Data Mining and Machine Learning: Fundamental Concepts and Algorithms

dataminingbook.info

Mohammed J. Zaki[1] Wagner Meira Jr.[2]

[1]Department of Computer Science
Rensselaer Polytechnic Institute, Troy, NY, USA

[2]Department of Computer Science
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Chapter 17: Clustering Validation

Clustering Validation and Evaluation

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.

Validity measures can be divided into three main types:

External: External validation measures employ criteria that are not inherent to the dataset, e.g., class labels.

Internal: Internal validation measures employ criteria that are derived from the data itself, e.g., intracluster and intercluster distances.

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

External Measures

External measures assume that the correct or ground-truth clustering is known *a priori*, which is used to evaluate a given clustering.

Let $\boldsymbol{D} = \{\boldsymbol{x}_i\}_{i=1}^n$ 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., $T_j = \{\boldsymbol{x}_i \in \boldsymbol{D} | y_i = j\}$. We refer to $\mathcal{T}$ as the ground-truth *partitioning*, and to each T_i as a *partition*.

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 $\boldsymbol{x}_i$.

External Measures

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.

All of the external measures rely on the $r \times k$ *contingency table* $\boldsymbol{N}$ that is induced by a clustering $\mathcal{C}$ and the ground-truth partitioning $\mathcal{T}$, defined as follows

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

The count n_{ij} denotes the number of points that are common to cluster C_i and ground-truth partition T_j.

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 $\boldsymbol{x}_i \in \boldsymbol{D}$ and incrementing the corresponding count $n_{y_i \hat{y}_i}$.

Matching Based Measures: Purity

Purity quantifies the extent to which a cluster C_i contains entities from only one partition:

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

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.

Matching Based Measures: 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.

Let G be a bipartite graph over the vertex set $V = \mathcal{C} \cup \mathcal{T}$, and let the edge set be $E = \{(C_i, T_j)\}$ with edge weights $w(C_i, T_j) = n_{ij}$. 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 weight matching* in G is given as:

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

where $w(M)$ is the sum of the sum of all the edge weights in matching M, given as $w(M) = \sum_{e \in M} w(e)$

Matching Based Measures: 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:

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

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 $m_{j_i} = |T_{j_i}|$.

Matching Based Measures: F-measure

The F-measure is the harmonic mean of the precision and recall values for each C_i

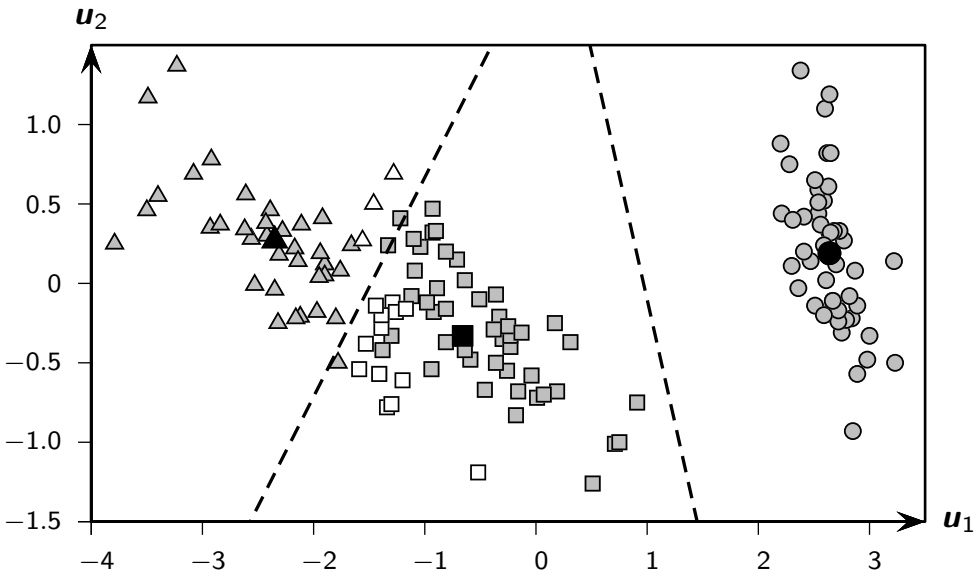
$$F_i = \frac{2}{\frac{1}{prec_i} + \frac{1}{recall_i}} = \frac{2 \cdot prec_i \cdot recall_i}{prec_i + recall_i} = \frac{2\, n_{ij_i}}{n_i + m_{j_i}}$$

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

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

K-means: Iris Principal Components Data

Good Case



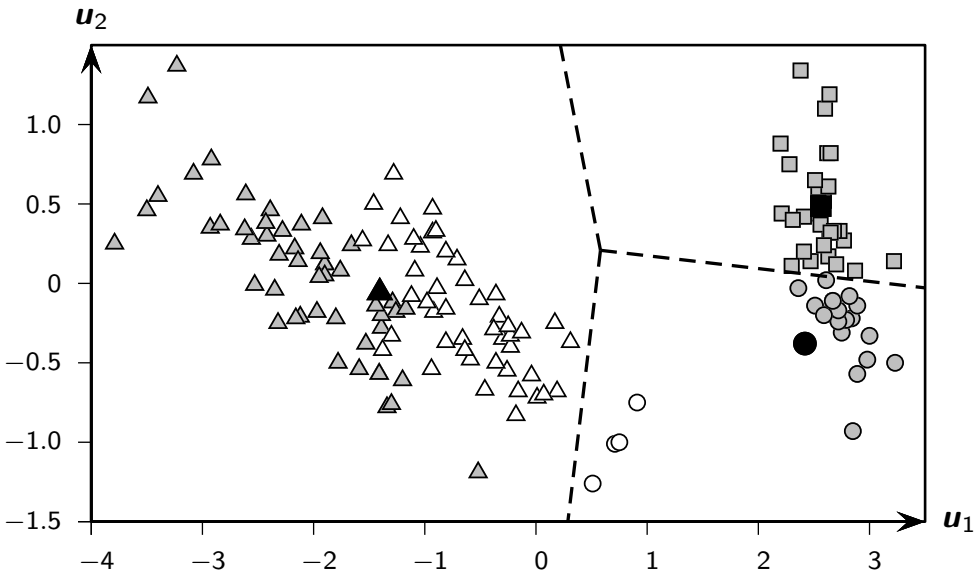
Contingency table:

	iris-setosa T_1	iris-versicolor T_2	iris-virginica 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	$n = 100$

$purity = 0.887$, $match = 0.887$, $F = 0.885$.

K-means: Iris Principal Components Data

Bad Case



Contingency table:

	iris-setosa T_1	iris-versicolor T_2	iris-virginica T_3	n_i
C_1(*squares*)	30	0	0	30
C_2(*circles*)	20	4	0	24
C_3(*triangles*)	0	46	50	96
m_j	50	50	50	$n = 150$

$purity = 0.667,\ match = 0.560,\ F = 0.658$

Entropy-based Measures: Conditional Entropy

The entropy of a clustering $\mathcal{C}$ and partitioning $\mathcal{T}$ is given as

$$H(\mathcal{C}) = -\sum_{i=1}^{r} p_{C_i} \log p_{C_i} \qquad H(\mathcal{T}) = -\sum_{j=1}^{k} p_{T_j} \log p_{T_j}$$

where $p_{C_i} = \frac{n_i}{n}$ and $p_{T_j} = \frac{m_j}{n}$ are the probabilities of cluster C_i and 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

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

Entropy-based Measures: Conditional Entropy

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

$$H(\mathcal{T}|\mathcal{C}) = \sum_{i=1}^{r} \frac{n_i}{n} H(\mathcal{T}|C_i) = \boxed{-\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij} \log\left(\frac{p_{ij}}{p_{C_i}}\right)}$$

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

where $p_{ij} = \frac{n_{ij}}{n}$ is the probability that a point in cluster i also belongs to partition and 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 $\mathcal{T}$.

$H(\mathcal{T}|\mathcal{C}) = 0$ if and only if $\mathcal{T}$ is completely determined by $\mathcal{C}$, corresponding to the ideal clustering. If $\mathcal{C}$ and $\mathcal{T}$ are independent of each other, then $H(\mathcal{T}|\mathcal{C}) = H(\mathcal{T})$.

Entropy-based Measures: 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

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

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.

The *normalized mutual information* (NMI) is defined as the geometric mean:

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

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

Entropy-based Measures: 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

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

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 $\mathcal{C}$.

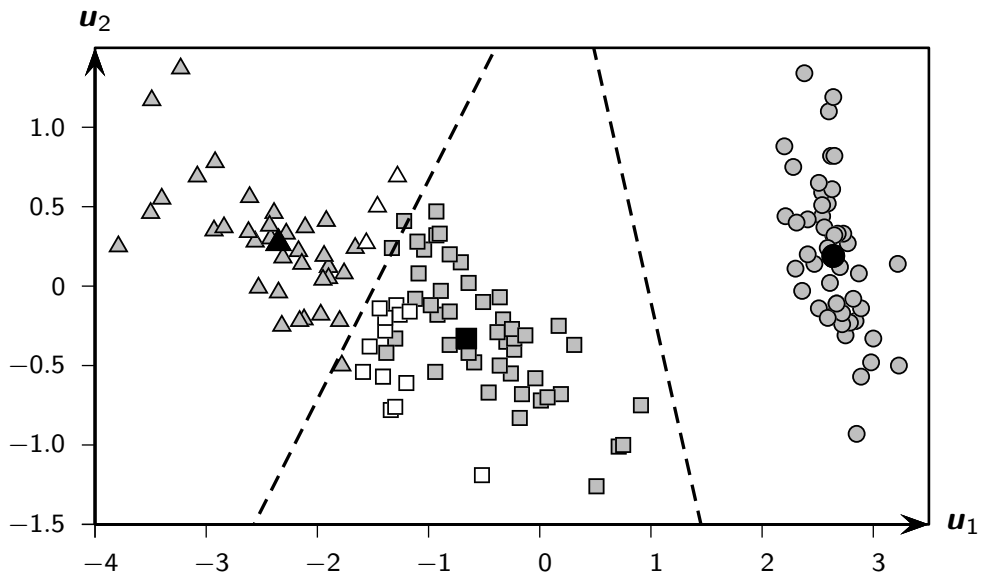
VI can also be expressed as:

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

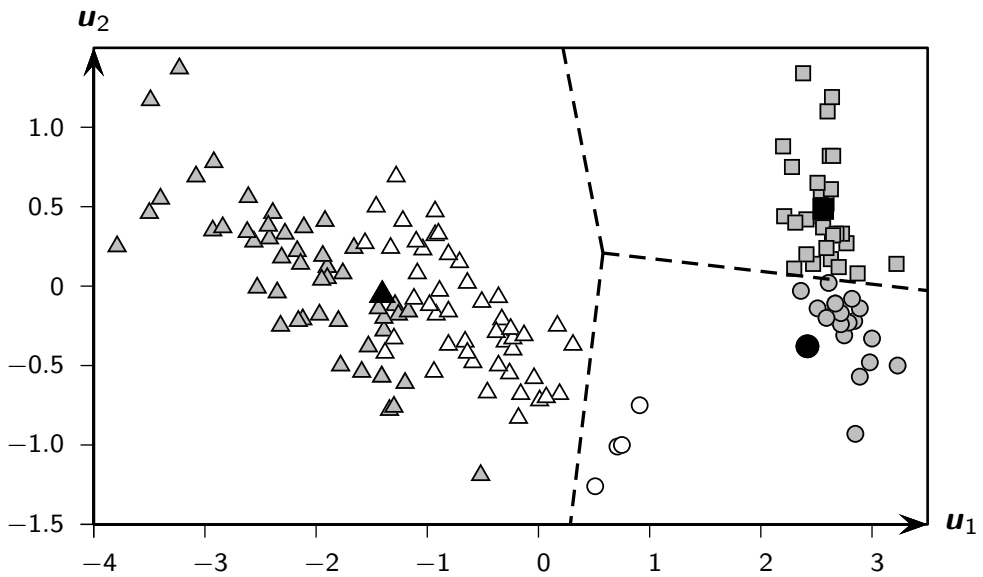
$$VI(\mathcal{C},\mathcal{T}) = 2H(\mathcal{T},\mathcal{C}) - H(\mathcal{T}) - H(\mathcal{C})$$

K-means: Iris Principal Components Data

Good Case



(a) K-means: good



(b) K-means: bad

	purity	*match*	*F*	$H(\mathcal{T}\|\mathcal{C})$	*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	1.200

Pairwise Measures

Given clustering $\mathcal{C}$ and ground-truth partitioning $\mathcal{T}$, let $\boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{D}$ 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 $\boldsymbol{x}_i$.

If both $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ 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: $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to the same partition in $\mathcal{T}$, and they are also in the same cluster in $\mathcal{C}$. The number of true positive pairs is given as

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

False Negatives: $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to the same partition in $\mathcal{T}$, but they do not belong to the same cluster in $\mathcal{C}$. The number of all false negative pairs is given as

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

Pairwise Measures

False Positives: $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ do not belong to the same partition in $\mathcal{T}$, but they do belong to the same cluster in $\mathcal{C}$. The number of false positive pairs is given as

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

True Negatives: $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ neither belong to the same partition in $\mathcal{T}$, nor do they belong to the same cluster in $\mathcal{C}$. The number of such true negative pairs is given as

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

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

$$N = TP + FN + FP + TN$$

Pairwise Measures: TP, TN, FP, FN

They can be computed efficiently using the contingency table $\boldsymbol{N} = \{n_{ij}\}$. The number of true positives is given as

$$TP = \frac{1}{2}\left(\left(\sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^2\right) - n\right)$$

The false negatives can be computed as

$$FN = \frac{1}{2}\left(\sum_{j=1}^{k} m_j^2 - \sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^2\right)$$

The number of false positives are:

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

Finally, the number of true negatives can be obtained via

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

Pairwise Measures: Jaccard Coefficient, Rand Statistic

Jaccard Coefficient: measures the fraction of true positive point pairs, but after ignoring the true negative:

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

Rand Statistic: measures the fraction of true positives and true negatives over all point pairs:

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

Pairwise Measures: FM Measure

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

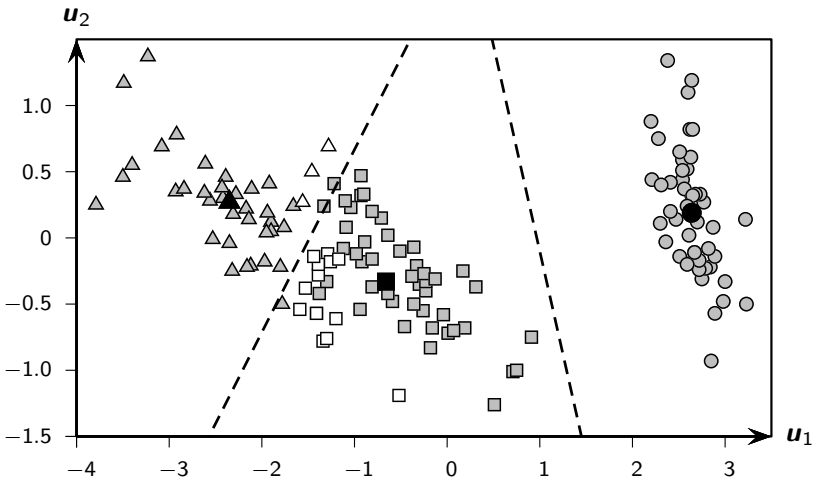
$$prec = TP/TP + FP \qquad\qquad recall = TP/TP + FN$$

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

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

K-means: Iris Principal Components Data

Good Case



Contingency table:

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

The number of true positives is:

$$TP = \binom{47}{2} + \binom{14}{2} + \binom{50}{2} + \binom{3}{2} + \binom{36}{2} = 3030$$

Likewise, we have $FN = 645$, $FP = 766$, $TN = 6734$, and $N = \binom{150}{2} = 11175$.
We therefore have: $Jaccard = 0.682$, $Rand = 0.887$, $FM = 0.811$.
For the "bad" clustering, we have: $Jaccard = 0.477$, $Rand = 0.717$, $FM = 0.657$.

Correlation Measures: Hubert statistic

Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be two symmetric $n \times n$ matrices, and let $N = \binom{n}{2}$. Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$ denote the vectors obtained by linearizing the upper triangular elements (excluding the main diagonal) of $\boldsymbol{X}$ and $\boldsymbol{Y}$.

Let μ_X denote the element-wise mean of $\boldsymbol{x}$, given as

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

and let $\boldsymbol{z}_x$ denote the centered $\boldsymbol{x}$ vector, defined as $\boldsymbol{z}_x = \boldsymbol{x} - 1 \cdot \mu_X$

The Hubert statistic is defined as

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

The normalized Hubert statistic is defined as the element-wise correlation

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

Correlation-based Measure: Discretized Hubert Statistic

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

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

Let $\boldsymbol{t}, \boldsymbol{c} \in \mathbb{R}^N$ denote the N-dimensional vectors comprising the upper triangular elements (excluding the diagonal) of $\boldsymbol{T}$ and $\boldsymbol{C}$. Let $\boldsymbol{z}_t$ and $\boldsymbol{z}_c$ denote the centered $\boldsymbol{t}$ and $\boldsymbol{c}$ vectors.

The discretized Hubert statistic is computed by setting $\boldsymbol{x} = \boldsymbol{t}$ and $\boldsymbol{y} = \boldsymbol{c}$:

$$\Gamma = \frac{1}{N} \boldsymbol{t}^T \boldsymbol{c} = \frac{TP}{N}$$

The normalized version of the discretized Hubert statistic is simply the correlation between $\boldsymbol{t}$ and $\boldsymbol{c}$

$$\Gamma_n = \frac{\boldsymbol{z}_t^T \boldsymbol{z}_c}{\|\boldsymbol{z}_t\| \cdot \|\boldsymbol{z}_c\|} = \frac{\frac{TP}{N} - \mu_T \mu_C}{\sqrt{\mu_T \mu_C (1-\mu_T)(1-\mu_C)}}$$

where $\mu_T = \frac{TP+FN}{N}$ and $\mu_C = \frac{TP+FP}{N}$.

Internal Measures

Internal evaluation measures do not have recourse to the ground-truth partitioning. 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:

$$\boldsymbol{W} = \left\{ \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \right\}_{i,j=1}^{n} \tag{1}$$

where $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ is the Euclidean distance between $\boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{D}$.

The proximity matrix $\boldsymbol{W}$ is the adjacency matrix of the weighted complete graph G over the n points, that is, with nodes $V = \{\boldsymbol{x}_i \mid \boldsymbol{x}_i \in \boldsymbol{D}\}$, edges $E = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) \mid \boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{D}\}$, and edge weights $w_{ij} = \boldsymbol{W}(i,j)$ for all $\boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{D}$.

Internal Measures

The clustering $\mathcal{C}$ can be considered as a k-way cut in G. 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

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

We denote by $\overline{S} = V - S$ the complementary set of vertices.

The sum of all the intracluster and intercluster weights are given as

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

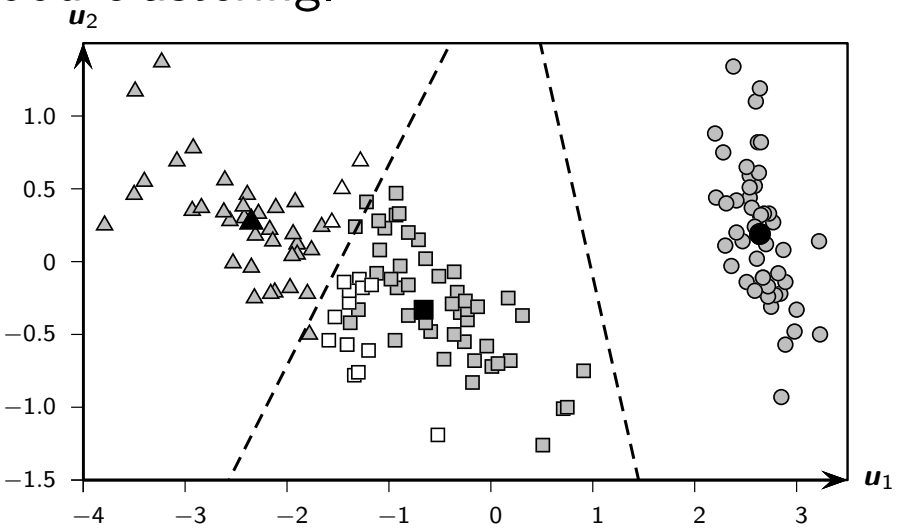
The number of distinct intracluster and intercluster edges is given as

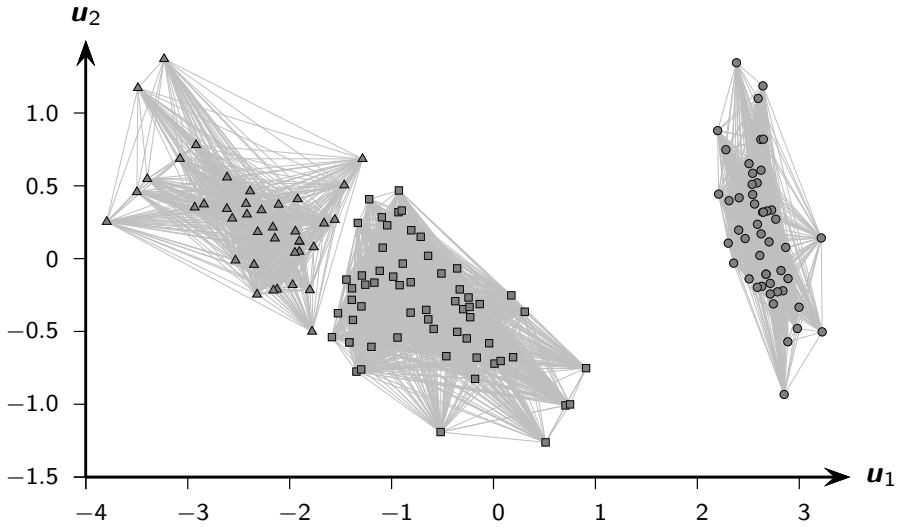
$$N_{in} = \sum_{i=1}^{k} \binom{n_i}{2} \qquad N_{out} = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} n_i \cdot n_j$$

Clusterings as Graphs: Iris

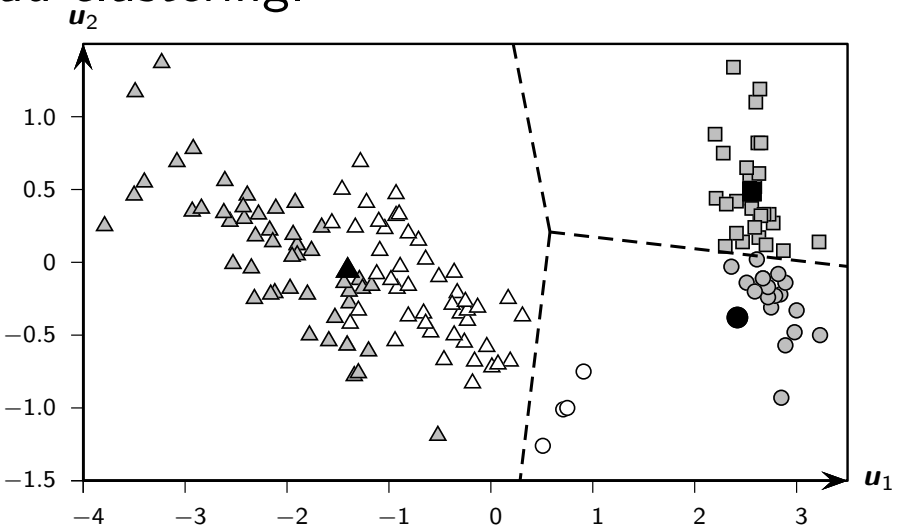
Only intracluster edges shown.

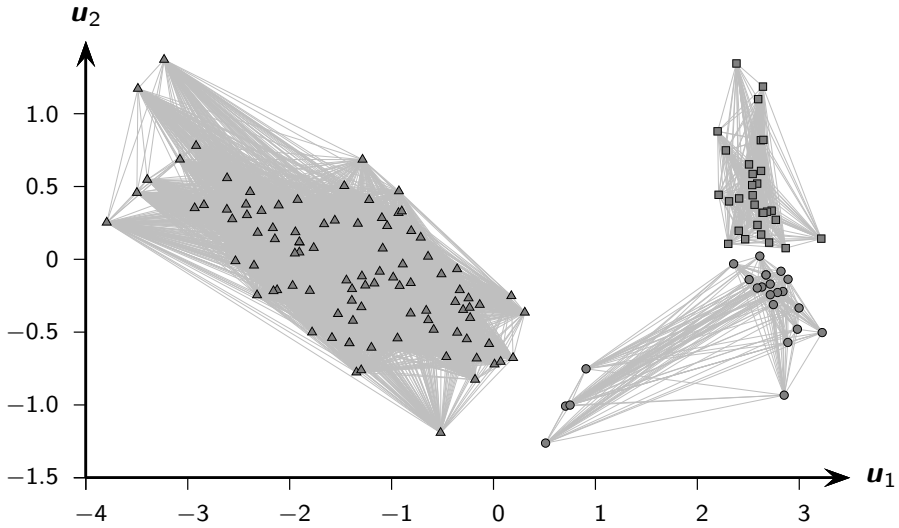
Good clustering.





Bad clustering.





Internal Measures: BetaCV and C-index

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.

C-index: Let $W_{\min}(N_{in})$ be the sum of the smallest N_{in} distances in the proximity matrix $\boldsymbol{W}$, 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 $\boldsymbol{W}$.

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

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

The C-index lies in the range $[0, 1]$. The smaller the C-index, the better the clustering.

Internal Measures: Normalized Cut and Modularity

Normalized Cut Measure: The normalized cut objective 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. The higher the normalized cut value the better.

Modularity: The modularity objective is given as

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

The smaller the modularity measure the better the clustering.

Internal Measures: 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

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

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

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

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

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

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.

Internal Measures: Davies-Bouldin Index

Let μ_i denote the cluster mean

$$\mu_i = \frac{1}{n_i} \sum_{\boldsymbol{x}_j \in C_i} \boldsymbol{x}_j$$

Let σ_{μ_i} denote the dispersion or spread of the points around the cluster mean

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

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

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

The smaller the DB value the better the clustering

Silhouette Coefficient

Define the silhoutte coefficient of a point $\boldsymbol{x}_i$ as

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

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

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

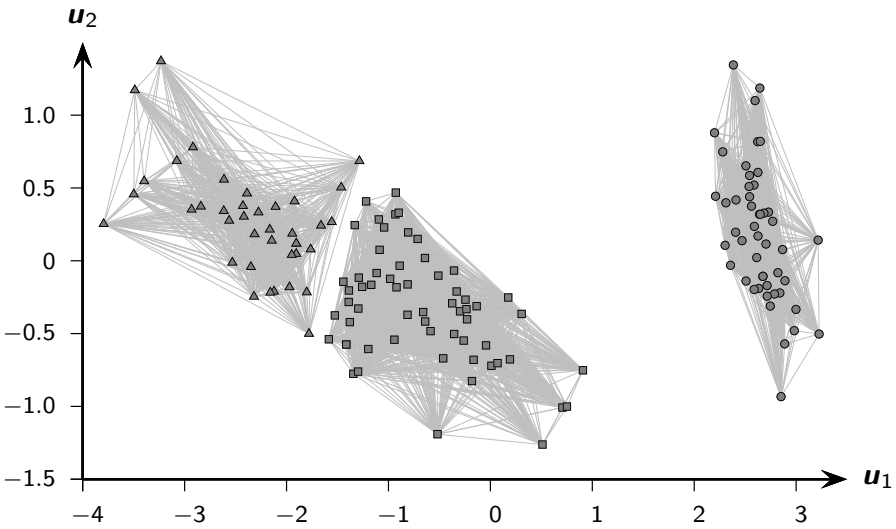
and $\mu_{out}^{\min}(\boldsymbol{x}_i)$ is the mean of the distances from $\boldsymbol{x}_i$ to points in the closest cluster:

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

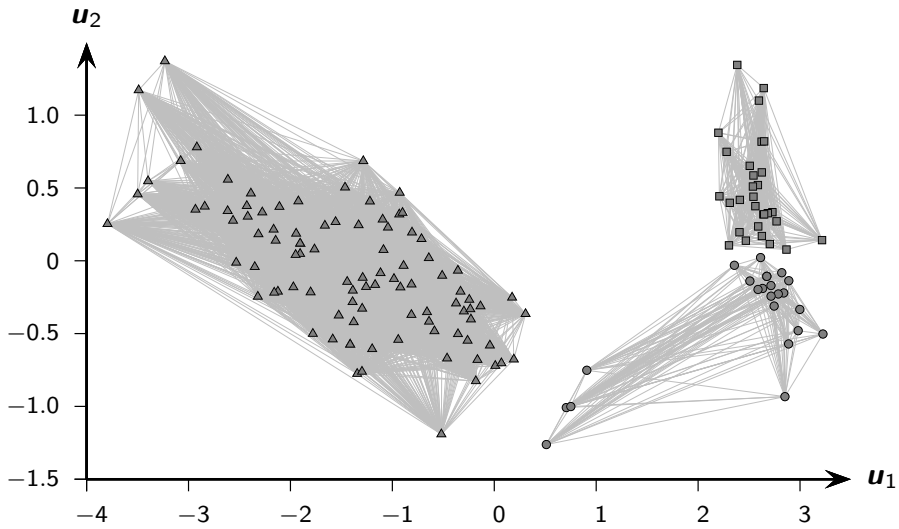
The s_i value lies in the interval $[-1, +1]$. A value close to $+1$ indicates that $\boldsymbol{x}_i$ is much closer to points in its own cluster, a value close to zero indicates $\boldsymbol{x}_i$ is close to the boundary, and a value close to -1 indicates that $\boldsymbol{x}_i$ is much closer to another cluster, and therefore may be mis-clustered.

The silhouette coefficient is the mean s_i value: $SC = \frac{1}{n}\sum_{i=1}^{n} s_i$. A value close to $+1$ indicates a good clustering.

Iris Data: Good vs. Bad Clustering



(a) Good



(b) Bad

	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

Relative Measures: Silhouette Coefficient

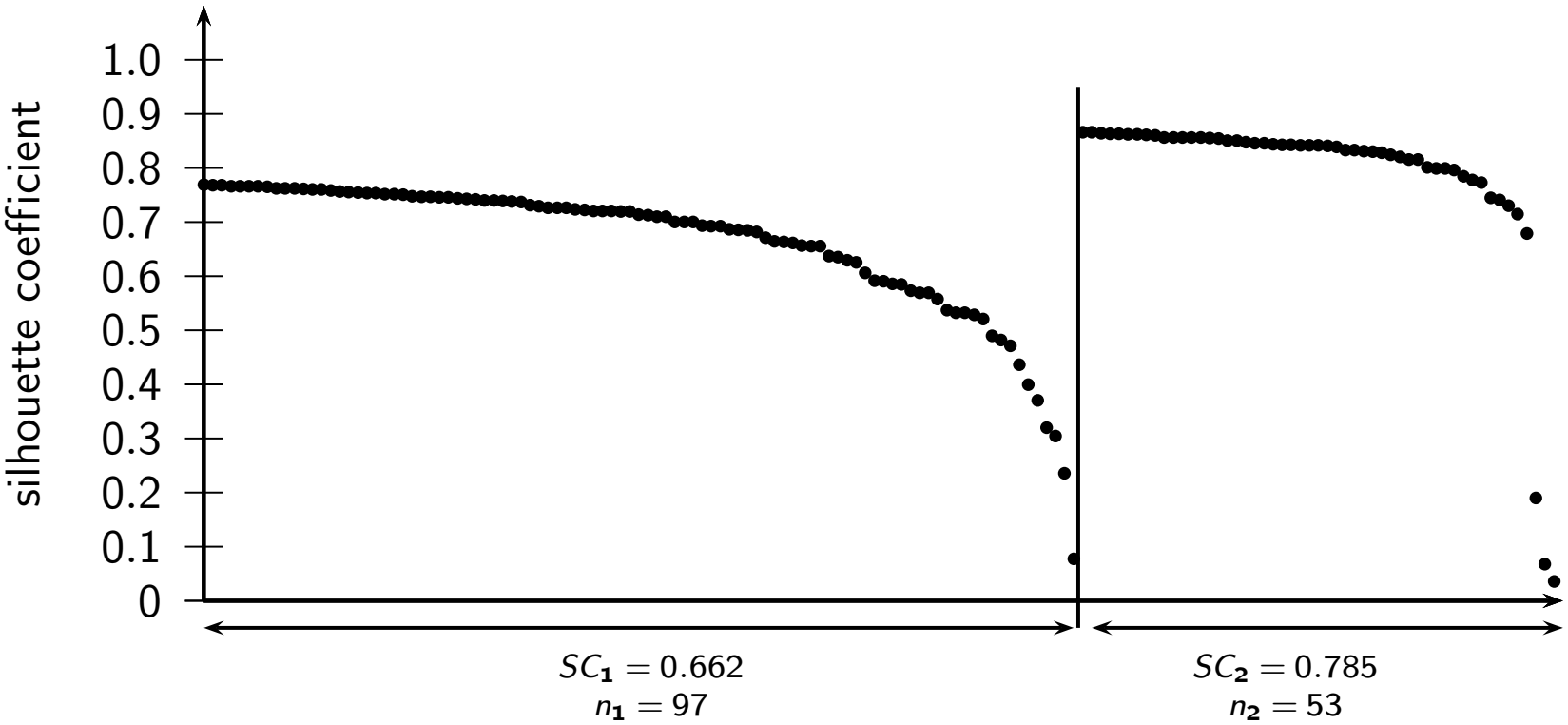
The silhouette coefficient for each point s_j, and the average SC value can be used to estimate the number of clusters in the data.

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:

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

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

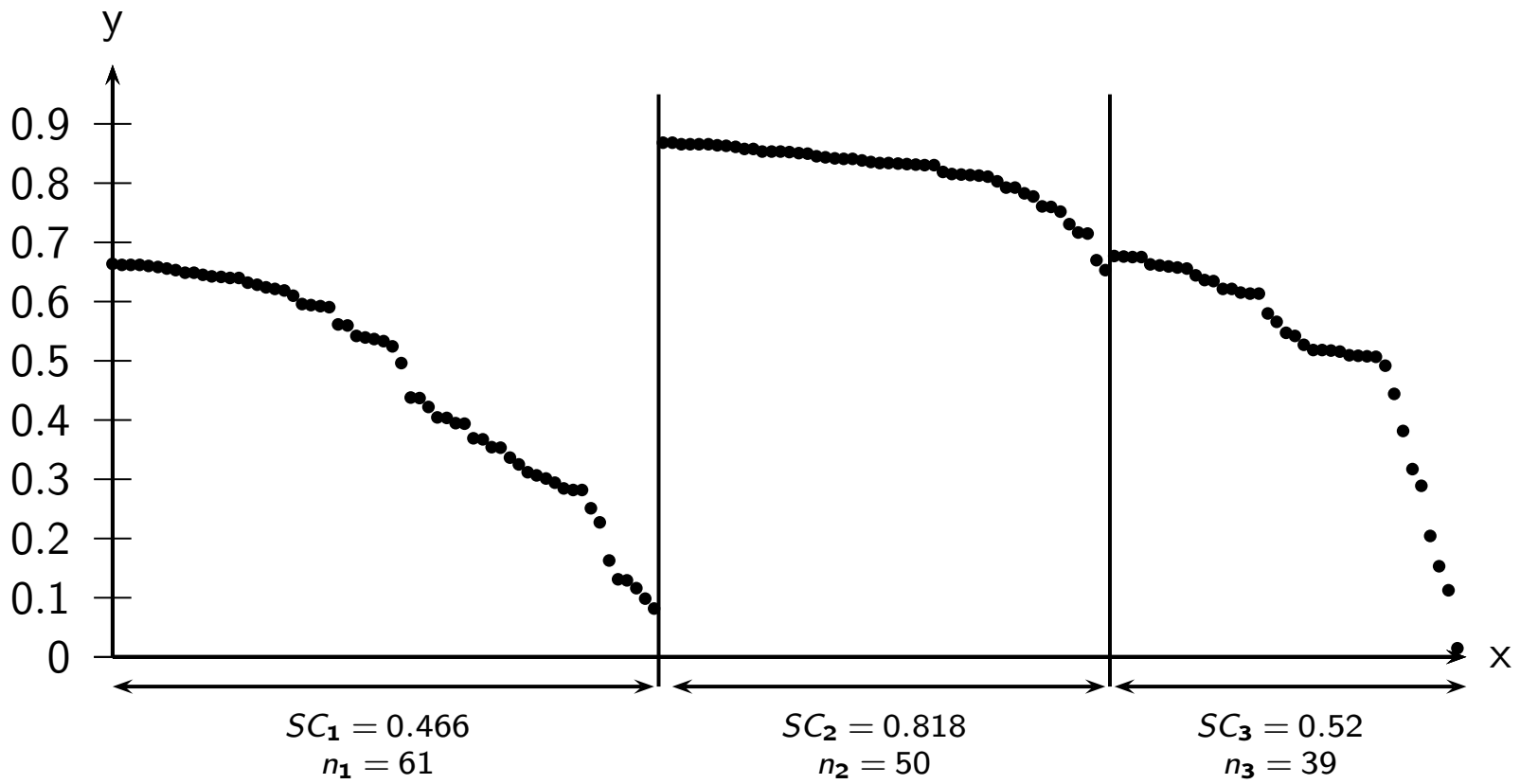
Iris K-means: Silhouette Coefficient Plot ($k = 2$)



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

$k = 2$ yields the highest silhouette coefficient, with the two clusters essentially well separated. C_1 starts out with high s_i values, which gradually drop as we get to border points. 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.

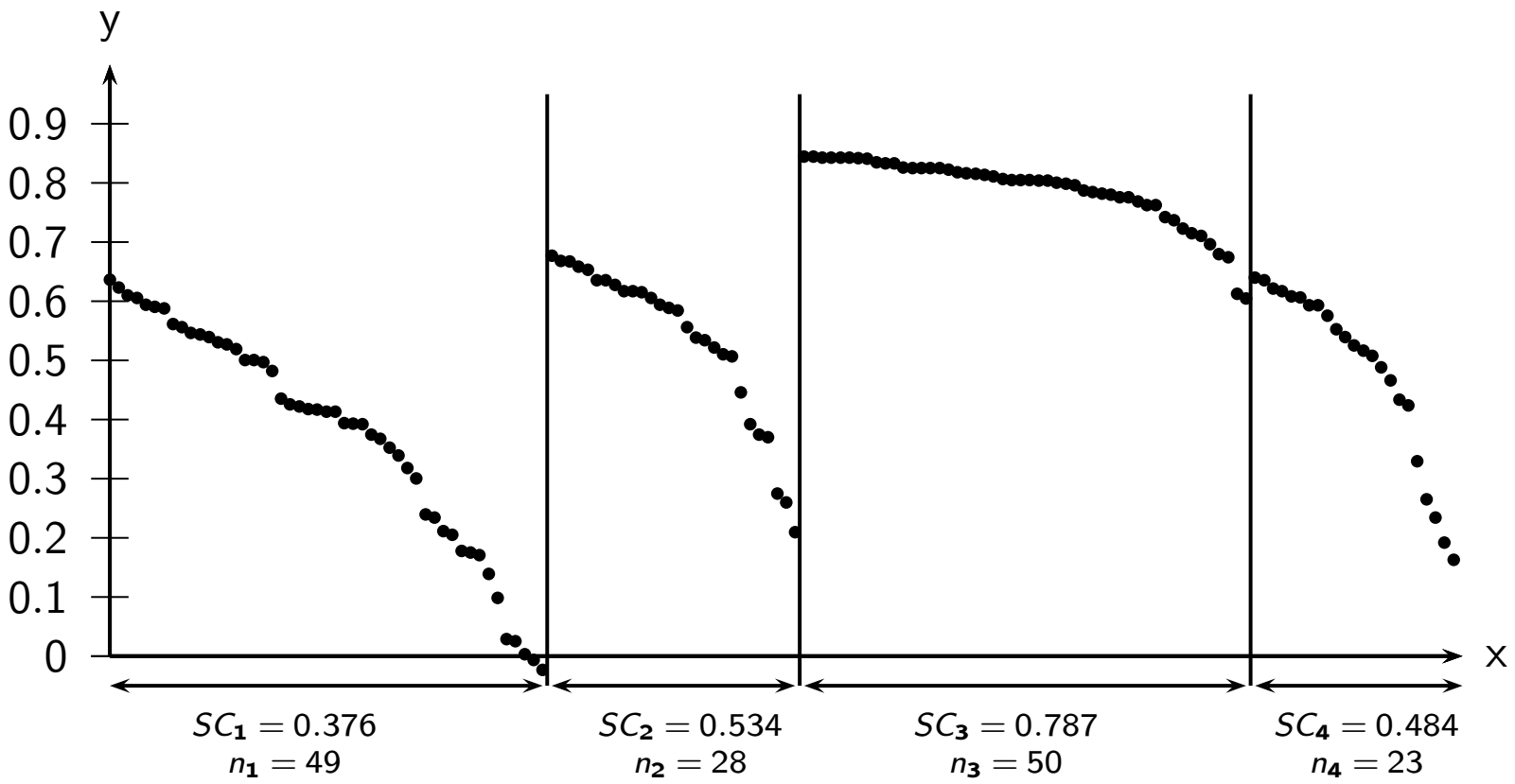
Iris K-means: Silhouette Coefficient Plot ($k = 3$)



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

C_1 from $k = 2$ 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.

Iris K-means: Silhouette Coefficient Plot ($k = 4$)



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

C_3 is the well separated cluster, corresponding to C_2 (in $k = 2$ and $k = 3$), and the remaining clusters are essentially subclusters of C_1 for $k = 2$. Cluster C_1 also has two points with negative s_i values, indicating that they are probably misclustered.

Relative Measures: Calinski–Harabasz Index

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

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

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

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

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

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

Relative Measures: Calinski–Harabasz Index

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

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

where tr is the trace of the matrix.

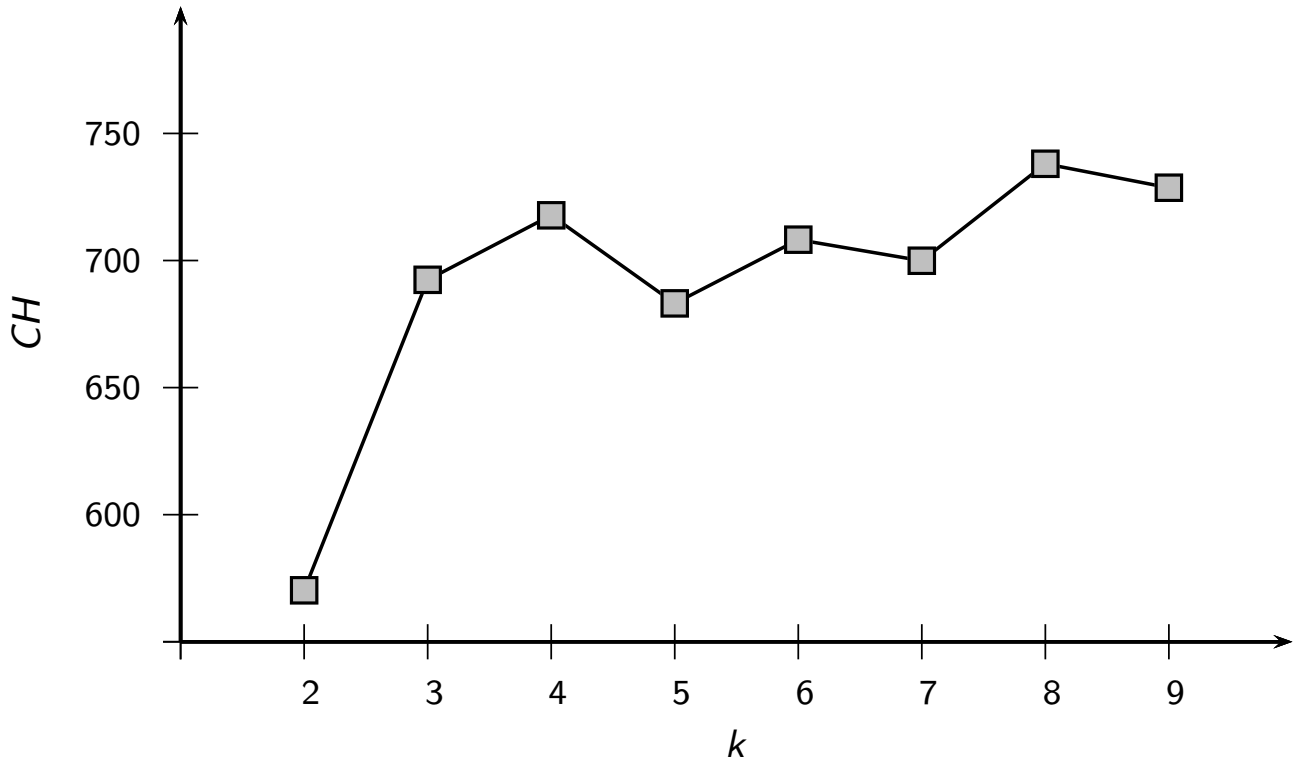
We plot the CH values and look for a large increase in the value followed by little or no gain. We choose the value $k > 3$ that minimizes the term

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

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.

Calinski–Harabasz Variance Ratio

CH ratio for various values of k on the Iris principal components data, using the K-means algorithm, with the best results chosen from 200 runs.



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	–

$\Delta(k)$ suggests $k = 3$ as the best (lowest) value.

Relative Measures: Gap Statistic

The gap statistic compares the sum of intracluster weights W_{in} 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. Let $W_{in}^k(\boldsymbol{D})$ denote the sum of intracluster weights (over all clusters) for $\mathcal{C}_k$ on the input dataset $\boldsymbol{D}$.

We would like to compute the probability of the observed W_{in}^k value under the null hypothesis. To obtain an empirical distribution for W_{in}, we resort to Monte Carlo simulations of the sampling process.

Relative Measures: Gap Statistic

We generate t random samples comprising n points. Let $\boldsymbol{R}_i \in \mathbb{R}^{n \times d}$, $1 \leq i \leq t$ denote the ith sample. Let $W_{in}^k(\boldsymbol{R}_i)$ denote the sum of intracluster weights for a given clustering of $\boldsymbol{R}_i$ into k clusters.

From each sample dataset $\boldsymbol{R}_i$, we generate clusterings for different values of k, and record the intracluster values $W_{in}^k(\boldsymbol{R}_i)$.

Let $\mu_W(k)$ and $\sigma_W(k)$ denote the mean and standard deviation of these intracluster weights for each value of k. The *gap statistic* for a given k is then defined as

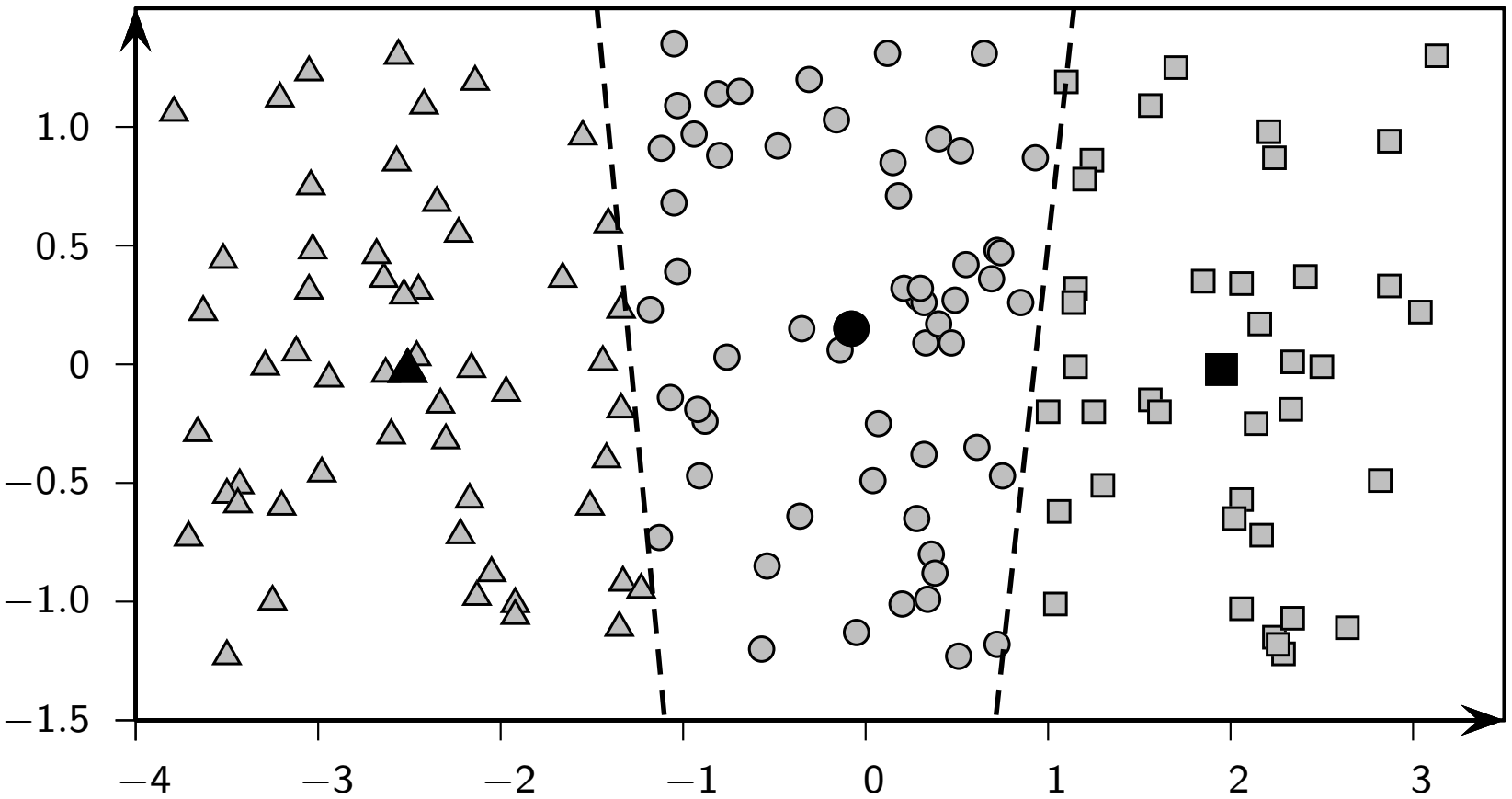
$$gap(k) = \mu_W(k) - \log W_{in}^k(\boldsymbol{D})$$

Choose k as follows:

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

Gap Statistic: Randomly Generated Data

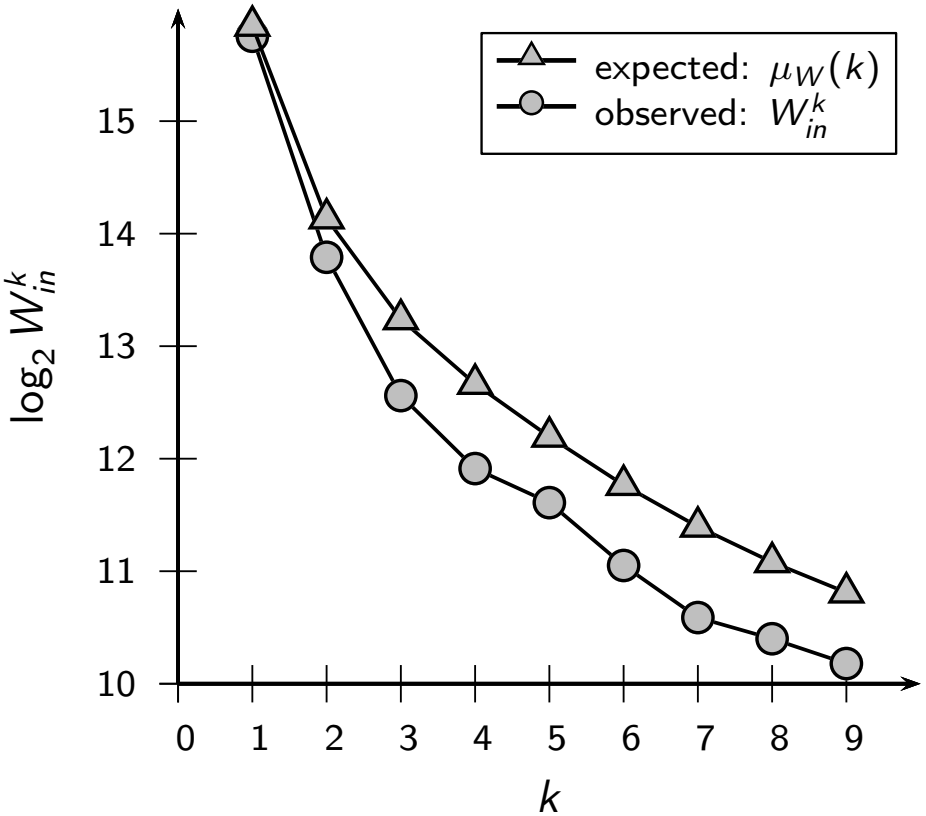
A random sample of $n = 150$ points, which does not have any apparent cluster structure.



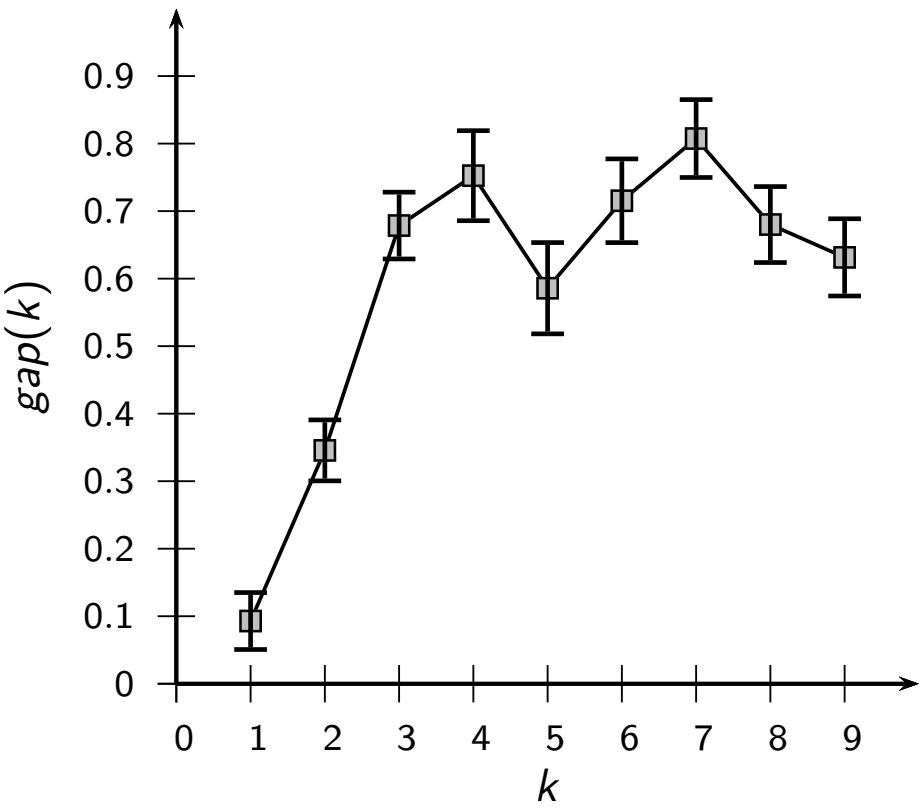
(a) Randomly generated data ($k = 3$)

Gap Statistic: Intracluster Weights and Gap Values

We generate $t = 200$ random datasets, and compute both the expected and the observed (Iris) intracluster weight $\mu_W(k)$, for each value of k. The observed $W_{in}^k(\boldsymbol{D})$ values are smaller than the expected values $\mu_W(k)$.



(b) Intracluster weights



(c) Gap statistic

Gap Statistic 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

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 relax the gap test to be within two standard deviations, then the optimal value is $k = 3$ because

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

Cluster Stability

The main idea behind cluster stability is that the clusterings obtained from several datasets sampled from the same underlying distribution as $\boldsymbol{D}$ should be similar or "stable."

Stability can be used to find a good value for k, the correct number of clusters.

We generate t samples of size n by sampling from $\boldsymbol{D}$ with replacement. Let $\mathcal{C}_k(\boldsymbol{D}_i)$ denote the clustering obtained from sample $\boldsymbol{D}_i$, for a given value of k.

Next, we compare the distance between all pairs of clusterings $\mathcal{C}_k(\boldsymbol{D}_i)$ and $\mathcal{C}_k(\boldsymbol{D}_j)$ using several of the external cluster evaluation measures. 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 obtained from the resampled datasets is the best choice for k because it exhibits the most stability.

Clustering Stability Algorithm

ClusteringStability $(A, t, k^{\max}, \boldsymbol{D})$:

$n \leftarrow |\boldsymbol{D}|$

for $i = 1, 2, \ldots, t$ **do**

  $\boldsymbol{D}_i \leftarrow$ sample n points from $\boldsymbol{D}$ with replacement

for $i = 1, 2, \ldots, t$ **do**

  for $k = 2, 3, \ldots, k^{\max}$ **do**

    $\mathcal{C}_k(\boldsymbol{D}_i) \leftarrow$ cluster $\boldsymbol{D}_i$ into k clusters using algorithm A

foreach *pair* $\boldsymbol{D}_i, \boldsymbol{D}_j$ *with* $j > i$ **do**

  $\boldsymbol{D}_{ij} \leftarrow \boldsymbol{D}_i \cap \boldsymbol{D}_j$ `// create common dataset`

  for $k = 2, 3, \ldots, k^{\max}$ **do**

    $d_{ij}(k) \leftarrow d\big(\mathcal{C}_k(\boldsymbol{D}_i), \mathcal{C}_k(\boldsymbol{D}_j), \boldsymbol{D}_{ij}\big)$ `// distance between clusterings`

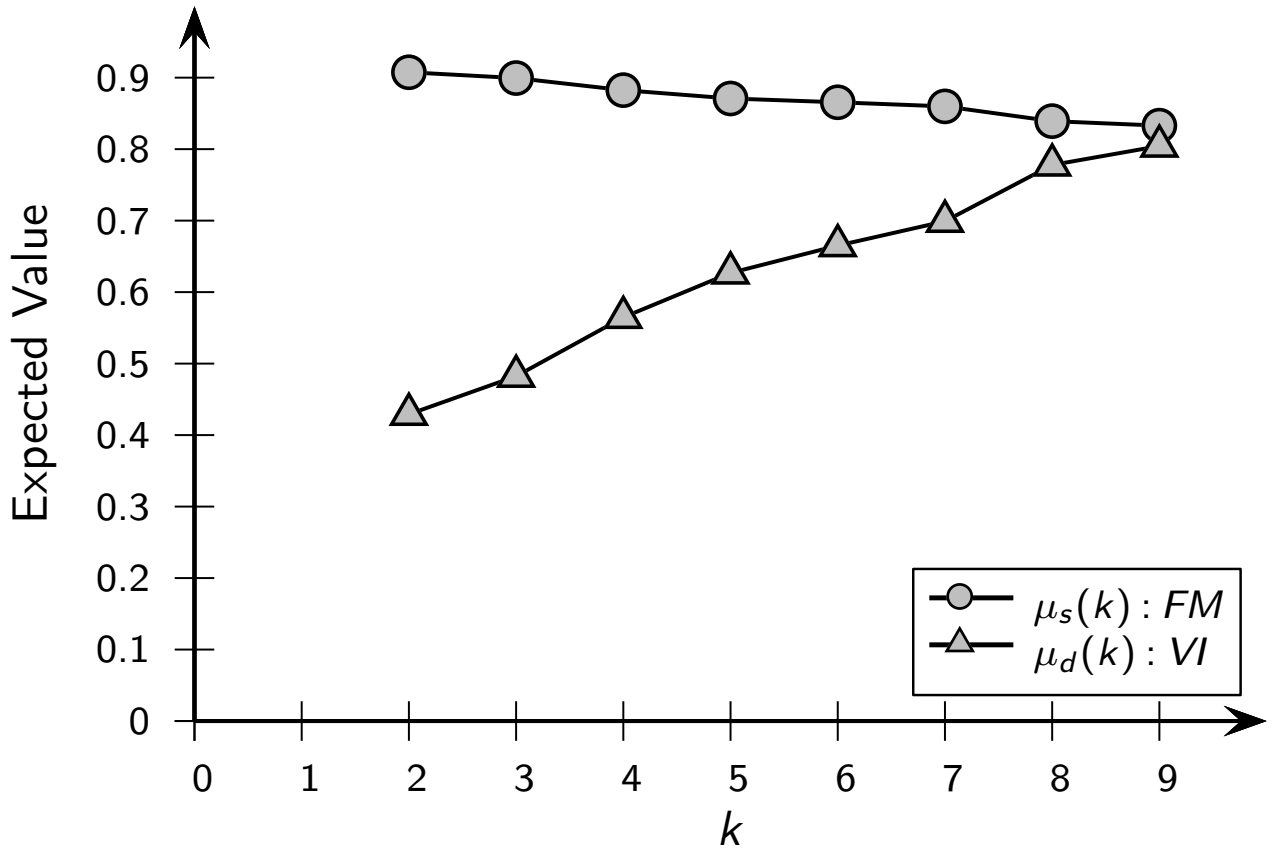
$\mu_d(k) \leftarrow \frac{2}{t(t-1)} \sum_{i=1}^{t} \sum_{j>i} d_{ij}(k)$

$k^* \leftarrow \arg\min_k \{\mu_d(k)\}$

Clustering Stability: Iris Data

$t = 500$ bootstrap samples; best K-means from 100 runs

Both the Variation of Information and the Fowlkes-Mallows measures indicate that $k = 2$ is the best value. VI indicates the least expected distance between pairs of clusterings, and FM indicates the most expected similarity between clusterings.



Clustering Tendency: Spatial Histogram

Clustering tendency or clusterability aims to determine whether the dataset $\boldsymbol{D}$ has any meaningful groups to begin with.

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 $\boldsymbol{D}$

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

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

Clustering Tendency: Spatial Histogram

We generate t random samples, each comprising n points within the same d-dimensional space as the input dataset $\boldsymbol{D}$. Let $\boldsymbol{R}_j$ denote the jth such random sample. We then compute the corresponding EPMF $g_j(\boldsymbol{i})$ for each $\boldsymbol{R}_j$, $1 \leq j \leq t$.

We next 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

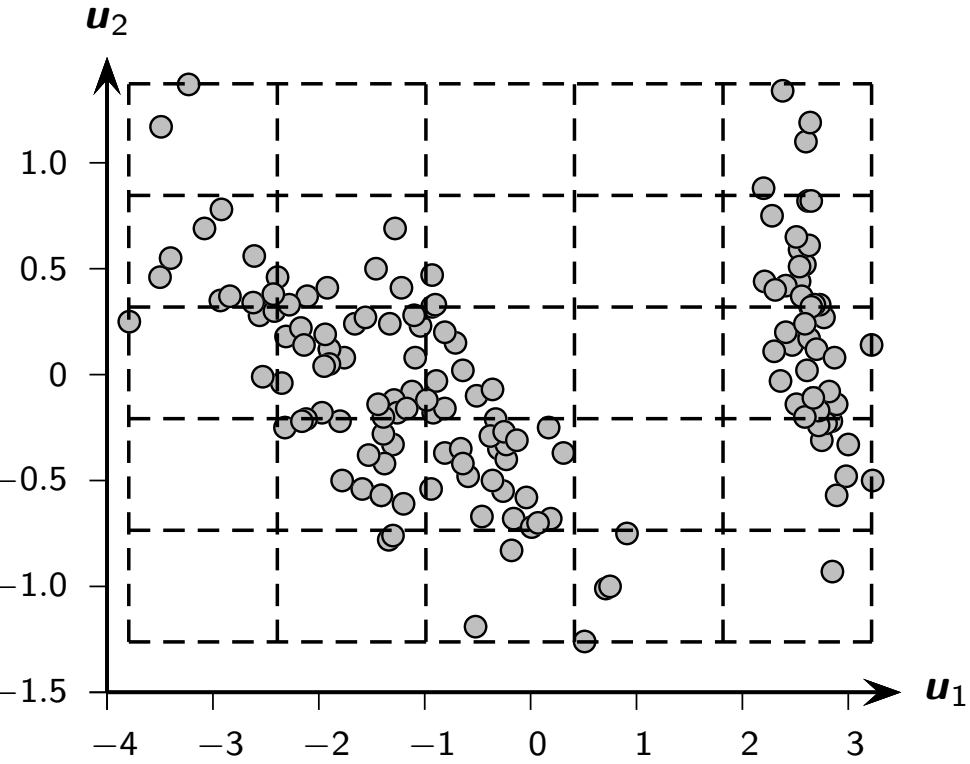
$$KL(f|g_j) = \sum_{\boldsymbol{i}} f(\boldsymbol{i}) \log \left(\frac{f(\boldsymbol{i})}{g_j(\boldsymbol{i})} \right)$$

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 $\boldsymbol{D}$ differs from a random dataset.

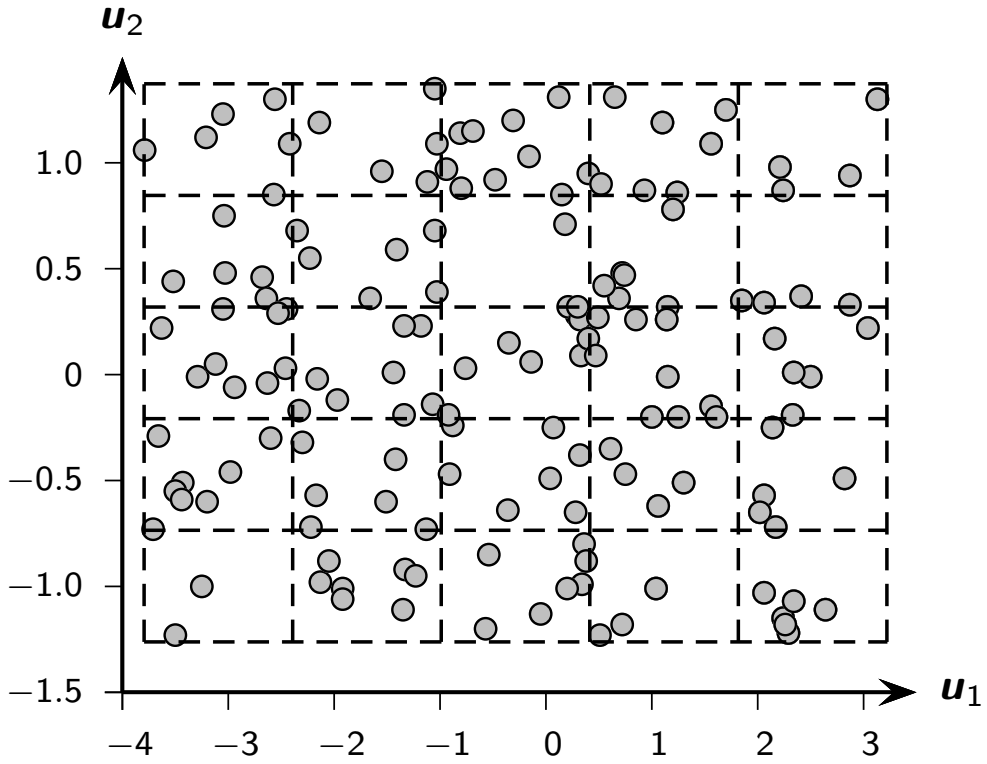
Its main limitation is that the number of cells (b^d) increases exponentially with the dimensionality, and, with a fixed sample size n, most of the cells will have none or one point, making it hard to estimate the divergence. The method is also sensitive to the choice of parameter b.

Spatial Histogram: Iris PCA Data versus Uniform

Uniform has $n = 150$ points



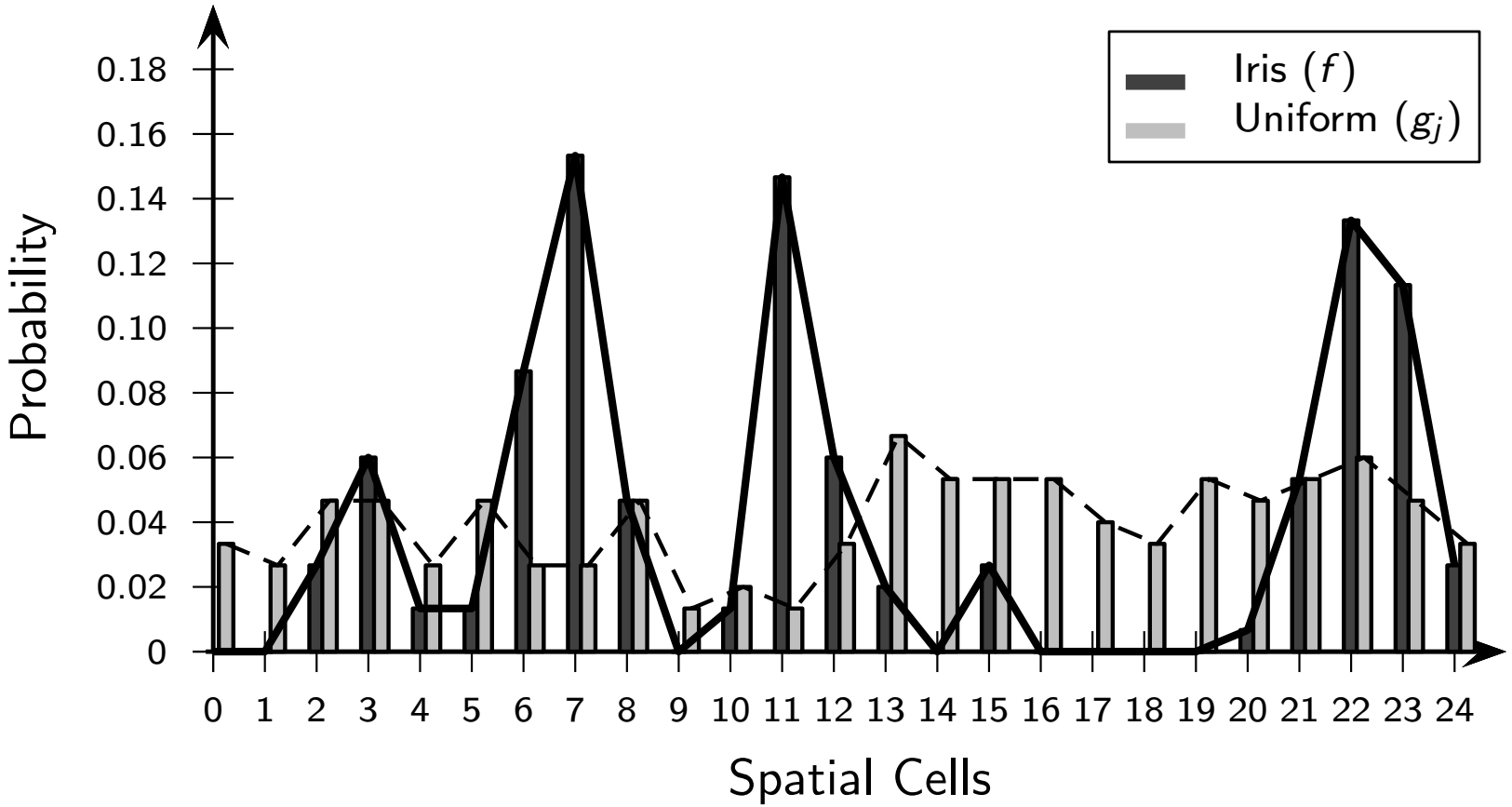
(a) Iris: spatial cells



(b) Uniform: spatial cells

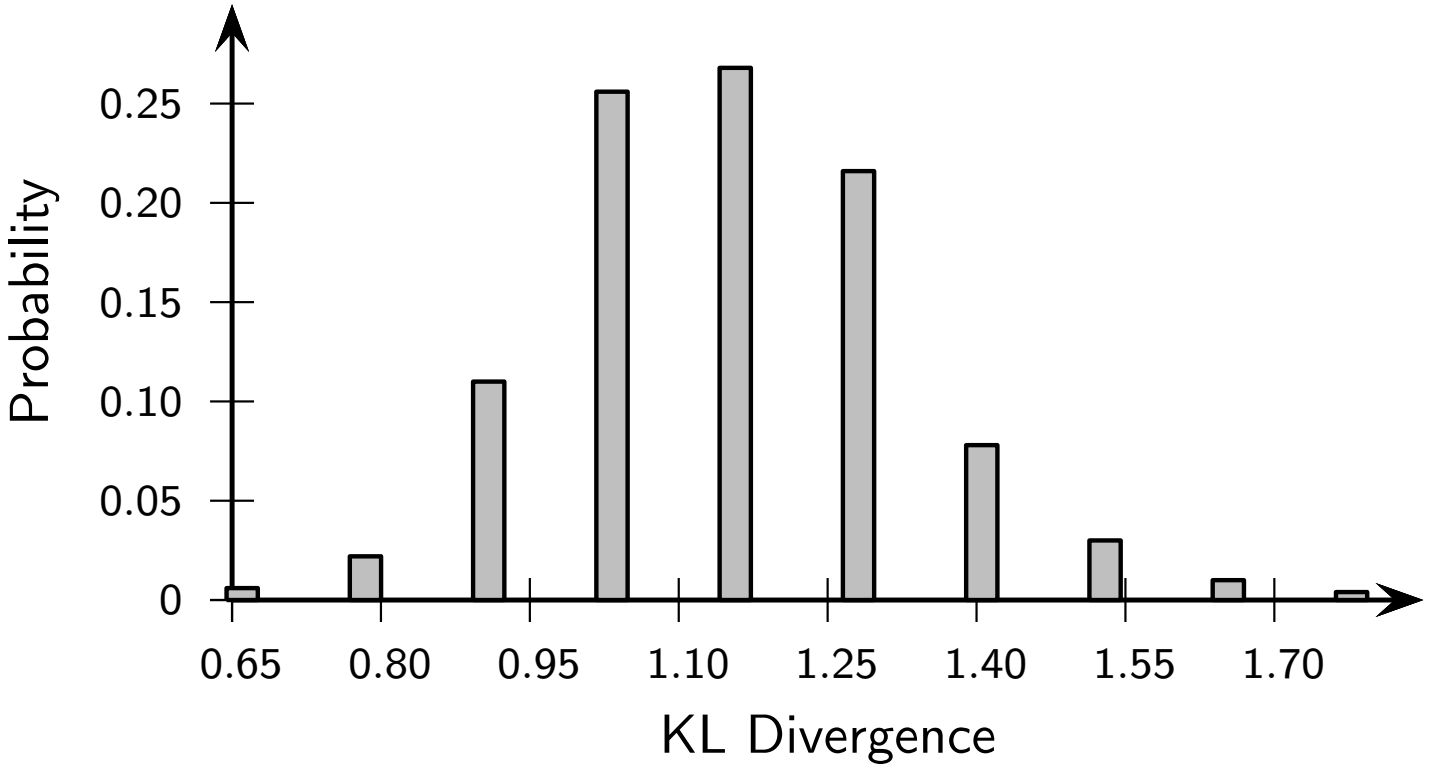
Spatial Histogram: Empirical PMF

5 bins results in 25 spatial cells



(c) Empirical probability mass function

Spatial Histogram: KL Divergence Distribution



(d) KL-divergence distribution

We generated $t = 500$ random samples from the null distribution, and computed the KL divergence from f to g_j for each $1 \leq j \leq t$.

The mean KL value is $\mu_{KL} = 1.17$, with a standard deviation of $\sigma_{KL} = 0.18$, that is, Iris PCA is clusterable.

Clustering Tendency: Distance Distribution

We can compare the pairwise point distances from $\boldsymbol{D}$, with those from the randomly generated samples $\boldsymbol{R}_i$ from the null distribution.

We create the EPMF from the proximity matrix $\boldsymbol{W}$ for $\boldsymbol{D}$ by binning the distances into b bins:

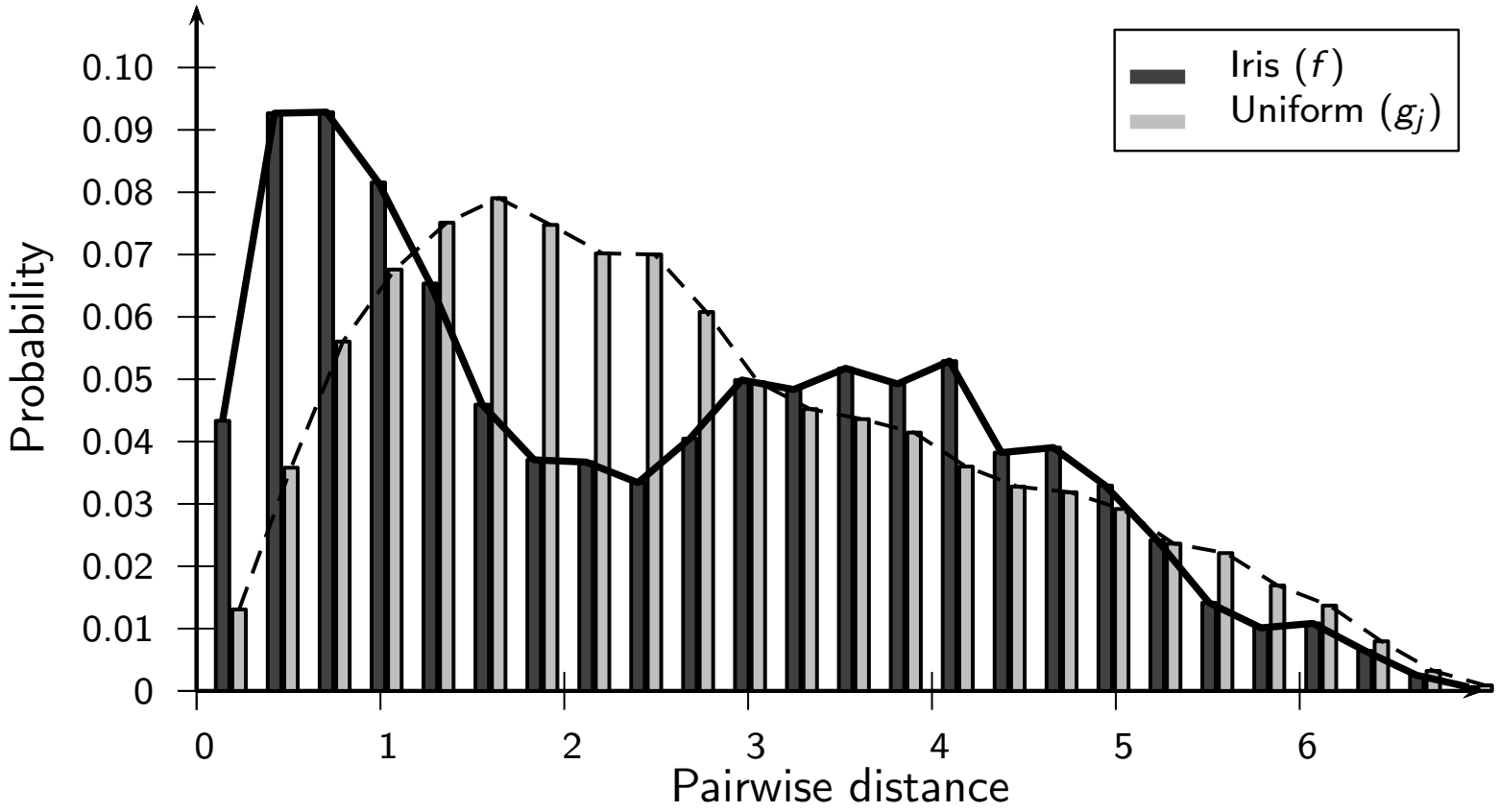
$$f(i) = P(w_{pq} \in \text{ bin } i \mid \boldsymbol{x}_p, \boldsymbol{x}_q \in \boldsymbol{D}, p < q) = \frac{\left|\{w_{pq} \in \text{ bin } i\}\right|}{n(n-1)/2}$$

Likewise, for each of the samples $\boldsymbol{R}_j$, we determine the EPMF for the pairwise distances, denoted g_j.

Finally, we compute the KL divergences between f and g_j. The expected divergence indicates the extent to which $\boldsymbol{D}$ differs from the null (random) distribution.

Iris PCA Data × Uniform: Distance Distribution

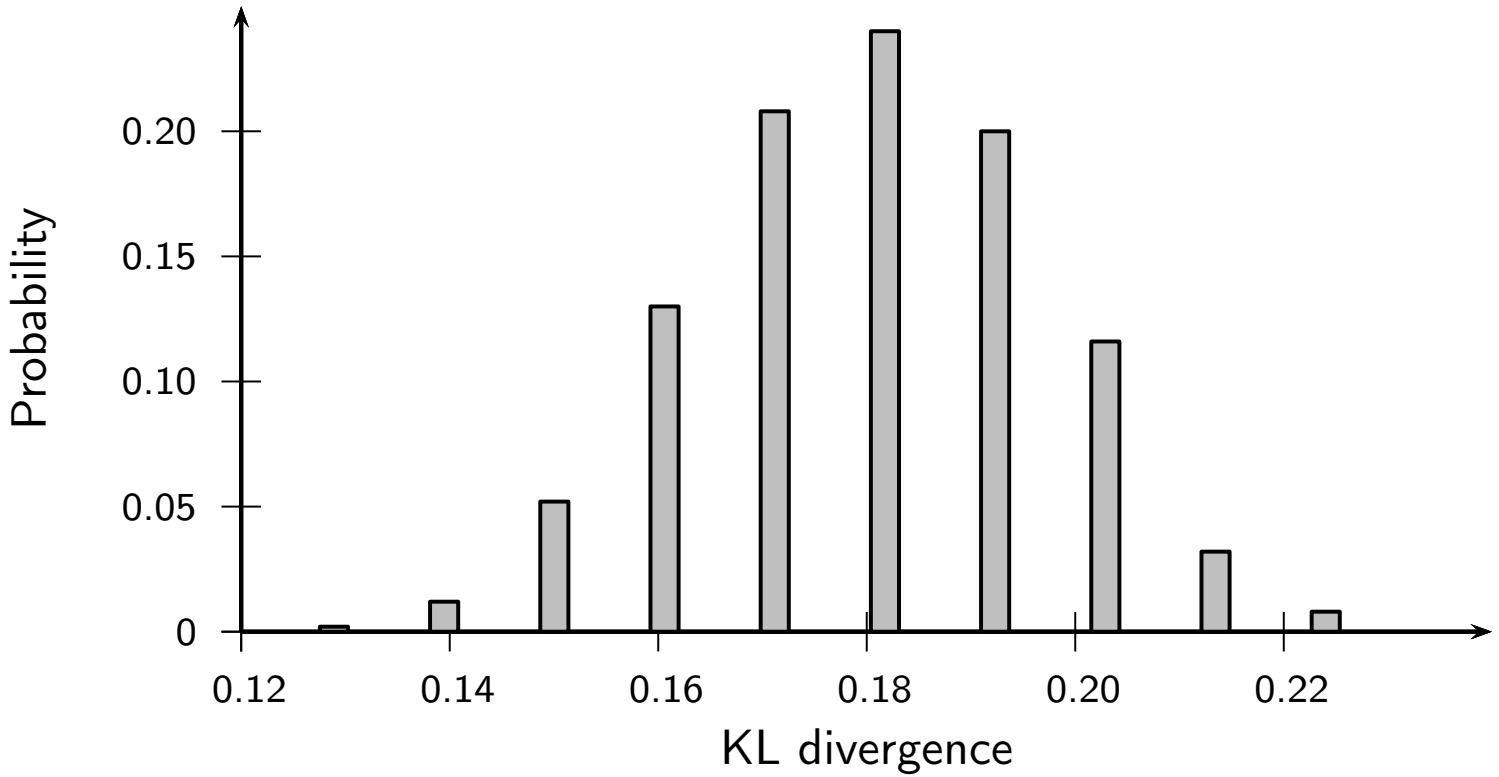
The distance distribution is obtained by binning the edge weights between all pairs of points using $b = 25$ bins.



(a)

Iris PCA Data $\times$ Uniform: Distance Distribution

We compute the KL divergence from $\boldsymbol{D}$ to each $\boldsymbol{R}_j$, over $t = 500$ samples. 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.



(b)

We conclude that, at least for the Iris dataset, the distance distribution is not as discriminative as the spatial histogram approach for clusterability analysis.

Clustering Tendency: Hopkins Statistic

Given a dataset $\boldsymbol{D}$ comprising n points, we generate t uniform subsamples $\boldsymbol{R}_i$ of m points each, sampled from the same dataspace as $\boldsymbol{D}$.

We also generate t subsamples of m points directly from $\boldsymbol{D}$, using sampling without replacement. Let $\boldsymbol{D}_i$ denote the ith direct subsample.

Next, we compute the minimum distance between each point $\boldsymbol{x}_j \in \boldsymbol{D}_i$ and points in $\boldsymbol{D}$

$$\delta_{\min}(\boldsymbol{x}_j) = \min_{\boldsymbol{x}_i \in \boldsymbol{D}, \boldsymbol{x}_i \neq \boldsymbol{x}_j} \left\{ \|\boldsymbol{x}_j - \boldsymbol{x}_i\| \right\}$$

We also compute the minimum distance $\delta_{\min}(\boldsymbol{y}_j)$ between a point $\boldsymbol{y}_j \in \boldsymbol{R}_i$ and points in $\boldsymbol{D}$.

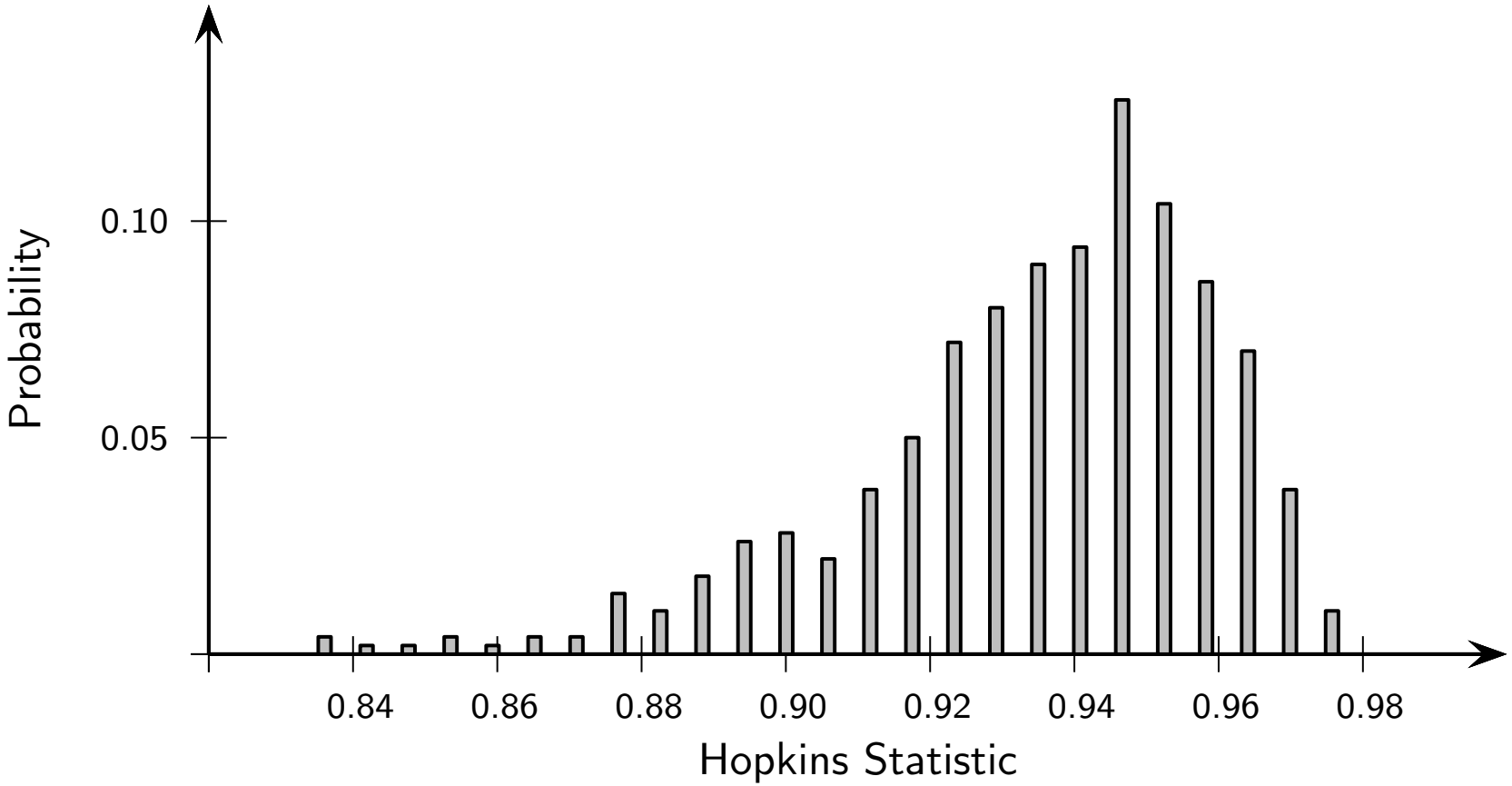
The Hopkins statistic (in d dimensions) for the ith pair of samples $\boldsymbol{R}_i$ and $\boldsymbol{D}_i$ is then defined as

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

If the data is well clustered we expect $\delta_{\min}(\boldsymbol{x}_j)$ values to be smaller compared to the $\delta_{\min}(\boldsymbol{y}_j)$ values, and in this case HS_i tends to 1.

Iris PCA Data $\times$ Uniform: Hopkins Statistic Distribution

Number of sample pairs $t = 500$, subsample size $m = 30$.



The Hopkins statistic has $\mu_{HS} = 0.935$ and $\sigma_{HS} = 0.025$.

Given the high value of the statistic, we conclude that the Iris dataset has a good clustering tendency.

Data Mining and Machine Learning: Fundamental Concepts and Algorithms

dataminingbook.info

Mohammed J. Zaki[1] Wagner Meira Jr.[2]

[1]Department of Computer Science
Rensselaer Polytechnic Institute, Troy, NY, USA

[2]Department of Computer Science
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Chapter 17: Clustering Validation