Multivariate Verfahren 2

cluster analysis

Helmut Waldl

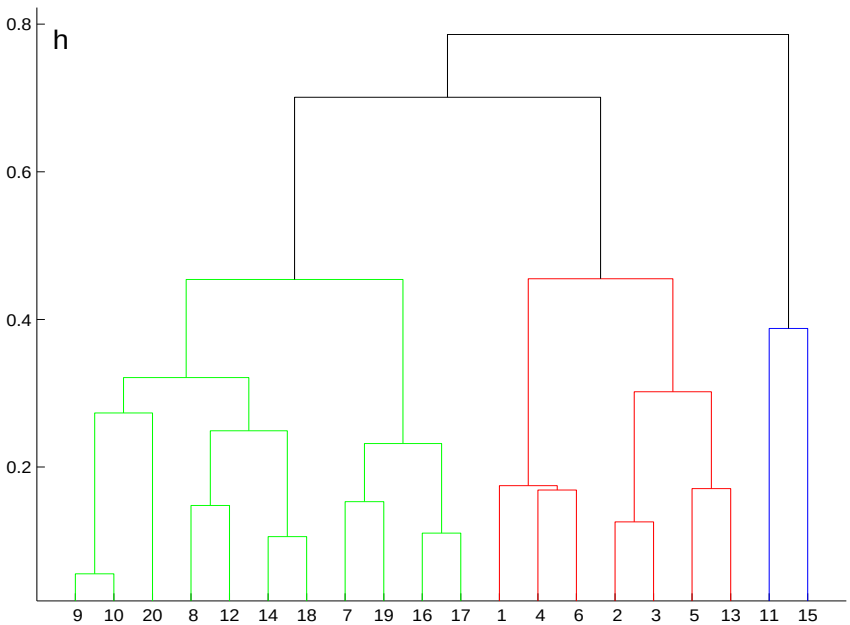
June 18th and 19th 2012

cluster analysis

hierarchical classification

A sequence of partitions of the set of objects $I = \{1, \ldots, N\}$ is generated. Thereby the number of classes is decreased (agglomerative methods) or increased (divisive methods) step by step. An increase in the number of classes involves an increase in the homogeneity within the classes.

With agglomerative clustering the classes are joined successively, with divisive methods the classes are partitioned successively.



Representation of the clustering methods with a **dendogram**

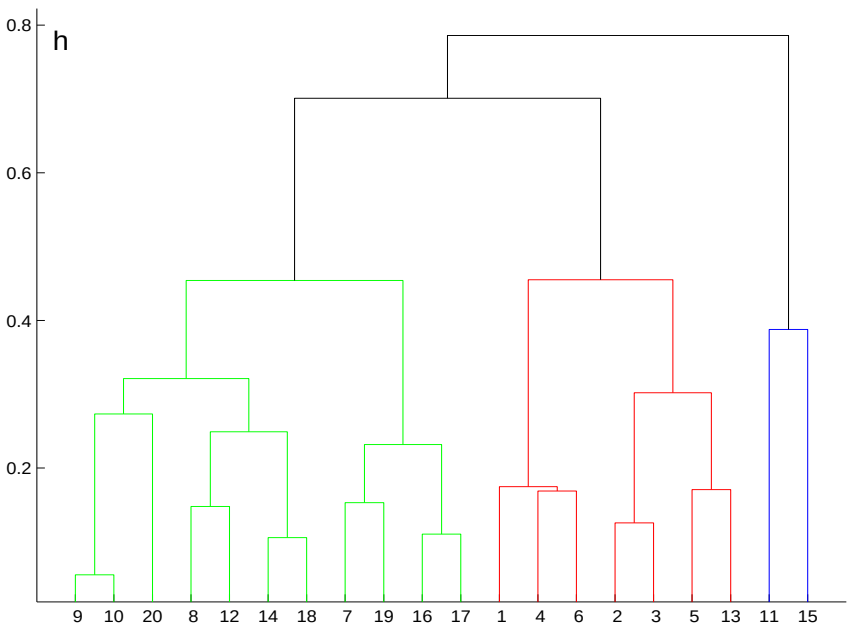
h measures the homogeneity of the classes. The smaller h is at the merging of two classes, the more similar the objects are in the classes.

cluster analysis

hierarchical classification

A sequence of partitions of the set of objects $I = \{1, \ldots, N\}$ is generated. Thereby the number of classes is decreased (agglomerative methods) or increased (divisive methods) step by step. An increase in the number of classes involves an increase in the homogeneity within the classes.

With agglomerative clustering the classes are joined successively, with divisive methods the classes are partitioned successively.



Representation of the clustering methods with a **dendogram**

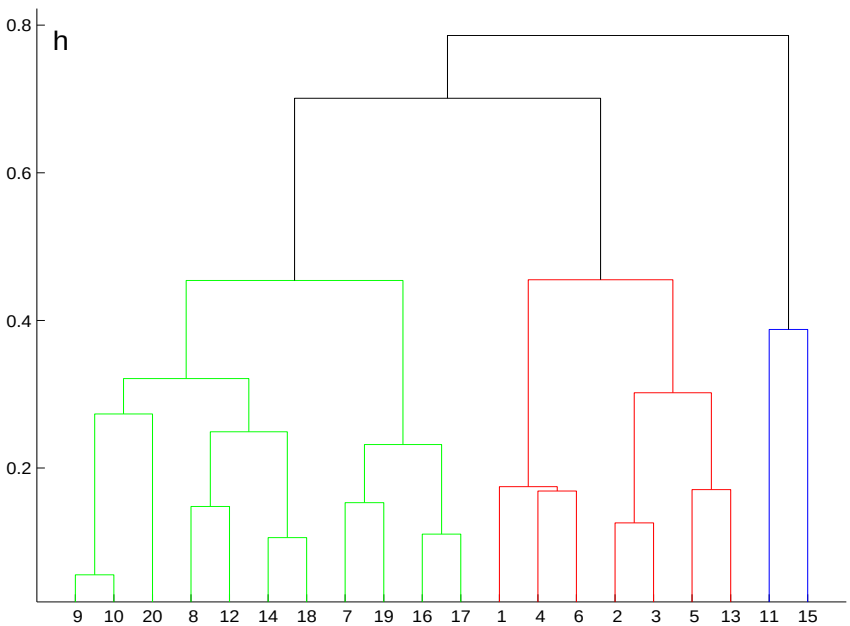
h measures the homogeneity of the classes. The smaller h is at the merging of two classes, the more similar the objects are in the classes.

cluster analysis

hierarchical classification

A sequence of partitions of the set of objects $I = \{1, \ldots, N\}$ is generated. Thereby the number of classes is decreased (agglomerative methods) or increased (divisive methods) step by step. An increase in the number of classes involves an increase in the homogeneity within the classes.

With agglomerative clustering the classes are joined successively, with divisive methods the classes are partitioned successively.



Representation of the clustering methods with a **dendogram**

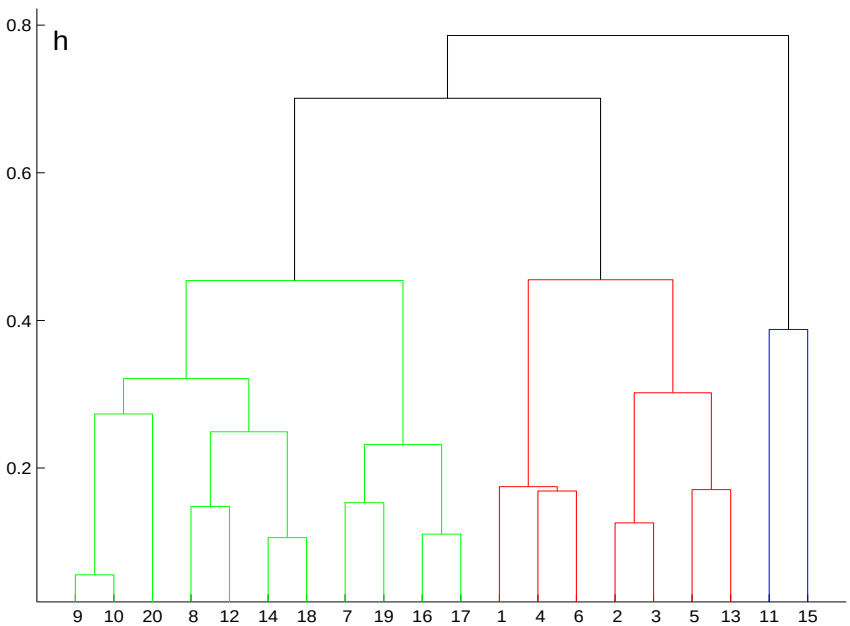
h measures the homogeneity of the classes. The smaller h is at the merging of two classes, the more similar the objects are in the classes.

cluster analysis

hierarchical classification

A sequence of partitions of the set of objects $I = \{1, \ldots, N\}$ is generated. Thereby the number of classes is decreased (agglomerative methods) or increased (divisive methods) step by step. An increase in the number of classes involves an increase in the homogeneity within the classes.

With agglomerative clustering the classes are joined successively, with divisive methods the classes are partitioned successively.



Representation of the clustering methods with a **dendogram**

h measures the homogeneity of the classes. The smaller h is at the merging of two classes, the more similar the objects are in the classes.

cluster analysis

hierarchical classification

Advantage of hierarchical methods: We don't have to provide the number of classes, we also do not need to assess a homogeneity-limit.

Different classification methods and distance measures obviously generate different dendograms.

Hierarchies

Let $I = \{1, \ldots, N\} = \{I_1, \ldots, I_N\}$, $\mathcal{P}_*(I) \ldots$ power set of I without $\{\}$

Definition: A system of sets $\mathcal{H} \subset \mathcal{P}_*(I)$ is called **hierarchy** of I, if for 2 different sets $B, C \in \mathcal{H}$ exactly one of the following 3 possibilities is true:

$$B \cap C = \{\} \qquad \text{or} \qquad B \subset C \qquad \text{or} \qquad C \subset B$$

Definition: A hierarchy $\mathcal{H}$ that contains both the set of objects I and all objects $I_1, \ldots, I_N$ as classes is called **total hierarchy**

cluster analysis

hierarchical classification

Advantage of hierarchical methods: We don't have to provide the number of classes, we also do not need to assess a homogeneity-limit.

Different classification methods and distance measures obviously generate different dendograms.

Hierarchies

Let $I = \{1, \ldots, N\} = \{I_1, \ldots, I_N\}$, $\mathcal{P}_*(I) \ldots$ power set of I without $\{\}$

Definition: A system of sets $\mathcal{H} \subset \mathcal{P}_*(I)$ is called **hierarchy** of I, if for 2 different sets $B, C \in \mathcal{H}$ exactly one of the following 3 possibilities is true:

$$B \cap C = \{\} \qquad \text{or} \qquad B \subset C \qquad \text{or} \qquad C \subset B$$

Definition: A hierarchy $\mathcal{H}$ that contains both the set of objects I and all objects $I_1, \ldots, I_N$ as classes is called **total hierarchy**

cluster analysis

hierarchical classification

Advantage of hierarchical methods: We don't have to provide the number of classes, we also do not need to assess a homogeneity-limit.

Different classification methods and distance measures obviously generate different dendograms.

Hierarchies

Let $I = \{1, \ldots, N\} = \{I_1, \ldots, I_N\}$, $\mathcal{P}_*(I) \ldots$ power set of I without $\{\}$

Definition: A system of sets $\mathcal{H} \subset \mathcal{P}_*(I)$ is called **hierarchy** of I, if for 2 different sets $B, C \in \mathcal{H}$ exactly one of the following 3 possibilities is true:

$$B \cap C = \{\} \qquad \text{or} \qquad B \subset C \qquad \text{or} \qquad C \subset B$$

Definition: A hierarchy $\mathcal{H}$ that contains both the set of objects I and all objects $I_1, \ldots, I_N$ as classes is called **total hierarchy**

cluster analysis

hierarchical classification

Advantage of hierarchical methods: We don't have to provide the number of classes, we also do not need to assess a homogeneity-limit.

Different classification methods and distance measures obviously generate different dendograms.

Hierarchies

Let $I = \{1, \ldots, N\} = \{I_1, \ldots, I_N\}$, $\mathcal{P}_*(I) \ldots$ power set of I without $\{\}$

Definition: A system of sets $\mathcal{H} \subset \mathcal{P}_*(I)$ is called **hierarchy** of I, if for 2 different sets $B, C \in \mathcal{H}$ exactly one of the following 3 possibilities is true:

$$B \cap C = \{\} \qquad \text{or} \qquad B \subset C \qquad \text{or} \qquad C \subset B$$

Definition: A hierarchy $\mathcal{H}$ that contains both the set of objects I and all objects $I_1, \ldots, I_N$ as classes is called **total hierarchy**

cluster analysis

hierarchical classification

Advantage of hierarchical methods: We don't have to provide the number of classes, we also do not need to assess a homogeneity-limit.

Different classification methods and distance measures obviously generate different dendograms.

Hierarchies

Let $I = \{1, \ldots, N\} = \{I_1, \ldots, I_N\}$, $\mathcal{P}_*(I) \ldots$ power set of I without $\{\}$

Definition: A system of sets $\mathcal{H} \subset \mathcal{P}_*(I)$ is called **hierarchy** of I, if for 2 different sets $B, C \in \mathcal{H}$ exactly one of the following 3 possibilities is true:

$$B \cap C = \{\} \qquad \text{or} \qquad B \subset C \qquad \text{or} \qquad C \subset B$$

Definition: A hierarchy $\mathcal{H}$ that contains both the set of objects I and all objects $I_1, \ldots, I_N$ as classes is called **total hierarchy**

cluster analysis

hierarchical classification

Advantage of hierarchical methods: We don't have to provide the number of classes, we also do not need to assess a homogeneity-limit.

Different classification methods and distance measures obviously generate different dendograms.

Hierarchies

Let $I = \{1, \ldots, N\} = \{I_1, \ldots, I_N\}$, $\mathcal{P}_*(I) \ldots$ power set of I without $\{\}$

Definition: A system of sets $\mathcal{H} \subset \mathcal{P}_*(I)$ is called **hierarchy** of I, if for 2 different sets $B, C \in \mathcal{H}$ exactly one of the following 3 possibilities is true:

$$B \cap C = \{\} \qquad \text{or} \qquad B \subset C \qquad \text{or} \qquad C \subset B$$

Definition: A hierarchy $\mathcal{H}$ that contains both the set of objects I and all objects $I_1, \ldots, I_N$ as classes is called **total hierarchy**

cluster analysis

hierarchical classification

Definition: **Index of a hierarchy** $\mathcal{H}$: a nonnegative function defined for all classes $C \in \mathcal{H}$ with the properties:

$$h(C) \leq h(B) \qquad \Longleftarrow \qquad C \subseteq B$$

$$h(C) = 0 \qquad \Longleftrightarrow \qquad C \text{ has only equivalent objects } (x_n = x_m)$$

h measures the homogeneity of the classes. The smaller h the more homogeneous the classes.

There are classification methods (e.g. centroid method) where the measure of homogeneity is no index. Then also $h(C) > h(B)$ is possible if $C \subsetneq B$.

cluster analysis

hierarchical classification

Definition: **Index of a hierarchy** $\mathcal{H}$: a nonnegative function defined for all classes $C \in \mathcal{H}$ with the properties:

$$h(C) \leq h(B) \qquad \Longleftarrow \qquad C \subseteq B$$

$$h(C) = 0 \qquad \Longleftrightarrow \qquad C \text{ has only equivalent objects } (x_n = x_m)$$

h measures the homogeneity of the classes. The smaller h the more homogeneous the classes.

There are classification methods (e.g. centroid method) where the measure of homogeneity is no index. Then also $h(C) > h(B)$ is possible if $C \subsetneq B$.

cluster analysis

hierarchical classification

Definition: **Index of a hierarchy** $\mathcal{H}$: a nonnegative function defined for all classes $C \in \mathcal{H}$ with the properties:

$$h(C) \leq h(B) \qquad \Longleftarrow \qquad C \subseteq B$$

$$h(C) = 0 \qquad \Longleftrightarrow \qquad C \text{ has only equivalent objects } (x_n = x_m)$$

h measures the homogeneity of the classes. The smaller h the more homogeneous the classes.

There are classification methods (e.g. centroid method) where the measure of homogeneity is no index. Then also $h(C) > h(B)$ is possible if $C \subsetneq B$.

cluster analysis

agglomerative methods

Assumption: for arbitrary nonempty subsets of objects a distance measure D or a similarity measure S must be specified.

Algorithmic course:

1. $\nu = 0,\ C^0 = \{\{I_1\}, \ldots, \{I_N\}\}$ ... finest partition of I
2. We get the partition C^ν $(\nu \geq 1)$ from $C^{\nu-1}$ by merging those classes of $C^{\nu-1}$ that have minimal D or maximal S.
3. iterate 2) until $C^\nu = \{I\}$

Thereby a total hierarchy is generated. The hierarchy may be indexed in the following way:
To the class that is generated in the ν-th iteration by merging C_j and C_k the following index is assigned:

$$h_\nu = D_\nu = \min_{k \neq j} D(C_k, C_j) \qquad \nu \geq 1$$

$$\text{or} \qquad h_\nu = S_\nu = \max_{k \neq j} S(C_k, C_j)$$

cluster analysis

agglomerative methods

Assumption: for arbitrary nonempty subsets of objects a distance measure D or a similarity measure S must be specified.

Algorithmic course:

1. $\nu = 0,\ \mathcal{C}^0 = \{\{I_1\}, \ldots, \{I_N\}\}$ ... finest partition of I
2. We get the partition $\mathcal{C}^\nu$ $(\nu \geq 1)$ from $\mathcal{C}^{\nu-1}$ by merging those classes of $\mathcal{C}^{\nu-1}$ that have minimal D or maximal S.
3. iterate 2) until $\mathcal{C}^\nu = \{I\}$

Thereby a total hierarchy is generated. The hierarchy may be indexed in the following way:
To the class that is generated in the ν-th iteration by merging C_j and C_k the following index is assigned:

$$h_\nu = D_\nu = \min_{k \neq j} D(C_k, C_j) \qquad \nu \geq 1$$

$$\text{or} \qquad h_\nu = S_\nu = \max_{k \neq j} S(C_k, C_j)$$

cluster analysis

agglomerative methods

Assumption: for arbitrary nonempty subsets of objects a distance measure D or a similarity measure S must be specified.

Algorithmic course:

1. $\nu = 0$, $\mathcal{C}^0 = \{\{I_1\}, \ldots, \{I_N\}\}$ ... finest partition of I
2. We get the partition $\mathcal{C}^\nu$ ($\nu \geq 1$) from $\mathcal{C}^{\nu-1}$ by merging those classes of $\mathcal{C}^{\nu-1}$ that have minimal D or maximal S.
3. iterate 2) until $\mathcal{C}^\nu = \{I\}$

Thereby a total hierarchy is generated. The hierarchy may be indexed in the following way:
To the class that is generated in the ν-th iteration by merging C_j and C_k the following index is assigned:

$$h_\nu = D_\nu = \min_{k \neq j} D(C_k, C_j) \qquad \nu \geq 1$$

$$\text{or} \qquad h_\nu = S_\nu = \max_{k \neq j} S(C_k, C_j)$$

cluster analysis

agglomerative methods

Assumption: for arbitrary nonempty subsets of objects a distance measure D or a similarity measure S must be specified.

Algorithmic course:

1. $\nu = 0$, $\mathcal{C}^0 = \{\{I_1\}, \ldots, \{I_N\}\}$ ... finest partition of I
2. We get the partition $\mathcal{C}^\nu$ ($\nu \geq 1$) from $\mathcal{C}^{\nu-1}$ by merging those classes of $\mathcal{C}^{\nu-1}$ that have minimal D or maximal S.
3. iterate 2) until $\mathcal{C}^\nu = \{I\}$

Thereby a total hierarchy is generated. The hierarchy may be indexed in the following way:
To the class that is generated in the ν-th iteration by merging C_j and C_k the following index is assigned:

$$h_\nu = D_\nu = \min_{k \neq j} D(C_k, C_j) \qquad \nu \geq 1$$

$$\text{or} \qquad h_\nu = S_\nu = \max_{k \neq j} S(C_k, C_j)$$

cluster analysis

agglomerative methods

Assumption: for arbitrary nonempty subsets of objects a distance measure D or a similarity measure S must be specified.

Algorithmic course:

1. $\nu = 0$, $\mathcal{C}^0 = \{\{I_1\}, \ldots, \{I_N\}\}$ ... finest partition of I
2. We get the partition $\mathcal{C}^\nu$ ($\nu \geq 1$) from $\mathcal{C}^{\nu-1}$ by merging those classes of $\mathcal{C}^{\nu-1}$ that have minimal D or maximal S.
3. iterate 2) until $\mathcal{C}^\nu = \{I\}$

Thereby a total hierarchy is generated. The hierarchy may be indexed in the following way:
To the class that is generated in the ν-th iteration by merging C_j and C_k the following index is assigned:

$$h_\nu = D_\nu = \min_{k \neq j} D(C_k, C_j) \qquad \nu \geq 1$$

$$\text{or} \qquad h_\nu = S_\nu = \max_{k \neq j} S(C_k, C_j)$$

cluster analysis

agglomerative methods

Assumption: for arbitrary nonempty subsets of objects a distance measure D or a similarity measure S must be specified.

Algorithmic course:

1. $\nu = 0$, $\mathcal{C}^0 = \{\{I_1\}, \ldots, \{I_N\}\}$ ... finest partition of I
2. We get the partition $\mathcal{C}^\nu$ $(\nu \geq 1)$ from $\mathcal{C}^{\nu-1}$ by merging those classes of $\mathcal{C}^{\nu-1}$ that have minimal D or maximal S.
3. iterate 2) until $\mathcal{C}^\nu = \{I\}$

Thereby a total hierarchy is generated. The hierarchy may be indexed in the following way:
To the class that is generated in the ν-th iteration by merging C_j and C_k the following index is assigned:

$$h_\nu = D_\nu = \min_{k \neq j} D(C_k, C_j) \qquad \nu \geq 1$$

$$\text{or} \qquad h_\nu = S_\nu = \max_{k \neq j} S(C_k, C_j)$$

cluster analysis

agglomerative methods

Furthermore we set $h_0 = D_0 = 0$ or $h_0 = S_0 = \max_{n \in I}\{S_{nn}\}$

If in the above algorithm the pair of classes with minimal distance or maximal similarity is not unique then we merge the classes such that as many pairs as possible may be joined (but always merge just two classes).

Second possibility: all classes C_k, C_j are merged for which a chain of classes with minimal distance or maximal similarity exists, i.e.

$$D(C_k, C_{k_1}) = D(C_{k_1}, C_{k_2}) = \ldots = D(C_{k_t}, C_j) = \min_{C_1, C_2 \in \mathcal{C}^{\nu-1}} D(C_1, C_2)$$

$$\text{or} \quad S(C_k, C_{k_1}) = S(C_{k_1}, C_{k_2}) = \ldots = S(C_{k_t}, C_j) = \max_{C_1, C_2 \in \mathcal{C}^{\nu-1}} S(C_1, C_2)$$

cluster analysis

agglomerative methods

Furthermore we set $h_0 = D_0 = 0$ or $h_0 = S_0 = \max_{n \in I}\{S_{nn}\}$

If in the above algorithm the pair of classes with minimal distance or maximal similarity is not unique then we merge the classes such that as many pairs as possible may be joined (but always merge just two classes).

Second possibility: all classes C_k, C_j are merged for which a chain of classes with minimal distance or maximal similarity exists, i.e.

$$D(C_k, C_{k_1}) = D(C_{k_1}, C_{k_2}) = \ldots = D(C_{k_t}, C_j) = \min_{C_1, C_2 \in \mathcal{C}^{\nu-1}} D(C_1, C_2)$$

$$\text{or} \quad S(C_k, C_{k_1}) = S(C_{k_1}, C_{k_2}) = \ldots = S(C_{k_t}, C_j) = \max_{C_1, C_2 \in \mathcal{C}^{\nu-1}} S(C_1, C_2)$$

cluster analysis

agglomerative methods

Furthermore we set $h_0 = D_0 = 0$ or $h_0 = S_0 = \max_{n \in I}\{S_{nn}\}$

If in the above algorithm the pair of classes with minimal distance or maximal similarity is not unique then we merge the classes such that as many pairs as possible may be joined (but always merge just two classes).

Second possibility: all classes C_k, C_j are merged for which a chain of classes with minimal distance or maximal similarity exists, i.e.

$$D(C_k, C_{k_1}) = D(C_{k_1}, C_{k_2}) = \ldots = D(C_{k_t}, C_j) = \min_{C_1, C_2 \in \mathcal{C}^{\nu-1}} D(C_1, C_2)$$

$$\text{or} \quad S(C_k, C_{k_1}) = S(C_{k_1}, C_{k_2}) = \ldots = S(C_{k_t}, C_j) = \max_{C_1, C_2 \in \mathcal{C}^{\nu-1}} S(C_1, C_2)$$

cluster analysis

agglomerative methods

Recursion formula for the computation of class distances for new classes:

C_v and C_w are merged to class C, then

$$D(C, C_k) = \alpha_v \cdot D(C_v, C_k) + \alpha_w \cdot D(C_w, C_k) + \beta \cdot D(C_v, C_w) +$$

$$+ \gamma \cdot |D(C_v, C_k) - D(C_w, C_k)|$$

If $\alpha_v + \alpha_w + \beta = 1$ the distance measure D may be replaced by a similarity measure S.

Recursion formula for the computation of class distances for new classes:

C_v and C_w are merged to class C, then

$$D(C, C_k) = \alpha_v \cdot D(C_v, C_k) + \alpha_w \cdot D(C_w, C_k) + \beta \cdot D(C_v, C_w) +$$

$$+ \gamma \cdot |D(C_v, C_k) - D(C_w, C_k)|$$

If $\alpha_v + \alpha_w + \beta = 1$ the distance measure D may be replaced by a similarity measure S.

cluster analysis

agglomerative methods

Recursion formula for the computation of class distances for new classes:

C_v and C_w are merged to class C, then

$$D(C, C_k) = \alpha_v \cdot D(C_v, C_k) + \alpha_w \cdot D(C_w, C_k) + \beta \cdot D(C_v, C_w) +$$

$$+ \gamma \cdot |D(C_v, C_k) - D(C_w, C_k)|$$

If $\alpha_v + \alpha_w + \beta = 1$ the distance measure D may be replaced by a similarity measure S.

cluster analysis

special agglomerative methods

Single linkage clustering (nearest-neighbor clustering, minimum distance method)

Recursion parameters for the computation of distances for new classes:

$$\alpha_v = \alpha_w = \frac{1}{2} \qquad \beta = 0 \qquad \gamma = -\frac{1}{2}$$

The distance between two classes C_j, C_k equals the minimum distance between an object in C_j and an object in C_k:

$$D(C_k, C_j) = \min_{\substack{n \in C_k \\ m \in C_j}} \{d_{nm}\}$$

In the ν-th iteration the classes $C_v, C_w \in \mathcal{C}^{\nu-1}$ are merged for which we have

$$h_\nu = D_\nu = D(C_v, C_w) = \min_{k \neq j} \min_{\substack{n \in C_j \\ m \in C_k}} \{d_{nm}\}$$

cluster analysis

special agglomerative methods

Single linkage clustering (nearest-neighbor clustering, minimum distance method)

Recursion parameters for the computation of distances for new classes:

$$\alpha_v = \alpha_w = \frac{1}{2} \qquad \beta = 0 \qquad \gamma = -\frac{1}{2}$$

The distance between two classes C_j, C_k equals the minimum distance between an object in C_j and an object in C_k:

$$D(C_k, C_j) = \min_{\substack{n \in C_k \\ m \in C_j}} \{d_{nm}\}$$

In the ν-th iteration the classes $C_v, C_w \in \mathcal{C}^{\nu-1}$ are merged for which we have

$$h_\nu = D_\nu = D(C_v, C_w) = \min_{k \neq j} \min_{\substack{n \in C_j \\ m \in C_k}} \{d_{nm}\}$$

cluster analysis

special agglomerative methods

Single linkage clustering (nearest-neighbor clustering, minimum distance method)

Recursion parameters for the computation of distances for new classes:

$$\alpha_v = \alpha_w = \frac{1}{2} \qquad \beta = 0 \qquad \gamma = -\frac{1}{2}$$

The distance between two classes C_j, C_k equals the minimum distance between an object in C_j and an object in C_k:

$$D(C_k, C_j) = \min_{\substack{n \in C_k \\ m \in C_j}} \{d_{nm}\}$$

In the ν-th iteration the classes $C_v, C_w \in \mathcal{C}^{\nu-1}$ are merged for which we have

$$h_\nu = D_\nu = D(C_v, C_w) = \min_{k \neq j} \min_{\substack{n \in C_j \\ m \in C_k}} \{d_{nm}\}$$

cluster analysis

special agglomerative methods

Single linkage clustering (nearest-neighbor clustering, minimum distance method)

Recursion parameters for the computation of distances for new classes:

$$\alpha_v = \alpha_w = \frac{1}{2} \qquad \beta = 0 \qquad \gamma = -\frac{1}{2}$$

The distance between two classes C_j, C_k equals the minimum distance between an object in C_j and an object in C_k:

$$D(C_k, C_j) = \min_{\substack{n \in C_k \\ m \in C_j}} \{d_{nm}\}$$

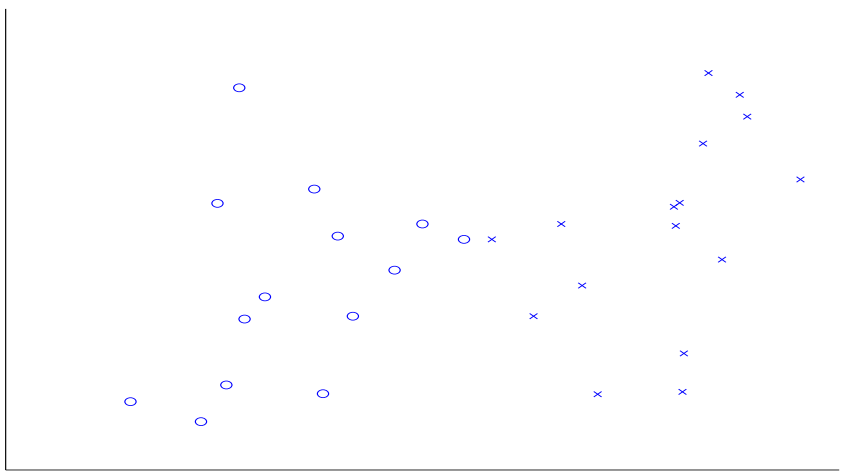
In the ν-th iteration the classes $C_v, C_w \in \mathcal{C}^{\nu-1}$ are merged for which we have

$$h_\nu = D_\nu = D(C_v, C_w) = \min_{k \neq j} \min_{\substack{n \in C_j \\ m \in C_k}} \{d_{nm}\}$$

cluster analysis

special agglomerative methods

With this method classes are also merged if only one single object of one class is close to a single object of the other class. The other objects may be a long way away from each other.

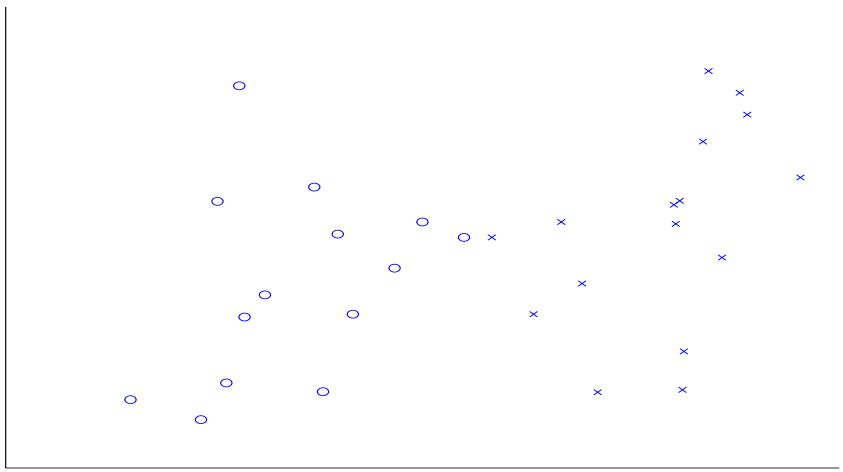


disadvantage: Classes that are connected by "bridges" are not detected ("generation of chains")

cluster analysis

special agglomerative methods

With this method classes are also merged if only one single object of one class is close to a single object of the other class. The other objects may be a long way away from each other.



disadvantage: Classes that are connected by "bridges" are not detected ("generation of chains")

cluster analysis

special agglomerative methods

Complete linkage clustering (furthest-neighbor clustering, maximum distance method)

Recursion parameters for the computation of distances for new classes:

$$\alpha_v = \alpha_w = \frac{1}{2} \qquad \beta = 0 \qquad \gamma = \frac{1}{2}$$

The distance between two classes C_j, C_k equals the maximum distance between an object in C_j and an object in C_k:

$$D(C_k, C_j) = \max_{\substack{n \in C_k \\ m \in C_j}} \{d_{nm}\}$$

cluster analysis

special agglomerative methods

Complete linkage clustering (furthest-neighbor clustering, maximum distance method)

Recursion parameters for the computation of distances for new classes:

$$\alpha_v = \alpha_w = \frac{1}{2} \qquad \beta = 0 \qquad \gamma = \frac{1}{2}$$

The distance between two classes C_j, C_k equals the maximum distance between an object in C_j and an object in C_k:

$$D(C_k, C_j) = \max_{\substack{n \in C_k \\ m \in C_j}} \{d_{nm}\}$$

cluster analysis

special agglomerative methods

Complete linkage clustering (furthest-neighbor clustering, maximum distance method)

Recursion parameters for the computation of distances for new classes:

$$\alpha_v = \alpha_w = \frac{1}{2} \qquad \beta = 0 \qquad \gamma = \frac{1}{2}$$

The distance between two classes C_j, C_k equals the maximum distance between an object in C_j and an object in C_k:

$$D(C_k, C_j) = \max_{\substack{n \in C_k \\ m \in C_j}} \{d_{nm}\}$$

cluster analysis

special agglomerative methods

In the ν-th iteration the classes $C_v, C_w \in \mathcal{C}^{\nu-1}$ are merged for which we have

$$h_\nu = D_\nu = D(C_v, C_w) = \min_{k \neq j} \max_{\substack{n \in C_j \\ m \in C_k}} \{d_{nm}\}$$

All objects that are in the same class after the ν-th iteration have at most a distance of D_ν to each other.

As with single linkage clustering we do not need an index, the knowledge of the distance ranking suffices.

cluster analysis

special agglomerative methods

In the ν-th iteration the classes $C_v, C_w \in \mathcal{C}^{\nu-1}$ are merged for which we have

$$h_\nu = D_\nu = D(C_v, C_w) = \min_{k \neq j} \max_{\substack{n \in C_j \\ m \in C_k}} \{d_{nm}\}$$

All objects that are in the same class after the ν-th iteration have at most a distance of D_ν to each other.

As with single linkage clustering we do not need an index, the knowledge of the distance ranking suffices.

cluster analysis

special agglomerative methods

In the ν-th iteration the classes $C_v, C_w \in \mathcal{C}^{\nu-1}$ are merged for which we have

$$h_\nu = D_\nu = D(C_v, C_w) = \min_{k \neq j} \max_{\substack{n \in C_j \\ m \in C_k}} \{d_{nm}\}$$

All objects that are in the same class after the ν-th iteration have at most a distance of D_ν to each other.

As with single linkage clustering we do not need an index, the knowledge of the distance ranking suffices.

Average linkage clustering

Recursion parameters for the computation of distances for new classes:

$$\alpha_v = \frac{n_v}{n_v + n_w} \qquad \alpha_w = \frac{n_w}{n_v + n_w} \qquad \beta = \gamma = 0$$

where n_v is the number of objects in class C_v
The distance between two classes C_j, C_k equals the average of all distances between objects in C_j and C_k:

$$D(C_k, C_j) = \frac{1}{n_k \cdot n_j} \sum_{n \in C_k} \sum_{m \in C_j} d_{nm}$$

Average linkage clustering

Recursion parameters for the computation of distances for new classes:

$$\alpha_v = \frac{n_v}{n_v + n_w} \qquad \alpha_w = \frac{n_w}{n_v + n_w} \qquad \beta = \gamma = 0$$

where n_v is the number of objects in class C_v

The distance between two classes C_j, C_k equals the average of all distances between objects in C_j and C_k:

$$D(C_k, C_j) = \frac{1}{n_k \cdot n_j} \sum_{n \in C_k} \sum_{m \in C_j} d_{nm}$$

Average linkage clustering

Recursion parameters for the computation of distances for new classes:

$$\alpha_v = \frac{n_v}{n_v + n_w} \qquad \alpha_w = \frac{n_w}{n_v + n_w} \qquad \beta = \gamma = 0$$

where n_v is the number of objects in class C_v
The distance between two classes C_j, C_k equals the average of all distances between objects in C_j and C_k:

$$D(C_k, C_j) = \frac{1}{n_k \cdot n_j} \sum_{n \in C_k} \sum_{m \in C_j} d_{nm}$$

cluster analysis

special agglomerative methods

In the ν-th iteration the classes $C_v, C_w \in \mathcal{C}^{\nu-1}$ are merged for which we have

$$h_\nu = D_\nu = D(C_v, C_w) = \min_{k \neq j} \frac{1}{n_k \cdot n_j} \sum_{\substack{n \in C_k \\ m \in C_j}} d_{nm}$$

Effect of averaging: In one class there may be quite distant objects if this is compensated for many very close objects of the same class.

cluster analysis

special agglomerative methods

In the ν-th iteration the classes $C_v, C_w \in \mathcal{C}^{\nu-1}$ are merged for which we have

$$h_\nu = D_\nu = D(C_v, C_w) = \min_{k \neq j} \frac{1}{n_k \cdot n_j} \sum_{\substack{n \in C_k \\ m \in C_j}} d_{nm}$$

Effect of averaging: In one class there may be quite distant objects if this is compensated for many very close objects of the same class.

cluster analysis

special agglomerative methods

centroid method (all variates should be quantitative)

Recursion parameters for the computation of distances for new classes:

$$\alpha_v = \frac{n_v}{n_v + n_w} \qquad \alpha_w = \frac{n_w}{n_v + n_w} \qquad \beta = -\frac{n_v \cdot n_w}{n_v + n_w} \qquad \gamma = 0$$

Each class is represented by the class centroid $\bar{x}_k = \frac{1}{n_k} \sum_{n \in C_k} x_n$

The distance between two classes C_j, C_k equals the squared Euclidean distance of the class centroids.

In the ν-th iteration the classes $C_v, C_w \in \mathcal{C}^{\nu-1}$ are merged for which we have

$$h_\nu = D_\nu = D(C_v, C_w) = \min_{k \neq j} \|\bar{x}_j - \bar{x}_k\|^2$$

cluster analysis

special agglomerative methods

centroid method (all variates should be quantitative)

Recursion parameters for the computation of distances for new classes:

$$\alpha_v = \frac{n_v}{n_v + n_w} \qquad \alpha_w = \frac{n_w}{n_v + n_w} \qquad \beta = -\frac{n_v \cdot n_w}{n_v + n_w} \qquad \gamma = 0$$

Each class is represented by the class centroid $\bar{x}_k = \frac{1}{n_k} \sum_{n \in C_k} x_n$

The distance between two classes C_j, C_k equals the squared Euclidean distance of the class centroids.

In the ν-th iteration the classes $C_v, C_w \in \mathcal{C}^{\nu-1}$ are merged for which we have

$$h_\nu = D_\nu = D(C_v, C_w) = \min_{k \neq j} \|\bar{x}_j - \bar{x}_k\|^2$$

cluster analysis

special agglomerative methods

centroid method (all variates should be quantitative)

Recursion parameters for the computation of distances for new classes:

$$\alpha_v = \frac{n_v}{n_v + n_w} \qquad \alpha_w = \frac{n_w}{n_v + n_w} \qquad \beta = -\frac{n_v \cdot n_w}{n_v + n_w} \qquad \gamma = 0$$

Each class is represented by the class centroid $\bar{x}_k = \frac{1}{n_k} \sum_{n \in C_k} x_n$

The distance between two classes C_j, C_k equals the squared Euclidean distance of the class centroids.

In the ν-th iteration the classes $C_v, C_w \in \mathcal{C}^{\nu-1}$ are merged for which we have

$$h_\nu = D_\nu = D(C_v, C_w) = \min_{k \neq j} \|\bar{x}_j - \bar{x}_k\|^2$$

cluster analysis

special agglomerative methods

centroid method (all variates should be quantitative)

Recursion parameters for the computation of distances for new classes:

$$\alpha_v = \frac{n_v}{n_v + n_w} \qquad \alpha_w = \frac{n_w}{n_v + n_w} \qquad \beta = -\frac{n_v \cdot n_w}{n_v + n_w} \qquad \gamma = 0$$

Each class is represented by the class centroid $\bar{x}_k = \frac{1}{n_k} \sum_{n \in C_k} x_n$

The distance between two classes C_j, C_k equals the squared Euclidean distance of the class centroids.

In the ν-th iteration the classes $C_v, C_w \in \mathcal{C}^{\nu-1}$ are merged for which we have

$$h_\nu = D_\nu = D(C_v, C_w) = \min_{k \neq j} \|\bar{x}_j - \bar{x}_k\|^2$$

cluster analysis

special agglomerative methods

centroid method (all variates should be quantitative)

Recursion parameters for the computation of distances for new classes:

$$\alpha_v = \frac{n_v}{n_v + n_w} \qquad \alpha_w = \frac{n_w}{n_v + n_w} \qquad \beta = -\frac{n_v \cdot n_w}{n_v + n_w} \qquad \gamma = 0$$

Each class is represented by the class centroid $\bar{x}_k = \frac{1}{n_k} \sum_{n \in C_k} x_n$

The distance between two classes C_j, C_k equals the squared Euclidean distance of the class centroids.

In the ν-th iteration the classes $C_v, C_w \in \mathcal{C}^{\nu-1}$ are merged for which we have

$$h_\nu = D_\nu = D(C_v, C_w) = \min_{k \neq j} \|\bar{x}_j - \bar{x}_k\|^2$$

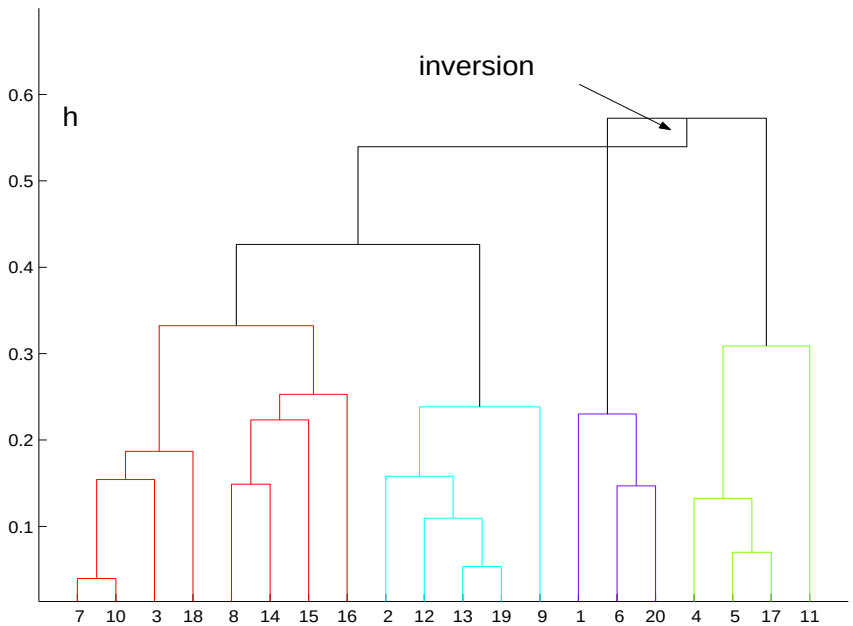
cluster analysis

special agglomerative methods

Here the index condition

$$C \subseteq B \qquad \Longrightarrow \qquad h(C) \leq h(B) \qquad \text{for } C, B \in \mathcal{H}$$

is not satisfied ($\mathcal{H}$ is a hierarchy).



It may happen that in the ν-th iteration the merged class $C_v \cup C_w$ is more homogeneous (measured in h_ν) than the classes C_v and C_w ("inversion").

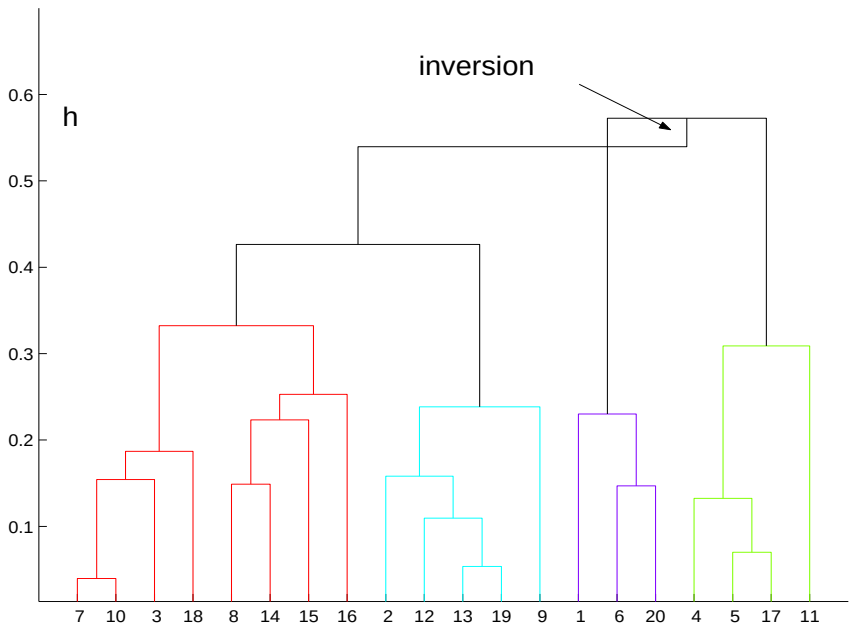
cluster analysis

special agglomerative methods

Here the index condition

$$C \subseteq B \qquad \Longrightarrow \qquad h(C) \leq h(B) \qquad \text{for } C, B \in \mathcal{H}$$

is not satisfied ($\mathcal{H}$ is a hierarchy).



It may happen that in the ν-th iteration the merged class $C_v \cup C_w$ is more homogeneous (measured in h_ν) than the classes C_v and C_w ("inversion").

cluster analysis

special agglomerative methods

Like with the average linkage clustering we may observe an effect of averaging. If we choose squared Euclidean distances with average linkage clustering then

$$d_{nm_{ALC}} = \frac{1}{n_k \cdot n_j} \sum_{\substack{n \in C_k \\ m \in C_j}} \|\bar{x}_k - \bar{x}_j\|^2 + s_k^2 + s_j^2$$

where s_k^2, s_j^2 are the variances in the classes C_k, C_j.

I.e. with average linkage clustering the distances of the centroids of two classes to be merged may be bigger than with the centroid method if the classes are very homogeneous.

cluster analysis

special agglomerative methods

Like with the average linkage clustering we may observe an effect of averaging. If we choose squared Euclidean distances with average linkage clustering then

$$d_{nm_{ALC}} = \frac{1}{n_k \cdot n_j} \sum_{\substack{n \in C_k \\ m \in C_j}} \|\bar{x}_k - \bar{x}_j\|^2 + s_k^2 + s_j^2$$

where s_k^2, s_j^2 are the variances in the classes C_k, C_j.

I.e. with average linkage clustering the distances of the centroids of two classes to be merged may be bigger than with the centroid method if the classes are very homogeneous.

cluster analysis

special agglomerative methods

Ward's method (quantitative variates)

Recursion parameters for the computation of distances for new classes:

$$\alpha_v = \frac{n_v + n_k}{n_v + n_w + n_k} \qquad \alpha_w = \frac{n_w + n_k}{n_v + n_w + n_k}$$

$$\beta = -\frac{n_k}{n_v + n_w + n_k} \qquad \gamma = 0$$

The homogeneity of a partition $\mathcal{C}^{\nu-1}$ is measured as the sum of the within class variances:

$$H(\mathcal{C}^{\nu-1}) = \sum_{k=1}^{g} \sum_{n \in C_k} \|x_n - \bar{x}_k\|^2$$

Ward's method (quantitative variates)

Recursion parameters for the computation of distances for new classes:

$$\alpha_v = \frac{n_v + n_k}{n_v + n_w + n_k} \qquad \alpha_w = \frac{n_w + n_k}{n_v + n_w + n_k}$$

$$\beta = -\frac{n_k}{n_v + n_w + n_k} \qquad \gamma = 0$$

The homogeneity of a partition $\mathcal{C}^{\nu-1}$ is measured as the sum of the within class variances:

$$H(\mathcal{C}^{\nu-1}) = \sum_{k=1}^{g} \sum_{n \in C_k} \|x_n - \bar{x}_k\|^2$$

cluster analysis

special agglomerative methods

Ward's method (quantitative variates)

Recursion parameters for the computation of distances for new classes:

$$\alpha_v = \frac{n_v + n_k}{n_v + n_w + n_k} \qquad \alpha_w = \frac{n_w + n_k}{n_v + n_w + n_k}$$

$$\beta = -\frac{n_k}{n_v + n_w + n_k} \qquad \gamma = 0$$

The homogeneity of a partition $\mathcal{C}^{\nu-1}$ is measured as the sum of the within class variances:

$$H(\mathcal{C}^{\nu-1}) = \sum_{k=1}^{g} \sum_{n \in C_k} \|x_n - \bar{x}_k\|^2$$

cluster analysis

special agglomerative methods

If $\mathcal{C}^\nu$ arises from $\mathcal{C}^{\nu-1}$ by combining the classes C_v and C_w to class C, then

$$H(\mathcal{C}^\nu) = \sum_{k \neq v,w} \sum_{n \in C_k} \|x_n - \bar{x}_k\|^2 + \sum_{n \in C} \|x_n - \bar{x}_c\|^2$$

The loss of homogeneity in the ν-th iteration should be minimal:

$$h_\nu = D(C_v, C_w) = \min_{k \neq j} \underbrace{\frac{n_k \cdot n_j}{n_k + n_j} \|\bar{x}_k - \bar{x}_j\|^2}_{D(C_k, C_j)}$$

The loss of homogeneity merging C_v and C_w in the ν-th iteration is:

$$H(\mathcal{C}^\nu) - H(\mathcal{C}^{\nu-1}) = \frac{n_v \cdot n_w}{n_v + n_w} \|\bar{x}_v - \bar{x}_w\|^2$$

cluster analysis

special agglomerative methods

If $\mathcal{C}^\nu$ arises from $\mathcal{C}^{\nu-1}$ by combining the classes C_v and C_w to class C, then

$$H(\mathcal{C}^\nu) = \sum_{k \neq v,w} \sum_{n \in C_k} \|x_n - \bar{x}_k\|^2 + \sum_{n \in C} \|x_n - \bar{x}_c\|^2$$

The loss of homogeneity in the ν-th iteration should be minimal:

$$h_\nu = D(C_v, C_w) = \min_{k \neq j} \underbrace{\frac{n_k \cdot n_j}{n_k + n_j} \|\bar{x}_k - \bar{x}_j\|^2}_{D(C_k, C_j)}$$

The loss of homogeneity merging C_v and C_w in the ν-th iteration is:

$$H(\mathcal{C}^\nu) - H(\mathcal{C}^{\nu-1}) = \frac{n_v \cdot n_w}{n_v + n_w} \|\bar{x}_v - \bar{x}_w\|^2$$

cluster analysis

special agglomerative methods

If $\mathcal{C}^\nu$ arises from $\mathcal{C}^{\nu-1}$ by combining the classes C_v and C_w to class C, then

$$H(\mathcal{C}^\nu) = \sum_{k \neq v,w} \sum_{n \in C_k} \|x_n - \bar{x}_k\|^2 + \sum_{n \in C} \|x_n - \bar{x}_c\|^2$$

The loss of homogeneity in the ν-th iteration should be minimal:

$$h_\nu = D(C_v, C_w) = \min_{k \neq j} \underbrace{\frac{n_k \cdot n_j}{n_k + n_j} \|\bar{x}_k - \bar{x}_j\|^2}_{D(C_k, C_j)}$$

The loss of homogeneity merging C_v and C_w in the ν-th iteration is:

$$H(\mathcal{C}^\nu) - H(\mathcal{C}^{\nu-1}) = \frac{n_v \cdot n_w}{n_v + n_w} \|\bar{x}_v - \bar{x}_w\|^2$$