkes–Mallows measure can be considered as the correlation between the pairwise indicator matrices for $\mathcal{C}$ and $\mathcal{T}$, respectively. Define $\mathbf{C}(i, j) = 1$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ (with $i \neq j$) are in the same cluster, and 0 otherwise. Define $\mathbf{T}$ similarly for the ground-truth partitions. Define $\langle \mathbf{C}, \mathbf{T} \rangle = \sum_{i,j=1}^{n} \mathbf{C}_{ij}\mathbf{T}_{ij}$. Show that $FM = \frac{\langle \mathbf{C},\mathbf{T} \rangle}{\sqrt{\langle \mathbf{T},\mathbf{T} \rangle \langle \mathbf{C},\mathbf{C} \rangle}}$

Q8. Show that the silhouette coefficient of a point lies in the interval $[-1, +1]$.

Q9. Show that the scatter matrix can be decomposed as $\mathbf{S} = \mathbf{S}_W + \mathbf{S}_B$, where $\mathbf{S}_W$ and $\mathbf{S}_B$ are the within-cluster and between-cluster scatter matrices.

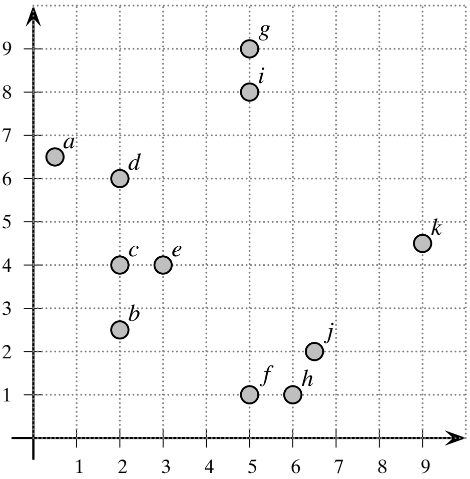


Figure 17.10. Data for Q10 .

Q10. Consider the dataset in Figure 17.10. Compute the silhouette coefficient for the point labeled c.

Q11. Describe how one may apply the gap statistic methodology for determining the parameters of density-based clustering algorithms, such as DBSCAN and DENCLUE (see Chapter 15).