Chapter 18

Clustering Validation

As we have discussed in previous chapters, there exist many different clustering methods, depending on the type of clusters sought, and depending on the data characteristics. A given clustering thus depends on the inherent data characteristics, as well as on the clustering method and its parameters. Cluster validation 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, e.g., the number of clusters, and *clustering tendency* assesses the suitability of applying clustering in the first place, i.e., 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 above 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, e.g., class labels for each point.

Internal: Internal validation measures employ criteria that are derived from the data itself. For instance, we can use intra-cluster and inter-cluster 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.

18.1 External Measures

As the name implies, external measures assume that the correct or ground-truth clustering is known *a priori*. 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. The true cluster labels then play the role of external information that will be used to evaluate the given clustering.

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

18.1.1 Contingency Table Based Measures

The given clustering C, and the ground-truth partitioning T, induce a $r \times k$ contingency table $\mathbf{N}$, defined as follows

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

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 T and 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 $n_{y_i \hat{y}_i}$.

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 C is defined as the weighted sum of the cluster-wise purity values

$$purity = \sum_{i=1}^{r} \frac{n_i}{n} purity_i = \frac{1}{n} \sum_{i=1}^{r} \max_{j=1}^{k} \{n_{ij}\} \tag{18.1}$$

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

Maximum Matching: The maximum matching measure ensures that only one cluster can match with a given partition, unlike purity, where two different clusters may share the same majority partition. We treat the contingency table as a complete weighted bipartite graph $G = (V, E)$, where each partition and cluster is a node, i.e., $V = T \cup C$, and there exists an edge $(C_i, T_j) \in E$, with weight $w(C_i, T_j) = n_{ij}$, for all $C_i \in C$ and $T_j \in T$. A *matching* M in G is a subset of E, such that the edges in M are pair-wise non-adjacent, i.e., they do not have a common vertex. The weight of the matching M is simply the sum of all the edge weights in M, given as $w(M) = \sum_{e \in M} w(e)$. A *maximum weight matching* M^* in G is a matching with maximum weight, $M^* = \arg\max_M \{w(M)\}$, and it can be computed in time $O(|V|^2 \cdot |E|) = O((r+k)^2 rk) == O(r^3 k + 2r^2 k^2 + rk^3)$, which is equivalent to $O(k^4)$ if $r = O(k)$. The maximum matching measure is defined as

$$match = \frac{w(M^*)}{n}$$

F-Measure: F-measure is the harmonic mean of the precision and recall values for each cluster. Given cluster C_i, let j_i denote the partition that contains the maximum number of points from C_i, i.e., $j_i = \max_{j=1}^{k}\{n_{ij}\}$. The *precision* of a cluster C_i is the same as its purity

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

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

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

It measures the fraction of point in partition T_{j_i} shared in common with cluster C_i.

The F-measure for cluster C_i is then given as

$$F_i = \frac{2}{\frac{1}{prec_i} + \frac{1}{recall_i}} = \frac{2 \cdot prec_i \cdot recall_i}{prec_i + recall_i} = \frac{2\, n_{ij_i}}{n_i + m_{j_i}} \tag{18.2}$$

The F-measure for the clustering C is the mean of cluster-wise F-measure values

$$F = \frac{1}{r}\sum_{i=1}^{r} F_i \tag{18.3}$$

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 one.

Conditional Entropy: The entropy of a clustering C is defined as

$$H(C) = -\sum_{i=1}^{r} p_{C_i} \log_2 p_{C_i}$$

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

$$H(T) = -\sum_{j=1}^{k} p_{T_j} \log_2 p_{T_j}$$

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

The cluster-specific entropy of T, i.e., the conditional entropy of T with respect to cluster C_i, measures the amount of uncertainty in the cluster labels with respect to the ground-truth partitions, and is defined as

$$H(T|C_i) = -\sum_{j=1}^{k} \left(\frac{n_{ij}}{n_i}\right) \log_2 \left(\frac{n_{ij}}{n_i}\right)$$

The conditional entropy of T given cluster C is then defined as the weighted sum

$$\begin{aligned} H(T|C) &= \sum_{i=1}^{r} \frac{n_i}{n} H(T|C_i) = -\sum_{i=1}^{r}\sum_{j=1}^{k} \frac{n_{ij}}{n} \log_2 \left(\frac{n_{ij}}{n_i}\right) \\ &= -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij} \log_2 \left(\frac{p_{ij}}{p_{C_i}}\right) \end{aligned} \tag{18.4}$$

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 more the conditional entropy. For a perfect clustering, the conditional entropy value is zero, whereas the worst possible conditional entropy value is $\log_2 k$. Furthermore,

expanding (18.4), we can see that

$$H(T|C) = -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij}\left(\log_2 p_{ij} - \log_2 p_{C_i}\right) \tag{18.5}$$

$$= -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij}\log_2 p_{ij} + \sum_{i=1}^{r}\left(\log_2 p_{C_i}\sum_{j=1}^{k} p_{ij}\right) \tag{18.6}$$

$$= -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij}\log_2 p_{ij} + \sum_{i=1}^{r} p_{C_i}\log_2 p_{C_i} \tag{18.7}$$

$$= H(C,T) - H(C) \tag{18.8}$$

where $H(C,T) = -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij}\log_2 p_{ij}$ is the joint entropy of C and T. The conditional entropy $H(T|C)$ thus measures remaining entropy of T given the clustering C. In particular, if $H(T|C) = 0$, it implies that T is completely determined by C, corresponding to the ideal clustering. On the other hand, if C and T are independent of each other, then $H(T|C) = H(T)$, which means that C provides no information about T.

Normalized Mutual Information: The *mutual information* tries to quantify the amount of shared information between the clustering C and partitioning T, defined as

$$I(C,T) = \sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij}\log_2\left(\frac{p_{ij}}{p_{C_i}\cdot p_{T_j}}\right) \tag{18.9}$$

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

Expanding (18.9) we observe that $I(C,T) = H(C) + H(T) - H(C,T)$. Using (18.8), we obtain the two equivalent expressions

$$I(C,T) = H(C) - H(C|T)$$
$$I(C,T) = H(T) - H(T|C)$$

Finally, since $H(C|T) \geq 0$ and $H(T|C) \geq 0$, we have the inequalities $I(C,T) \leq H(C)$ and $I(C,T) \leq H(T)$. We can obtain a normalized version of mutual information by considering the ratios $I(C,T)/H(C)$ and $I(C,T)/H(T)$. The *normalized mutual information* is defined as the geometric mean of these two ratios

$$NMI(C,T) = \sqrt{\frac{I(C,T)}{H(C)}\cdot\frac{I(C,T)}{H(T)}} = \frac{I(C,T)}{\sqrt{H(C)\cdot H(T)}} \tag{18.10}$$

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

Variation of Information: This criteria is based on the mutual information between the clustering C and the ground-truth partitioning T, and their entropies, given as

$$\begin{aligned} VI(C,T) &= (H(T) - I(C,T)) + (H(C) - I(C,T)) \\ &= H(T) + H(C) - 2I(C,T) \end{aligned} \tag{18.11}$$

The VI value is zero only when C and T are identical. Thus, the lower the VI value the better the clustering C.

Using the equivalence $I(C,T) = H(T) - H(T|C) = H(C) - H(C|T)$, we can also express (18.11) as

$$VI(C,T) = H(T|C) + H(C|T)$$

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

$$VI(C,T) = 2H(T,C) - H(T) - H(C)$$

Example 18.1: Figure 18.1 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 18.1a is a better clustering than that in Figure 18.1b. We will now examine how the different contingency table based measures can be used to evaluate these two clusterings.

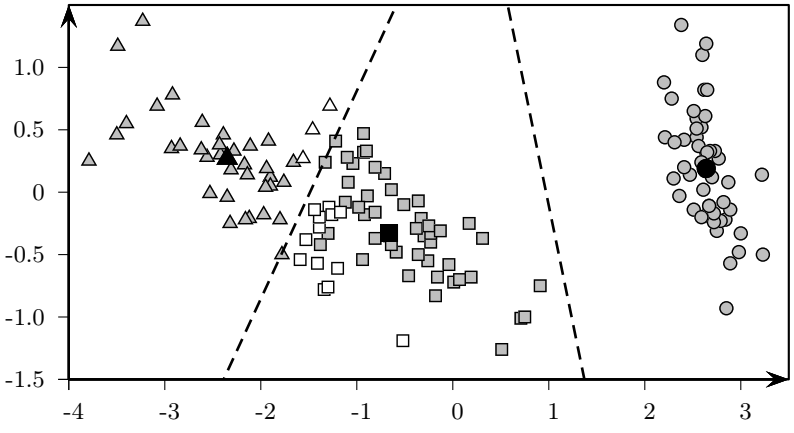
For the clustering in Figure 18.1a, we have the following contingency table

	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	$n = 100$

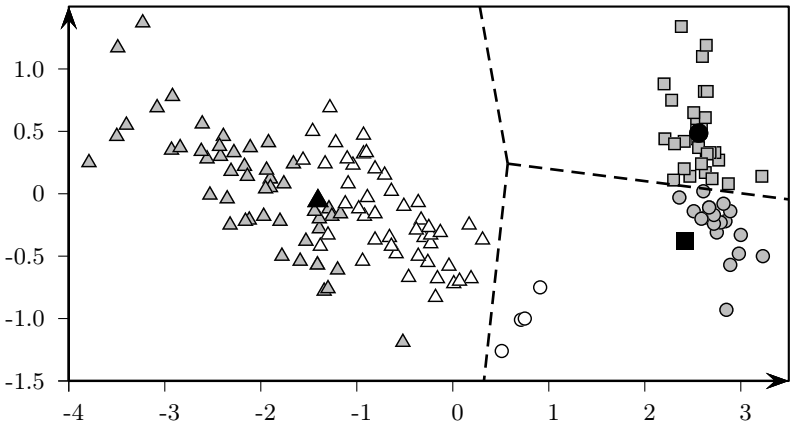
To compute purity, we first compute for each cluster, the partition with the maximum overlap. We have the correspondence (C_1, T_2), (C_2, T_1), and (C_3, T_3). Thus, purity is given as

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

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



(a) K-Means: Good



(b) K-Means: Bad

Figure 18.1: K-means: Iris Principal Components Dataset

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

$$prec_1 = \frac{47}{61} = 0.77$$

and its recall is given as

$$recall_1 = \frac{47}{50} = 0.94$$

We can then obtain the value for the F-measure for C_1 as follows

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

We can also directly compute F_1 using (24.5)

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

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

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

Consider the conditional entropy for cluster C_1, given as

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

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

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

To compute the normalized mutual information, note that

$$\begin{aligned} H(T) &= -3\left(\frac{50}{150}\log_2\left(\frac{50}{150}\right)\right) = 1.585 \\ H(C) &= -\left(\frac{61}{150}\log_2\left(\frac{61}{150}\right) + \frac{50}{150}\log_2\left(\frac{50}{150}\right) + \frac{39}{150}\log_2\left(\frac{39}{150}\right)\right) \\ &= 0.528 + 0.528 + 0.505 = 1.561 \\ I(C,T) &= \frac{47}{150}\log_2\left(\frac{47\cdot 150}{61\cdot 50}\right) + \frac{14}{150}\log_2\left(\frac{14\cdot 150}{61\cdot 50}\right) + \frac{50}{150}\log_2\left(\frac{50\cdot 150}{50\cdot 50}\right) \\ &\quad + \frac{3}{150}\left(\log_2\frac{3\cdot 150}{39\cdot 50}\right) + \frac{36}{150}\log_2\left(\frac{36\cdot 150}{39\cdot 50}\right) \\ &= 0.379 - 0.05 + 0.528 - 0.042 + 0.353 = 1.167 \end{aligned}$$

Thus, the NMI and VI values are

$$\begin{aligned} NMI(C,T) &= \frac{1.167}{\sqrt{1.585\times 1.561}} = 0.742 \\ VI(C,T) &= 1.585 + 1.561 - 2\cdot 1.167 = 0.812 \end{aligned}$$

For the clustering in Figure 18.1b, we have the following contingency table

	iris-setosa T_1	**iris-versicolor** T_2	**iris-virginica** 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, namely (C_1, T_1), (C_2, T_2), and (C_3, T_3), with

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

The table below compares the different contingency based measures on the two clusterings in Figure 18.1: (a) Good, and (b) Bad.

	$purity$	$match$	F	$H(T\|C)$	NMI	VI
$(a) Good$	0.887	0.887	0.885	0.418	0.742	0.812
$(b) Bad$	0.667	0.560	0.658	0.743	0.587	1.20

As expected, the good clustering in Figure 18.1a has higher scores for the purity, maximum matching, F-measure, and normalized mutual information, and it has lower scores for conditional entropy and variation of information.

18.1.2 Pair-wise Measures

Given clustering C and ground-truth partitioning T, the pair-wise 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. 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, i.e., $\hat{y}_i = \hat{y}_j$, we call it a *positive* event, and if they do not belong to the same cluster, i.e., $\hat{y}_i \neq \hat{y}_j$, we call it 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 T, and they are also in the same cluster in C. This is a true positive pair, since 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

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

- *False Negatives:* $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same partition in T, but they do not belong to the same cluster in 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

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

- *False Positives:* $\mathbf{x}_i$ and $\mathbf{x}_j$ do not belong to the same partition in T, but they do belong to the same cluster in C. This pair is a false positive, since the positive event, $\hat{y}_i = \hat{y}_j$, is actually false, i.e., 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

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

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

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

Since there are $N = \binom{n}{2} = \frac{n(n-1)}{2}$ point pairs, we have the following identity

$$N = TP + FN + FP + TN \tag{18.12}$$

A naive computation of the above 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

$$\begin{aligned} TP &= \sum_{i=1}^{r}\sum_{j=1}^{k}\binom{n_{ij}}{2} = \sum_{i=1}^{r}\sum_{j=1}^{k}\frac{n_{ij}(n_{ij}-1)}{2} = \frac{1}{2}\left(\sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^2 - \sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}\right) \\ &= \frac{1}{2}\left(\sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^2 - n\right) \end{aligned} \tag{18.13}$$

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

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. The pairs that

remain are incorrectly labeled as negative pairs, i.e., they are false negatives

$$FN = \sum_{j=1}^{k} \binom{m_j}{2} - TP = \frac{1}{2}\left(\sum_{j=1}^{k} m_j^2 - \sum_{j=1}^{k} m_j - \sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^2 + n\right)$$
$$= \frac{1}{2}\left(\sum_{j=1}^{k} m_j^2 - \sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^2\right) \tag{18.14}$$

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

$$FP = \sum_{i=1}^{r} \binom{n_i}{2} - TP = \frac{1}{2}\left(\sum_{i=1}^{r} n_i^2 - \sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^2\right) \tag{18.15}$$

Finally, the number of true negatives can be obtained via (18.12) as follows

$$TN = N - (TP + FN + FP) = \frac{1}{2}\left(n^2 - \sum_{i=1}^{r} n_i^2 - \sum_{j=1}^{k} m_j^2 + \sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^2\right) \tag{18.16}$$

Each of the four values can be computed in $O(rk)$ time. Since 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.

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

$$Jaccard = \frac{TP}{TP + FN + FP} \tag{18.17}$$

For a perfect clustering C (i.e., total agreement with the partitioning T), the Jaccard Coefficient has value one, since in that case there are no false positives or false negatives. Jaccard Coefficient is asymmetric in terms of the true positives and negatives, since it ignores the true negatives.

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

$$Rand = \frac{TP + TN}{N} \tag{18.18}$$

In other words, the Rand statistic measures the fraction of point pairs where C and T agree that they belong together or do not belong together. The highest value of the statistic is one, for the clustering that is perfect. Rand statistic is a symmetric measure.

Fowlkes-Mallows Measure: Define the overall precision and recall values for a clustering C, as follows

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

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

The Fowlkes-Mallows measure is then defined as the geometric mean of the overall precision and recall

$$FM = \sqrt{prec \cdot recall} = \frac{TP}{\sqrt{(TP + FN)(TP + FP)}} \tag{18.19}$$

The FM measure is also asymmetric in terms of the true positives and negatives, since it ignores the true negatives. It's highest value is also one, achieved when there are no false positives or negatives.

Example 18.2: Let us continue with Example 18.1. Consider again the contingency table for the clustering in Figure 18.1a

	iris-setosa T_1	**iris-versicolor** T_2	**iris-virginica** T_3
C_1	0	47	14
C_2	50	0	0
C_3	0	3	36

Using (18.13), 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 (18.14), (18.15), and (18.16), we obtain $FN = 645$, $FP = 766$ and $TN = 6734$. Note that there are a total of $N = \binom{150}{2} = 11175$ point pairs.

We can now compute the different pair-wise measures for clustering evaluation. The Jaccard coefficient (18.17), Rand statistic (18.18), and Fowlkes-Mallows measure (18.19), 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 18.1b from Example 18.1, we obtain $TP = 2891$, $FN = 784$, $FP = 2380$ and $TN = 5120$. The table below compares the different contingency based measures on the two clusterings in Figure 18.1: (a) Good, and (b) Bad.

	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 18.1a has higher scores for all the three measures.

18.1.3 Correlation Measures

Let $\mathbf{X}$ and $\mathbf{Y}$ be two $n \times n$ matrices (not necessarily symmetric), and let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{n^2}$ denote the vectors obtained by linearizing (e.g., by concatenating all the rows and taking the transpose of) $\mathbf{X}$ and $\mathbf{Y}$, respectively. Let μ_X denote the element-wise mean of $\mathbf{X}$, given as

$$\mu_X = \frac{1}{n^2} \sum_{i,j=1}^{n} \mathbf{X}(i,j) = \frac{1}{n^2} \mathbf{x}^T \mathbf{x}$$

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

$$\mathbf{z}_x = \mathbf{x} - \mathbf{1}_{n^2} \cdot \mu_X$$

where $\mathbf{1}_{n^2} \in \mathbb{R}^{n^2}$ is the vector of all ones. Likewise, let μ_Y be the mean of $\mathbf{Y}$, and $\mathbf{z}_y$ the centered $\mathbf{y}$ vector.

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

$$\Gamma = \frac{1}{n^2} \sum_{i,j=1}^{n} \mathbf{X}(i,j) \cdot \mathbf{Y}(i,j) = \frac{1}{n^2} \mathbf{x}^T \mathbf{y} \tag{18.20}$$

It can be seen as the averaged dot product of $\mathbf{x}$ and $\mathbf{y}$.

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

$$\Gamma_n = \frac{\sum_{i,j=1}^{n} \mathbf{X}(i,j) \cdot \mathbf{Y}(i,j)}{\sqrt{\sum_{i,j=1}^{n} (\mathbf{X}(i,j) - \mu_X)^2 \quad \sum_{i,j=1}^{n} (\mathbf{Y}[i] - \mu_Y)^2}} = \frac{\sigma_{XY}}{\sqrt{\sigma_X^2 \sigma_Y^2}} \tag{18.21}$$

where σ_X^2 and σ_Y^2 are the variances, and σ_{XY} the covariance, for the vectors $\mathbf{x}$ and $\mathbf{y}$ (or the element-wise matrices $\mathbf{X}$ and $\mathbf{Y}$), defined as

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

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

$$\sigma_{XY} = \frac{1}{n^2}\sum_{i,j=1}^{n}(\mathbf{X}(i,j)-\mu_X)(\mathbf{Y}(i,j)-\mu_Y) = \frac{1}{n^2}\mathbf{z}_x^T\mathbf{z}_y$$

Thus the normalized Hubert's statistic can be rewritten as

$$\Gamma_n = \frac{\mathbf{z}_x^T\mathbf{z}_y}{\|\mathbf{z}_x\|\cdot\|\mathbf{z}_y\|} = \cos\theta \tag{18.22}$$

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 $+1$.

Note that when $\mathbf{X}$ and $\mathbf{Y}$ are symmetric matrices, only their upper triangular elements are considered when computing Γ and Γ_n. The Hubert's statistics, Γ and Γ_n, can be used as external evaluation measures, with appropriately defined matrices $\mathbf{X}$ and $\mathbf{Y}$, as described next.

Discretized Hubert's Statistic: Let $\mathbf{T}$ and $\mathbf{C}$ be the $n \times n$ matrices defined as

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

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's Γ (Gamma) statistic is computed via (18.20), by setting $\mathbf{x} = \mathbf{t}$ and $\mathbf{y} = \mathbf{c}$,

$$\Gamma = \frac{1}{N}\mathbf{t}^T\mathbf{c} = \frac{TP}{N} \tag{18.23}$$

Since the i-th element of $\mathbf{t}$ is one only when the i-th pair of points belong to the same partition, and likewise, the i-th element of $\mathbf{c}$ is one only when the i-th pair of points also belong 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 T and C, the higher the Γ value.

Normalized Discretized Hubert's Statistic: The normalized version of the Discretized Hubert's statistic is simply the correlation between $\mathbf{t}$ and $\mathbf{c}$ (18.22)

$$\Gamma_n = \frac{\mathbf{z}_t^T \mathbf{z}_c}{\|\mathbf{z}_t\| \cdot \|\mathbf{z}_c\|} = \cos\theta \tag{18.24}$$

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

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

Substituting these into the numerator in (18.24), we get

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

where $\mathbf{1} \in \mathbb{R}^N$ is the vector of all ones. 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

$$\begin{aligned}
\mathbf{z}_t^T\mathbf{z}_t &= \mathbf{t}^T\mathbf{t} - N\mu_T^2 = N\mu_T - N\mu_T^2 = N\mu_T(1-\mu_T) \\
\mathbf{z}_c^T\mathbf{z}_c &= \mathbf{c}^T\mathbf{c} - N\mu_C^2 = N\mu_C - N\mu_C^2 = N\mu_C(1-\mu_C)
\end{aligned}$$

Thus, the normalized, discretized Hubert's statistic is given as

$$\Gamma_n = \frac{\frac{TP}{N} - \mu_T\mu_C}{\sqrt{\mu_T\mu_C(1-\mu_T)(1-\mu_C)}} \tag{18.25}$$

since $\mu_T = \frac{TP+FN}{N}$ 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, i.e., when $FN = FP = 0$. The minimum value of $\Gamma_n = -1$ is when there are no true positives and negatives, i.e., when $TP = TN = 0$.

Example 18.3: Continuing Example 18.2, for the good clustering in Figure 18.1a, we have

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

From these values, we have

$$\mu_T = \frac{TP + FN}{N} = \frac{3675}{11175} = 0.33$$
$$\mu_C = \frac{TP + FP}{N} = \frac{3796}{11175} = 0.34$$

Using (18.23) and (18.25), we have

$$\Gamma = \frac{3030}{11175} = 0.271$$
$$\Gamma_n = \frac{0.27 - 0.33 \cdot 0.34}{\sqrt{0.33 \cdot 0.34 \cdot (1 - 0.33) \cdot (1 - 0.34)}} = \frac{0.159}{0.222} = 0.717$$

Likewise, for the bad clustering in Figure 18.1b, we have

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

and the values for the discretized Hubert's statistics are given as

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

We can observe that the normalized Hubert's statistic is more discerning than the unnormalized version, i.e., 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 different.

18.2 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 intra-cluster similarity or compactness, contrasted with notions of inter-cluster separation. Most of the internal measures are based on the $n \times n$ *distance matrix*, also called the *proximity matrix*, of all pair-wise distances among the n points

$$\mathbf{W} = \Big\{\delta(\mathbf{x}_i, \mathbf{x}_j)\Big\}_{i,j=1}^{n} \tag{18.26}$$

where

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

is the Euclidean distance between $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{D}$. Since $\mathbf{W}$ 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, i.e., with nodes $V = \{\mathbf{x}_i \in \mathbf{D}\}$, edges $E = \{(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i, \mathbf{x}_j \in \mathbf{D}\}$, and edge weights $w_{ij} = \mathbf{W}(\mathbf{x}_i, \mathbf{x}_j)$ 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 17.

Assume that we are given a clustering $C = \{C_1 \dots C_k\}$ comprising k clusters, with cluster C_i containing $n_i = |C_i|$ points. Let $\hat{y}_i \in \{1, 2, \cdots, k\}$ denote the cluster label for point $\mathbf{x}_i$. The clustering C can be considered as a k-way cut in G, since $C_i \neq \emptyset$ 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, i.e., $\overline{S} = V - S$.

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

$$W_{in} = \frac{1}{2} \sum_{i=1}^{k} W(C_i, C_i) \tag{18.27}$$

We divide by 2, since 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 inter-cluster weights is given as

$$W_{out} = \frac{1}{2} \sum_{i=1}^{k} W(C_i, \overline{C_i}) = \sum_{i=1}^{k} \sum_{j>i} W(C_i, C_j) \tag{18.28}$$

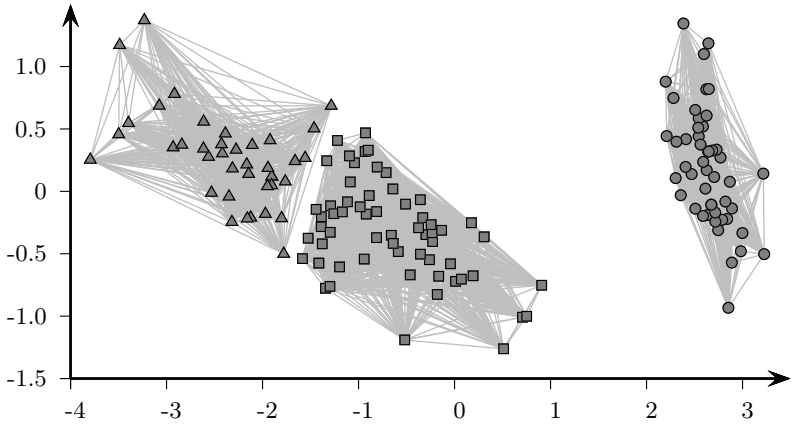
Here we divide by 2, since each edge is counted twice in the summation across clusters. The number of distinct intra-cluster edges, denoted N_{in}, and inter-cluster edges, denoted N_{out}, are given as

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

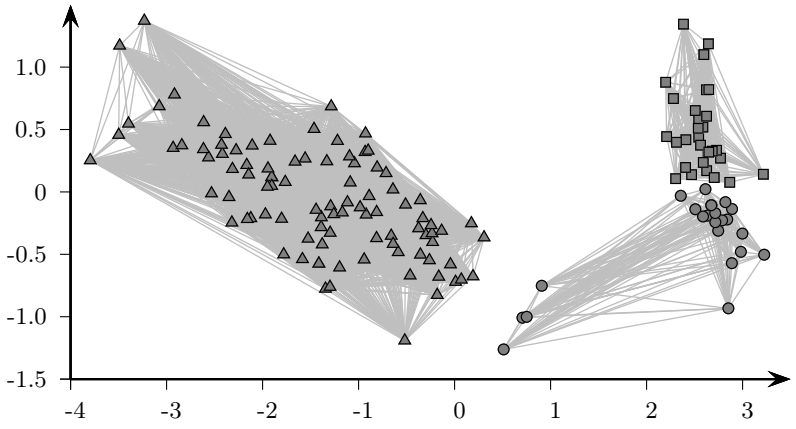
$$N_{out} = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} n_i \cdot n_j = \frac{1}{2} \sum_{i=1}^{k} \sum_{\substack{j=1 \\ j \neq i}}^{k} n_i \cdot n_j$$

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

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



(a) K-Means: Good



(b) K-Means: Bad

Figure 18.2: Clusterings as Graphs: Iris

Example 18.4: Figure 18.2 shows the graphs corresponding to the two K-means clusterings shown in Figure 18.1. 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 intra-cluster edges are shown (with inter-cluster edges omitted) to avoid clutter. Since internal measures do not have access to a ground truth labeling, the goodness of a clustering is measured based on intra-cluster and inter-cluster statistics.

BetaCV Measure: The BetaCV measure is the ratio of the mean intra-cluster distance to the mean inter-cluster 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, since it indicates that intra-cluster distances are on average smaller than inter-cluster 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 intra-cluster edges, or point pairs. Let $W_{\max}(N_{in})$ be the sum of the largest N_{in} distances in $\mathbf{W}$.

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 intra-cluster distances (18.27). C-index lies in the range $[0, 1]$. The smaller the C-index the better the clustering, since it indicates more compact clusters, with relatively smaller distances within clusters rather than between clusters.

Normalized Cut Measure: The normalized cut objective (17.24) 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, i.e., the total weights on edges with at least one end in the cluster. However, since we are using the proximity or distance matrix $\mathbf{W}$, instead of the affinity or similarity matrix $\mathbf{A}$, the 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

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

We can see that NC is maximized when the ratios $\frac{W(C_i, C_i)}{W(C_i, \overline{C_i})}$ (across the k clusters) are as small as possible, which happens when the intra-cluster distances are much smaller compared to inter-cluster distances. The maximum possible value of NC is k.

Modularity: The modularity objective for graph clustering (17.33) 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 (18.27) and (18.28). 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 intra-cluster 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, to the maximum distance between a point pair from the same cluster. More formally, we have

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

where $W_{out}^{\min}$ is the minimum inter-cluster 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 intra-cluster distance

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

The larger the Dunn index the better the clustering, since it means even the closest distance between points in different clusters is much larger than the furthest distance between points in the same cluster.

Davies-Bouldin Index: Let μ_i denote the cluster mean, given as

$$\mu_i = \frac{1}{n_i} \sum_{\mathbf{x}_j \in C_i} \mathbf{x}_j \tag{18.29}$$

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

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

where $var(C_i)$ is the total variance (1.9) of cluster C_i.

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

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

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, since 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

$$s_i = \frac{\mu_{out}^{\min}(\mathbf{x}_i) - \mu_{in}(\mathbf{x}_i)}{\max\left\{\mu_{out}^{\min}(\mathbf{x}_i), \mu_{in}(\mathbf{x}_i)\right\}} \tag{18.30}$$

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

$$\mu_{out}^{\min}(\mathbf{x}_i) = \min_{\substack{j=1 \\ j \neq \hat{y}_i}}^{k} \left\{ \frac{\sum_{\mathbf{y} \in C_j} \delta(\mathbf{x}_i, \mathbf{y})}{n_j} \right\}$$

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

$$SC = \frac{1}{n}\sum_{i=1}^{n} s_i \tag{18.31}$$

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

Hubert's Statistic: The Hubert's Γ statistic (18.20), and its normalized version Γ_n (18.22), 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

$$\mathbf{Y} = \left\{\delta(\mu_{C_{\hat{y}_i}}, \mu_{C_{\hat{y}_j}})\right\}_{i,j=1}^{n} \tag{18.32}$$

where μ_{C_i} denotes the mean for cluster C_i (18.29). Since both $\mathbf{W}$ and $\mathbf{Y}$ are symmetric, both Γ and Γ_n are computed over their upper triangular elements.

Example 18.5: Consider the two clusterings for the Iris principal components dataset shown in Figure 18.1, along with their corresponding graph representations in Figure 18.2. Let us evaluate these two clusterings using internal measures.

The good clustering shown in Figure 18.1a and Figure 18.2a has clusters with the following sizes

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

Thus, the number of intra-cluster and inter-cluster 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 inter-cluster weight matrix

$$\left(\begin{array}{c|ccc} W & C_1 & C_2 & C_3 \\ \hline C_1 & 3265.69 & 10402.30 & 4418.62 \\ C_2 & 10402.30 & 1523.10 & 9792.45 \\ C_3 & 4418.62 & 9792.45 & 1252.36 \end{array}\right) \tag{18.33}$$

Thus, the sum of all the intra-cluster and inter-cluster 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 inter-cluster weight matrix (18.33)

$$\begin{aligned}
W(C_1, \overline{C_1}) &= 10402.30 + 4418.62 = 14820.91 \\
W(C_2, \overline{C_2}) &= 10402.30 + 9792.45 = 20194.75 \\
W(C_3, \overline{C_3}) &= 4418.62 + 9792.45 = 14211.07 \\
W(C_1, V) &= 3265.69 + W(C_1, \overline{C_1}) = 18086.61 \\
W(C_2, V) &= 1523.10 + W(C_2, \overline{C_2}) = 21717.85 \\
W(C_3, V) &= 1252.36 + W(C_3, \overline{C_3}) = 15463.43 \\
W(V, V) &= W(C_1, V) + W(C_2, V) + W(C_3, V) = 55267.89
\end{aligned}$$

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 inter-cluster distances

$$\begin{pmatrix} W^{\min} & C_1 & C_2 & C_3 \\ \hline C_1 & 0 & 1.62 & 0.198 \\ C_2 & 1.62 & 0 & 3.49 \\ C_3 & 0.198 & 3.49 & 0 \end{pmatrix} \quad \begin{pmatrix} W^{\max} & C_1 & C_2 & C_3 \\ \hline C_1 & 2.50 & 4.85 & 4.81 \\ C_2 & 4.85 & 2.33 & 7.06 \\ C_3 & 4.81 & 7.06 & 2.55 \end{pmatrix}$$

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

$$\mu_1 = \begin{pmatrix} -0.664 \\ -0.33 \end{pmatrix} \qquad \mu_2 = \begin{pmatrix} 2.64 \\ 0.19 \end{pmatrix} \qquad \mu_3 = \begin{pmatrix} -2.35 \\ 0.27 \end{pmatrix}$$
$$s_{\mu_1} = 0.723 \qquad s_{\mu_2} = 0.512 \qquad s_{\mu_3} = 0.695$$

and the DB_{ij} values for pairs of clusters

$$\begin{pmatrix} DB_{ij} & C_1 & C_2 & C_3 \\ \hline C_1 & - & 0.369 & 0.794 \\ C_2 & 0.369 & - & 0.242 \\ C_3 & 0.794 & 0.242 & - \end{pmatrix}$$

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

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

The silhouette coefficient (18.30) for a chosen point, say $\mathbf{x}_1$, is given as

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

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

Hubert's statistic can be computed by taking the dot product over the upper triangular elements of the proximity matrix $\mathbf{W}$ (18.26), and the $n \times n$ matrix of distances among cluster means $\mathbf{Y}$ (18.32), 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's statistic can be obtained as the correlation between $\mathbf{w}$ and $\mathbf{y}$ (18.22)

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

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

The table below summarizes the various internal measure values for the good and bad clusterings shown in Figure 18.1 and Figure 18.2.

	BetaCV	Cindex	NC	Q	Dunn	DB	SC	Γ	Γ_n
(a) Good	0.24	0.034	2.67	-0.23	0.08	0.65	0.60	8.19	0.92
(b) Bad	0.33	0.08	2.56	-0.20	0.03	1.11	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's 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.

18.3 Relative Measures

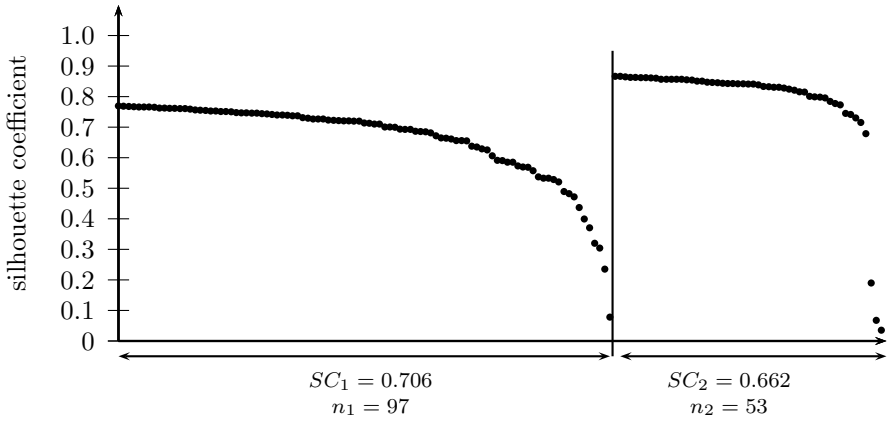
Relative measures are used to compare different clusterings obtained by varying different parameters for the same algorithm, for example, to choose the number of clusters k.

Silhouette Coefficient: The silhouette coefficient for each point s_j (18.30), and the average SC value (18.31), 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 cluster-wise SC values

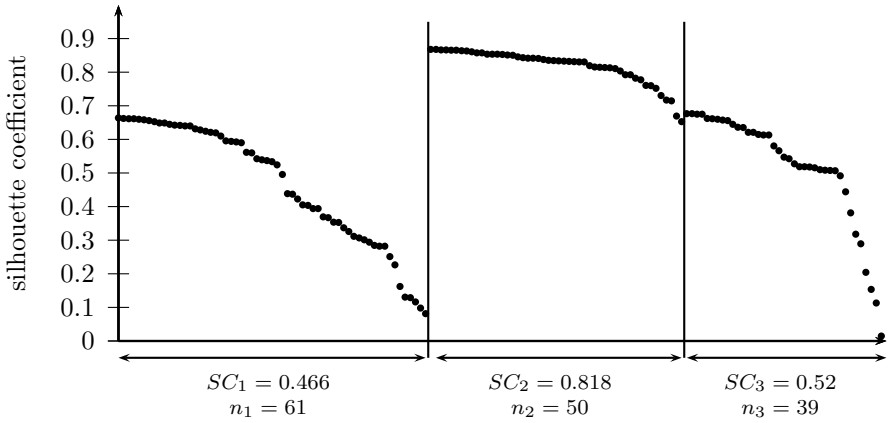
$$SC_i = \frac{1}{n_i} \sum_{\mathbf{x}_j \in C_i} s_j$$

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 ($1 \leq i \leq k$).

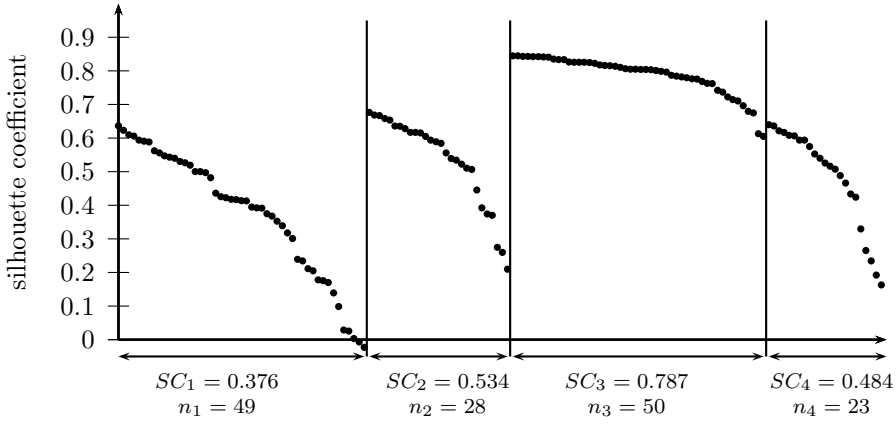
Example 18.6: Figure 18.3 shows the silhouette coefficient plot for the best clustering results for the K-means algorithm on the Iris principal components dataset



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



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



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

Figure 18.3: Iris K-means: Silhouette Coefficient Plot

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 cluster-wise average (SC_i, for $1 \leq i \leq k$) silhouette coefficient values, along with the cluster sizes, are also shown.

Figure 18.3a 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 point-wise scores are all high, except for the last three points, suggesting that the points are well clustered. The silhouette plot in Figure 18.3b, with $k = 3$, corresponds to the "good" clustering shown in Figure 18.1a. We can see that cluster C_1 from Figure 18.3a 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 18.3c. Here C_3 is the well separated cluster, corresponding to C_2 above, and the remaining clusters are essentially sub-clusters of C_1 for $k = 2$ (Figure 18.3a). Cluster C_1 also has two points with negative s_i values, indicating that they are probably mis-clustered.

Since $k = 2$ 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 $n \times d$ dataset $\mathbf{D}$, the scatter matrix for $\mathbf{D}$ is given as

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

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

$$\begin{aligned}
\mathbf{S}_W &= \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in C_i} (\mathbf{x}_j - \boldsymbol{\mu}_i)(\mathbf{x}_j - \boldsymbol{\mu}_i)^T \\
\mathbf{S}_B &= \sum_{i=1}^{k} n_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T \\
\mathbf{S} &= \mathbf{S}_W + \mathbf{S}_B
\end{aligned} \tag{18.34}$$

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 Variance Ratio Criteria 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 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 large changes at successive values of k. 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 the increase in $CH(k)$ is the most, when comparing successive differences. This is most likely to happen when there is either no improvement or even a decrease in the $CH(k+1)$ value, compared to $CH(k-1)$.

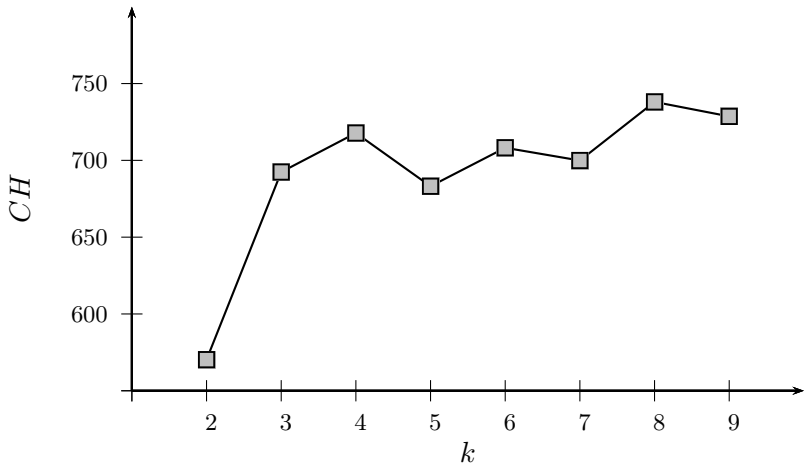


Figure 18.4: Calinski-Harabasz Variance Ratio Criteria

Example 18.7: Figure 18.4 shows the CH ratio for various values of k on the Iris principal components dataset, using the K-means algorithm (best run chosen from 200 initializations).

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 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 intra-cluster weights W_{in} (18.27) for different values of k, with the expected values assuming no apparent clustering structure, which forms the null hypothesis.

Let $C(k)$ be the clustering obtained for a specified value of k, using the chosen clustering algorithm. Let $W_{in}(k)$ denote the sum of intra-cluster weights (over all clusters) for $C(k)$. We would like to compute the probability of the $W_{in}(k)$ value, or some related statistic, under the null hypothesis. Unfortunately, the sampling distribution of W_{in} is not known. Furthermore, it depends on the number of clusters k, the number of points n, and other characteristics of the input data $\mathbf{D}$. 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 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 j-th dimension) uniformly at random within the given range. Let $\mathbf{R}_i$, $1 \leq i \leq t$ denote the i-th sample. From each sample dataset $\mathbf{R}_i$, we generate clusterings for different values of k using the same algorithm, and record the intra-cluster values $W_{in}(k, i)$, which comprise the null distributions for W_{in} for different values of k. Let $\mu_W(k)$ and $\sigma_W(k)$ denote the mean and standard

deviation of the intra-cluster weights under the null hypothesis, given as

$$\mu_W(k) = \frac{1}{t}\sum_{i=1}^{t} \log_2 W_{in}(k,i)$$

$$\sigma_W(k) = \sqrt{\frac{1}{t}\sum_{i=1}^{t}\Big(\log_2 W_{in}(k,i) - \mu_W(k)\Big)^2}$$

Since the values W_{in} can be quite large, we use $\log_2 W_{in}$ in the computations above.

The gap statistic is then defined as

$$gap(k) = \mu_W(K) - \log_2 W_{in}(k)$$

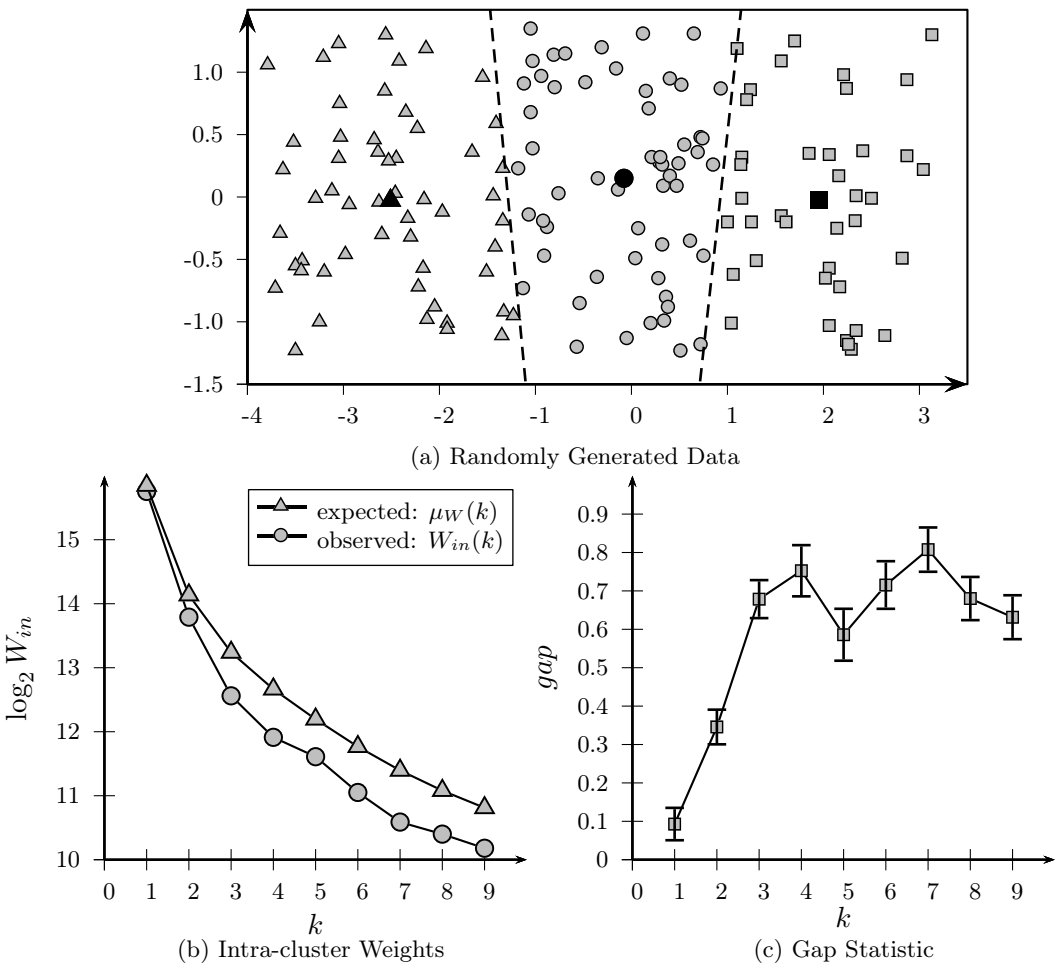
We can select the value of k that yields the largest gap statistic, since that indicates a clustering structure far away from the uniform distribution of points. A more robust approach is to choose k as follows

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

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

Example 18.8: 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 18.5a, 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 $\log_2 W_{in}(k,i)$ value.

For Monte Carlo sampling, we generated $t = 200$ such random datasets, and computed the mean or expected intra-cluster weights $\mu_W(k)$ under the null hypothesis. Figure 18.5b shows the expected intra-cluster weights for different values of k. It also shows the observed values of $\log_2 W_{in}$ 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, form which the W_{in} values are computed. We can see that the observed $W_{in}(k)$ values are smaller than the expected values $\mu_W(k)$. From these values, we then compute the gap statistics $gap(k)$, which are plotted in Figure 18.5c. The following table lists the gap values and the standard deviation for different values



(a) Randomly Generated Data

(b) Intra-cluster Weights

(c) Gap Statistic

Figure 18.5: Gap Statistic: (a) Randomly generated data, (b) intra-cluster weights for different k, and (c) gap statistic as a function of k.

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 the values above the optimal value for the number of clusters is $k = 4$, since

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

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

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

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

18.3.1 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". Of course, since the underlying joint probability distribution for $\mathbf{D}$ is unknown, we can try generating samples via a variety of methods, including random perturbations of $\mathbf{D}$, sub-sampling from $\mathbf{D}$, and bootstrap resampling from $\mathbf{D}$. The cluster stability approach can be used to find good parameter values for a given clustering algorithm on the dataset $\mathbf{D}$. We will focus on the task of finding a good value for k.

Algorithm 18.1 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 $\mathbf{D}$. Initially, we generate t samples of size n by bootstrap sampling from $\mathbf{D}$ with replacement. Sampling with replacement allows the same point to be chosen possibly multiple times, and thus each sample $\mathbf{D}_i$ will be different. Other sampling approaches are also possible. For example, we may choose to sample $m = \alpha \cdot n$ points without replacement, with $\alpha \in (0, 1)$, representing the sample size as a fraction of n. Typically α is chosen to be in the range $[0.8, 0.9]$, so that each subsample $\mathbf{D}_i$ retains most of the properties of the input dataset $\mathbf{D}$. Another alternative approach is to obtain each $\mathbf{D}_i$ by adding small random perturbations to points in $\mathbf{D}$.

Next, for each sample $\mathbf{D}_i$ we run the clustering algorithm A with different cluster values k ranging from 2 to $k^{\max}$. All the t different clusterings (over the t samples) for a given value of k are stored in the set $\mathcal{C}_k$. Next, the method compares the distance between all pairs of clusterings $C^i, C^j \in \mathcal{C}_k$ via some distance function $d(C^i, C^j)$. Several of the external cluster evaluation measures above can be used as distance measures, by setting $C = C^i$ and $T = C^j$, or vice-versa. From these pairwise distances, we compute the expected distance for each value of k (line 13). Here we assume that d is a symmetric function, i.e., $d(C^i, C^j) = d(C^j, C^i)$, and further

Algorithm 18.1: Clustering Stability Algorithm for Choosing k

```
CLUSTERING STABILITY (A, t, k^max, D):
1  n ← |D|
   // Generate t samples
2  for i ∈ [1, t] do
3      D_i ← Sample n points from D with replacement
   // Generate clusterings for different values of k
4  for i ∈ [1, t] do
5      C_k ← ∅  for k ∈ [2, k^max] do
6          C^i = A(k, D_i) // C is a clustering into k groups
7          C_k ← C_k ∪ {C^i}
   // Compute mean difference between clusterings for each k
8  foreach pair of clusterings C^i, C^j ∈ C_k with j > i do
9      D_ij ← D_i ∩ D_j // create common dataset
10     for k ∈ [2, k^max] do
11         d_ij(k) ← d(C^i, C^j, D_ij) // distance between clusterings
12 for k ∈ [2, k^max] do
13     μ_d(k) = 2/(t(t−1)) Σ_{i=1}^{t} Σ_{j>i} d_ij(k) // expected pair-wise distance
   // Choose best k
14 k* ← arg min_k { μ_d(k) }
```

that $d(C_i, C_i) = 0$. If d is not symmetric, then the expected difference should be computed over all ordered pairs, i.e., $\mu_d(k) = \frac{1}{t(t-1)} \sum_{i=1}^{r} \sum_{j \neq i} d_{ij}(k)$. Finally, the value k^* that exhibits the least deviation between the clusterings obtained from the random samples is the best choice for k, since it exhibits the most stability.

Instead of the distance function d, we can just as easily evaluate clustering stability via a similarity measure $s(C^i, C^j)$, in which case, after computing the average similarity $\mu_s(k)$ between pairs of clusterings for a given k, we can choose the best value k^* as the one that maximizes the expected similarity, i.e., $k^* = \arg\max_k \{\mu_s(k)\}$. In general, those external measures that yield lower values for better agreement between C^i and C^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's Γ statistics, and so on.

There is, however, one complication when evaluating the distance (or the similarity) between a pair of clusterings C^i and C^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, since

each sample $\mathbf{D}_i$ is different. Before computing the distance between the two clusterings, we have to restrict C^i and C^j only to the points common to both $\mathbf{D}_i$ and $\mathbf{D}_j$, denoted as $\mathbf{D}_{ij}$. Since sampling with replacement (when using bootstrap sampling) allows multiple instances of the same point, we 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

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

That is, the common dataset $\mathbf{D}_{ij}$ is created by selecting the least number of instances of each point $\mathbf{x}_a$. Thus, if $\mathbf{x}_a$ does not occur in either $\mathbf{D}_i$ or $\mathbf{D}_j$, then it will not be in $\mathbf{D}_{ij}$. Once the clusterings C^i and C^j have been restricted to the common points, we can compute the distance (or similarity) between them.

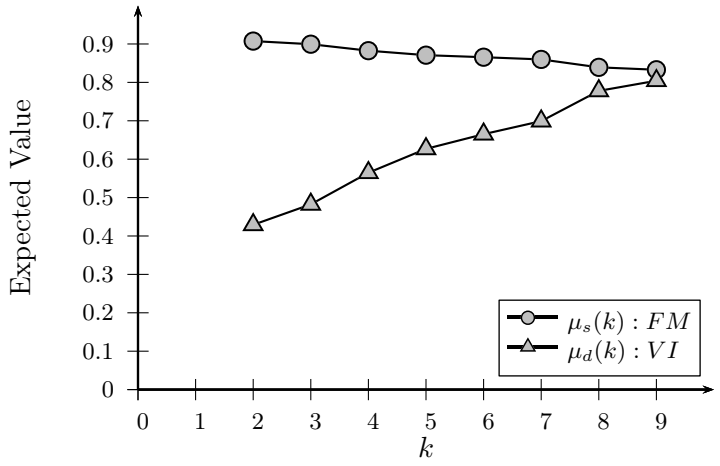


Figure 18.6: Clustering Stability: Iris Dataset

Example 18.9: 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 (18.11) between each pair of clusterings $C^i, C^j \in \mathcal{C}_k$. We also used the Fowlkes-Mallows measure (18.19) as an example of a similarity measure. The expected values of the pair-wise distance $\mu_d(k)$ for the VI measure, and the pair-wise similarity $\mu_s(k)$ for the FM measure are plotted in Figure 18.6. Both the measures indicate that $k = 2$ is the best value, since 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.

18.3.2 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, e.g., 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. We shall now look at some approaches to answer whether the data is clusterable or not.

Spatial Histogram: One simple approach is to contrast the d-dimensional spatial histogram of the given dataset $\mathbf{D}$ against those from samples generated randomly in the same data space.

Let $X_1, X_2, \cdots, X_d$ denote the d dimensions for $\mathbf{D}$. 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 these d-histograms, 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

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

where $\mathbf{i} = (i_1, i_2, \cdots, i_d)$ 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 j-th such random sample. We can then compute the corresponding EPMF $g^j(\mathbf{i})$ for each $\mathbf{R}_j$, $1 \leq j \leq t$.

Finally, we can compute how much the distribution f differs from each of the g^j, using the Kullback-Leibler (KL) divergence from f to g^j, defined as

$$KL(f|g^j) = \sum_{\mathbf{i}} f(\mathbf{i}) \log_2 \left(\frac{f(\mathbf{i})}{g^j(\mathbf{i})} \right) \tag{18.35}$$

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 increases exponentially (b^d), 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 dataset $\mathbf{D}$, and see how it differs from the PDF for the random datasets. However, the curse of dimensionality also causes problems for density estimation.

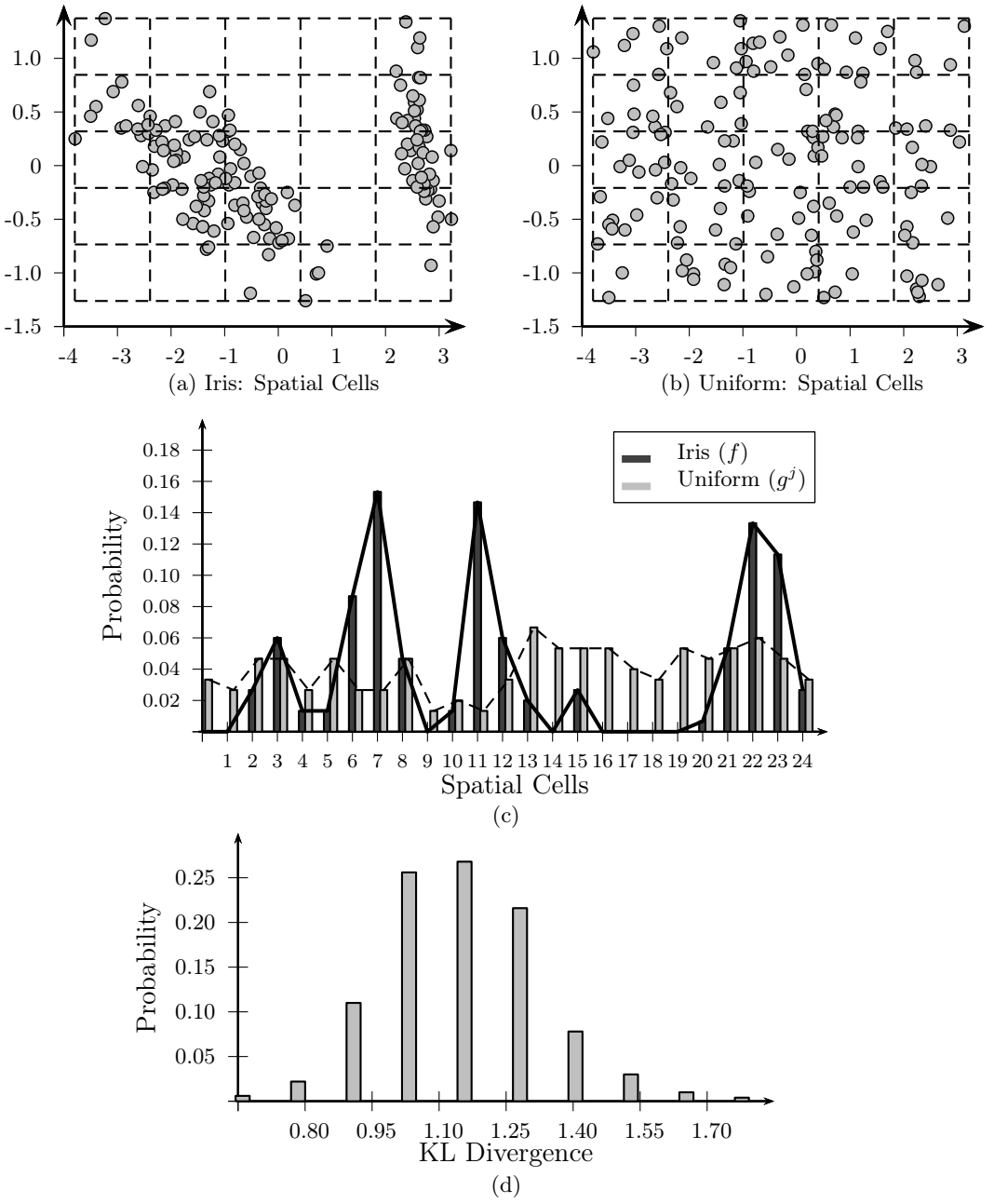


Figure 18.7: Iris Dataset: Spatial Histogram

Example 18.10: Figure 18.7c 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 $\mathbf{D}$, and the random sample $\mathbf{R}$, are shown in Figures 18.7a and 18.7b, respectively. The cells are numbered starting from zero, 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 18.7c.

We generated $t = 500$ random samples from the null (uniform) distribution, and computed the KL divergence from f to g^j for each $1 \leq j \leq t$. The distribution of the KL values is plotted in Figure 18.7d. 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 pair-wise point distances from $\mathbf{D}$, with those from the randomly generated samples $\mathbf{R}_i$ from the null distribution. That is, we create the EPMF from the proximity matrix $\mathbf{W}$ for $\mathbf{D}$ (**??**) by binning the distances into b bins

$$f_d(i) = P(w_{ab} \in \text{ bin } i | \mathbf{x}_a, \mathbf{x}_b \in \mathbf{D}, a > b) = \frac{|\{w_{ab} \in \text{ bin } i\}|}{n(n-1)/2}$$

Likewise, for each of the samples $\mathbf{R}_j$, we can determine the EPMF for the pair-wise distances, denoted g_d^j. Finally, we can compute the KL divergences between f_d and g_d^j using (18.35). The expected divergence indicates the extent to which $\mathbf{D}$ differs from the null (random) distribution.

Example 18.11: Figure 18.8a shows the distance distribution for the Iris principal components dataset $\mathbf{D}$, as well as one of the random samples $\mathbf{R}_j$. 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 $\mathbf{D}$ to each $\mathbf{R}_j$, over $t = 500$ samples. The distribution of the KL divergences is shown in Figure 18.8b. 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.

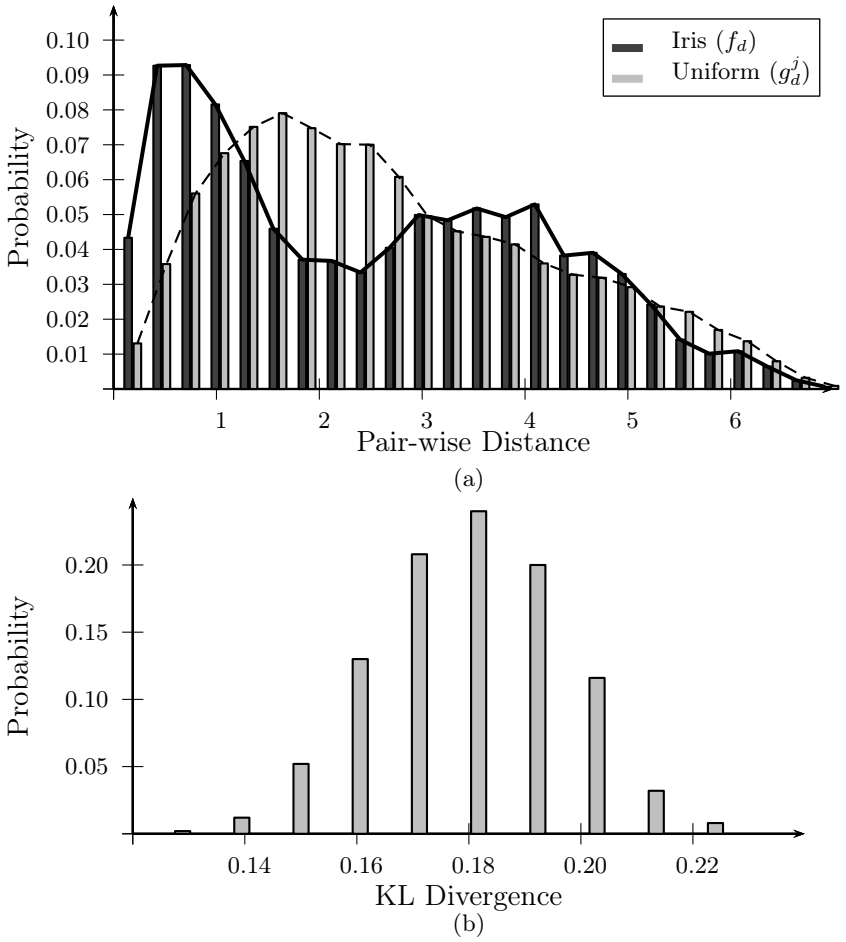


Figure 18.8: Iris Dataset: Distance Distribution

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 $\mathbf{D}$, but are generated uniformly at random along each dimension. Furthermore, we also generate t subsamples of m points directly from $\mathbf{D}$, using sampling without replacement. Let $\mathbf{D}_i$ denote the i-th 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 i-th pair of samples $\mathbf{R}_i$ and $\mathbf{D}_i$

is then defined as

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

This statistic compares the nearest-neighbor distribution of randomly generated points to the same distribution for random subsets of points from $\mathbf{D}$. 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 one. 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 zero, 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 if $\mathbf{D}$ is clusterable or not.

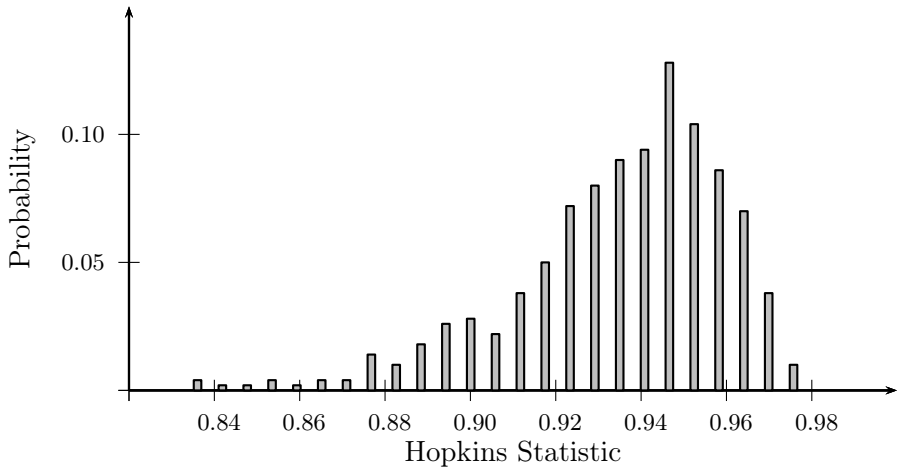


Figure 18.9: Iris Dataset: Hopkins Statistic Distribution

Example 18.12: Figure 18.9 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 $\mathbf{D}$. We used $m = 30$, using 20% of the points in $\mathbf{D}$, 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.