G. Rohwer, C. Dudel Wintersemester 2010/11

Methoden der Datenrepräsentation und Klassifikation

Kapitel 6: Hierarchische Klassifikation

6 Hierarchische Klassifikation

In diesem Kapitel besprechen wir hierarchische Klassifikationsverfahren, die nicht unmittelbar eine bestimmte Partition erzeugen, sondern deren Zweck primär darin besteht, Einsichten in die Struktur einer Abstandsmatrix zu gewinnen. Erst in einem nachträglichen Schritt können dann auch Partitionen gebildet werden.[1] Wir beginnen im ersten Abschnitt mit divisiven (zerlegenden) Verfahren, bei denen die Gesamtmenge der Objekte sukzessive in Teilmengen zerlegt wird. Dann werden im zweiten Abschnitt agglomerativen Verfahren besprochen, bei denen die einzelnen Objekte sukzessive zu Clustern zusammengefasst werden. Schließlich wird im dritten Abschnitt gezeigt, dass hierarchische Klassifikationsverfahren auch als Methoden zur Konstruktion von repräsentierenden Modellen für Abstandsmatrizen verstanden werden können. Diese Betrachtungsweise führt zur Konstruktion sogenannter ultrametrischer Baummodelle.

[1] Übersichten zu den verschiedenen hierarchischen Methoden findet man u.a. bei Gordon (1987); Blashfield und Aldenderfer (1988); Jain und Dubes (1988: 58ff.). Einen neueren Überblick, der auch Erweiterungen des hierarchischen Ansatzes einbezieht, geben Barthélemy, Brucker und Osswald (2007).

Notationen. $\Omega = \{\omega_1, \ldots, \omega_n\}$ ist die Menge der Objekte, die von beliebiger Art sein können. Es wird angenommen, dass eine Abstandsmatrix $\mathbf{D} = (d_{ij})$ gegeben ist, die für jeweils zwei Elemente $\omega_i, \omega_j \in \Omega$ einen Abstand d_{ij} angibt. Für Teilmengen von Ω, die als Cluster betrachtet werden können, wird meistens der Buchstabe C, für Partitionen wird der Buchstabe P verwendet.

6.1 Divisive Verfahren

In diesem Abschnitt besprechen wir einige divisive Verfahren, bei denen die Gesamtmenge der Objekte sukzessive in Teilmengen zerlegt wird. Um den Ansatz zu illustrieren, beginnen wir mit einer Methode, die sich an der Idee orientiert, dass die jeweils gebildeten Cluster einen möglichst kleinen Durchmesser haben sollen. Dann folgen Hinweise auf einige andere divisive Verfahren der Clusteranalyse.[2]

1. Cluster mit minimalen Durchmessern

Zur Erinnerung: Wenn ein Cluster $C \subseteq \Omega$ gegeben ist, wird $d(C) := \max\{d_{ij} \,|\, \omega_i, \omega_j \in C\}$ als Durchmesser des Clusters bezeichnet. Ein divisives Verfahren, das sich an Clusterdurchmessern orientiert, kann folgendermaßen beschrieben werden:[3]

(1) Setze $k = 1$ und bilde die Partition $P_k = \{\Omega\}$.

(2) Bestimme dasjenige Cluster $C_k \in P_k$, das den größten Durchmesser hat.

(3) Aufteilung von C_k in zwei disjunkte Teilmengen $C_{k'}$ und $C_{k''}$, so dass $\max\{d(C_{k'}), d(C_{k''})\}$ minimal ist.

(4) Bilde aus P_k eine neue Partition P_{k+1}, indem anstelle von C_k die beiden Teilmengen $C_{k'}$ und $C_{k''}$ verwendet werden.

(5) Setze $k \leftarrow k + 1$; dann Fortsetzung bei (2).

Zur Illustration verwenden wir die Abstandsmatrix aus Tabelle **??**-3 für die Berufsstrukturdaten. Box 6.1-1 zeigt das Ergebnis, das mit der TDA-Prozedur `hcld` erzeugt wurde.[4] Im ersten Schritt wird die Ausgangsmenge (das Gesamtcluster C_1) in zwei Cluster aufgeteilt: in C_2, das zwei Elemente enthält und einen Durchmesser $d(C_2) = 0.1551$ hat, und in C_3, das 6

[2] Es gibt zahlreiche unterschiedliche Verfahren. Man vgl. etwa Guénoche, Hansen und Jaumard (1991); Hansen, Jaumard und Mladenovic (1998).

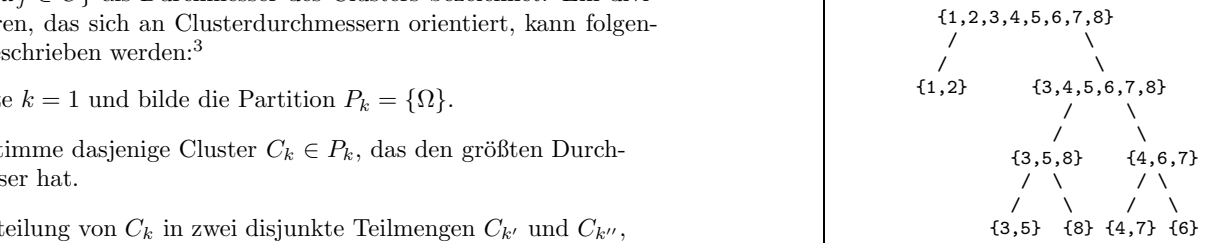
[3] Eine Diskussion von Rechenverfahren für dieses Verfahren findet man bei Guénoche, Hansen und Jaumard (1991).

[4] Das TDA-Skript ist `hcld2.cf`.

Box 6.1-1 Ergebnis einer divisiven Klassifikation der Berufsstrukturdaten mit der TDA-Prozedur `hcld`.

l	k	|C_l|	|C_k|	d(C_l)	d(C_k)	Elemente von C_l
2	1	2	8	0.1551	0.4230	1 2
3	1	6	8	0.2624	0.4230	3 4 5 6 7 8
4	3	3	6	0.1457	0.2624	3 5 8
5	3	3	6	0.1769	0.2624	4 6 7
6	5	2	3	0.1002	0.1769	4 7
7	5	1	3	0.0000	0.1769	6
8	2	1	2	0.0000	0.1551	1
9	2	1	2	0.0000	0.1551	2
10	4	2	3	0.0520	0.1457	3 5
11	4	1	3	0.0000	0.1457	8
12	6	1	2	0.0000	0.1002	4
13	6	1	2	0.0000	0.1002	7
14	10	1	2	0.0000	0.0520	3
15	10	1	2	0.0000	0.0520	5

Hierarchische Struktur



Elemente enthält und einen Durchmesser $d(C_3) = 0.2624$ hat. Die Tabelle zeigt dann, wie der Aufteilungsprozess fortgesetzt wird, bis schließlich nur noch einelementige Cluster vorhanden sind.

2. Varianten divisiver Verfahren

Es gibt viele mehr oder weniger unterschiedliche Varianten divisiver Verfahren der hierarchischen Klassifikation. In allen Varianten werden ausgehend von einem Ausgangscluster, das alle Objekte umfasst, bisher gebildete Cluster in jeweils zwei neue Cluster aufgeteilt. Dafür gibt es zwei verschiedene Möglichkeiten: Man kann in jedem Schritt eins der bisher gebildeten Cluster auswählen, um es in zwei neue Cluster aufzuteilen; so wird beispielsweise bei dem in Abschnitt 6.1-1 besprochenen Verfahren

vorgegangen. Oder man kann in jedem Schritt alle der bisher gebildeten Cluster in zwei neue Cluster aufteilen.[5]

Weitere Unterschiede betreffen die Art und Weise, wie die Aufteilung eines Clusters in zwei neue Cluster vorgenommen wird. Wie man hierbei vorgehen kann, hängt insbesondere davon ab, ob von einer bereits definierten Abstandsmatrix ausgegangen wird oder von Variablen, durch die die Objekte charakterisiert werden.

Ausgangspunkt Abstandsmatrix. In diesem Fall kann man beispielsweise anstreben, dass Cluster mit möglichst kleinen Durchmessern entstehen sollen (wie bei dem Verfahren in Abschnitt 6.1-1). Ein weiteres, damit meistens konkurrierendes Kriterium bezieht sich darauf, dass möglichst gut separierte Cluster entstehen sollen.[6] Es ist auch ein Verfahren vorgeschlagen worden, das beide Kriterien kombiniert.[7]

Ausgangspunkt Variablen. Bei diesem Fall geht man davon aus, dass jedes Objekt ω_i durch Werte von m Variablen, also durch einen Vektor $\mathbf{x}_i = (x_{i1}, \ldots, x_{im})'$, charakterisiert wird. Dann gibt es zunächst folgende Alternative:

a) Monothetische Verfahren: Für die Aufteilung eines Clusters werden jeweils nur die Werte einer Variablen verwendet. Eine besonders einfache Variante entsteht, wenn alle Variablen binär sind. Dann wird jeweils eine Variable ausgewählt, deren Werte unmittelbar eine Aufteilung in zwei Teilcluster liefern. Eine Variante diese Verfahrens wird in Abschnitt 6.1-4 besprochen.

b) Polythetische Verfahren: Für die Aufteilung eines Clusters werden die Werte aller Variablen verwendet. Oft orientiert man sich dann an euklidischen Abständen und verwendet für die Aufteilung eines Clusters C in zwei Teilcluster C_1 und C_2 folgendes Varianzkriterium:

$$\sum_{k=1}^{2} \sum_{\omega_i \in C_k} ||\mathbf{x}_i - \bar{\mathbf{x}}_k||^2$$

Hierbei ist $\bar{\mathbf{x}}_k$ der Mittelwert der Variablen im Cluster C_k. Die Cluster werden dann so gebildet, dass dieses Kriterium möglichst klein wird.[8]

[5] Man vgl. beispielsweise Edwards und Cavalli-Sforza (1965).

[6] Dabei wird die Separiertheit von zwei Clustern C' und C'' durch $\min\{d_{ij} | \omega_i \in C', \omega_j \in C''\}$ erfasst.

[7] Vgl. Wang, Yan und Sriskandarajah (1996).

[8] Man vgl. beispielsweise Edwards und Cavalli-Sforza (1965). Varianten dieses Ansatzes bespricht Späth (1975: 147ff.).

3. Divisive Verfahren in R

Im Paket `cluster` gibt es zwei Befehle für divisive Verfahren: `diana` und `mona`. `diana` basiert auf einem Algorithmus, der relativ analog zu dem in Abschnitt 6.1-1 beschriebenen Verfahren funktioniert, allerdings wird bei Schritt (3) versucht, die Teilmengen so zu bestimmen, dass die Beiträge einzelner Objekte zu den durchschnittlichen Abständen in den Teilmengen möglichst klein werden[9] Ein Beispiel mit den Berufsstrukturdaten findet man in Box 6.1-2.

Zunächst werden die Berufsstrukturdaten geladen und Abstände zwischen den Ländern wie in Tabelle ??-3 berechnet. Anschließend wird dem Befehl `diana` die unter dem Namen `d` abgespeicherte Abstandsmatrix übergeben. Die Resultate sind unter dem Namen `cl` im Speicher abgelegt. Bevor die Ergebnisse über den Aufruf von `cl` angezeigt werden, werden den Objekten (Ländern) noch Namen zugeordnet. Beim Aufruf von `cl` wird unter dem Punkt `Merge` der Aufteilungsprozess als Matrix ausgewiesen, wobei die Darstellung deutlich von der in Box 6.1-1 abweicht. Zunächst ist zu beachten, dass der Prozess mit der letzten Zeile der Matrix beginnt und mit der ersten endet. Positive Einträge der Matrix stehen für Cluster, negative für einzelne Beobachtungen. In Zeile 7 der Matrix – also beim ersten Aufteilungsschritt – werden die Cluster 3 und 6 getrennt. Die zu den Clustern gehörenden Objekte ergeben sich aus zu den Clusternummern korrespondierenden Zeilen. Cluster 3 enthält beispielsweise die Objekte 1 (Türkei) und 2 (Griechenland). Das Cluster 6 enthält das Objekt 6 und das Cluster 5, welches wiederum Objekt 8 und Cluster 4 enthält. Diese Darstellung ist zwar relativ umständlich, orientiert sich aber an der Darstellung für agglomerative Verfahren, wodurch die Ergebnisse einfacher vergleichbar werden.

Unter dem Punkt `Height` sind die Clusterdurchmesser angegeben und unter dem Punkt `Order of Objects` eine Reihung der Objekte, wobei man diese beiden Punkte zusammen interpretieren sollte. Insgesamt sind sieben Durchmesser angegeben, wobei der i-te Durchmesser zwischen dem i-ten und dem $i+1$-ten Objekt der Ordnung anzuordnen ist. Der i-te Durchmesser ist dann genau der Durchmesser des Clusters, welches zum einen in die Objekte bis i, zum anderen in die Ojekte ab $i+1$ zerlegt wird. Dabei findet man beim ersten Aufteilungsschritt den größten Durchmesser und beim letzten den kleinsten. Der gröte Durchmesser beträgt im Beispiel 0.42 und liegt zwischen den Objekten 2 und 3. Dies bedeutet, dass beim ersten Aufteilungsschritt die Objekte in die beiden Cluster mit den Objekten $\{1, 2\}$ und $\{3, 4, 5, 6, 7, 8\}$ aufgeteilt werden und dass das Cluster, aus dem diese beiden gebildet werden, einen Durchmesser von 0.42 hat. Der zweitgrößte Durchmesser beträgt 0.26 und liegt zwischen den Objekten 8 und 6. Dies bedeutet, dass beim zweiten Aufteilungsschritt

[9] Vgl. Kaufman und Rousseeuw (1990: 271ff.).

Box 6.1-2 R-Befehle für eine divisive Klassifikation mit Berufsstrukturdaten.

```
> dat <- read.table("bs1.dat")
> names(dat) <- c("X","Y","Z","h")

> tab <- xtabs(h~.,dat)
> tab <- ftable(tab,row.vars="X",col.vars=c("Y","Z"))
> tab <- prop.table(tab,1)
> d <- dist(tab,method="manhattan")*0.5

> library(cluster)
> cl <- diana(d)

> cl$order.lab <- c ("Turkey","Greece"," Switzerland",
+                    "Great Britain","Germany","Sweden",
+                   "USA"," Japan")

> cl
Merge:
     [,1] [,2]
[1,]   -3   -5
[2,]   -4   -7
[3,]   -1   -2
[4,]    1    2
[5,]    4   -8
[6,]    5   -6
[7,]    3    6

Order of objects:
[1] Turkey        Greece         Switzerland  Great Britain Germany
[6] Sweden        USA            Japan

Height:
[1] 0.15517386 0.42311450 0.05213896 0.16966925 0.10016099 0.21345035
    0.26242049

Divisive coefficient:
[1] 0.6777413

Available components:
[1] "order"  "height" "dc"     "merge"  "diss"   "call"

> cutree(cl,k=2)
[1] 1 1 2 2 2 2 2 2
```

das Cluster mit den Objekten 3 bis 8 in zwei Cluster aufgeteilt wird, von denen eines das Objekt 6 enthält und das andere die übrigen Objekte. Der drittgrößte Durchmesser liegt zwischen den Objekten 7 und 8. Im dritten Schritt wird also das Cluster mit den Objekten 3, 4, 5, 7 und 8 aufgeteilt, wobei Objekt 8 in einem Cluster ist und die übrigen Objekte in dem anderen. Der unter `Divisive coefficient` angegebene Kennwert

ist $DC = \frac{1}{n}\sum_{i=1}^{n} 1 - \frac{l(\omega_i)}{\max\{\mathbf{D}\}}$, wobei $l(\omega_i)$ für ein Objekt ω_i den Durchmesser des letzten Clusters angibt, zu dem dieses Objekt gehört, bevor es als einziges Objekt von diesem Cluster getrennt wird. Beispielsweise ist für die Berufsstrukturdaten $l(\omega_6) = 0.26$. Die Werte fr diesen Koeffizienten liegen zwischen 0 und 1. Werte nahe bei 1 sollen ausdrücken, dass sich die Objekte „gut“in Cluster unterteilen lassen.

Schließlich wird unter dem Punkt `Available components` ausgegeben, welche Resultate sich unter Verwendung des Objektnamens in Kombination mit dem $-Operator anzeigen lassen. Beispielsweise kann man mittels `cl$dc` den Wert von DC aufrufen.

Mit dem Befehls `cutree` lässt sich die Zuordnung von Objekten zu Clustern bei einer Partition in genau k Cluster anzeigen. Dem Befehl wird zum einen das R-Objekt `cl` übergeben, zum anderen wird mittels des Arguments `k` festgelegt, welche Partition ausgegeben werden soll. Bei $k = 2$ sind die ersten beiden Objekte zusammen in einem Cluster mit der Nummer 1 und die übrigen Objekte in einem Cluster mit der Nummter 2. Die Zuweisung von Nummern zu Clustern hat in diesem Kontext keine Bedeutung und dient lediglich einer einfachen Darstellung.

4. Ein monothetisches Verfahren

Jetzt besprechen wir ein monothetisches Verfahren, bei dem angenommen wird, dass alle Variablen $X_1, \ldots, X_m$, binär sind. Zu überlegen ist, wie die jeweils zu verwendende Variable ausgewählt werden soll.[10] Hier folgen wir einem Vorschlag von Kaufman und Rousseeuw (1990: 298). Ausgegangen wird von einer Datenmatrix $\mathbf{X}$, die fr jedes Objekt ω_i die Werte der m Variablen als Zeilenvektoren $\mathbf{x}_i = (X_1(\omega_i), \ldots, X_m(\omega_i))$ enthält, wobei für jede Variable $X_j(\omega) \in \{0, 1\}$ gilt. Ist ein Cluster C gegeben, wird fr je zwei Variablen X_j und X_k zunächst fr alle vier möglichen Merkmalskombinationen die Häufigkeit des Auftretens festgestellt:

$$
\begin{aligned}
a_{jk} &= \sum_{\omega \in C} X_j(\omega) X_k(\omega) \\
b_{jk} &= \sum_{\omega \in C} X_j(\omega)(1 - X_k(\omega)) \\
c_{jk} &= \sum_{\omega \in C} (1 - X_j(\omega)) X_k(\omega) \\
d_{jk} &= \sum_{\omega \in C} (1 - X_j(\omega))(1 - X_k(\omega))
\end{aligned}
$$

Hieraus wird ein Maß berechnet, welches die Differenz zwischen der Anzahl der Beobachtungen, für die X_j und X_k dieselben Werte aufweisen, und der

[10] Einige der unterschiedlichen Möglichkeiten werden von Everitt (1993: 82ff.) besprochen.

Anzahl der Beobachtungen, bei denen dies nicht der Fall ist, erfasst:

$$A_{jk} = |a_{jk}d_{jk} - b_{jk}c_{jk}|$$

Das Cluster C wird dann anhand derjenigen Variable in zwei Gruppen aufgeteilt, für die der Wert $A_j = \sum_{j \neq k} A_{jk}$ am größten ist.

Um eine Menge von Objekten mit diesem Verfahren in Cluster zu zerlegen, kann wie folgt vorgegangen werden:

1. Setze $k = 1$ und bilde die Partition $P_k = \{\Omega\}$.
2. Wähle ein beliebiges Cluster $C_k \in P_k$, das mehr als ein Objekt enthält.
3. Berechne für alle Variablen X_j, für die $0 < \sum_{\omega \in C_k} X_j(\omega) < |C_k|$ ist, A_j über alle Elemente aus C_k.
4. Wähle die Variable X_j, für die A_j den größten Wert aufweist.
5. Bilde aus C_k zwei Teilmengen $C'_k = \{\omega \in C_k | X_j(\omega) = 1\}$ und $C''_k = \{\omega \in C_k | X_j(\omega) = 0\}$.
6. Wiederhole (2) bis (5) für alle Elemente von P_k.
7. Bilde aus P_k eine neue Partition P_{k+1}, bei der alle C_k durch C'_k und C''_k ersetzt werden.
8. Setze $k \leftarrow k + 1$; Fortsetzung bei (2), bis alle Cluster nur noch ein Objekt enthalten.

Dieses Verfahren kann in R mit dem Befehl `mona` ausgeführt werden. Zur Illustration verwenden wir die Archäologiedaten aus Abschnitt **??**-2. Box 6.1-3 zeigt die R-Befehle.

Zunächst werden die Daten geladen und die einzelnen Beobachtungen entsprechend der Typenbezeichnung in Tabelle **??**-1 benannt. Die Datenmatrix wird anschließend dem Befehl `mona` übergeben und die Ergebnisse werden unter dem Namen `cl` im Speicher abgelegt. Wird dies Objekt aufgerufen, wird zunächst unter `Revised data` die verwendete Datenmatrix angezeigt. Wenn in den ursprünglichen Daten fehlende Werte vorkommen, werden diese von der Prozedur automatisch durch Schätzwerte ersetzt, so dass die verwendete Datenmatrix nicht unbedingt der ursprünglichen entsprechen muss.[11]

Die hierauf folgenden Ausgaben lassen sich wie beim Befehl `diana` am besten zusammen interpretieren. Unter `Order of objects` ist wieder eine Objektreihung angegeben. Die beiden folgenden Ausgaben weisen $n -$

[11] Fr Details hierzu vgl. Kaufman und Rousseeuw (1990: 298f.) und die Dokumentation zum Befehl `mona`.

Box 6.1-3 R-Befehle für eine monothetische divisive Klassifikation mit den Archäologiedaten.

```
> dat <- read.table("bs1.dat")
> names(dat) <- c("X","Y","Z","h")

> tab <- xtabs(h~.,dat)
> tab <- ftable(tab,row.vars="X",col.vars=c("Y","Z"))
> tab <- prop.table(tab,1)
> d <- dist(tab,method="manhattan")*0.5

> library(cluster)
> cl <- diana(d)

> cl$order.lab <- c ("Turkey","Greece"," Switzerland",
+                    "Great Britain","Germany","Sweden",
+                    "USA"," Japan")

> cl
Merge:
     [,1] [,2]
[1,]   -3   -5
[2,]   -4   -7
[3,]   -1   -2
[4,]    1    2
[5,]    4   -8
[6,]    5   -6
[7,]    3    6

Order of objects:
[1] Turkey        Greece         Switzerland  Great Britain Germany
[6] Sweden        USA            Japan

Height:
[1] 0.15517386 0.42311450 0.05213896 0.16966925 0.10016099 0.21345035
    0.26242049

Divisive coefficient:
[1] 0.6777413

Available components:
[1] "order"  "height" "dc"     "merge"  "diss"   "call"

> cutree(cl,k=2)
[1] 1 1 2 2 2 2 2 2
```

1 Elemente auf, wobei der i-te Eintrag wieder zwischen dem i-ten und $i + 1$-ten Objekt der Reihung anzuordnen ist. Unter `Separation step` wird ausgegeben, bei welchem Schritt die Objekte links und rechts vom entsprechenden Eintrag in zwei Cluster zerteilt wurden. Der elfte Eintrag dieses Ergebnisses ist gleich 1 (dies kann man sich mit `which(cl$step==1)` anzeigen lassen). Der elfte Eintrag der Objektreihung ist S, der zwölfte

das Objekt P. Beim ersten Aufteilungsschritt werden also alle Objekte der Reihung bis S in einem Cluster zusammengefasst, alle Objekte ab P befinden sich in einem zweiten Cluster. Der elfte Eintrag von `Variable used` gibt an, welche Variable für diese Aufteilung verwendet wurde. Dies ist `V8`.

Im zweiten Schritt werden beide nun vorhandenen Cluster entsprechend dem oben beschriebenen Verfahrens aufgeteilt, so dass sich unter `Separation step` zweimal der Eintrag 2 findet. Dieser steht an 5.ter und 13.ter Stelle. Dies bedeutet, dass das größere der beiden vorhandenen Cluster zwischen B und D aufgeteilt wird, wobei Variable `V3` benutzt wird. Das kleinere Cluster wird zwischen P und T aufgeteilt, wobei die Variable `V1` verwendet wird.

6.2 Agglomerative Verfahren

In diesem Abschnitt besprechen wir agglomerative Verfahren, bei denen die einzelnen Objekte sukzessive zu Clustern zusammengefasst werden.

1. Ein allgemeiner Rahmen

Man kann sich vorstellen, dass durch ein agglomeratives Klassifikationsverfahren eine Folge von Partitionen $(P_0, P_1, \ldots, P_{n-1})$ der vorausgesetzten Objektmenge Ω entsteht. Begonnen wird mit der Partition $P_0 = \{\{\omega_1\}, \ldots, \{\omega_n\}\}$, deren Cluster jeweils nur ein Objekt enthalten. Bei jedem neuen Level werden dann zwei Cluster der vorangehenden Partition zu einem neuen Cluster zusammengefasst. Diese Betrachtungsweise führt zu einem allgemeinen Schema für agglomerative Verfahren:

(1) Beginne mit der Partition $P_0 = \{\{\omega_1\}, \ldots, \{\omega_n\}\}$.

(2) Wenn eine Partition $P_i = \{C_{i1}, \ldots, C_{in_i}\}$ gegeben ist, die $n_i = n - i$ Cluster enthält, bestimme zwei Cluster C_{ij} and C_{ik}, die zu einem neuen Cluster vereinigt werden sollen.

(3) Erzeuge eine neue Partition P_{i+1}, indem in der Partition P_i die Cluster C_{ij} und C_{ik} durch ein Cluster $C_{ij} \cup C_{ik}$ ersetzt werden.

(4) Solange die neue Partition mehr als ein Cluster enthält, wiederhole den Prozess bei Schritt (2).

2. SAHN-Algorithmen

Zu überlegen ist, wie im zweiten Schritt vorgegangen werden soll. Oft werden sog. *SAHN-Algorithmen* verwendet, eine Abkürzung für *sequential, agglomerative, hierarchical, non-overlapping*.[12] Die Idee besteht darin,

[12] Vgl. Jain und Dubes (1988: 79).

Box 6.2-1 Abstandsdefinitionen für einige SAHN-Methoden.

Single Link Methode

$$\rho(C_i, C_j \cup C_k) = \min\{\rho(C_i, C_j), \rho(C_i, C_k)\}$$

Complete Link Methode

$$\rho(C_i, C_j \cup C_k) = \max\{\rho(C_i, C_j), \rho(C_i, C_k)\}$$

WPGMA (Weighted Average) Methode

$$\rho(C_i, C_j \cup C_k) = \frac{1}{2}\left(\rho(C_i, C_j) + \rho(C_i, C_k)\right)$$

WPGMC (Weighted Centroid) Methode

$$\rho(C_i, C_j \cup C_k) = \frac{1}{2}\left(\rho(C_i, C_j) + \rho(C_i, C_k)\right) - \frac{1}{4}\rho(C_j, C_k)$$

UPGMA (Group Average) Methode

$$\rho(C_i, C_j \cup C_k) = \frac{n_j\,\rho(C_i, C_j) + n_k\,\rho(C_i, C_k)}{n_j + n_k}$$

UPGMC (Unweighted Centroid) Methode

$$\rho(C_i, C_j \cup C_k) = \frac{n_j\,\rho(C_i, C_j) + n_k\,\rho(C_i, C_k)}{n_j + n_k} - \frac{n_j\,n_k}{(n_j + n_k)^2}\rho(C_j, C_k)$$

Ward's (Minimum Variance) Methode

$$\rho(C_i, C_j \cup C_k) = \frac{(n_i + n_j)\,\rho(C_i, C_j) + (n_i + n_k)\,\rho(C_i, C_k) - n_i\,\rho(C_j, C_k)}{n_i + n_j + n_k}$$

eine Abstandsfunktion für Cluster zu definieren und dann jeweils diejenigen Cluster zusammenzufassen, die den geringsten Abstand aufweisen. Wenn also $P = \{C_1, \ldots, C_m\}$ eine Partition von Ω ist und $\rho(C_j, C_k)$ eine Abstandsfunktion für die Cluster, muss man zwei Cluster $C_{j'}$ und $C_{k'}$ finden, so dass

$$\rho(C_{j'}, C_{k'}) = \min_{j,k}\{\rho(C_j, C_k)\}$$

ist. Die SAHN-Verfahren verwenden eine rekursive Definition der Abstandsfunktion, wobei stets mit $\rho(\{j\}, \{k\}) := d_{jk}$ begonnen wird. In jedem Schritt der agglomerativen Prozedur hat man dann bereits eine Ab-

Tabelle 6.2-1 Einige SAHN-Methoden, die mit der Formel von Lance und Williams definiert werden können.

Methode	α_1	α_2	β	γ
Single Link	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$
Complete Link	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
WPGMA (Weighted Average)	$\frac{1}{2}$	$\frac{1}{2}$	0	0
WPGMC (Weighted Centroid)	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{4}$	0
UPGMA (Group Average)	$\frac{n_j}{n_j+n_k}$	$\frac{n_k}{n_j+n_k}$	0	0
UPGMC (Unweighted Centroid)	$\frac{n_j}{n_j+n_k}$	$\frac{n_k}{n_j+n_k}$	$\frac{-n_j\,n_k}{(n_j+n_k)^2}$	0
Ward's (Minimum Variance)	$\frac{n_i+n_j}{n_i+n_j+n_k}$	$\frac{n_i+n_k}{n_i+n_j+n_k}$	$\frac{-n_i}{n_i+n_j+n_k}$	0

standsfunktion ρ für die aktuelle Partition, um die beiden ähnlichsten Cluster, etwa C_j und C_k, zu finden. Somit genügt es, eine Abstandsfunktion ρ für die neue Partition zu definieren, die aus der Zusammenfassung von C_j und C_k entsteht, also Abstände $\rho(C_i, C_j \cup C_k)$ zu definieren. Oft werden die in Box 6.2-1 angegebenen Varianten verwendet (dabei bedeuten: $n_i := |C_i|$, $n_j := |C_j|$ und $n_k := |C_k|$).[13] In den Abkürzungen steht der erste Buchstabe für *weighted* oder *unweighted*, der letzte für *average* oder *centroid*; die mittleren Buchstaben PGM stehen für *pair group method*.

Lance und Williams (1966) haben folgende Formel entwickelt, mit der sich eine ganze Familie von SAHN-Algorithmen definieren lässt, die die oben angegebenen Varianten als Spezialfälle enthält:

$$\rho(C_i, C_j \cup C_k) = \tag{6.1}$$
$$\alpha_1\rho(C_i, C_j) + \alpha_2\rho(C_i, C_k) + \beta\rho(C_j, C_k) + \gamma|\rho(C_i, C_j) - \rho(C_i, C_k)|$$

Tabelle 6.2-1 zeigt Parameterwerte für die Standardvarianten.

Es ist auch bemerkenswert, dass einige der SAHN-Algorithmen (insbesondere das Single Link und das Complete Link Verfahren) nur ordinale Beziehungen der Abstände ausnutzen. D.h., wenn $\mathbf{D}' = (d'_{ij})$ durch eine monotone Transformation aus $\mathbf{D} = (d_{ij})$ hervorgeht,[14] entsteht aus beiden Abstandsmatrizen dieselbe hierarchische Klassifikation.[15]

3. Illustration mit Berufsstrukturdaten

Zur Illustration dient wieder die Abstandsmatrix aus Tabelle **??**-3 für die Berufsstrukturdaten. Zur Durchführung der Berechnungen verwenden

[13] In den Bezeichnungen folgen wir Jain und Dubes (1988: 80); man vgl. auch Anderberg (1973: Kap. 6) und Späth (1975: 170-2).

[14] D.h. $d_{ij} \le d_{kl} \Longrightarrow d'_{ij} \le d'_{kl}$.

[15] Vgl. Johnson (1967); Hubert (1973).

Box 6.2-2 Durch `hca1.cf` erzeugtes Ausgabefile `hca1.lev`.

```
Level  Niveau   Ci   Cj
------------------------
 1     0.0520    3    5
 2     0.1002    4    7
 3     0.1027    3    4
 4     0.1341    3    8
 5     0.1551    1    2
 6     0.1652    1    3
 7     0.1664    1    6
```

Box 6.2-3 Durch `hca1.cf` erzeugtes Ausgabefile `hca1.den`, das das Dendrogramm in Form einer Kantenliste zeigt.

```
  I    J     Niveau
--------------------
  3    9     0.0520
  5    9     0.0520
  4   10     0.1002
  7   10     0.1002
  9   11     0.0507
 10   11     0.0025
  8   12     0.1341
 11   12     0.0314
  1   13     0.1551
  2   13     0.1551
 12   14     0.0311
 13   14     0.0101
  6   15     0.1664
 14   15     0.0012
```

wir zunächst die TDA-Prozedur `hcls`; wir beginnen mit der Single-Link-Methode.[16]

Das in Box 6.2-2 dokumentierte Ausgabefile `hca1.lev` zeigt den Ablauf des agglomerativen Verfahrens. In einem ersten Schritt werden die Objekte mit den Nummern 3 und 5 bei einem Niveau 0.052 (das ist der Abstand der beiden Objekte in der Abstandsmatrix) zu einem Cluster zusammengefasst:

$$\rho(C_3, C_5) = 0.0520$$

Dann werden die Objekte mit den Nummern 4 und 7 bei einem Niveau 0.1002 zusammengefasst:

$$\rho(C_4, C_7) = 0.1002$$

[16] Verwendet wird das Skript `hca1.cf`.

Dann werden diese beiden Cluster zusammengefasst:

$$\rho(C_3 \cup C_5, C_4 \cup C_7) = \min\{\rho(C_3 \cup C_5, C_4), \rho(C_3 \cup C_5, C_7) = 0.1027$$

Und so weiter, bis alle Objekte in einem Cluster enthalten sind. (Cluster, die mehr als ein Objekt enthalten, werden durch die jeweils kleinste Nummer eines Elements identifiziert.)

4. Agglomerative Verfahren in R

Die beschriebenen SAHN-Verfahren können in `R` mit dem Befehl `hclust` ausgeführt werden. Ein Beispiel mit den Berufsstrukturdaten findet sich in Box 6.2-4. Zunächst wird wie in Kapitel 2 die Abstandsmatrix aus Tabelle **??**-3 erstellt. Anschließend wird diese Abstandsmatrix an den Befehl `hclust` unter Angabe des Arguments `method` übergeben. Über das letztgenannte Argument wird der zu verwendende SAHN-Algorithmus festgelegt, wobei hier durch `"single"` die Single Link-Methode gewählt wird. Werte für andere SAHN-Algorithmen finden sich in Box 6.2-5. Der Standardwert des Arguments ist `"complete"`. Ergebnisse des `hclust`-Befehls werden unter dem Namen `cl` abgespeichert.

Im Anschluss an diesen Befehl werden Bezeichnungen für die einzelnen Objekte angegeben. Diese werden von einigen Auswertungsprozeduren anstelle von Beobachtungsnummern angezeigt und erlauben eine einfache Interpretation der Ergebnisse.

Ruft man das Objekt `cl` direkt auf, erhält man lediglich einen Überblick über das gewählte SAHN-Verfahren, die verwendeten Abstände und die Anzahl der Objekte. Den Ablauf der einzelnen Fusionierungsschritte kann man über `cl$merge` anzeigen lassen. Für jeden Schritt des Verfahrens wird ausgewiesen, welche Cluster zusammengefügt werden. Clustern, die lediglich einzelne Beobachtungen enthalten, ist ein Minus-Zeichen vorangestellt. Im ersten Schritt werden also die Beobachtungen 3 (Schweiz) und 5 (Deutschland) zusammengefasst, im zweiten Schritt sind dies die Beobachtungen 4 (Großbritannien) und 7 (USA). Die Cluster, die bei den beiden ersten Fusionierungsschritten entstanden sind, werden im dritten Schritt zusammengefasst und so fort. Das Niveau der einzelnen Fusionierungsschritte kann über `cl$height` abgerufen werden.[17]

5. Dendrogramme

Ein praktisches Hilfsmittel, um das Ergebnis einer agglomerativen Klassifikation zu veranschaulichen, sind Dendrogramme. Es handelt sich um baumartige Graphen, die zeigen, wie die Objekte bzw. Cluster sukzessive zusammengefasst werden.[18]

[17] Wie bereits in Abschnitt 6.2-3 erläutert wurde, bezeichnet das Wort 'Niveau' hier den Abstand $\rho(C, C')$ für die Zusammenfassung der Cluster C und C' zu einem neuen Cluster $C \cup C'$.

Box 6.2-4 R-Befehle für eine hierarchische Klassifikation mit den Berufsstrukturdaten.

```
# Tabelle
> dat <- read.table("bs1.dat")
> names(dat) <- c("X","Y","Z","h")
> tab <- xtabs(h~.,dat)
> tab <- ftable(tab,row.vars="X",col.vars=c("Y","Z"))
> tab <- prop.table(tab,1)
> d <- dist(tab,method="manhattan")*0.5

# Hierarchische Klassifikation mit single-link
> cl <- hclust(d,method="single")
> cl$labels <-c ("Turkey","Greece"," Switzerland",
                 "Great Britain","Germany","Sweden",
                 "USA"," Japan")

# Anzeigen diverser Ergebnisse
> cl

Call:
hclust(d = d, method = "single")

Cluster method   : single
Distance         : manhattan
Number of objects: 8

> cl$merge
     [,1] [,2]
[1,]   -3   -5
[2,]   -4   -7
[3,]    1    2
[4,]   -8    3
[5,]   -1   -2
[6,]    4    5
[7,]   -6    6

> cl$height
[1] 0.05213896 0.10016099 0.10273233 0.13411477
[5] 0.15517386 0.16529048 0.16643096

# Partitionierung mit 3 Clustern anzeigen
> cutree(cl,k=3)
      Turkey        Greece   Switzerland  Great Britain
           1             1             2             2
     Germany        Sweden           USA         Japan
           2             3             2             2

# Dendrogramm
> plot(cl,xlab="",ylab="",main="Dendrogramm",sub="")
> rect.hclust(cl,k=3,border="grey")
```

[18] Als ein Graph betrachtet, kann ein Dendrogramm auch als eine Kantenliste dargestellt (und dann ggf. für weitere Analysezwecke verwendet) werden. Für unser Beispiel erhält

Box 6.2-5 Argumente für den R-Befehl `hclust` zur Auswahl eines SAHN-Algorithmus.

```
method="single"         # Single Link
method="complete"       # Complete Link
method="mcquitty"       # WPGMA / Weighted Average
method="median"         # WPGMC / Weighted Centroid
method="average"        # UPGMA / Group Average Method
method="centroid"       # UPGMC / Unweighted Centroid
method="ward"           # Wards Methode (Minimum Variance)
```

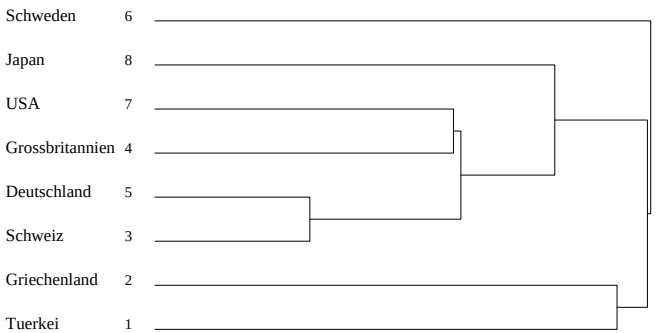


Abb. 6.2-1 Mit TDA (aus dem Ausgabefile `hca1.pcf`) erzeugtes Dendrogramm der hierarchischen Klassifikation mit den Berufsstrukturdaten.

Die Abbildungen 6.2-1 und 6.2-2 zeigen die mit TDA bzw.R erzeugten Dendrogramme für unser gegenwärtiges Beispiel. Das mit R erzeugte Dendrogramm vermittelt mehr Informationen. Um es zu erzeugen, wurde der in Box 6.2-4 angegebene `plot`-Befehl verwendet, der auf das Objekt `cl` angewendet wird. Die weiteren Argumente entfernen die Achsenbeschriftung (`xlab`, `ylab`), setzen den Titel der Grafik auf „Dendrogramm“und entfernen eine standardmäßig ausgegebene Bildunterschrift (`sub`). Die zuvor vergebenen Objektnamen werden automatisch verwendet. Über den anschließend benutzten Befehl `rect.clust` wird zusätzlich die oben bereits mit dem Befehl `cutree` angezeigte Partition der Beobachtungen in drei Cluster unter Verwendung von Boxen eingezeichnet. Über `border="grey"` wird als Farbe für diese Boxen grau gewählt.

man diese Darstellung durch das Ausgabefile `hca1.den` (Box 6.2-3).

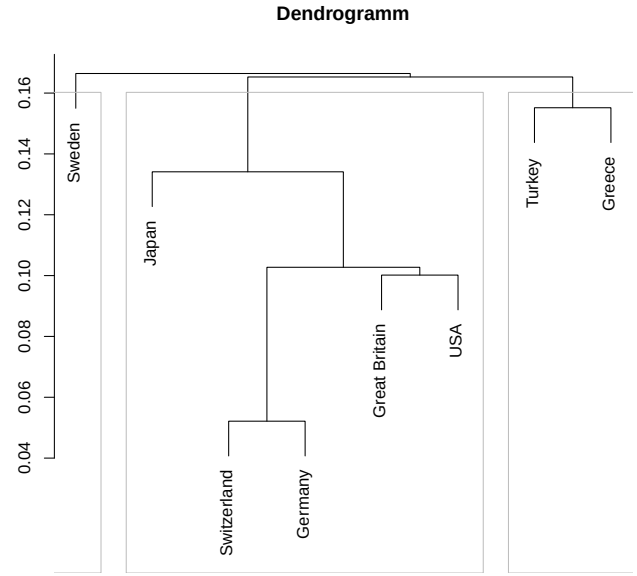


Abb. 6.2-2 Mit R (Box 6.2-4) erzeugtes Dendrogramm der hierarchischen Klassifikation mit den Berufsstrukturdaten.

6. Vergleiche der SAHN-Verfahren

Von *Bindungen* spricht man, wenn es in einer Abstandsmatrix zwei oder mehr unterschiedliche Paare von Objekten gibt, die den gleichen Abstand haben. Wenn es Bindungen gibt, hängen die Ergebnisse der SAHN-Algorithmen (mit Ausnahme der Single Link-Methode) auch von der zufälligen Ordnung der Daten ab.[19]

Auch wenn eine Abstandsmatrix keine Bindungen aufweist, führen die SAHN-Algorithmen meistens zu mehr oder weniger unterschiedlichen Ergebnissen.[20] Abbildung 6.2-3 illustriert das anhand der Berufsstrukturdaten. Ein theoretisch begründeter Vergleich setzt allerdings voraus, die Dendrogramme zunächst als unterschiedliche Modelle der gegebenen Abstandsmatrix zu verstehen.

Zu beachten ist auch, dass durch die SAHN-Algorithmen nicht immer ein (korrektes) Dendrogramm entsteht. Dies hängt sowohl von der Beschaffenheit der jeweils verwendeten Abstandsmatrix als auch vom Klassifikationsverfahren ab. Milligan (1979) hat gezeigt, dass bei den durch die

[19] Man vgl. die Diskussion bei Jain und Dubes (1988: 76ff.); Klemm (1995).

[20] Ausführliche Hinweise auf Eigenschaften der unterschiedlichen SAHN-Methoden findet man bei Sneath und Sokal (1973); Klemm (1995).

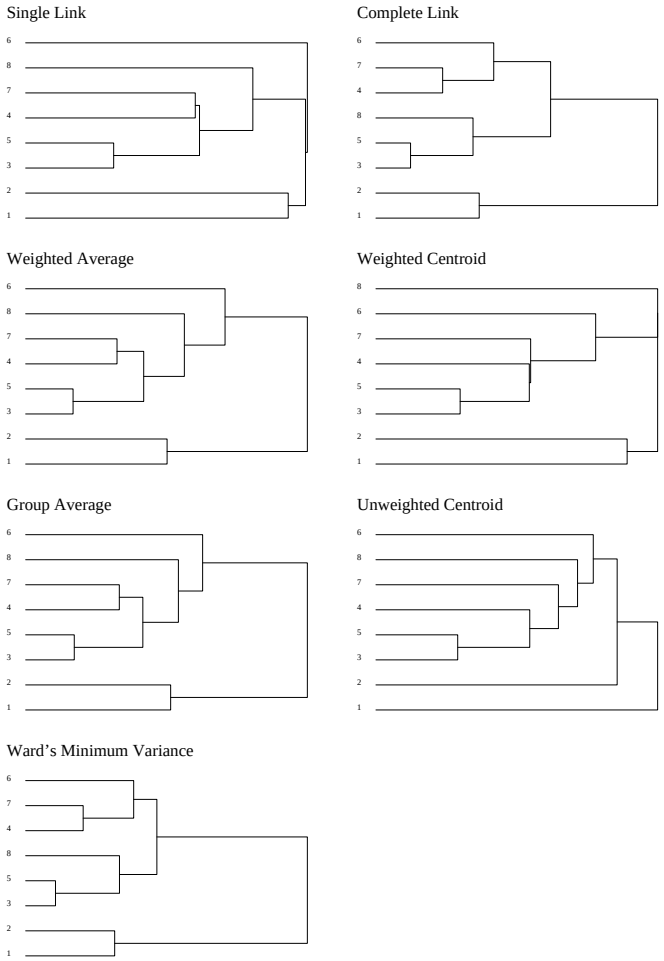


Abb. 6.2-3 Unterschiedliche Ergebnisse der SAHN-Algorithmen bei den Berufsstrukturdaten.

Formel (6.1) von Lance und Williams definierten Verfahren die für ein korrektes Dendrogramm erforderliche Monotoniebedingung dann erfüllt ist, wenn folgende Bedingungen gelten:[21]

$$\gamma \geq 0,\ \alpha_1 + \alpha_2 + \beta \geq 1,\ \alpha_1 \geq 0,\ \alpha_2 \geq 0 \quad \text{oder} \qquad (6.2)$$
$$\gamma < 0,\ \alpha_1 + \alpha_2 + \beta \geq 1,\ \alpha_1 \geq |\gamma|,\ \alpha_2 \geq |\gamma|$$

[21] Vgl. auch Jain und Dubes (1988: 83ff.).

In fünf der sieben in Tabelle 6.2-1 angegebenen Verfahren ist die Bedingung erfüllt. Bei den Centroid-Verfahren (WPGMC und UPGMC) ist sie nicht erfüllt. Tatsächlich tritt auch bei den Berufsstrukturdaten bei der WPGMC-Methode im letzten Schritt eine Verlezung der Monotoniebedingung auf.

7. Erzeugung von Partitionen

Ein hierarchisches Klassifikationsverfahren erzeugt offenbar nicht unmittelbar eine Einteilung in bestimmte Cluster. Um Partitionen zu erzeugen, kann man jedoch das Dendrogramm verwenden, denn für jedes Niveau zeigt es, welche Objekte bis zu diesem Niveau gemeinsamen Clustern angehören.

In Box 6.2-4 wird zur Illustration gezeigt, wie eine Partitionierung in drei Cluster vorgenommen werden kann. Verwendet wird der R-Befehl `cutree`. Diesem Befehl wird zunächst das Clusterobjekt `cl` übergeben, dann wird mit dem Argument `k` die Anzahl der Cluster angegeben (in unserem Beispiel `k=3`). Die Ausgabe verwendet für die einzelnen Beobachtungen die zuvor definierten Label und gibt jeweils an, zu welchem Cluster das Objekte gehört. Beispielsweise sind die Türkei und Griechenland zusammen in einem Cluster.

Wie Abbildung 6.2-1 zeigt, können die Cluster auch in einem Dendrogramm sichtbar gemacht werden. Dies wird in unserem Beispiel dadurch erreicht, dass im Anschluss an den `plot`-Befehl der Befehl `rect.clust` verwendet wird. Mit `border="grey"` wird als Farbe für die Boxen grau gewählt.

Es stellt natürlich die Frage, wie die Anzahl der Cluster für eine Klassifikation gewählt werden sollte. Zunächst ist sicherlich wichtig, dass die Cluster ausgehend von inhaltlichen Überlegungen überschaubar sind und sich gut interpretieren lassen. Darüber hinaus kann man sich auch an einigen formalen Kriterien orientieren, die zur Beurteilung unterschiedlicher Klassifikationsverfahren und resultierender Partitionen vorgeschlagen worden sind. Wir weisen kurz auf drei dieser Vorschläge hin.

1. Ein Ansatz findet sich bei Handl et al. (2005). Ausgehend von einer Abstandsmatrix **D** für eine Menge von Objekten $\Omega := \{\omega_1, \ldots, \omega_n\}$ sei für ω_i $O(\omega_i) = (\omega_1^i, \ldots, \omega_{n-1}^i)$ eine Ordnung aller übrigen Objekte nach Ähnlichkeit zu ω_i. ω_1^i ist dann genau die Beobachtung $\omega_j \in \Omega \backslash \omega_i$, für die $\min\{d_{ij} | \omega_j \in \Omega \backslash \omega_i\}$ gilt. ω_2^i ist die zu ω_i „zweit ähnlichste“Beobachtung und so fort. Nun sei $I(\omega_i, \omega_j^i)$ gleich 0, wenn bei einer gegebenen Partition C ω_i und ω_j^i zusammen im Cluster $C_k \in C$ sind, $1/j$ sonst. Dann kann eine Funktion

$$D(C) = \sum_{i=1}^{n} \sum_{j=1}^{l} I(\omega_i, \omega_j^i) \qquad (6.3)$$

definiert werden, die erfasst, wie viele der l ähnlichsten Beobachtungen zusammen in einem Cluster sind. Je höher der Wert dieser Funktion für eine Partition ist, desto eher sind ähnliche Beobachtungen in unterschiedlichen Clustern. Eine nach diesem Kriterium gute Lösung sollte also einen möglichst geringen Wert aufweisen.

2. Ein anderer Vorschlag stammt von Dunn (1974):

$$D(C) = \frac{\min_{C_k, C_l \in C, C_k \neq C_l} \{\min\{d_{ij} | \omega_i \in C_k, \omega_j \in C_l\}\}}{\max_{C_k \in C}\{d_{ij} | \omega_i, \omega_j \in C_k\}} \tag{6.4}$$

Im Zähler steht der kleinste Abstand, der zwischen zwei nicht dem selben Cluster zugeordneten Beobachtungen auftritt. Im Nenner findet sich der größte bei der Partition auftretende Clusterdurchmesser. Je höher dieser Kennwert, desto größer sind Unterschiede zwischen den Clustern verglichen mit Unterschieden innerhalb der Cluster. Im Gegensatz zum ersten vorgestellten Kriterium wird hier ein möglichst hoher Wert als vorteilhaft interpretiert.

3. Rousseeuw (1987) hat einen weiteren Vorschlag gemacht:

$$D(C) = \frac{1}{n} \sum_{i=1}^{n} \frac{b_i - a_i}{\max\{b_i, a_i\}} \tag{6.5}$$

Wenn Beobachtung $\omega_i \in C_k$ ist gilt hierbei

$$a_i = \sum_{\omega_j \in C_k \setminus \omega_i} \frac{d_{ij}}{|C_k| - 1} \tag{6.6}$$

und

$$b_i = \min_{C_l \in C \setminus C_k} \left\{ \sum_{\omega_j \in C_l} \frac{d_{ij}}{|C_l|} \right\} \tag{6.7}$$

a_i erfasst den durchschnittlichen Abstand zwischen ω_i und den übrigen Beobachtungen im selben Cluster. b_i gibt den durchschnittlichen Abstand von ω_i zu Beobachtungen im „ähnlichsten“Cluster wieder. Dieses Kriterium kann Werte zwischen -1 und 1 annehmen. Je näher der Wert an 1 ist, als desto besser kann die Clusterlösung angesehen werden.

Diese Kennwerte können in `R` mit Befehlen des Pakets `clValid` berechnet werden. Eine Illustration mit den Berufsstrukturdaten findet man in Box 6.2-6.

Als erstes werden die Berufsstrukturdaten geladen und so aufbereitet, dass sie als Matrix mit dem Namen `dat` im Speicher liegen. Hierauf folgend wird das Paket `clValid` aufgerufen, welches einen gleichnamigen Befehl zur Berechnung der besprochenen Kriterien enthält.

Diesem wird zunächst die Matrix `dat` übergeben. Über `nClust` wird

Box 6.2-6 R-Befehle zur Berechnung von Kennwerten zur Beurteilung von Klassifikationen.

```
> dat <- read.table("http://www.stat.rub.de/teaching/drk/bs1.dat")
> names(dat) <- c("X","Y","Z","h")
> tab <- xtabs(h~.,dat)
> tab <- ftable(tab,row.vars="X",col.vars=c("Y","Z"))
> tab <- prop.table(tab,1)
> dat <- matrix(tab,nrow=8)

> library(clValid)
> val <- clValid(dat,nClust=2:5,clMethods=c("hierarchical","kmeans",
+                "pam"),method="single",validation="internal",
+                       metric="manhattan",neighbSize=2)
> summary(val)

Clustering Methods:
 hierarchical kmeans pam

Cluster sizes:
 2 3 4 5

Validation Measures:
                                  2      3      4      5

hierarchical Connectivity  1.5000 2.5000 4.5000 6.0000
             Dunn          0.4265 0.7744 0.7270 0.7904
             Silhouette    0.1003 0.2664 0.1223 0.1412
kmeans       Connectivity  1.0000 2.5000 5.5000 6.0000
             Dunn          0.6299 0.7744 0.7904 0.7904
             Silhouette    0.4449 0.2664 0.1595 0.1412
pam          Connectivity  1.0000 2.5000 4.5000 6.5000
             Dunn          0.6299 0.7744 0.7270 0.7039
             Silhouette    0.4449 0.2664 0.1223 0.1700

Optimal Scores:

             Score  Method       Clusters
Connectivity 1.0000 kmeans       2
Dunn         0.7904 hierarchical 5
Silhouette   0.4449 kmeans       2
```

festgelegt, welche Partitionen untersucht werden sollen. Im Beispiel sind dies Partitionen mit 2 bis 5 Clustern. Darauf wird über `clMethods` bestimmt, welche Verfahren miteinander verglichen werden sollen. Hier sind es der *k-means* Algorithmus, das *k-medoids* Verfahren sowie ein (agglomeratives) hierarchisches Verfahren. Der bei der hierarchischen Klassifikation zu verwendende SAHN-Algorithmus wird über `method` ausgewählt. Über das Argument `validation` können verschiedene Kriterien zur Bewertung der Partitionen ausgesucht werden[22]. Die oben beschriebenen Verfahren

[22] Alle möglichen Varianten werden von Brock et al. (2008) beschrieben.

erhält man durch Angabe von `"internal"`. Zur Berechnung der Abstände zwischen den Beobachtungen wird mittels `metric="manhattan"` die City-Block-Metrik bestimmt. Schließlich wird durch `neighbSize=2` für das von Handl et al. (2005) vorgeschlagene Verfahren der Wert von l festgelegt.

Durch den Aufruf von `summary(val)` werden die Ergebnisse angezeigt. Als erstes werden die zu vergleichenden Verfahren und Partitionen ausgegeben. Anschließend werden für jede Verfahren-Clusterzahl-Kombination die errechneten Kennwerte angezeigt. `Connectivity` bezieht sich auf das von Handl et al. (2005) vorgeschlagene Verfahren, unter `Silhouette` findet sich der Wert des von Rousseeuw (1987) verwendeten Kriteriums. Beispielsweise erzielt eine aus einer hierarchischen Klassifikation mit Single Link stammende Lösung mit 2 Clustern beim Kriterium nach Dunn (1974) einen Wert von 0.4265.

Schließlich ist unter `Optimal Scores` für jedes der Kriterien genau die Verfahren-Clusterzahl Kombination angegeben, die den besten Wert aufweist. In zwei Fällen ist dies das *k-means*-Verfahren mit 2 Clustern. Bei Dunns Kriterium schneidet die hierarchische Klassifikation mit 5 Clustern am besten ab. Bei diesem Beispiel könnte man sich dann für eine Partition mit zwei Clustern entscheiden.

6.3 Ultrametrische Baummodelle

In diesem Abschnitt betrachten wir hierarchische Klassifikationsverfahren als Methoden zur Konstruktion von Modellen, die dem Zweck dienen sollen, Abstandsfunktionen zu repräsentieren.

1. Hierarchien und Bäume

Sei $\Omega = \{\omega_1, \ldots, \omega_n\}$ eine beliebige Objektmenge. Unter einer *Hierarchie* (für Ω) verstehen wir eine Menge $\mathcal{H}$, deren Elemente Teilmengen von Ω sind und die folgenden Bedingungen genügt:

a) $\Omega \in \mathcal{H}$, $\emptyset \notin \mathcal{H}$

b) für alle $\omega \in \Omega$: $\{\omega\} \in \mathcal{H}$

c) wenn $h_1, h_2 \in \mathcal{H}$, dann $h_1 \cap h_2 \in \{h_1, h_2, \emptyset\}$

Ist beispielssweise $\Omega = \{1, 2, 3, 4, 5\}$, wäre

$$\mathcal{H} := \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{2, 3\}, \{1, 2, 3, 4, 5\}\}$$

eine Hierarchie. Offenbar können auch noch andere Hierarchien gebildet werden.

Wenn $\mathcal{H}$ eine Hierarchie ist, kann sie auch durch einen Graphen repräsentiert werden. Man kann $\mathcal{H}$ mit der Knotenmenge des Graphen identifizieren und festlegen, dass zwei Knoten h_1 und h_2 durch eine Kante verbunden sind, wenn h_1 eine Teilmenge von h_2 ist und es keinen anderen Knoten h mit der Eigenschaft $h_1 \subset h \subset h_2$ gibt.

Graphen, durch die Hierarchien repräsentiert werden, sind Beispiele für eine besondere Art von Graphen, die man *Bäume* nennt. Wir verwenden folgende allgemeine Definition: Ein Graph $\mathcal{G} = (\mathcal{K}, \mathcal{L})$, wobei $\mathcal{K}$ die Knotenmenge und $\mathcal{L}$ die Kantenmenge bezeichnet, ist ein Baum, wenn $|\mathcal{K}| = |\mathcal{L}| + 1$ ist und je zwei Knoten durch genau einen Pfad verbunden sind.[23] Knoten, die den Grad 1 haben, werden als *Blätter* bezeichnet; wenn der Grad größer als 1 ist, spricht man von *internen Knoten*.

Man kann für einen Baum eine Orientierung einführen, indem man einen internen Knoten als *Wurzel des Baums* bestimmt. Dann gibt es von jedem Blatt des Baums genau einen Pfad zu seiner Wurzel. Zu beachten ist, dass es meistens unterschiedliche Möglichkeiten gibt, Wurzeln zu definieren. So könnte man in dem oben angeführten Beispiel sowohl den Knoten, der der Menge Ω entspricht, als auch den Knoten, der der Menge $\{2, 3\}$ entspricht, als eine Wurzel des Baums bestimmen.

2. Einfache Baummodelle

Sei $\mathcal{G} = (\mathcal{K}, \mathcal{L}, v)$ ein Baum mit einer Bewertungsfunktion $v : \mathcal{L} \longrightarrow \mathbf{R}$. Die Bewertungsfunktion ordnet jeder Kante $l \in \mathcal{L}$ eine positive Zahl $v(l)$ zu, die als Länge oder Wert der Kante interpretiert werden kann. Dann kann man für jeweils zwei Knoten $i, j \in \mathcal{K}$ einen Abstand

$$d_{ij}^v := \text{ Summe der Kantenbewertungen des Pfades von } i \text{ nach } j$$

definieren und erhält auf diese Weise eine Abstandsfunktion für die Knotenmenge des Baums.

Diese Möglichkeit, Abstandsfunktionen durch Bäume zu definieren, führt nun zu der Idee, Bäume als Modelle für vorgegebene Abstandsmatrizen zu verwenden. Wir beginnen mit einer einfachen Variante, die später modifiziert und erweitert wird. Als Ausgangspunkt wird angenommen, dass für eine Objektmenge $\Omega = \{\omega_1, \ldots, \omega_n\}$ Abstände gegeben sind; d_{ij} ist der Abstand zwischen ω_i und ω_j. Gesucht ist nun ein Baum $\mathcal{G}$ mit der Knotenmenge Ω und einer Kantenbewertung v, so dass die daraus resultierenden Abstände d_{ij}^v möglichst gut zu den vorausgesetzten Abständen d_{ij} passen.

Wenn man die Optimalitätsforderung („möglichst gut“) zunächst ignoriert, kann man einen sogenannten *minimalen aufspannenden Baum* (auch kurz *Minimalbaum* genannt[24]) verwenden. Um den Begriff und die Konstruktion zu erklären, verwenden wir als Beispiel die Berufsstrukturdaten aus Abschnitt **??**, und zwar die Abstandsmatrix in Tabelle **??**-3 für die

[23] Damit äquivalent ist die Definition, dass ein Baum ein zusammenhängender Graph ohne Zyklen ist.

[24] Im Englischen: *minimal spanning tree*.

Box 6.3-1 Kantenliste eines minimalen aufspannenden Baums für die Abstandsmatrix in Tabelle **??**-3.

```
 i    j     v(i,j)
-------------------
 3    5     0.0520
 4    7     0.1002
 4    5     0.1027
 5    8     0.1341
 1    2     0.1551
 2    8     0.1652
 4    6     0.1664
```

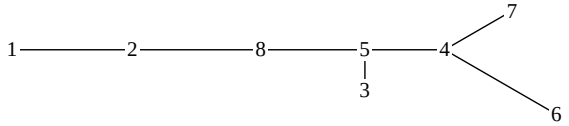


Abb. 6.3-1 Der Baum mit der Kantenliste in Box 6.3-1.

acht Länder; sie wird im Folgenden $\mathbf{D} = (d_{ij})$ genannt. Gesucht ist also ein Baum mit der Knotenmenge $\mathcal{K} = \{1, \ldots, 8\}$, so dass die Knoten den Ländern entsprechen. Man benötigt genau sieben Kanten, damit ein Baum entsteht. Welche soll man nehmen?

Bei der Konstruktion eines Minimalbaums wird angenommen, dass die Kantenbewertungen durch die Abstände d_{ij} vorgegeben sind. Wenn also für den zu konstruierenden Baum eine Kante verwendet wird, die die Knoten i und j verbindet, erhält sie die Bewertung d_{ij}. Also kann man sich an folgender Idee orientieren: Wähle die Kanten für den zu konstruierenden Baum so aus, dass die Summe der Kantenbewertungen minimal wird. Der auf diese Weise entstehende Baum wird ein Minimalbaum genannt.

Box 6.3-1 zeigt einen Minimalbaum für das Beispiel in Form einer Kantenliste.[25] Die angegebenen Kantenbewertungen entsprechen offenbar den Abständen in Tabelle **??**-3. Abbildung 6.3-1 zeigt den Baum in einer graphischen Darstellung, wobei die Längen der Kanten näherungsweise proportional zu ihren Bewertungen sein sollten.[26]

Allerdings ist ohne weiteres nicht klar, ob der Baum ein gutes Modell für die Abstandsmatrix $\mathbf{D}$ (in Tabelle **??**-3) ist. Denn der Baum repräsentiert zwar korrekt sieben Abstände; er impliziert aber (insgesamt 28) Abstände zwischen allen acht Knoten. Wenn man sie ausrechnet, erhält man eine neue Abstandsmatrix $\mathbf{D}^v = (d^v_{ij})$, die in Tabelle 6.3-1 gezeigt

[25] Erzeugt mit dem Skript `gm1.cf`.

[26] Erstellt mit dem Skript `gmplot1.cf`.

Tabelle 6.3-1 Aus dem Baum mit der Kantenliste in Box 6.3-1 gebildete Abstandsmatrix $\mathbf{D}^v$.

0.0000	0.1551	0.5064	0.5571	0.4544	0.7235	0.6573	0.3203
0.1551	0.0000	0.3513	0.4020	0.2993	0.5684	0.5022	0.1652
0.5064	0.3513	0.0000	0.1547	0.0520	0.3211	0.2549	0.1861
0.5571	0.4020	0.1547	0.0000	0.1027	0.1664	0.1002	0.2368
0.4544	0.2993	0.0520	0.1027	0.0000	0.2691	0.2029	0.1341
0.7235	0.5684	0.3211	0.1664	0.2691	0.0000	0.2666	0.4032
0.6573	0.5022	0.2549	0.1002	0.2029	0.2666	0.0000	0.3370
0.3203	0.1652	0.1861	0.2368	0.1341	0.4032	0.3370	0.0000

wird.[27] Vergleicht man $\mathbf{D}^v$ mit $\mathbf{D}$, findet man, dass es erhebliche Unterschiede gibt, was auch durch folgende Abstandsberechnungen sichtbar wird:[28]

$$\|\mathbf{D}^v - \mathbf{D}\| = 1.04 \quad \|\mathbf{D}^v\| = 2.70 \quad \|\mathbf{D}\| = 1.73$$

3. Minimalbaumberechnung mit R

Zur Berechnung von Minimalbäumen kann in `R` auf das Paket `vegan` zurückgegriffen werden. Ein Beispiel mit den Berufsstrukturdaten findet sich in Box 6.3-2. Zunächst werden die Berufsstrukturdaten wie bei den bisherigen Illustrationen in diesem Kapitel geladen und wird die Abstandsmatrix aus Tabelle **??**-3 erstellt. Anschließend wird das Paket `vegan` aufgerufen, welches den Befehl `spantree` enthält. Über diesen Befehl können Minimalbäume an Abstandsmatrizen angepasst werden. Bei diesem Beispiel wird die Abstandsmatrix der Länder an den Befehl übergeben und das Ergebnis unter dem Namen `baum` im Speicher abgelegt.

Wird `baum` direkt aufgerufen, ist die Kantenliste unter dem Punkt `$kid` und die Bewertung der Kanten unter `$dist` ablesbar. Bei `$kid` finden sich insgesamt $n - 1$ Einträge. Wenn Eintrag i den Wert j aufweist, bedeutet dies, dass eine Kante zwischen den Knoten $i + 1$ und j liegt. Der erste Eintrag von `$kid` bezieht sich also auf Beobachtung 2 und weist den Wert 1 auf, womit eine Kante die Knoten 1 und 2 verbindet. Der zweite Eintrag beträgt 5, was eine Kante zwischen den Knoten 3 und 5 anzeigt. Die sieben Einträge von `$dist` können dann den einzelnen Kanten aus `$kid` direkt zugeordnet werden. Beispielsweise bezieht sich die erste Bewertung auf den ersten Eintrag der Kantenliste. Somit beträgt der Abstand der Knoten 2 und 1 etwa 0.155.

Um eine grafische Darstellung zu erhalten, reicht es, das Objekt `baum` mit dem Befehl `plot` zu kombinieren. Im Beispiel werden zuvor allerdings

[27] Berechnet mit dem Skript `gm1a.cf`.

[28] Berechnet mit dem Skript `gm1b.cf`.

Box 6.3-2 R-Befehle für die Berechnung eines Minimalbaums für die Berufsstrukturdaten.

```
> dat <- read.table("http://www.stat.rub.de/teaching/drk/bs1.dat")
> names(dat) <- c("X","Y","Z","h")

> tab <- xtabs(h~.,dat)
> tab <- ftable(tab,row.vars="X",col.vars=c("Y","Z"))
> tab <- prop.table(tab,1)

> d <- dist(tab,method="manhattan")*0.5

# minimal spanning tree
> library(vegan)
> baum <- spantree(d)
> baum
$kid
[1] 1 5 5 8 4 4 2

$dist
[1] 0.15517386 0.05213896 0.10273233 0.13411477 0.16643096 0.10016099 0.16529048

$labels
NULL

$call
spantree(d = d)

attr(,"class")
[1] "spantree"

# Visualisierung
> baum$labels <- c("Turkey","Greece"," Switzerland",
+ "Great Britain","Germany","Sweden","USA"," Japan")
> plot(baum,type ="t",xlab="",ylab="")
Initial stress        : 0.00913
stress after  10 iters: 0.00457, magic = 0.500
stress after  20 iters: 0.00443, magic = 0.500
stress after  30 iters: 0.00441, magic = 0.500

# Implizierte Abst"ande
> d2 <- cophenetic(baum)
> d2
           1          2          3          4          5          6          7
2 0.15517386
3 0.50671807 0.35154421
4 0.55731144 0.40213758 0.15487128
5 0.45457911 0.29940525 0.05213896 0.10273233
6 0.72374240 0.56856854 0.32130224 0.16643096 0.26916329
7 0.65747244 0.50229857 0.25503228 0.10016099 0.20289332 0.26659196
8 0.32046434 0.16529048 0.18625373 0.23684710 0.13411477 0.40327806 0.33700810

> sqrt(sum((as.matrix(d)-as.matrix(d2))^2))
[1] 1.038197
> sqrt(sum((as.matrix(d2))^2))
[1] 2.70097
> sqrt(sum((as.matrix(d))^2))
[1] 1.731706
```

noch Bezeichnungen für die einzelnen Beobachtungen vergeben und beim Aufruf von `plot` wird über das Argument `type="t"` gesteuert, dass diese Bezeichnungen in der Grafik verwendet werden. Das Resultat sieht man

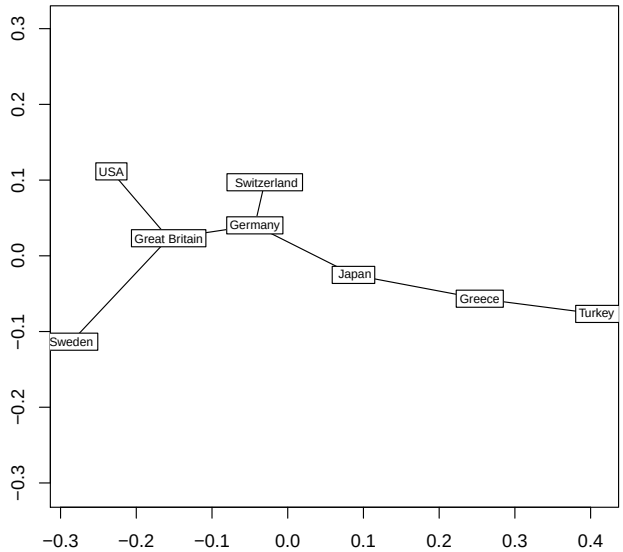


Abb. 6.3-2 Mit R (Box 6.3-2) erzeugter Minimalbaum für die Berufsstrukturdaten.

in Abbildung 6.3-2. Bei der Erzeugung der Grafik werden in einem ersten Schritt die durch den Baum implizierten Abstände zwischen den Beobachtungen errechnet. Anschließend wird über eine Variante der MDS eine zweidimensionale Konfiguration gesucht, die die implizierten Abstände möglichst gut repräsentiert. Der Wert der Stressfunktion ausgewählter Iterationsschritte der MDS wird nach dem Aufruf des `plot`-Befehls ausgegeben.

Die durch den Minimalbaum implizierten Abstände lassen sich über den Befehl `cophenetic` errechnen. Vergleicht man das Resultat aus Box 6.3-2 mit dem aus Tabelle 6.3-1, zeigen sich geringfügige Differenzen. Berechnet man aber wie am Ende der Box $||\mathbf{D}^v - \mathbf{D}||$, $||\mathbf{D}^v||$ und $||\mathbf{D}||$, zeigen sich praktisch die gleichen Ergebnisse wie in Abschnitt 6.2-3.

4. Hierarchische Klassifikationsschemas

Bei einfachen Baummodellen werden nur die Elemente einer vorgegebenen Objektmenge Ω als Knoten verwendet. Allgemeinere Baummodelle entstehen, wenn man Bäume verwendet, die darüber hinaus noch andere Knoten enthalten. Bei einem oft verwendeten Ansatz wird ein Baum verwendet, dessen Blätter den vorgegebenen Objekten entsprechen; dann werden interne Knoten hinzugefügt, so dass man eine möglichst gute Repräsentation der für die Objekte gegebenen Abstandsmatrix $\mathbf{D}$ durch die

durch den Baum induzierte Abstandsmatrix $\mathbf{D}^v$ erreicht.[29] Einen Spezialfall bilden hierarchische Klassifikationsschemas (HKS), mit denen wir uns im Folgenden beschäftigen.[30]

Wie bisher nehmen wir an, dass eine Objektmenge $\Omega = \{\omega_1, \ldots, \omega_n\}$ mit einer Abstandsmatrix $\mathbf{D} = (d_{ij})$ gegeben ist. Unter einem *hierarchischen Klassifikationsschema* (HKS) für Ω verstehen wir eine Hierarchie, für deren Elemente eine Indexfunktion definiert ist. Für die Hierarchie verwenden wir die Notation

$$\mathcal{H} := \{C_1, \ldots, C_n, \ldots, C_q\}$$

Die Elemente sind Teilmengen von Ω, und wir nehmen an, dass $C_i = \{\omega_i\}$ für $i = 1, \ldots, n$ und $C_q = \{\Omega\}$ ist. Außerdem gibt es für jedes Element $C \in \mathcal{H}$ einen Wert $\sigma(C) \geq 0$, wobei gilt, dass $\sigma(C_i) = 0$ für $i = 1, \ldots, n$. Die Funktion σ wird als *Indexfunktion* (der Hierarchie) bezeichnet.[31]

Wie diese Indexwerte entstehen, hängt vom Konstruktionsverfahren ab. Davon hängt auch ab, ob eine Indexfunktion entsteht, die folgende Monotoniebedingung erfüllt:

$$C \subset C' \implies \sigma(C) < \sigma(C') \tag{6.8}$$

Sie ist zum Beispiel erfüllt, wenn ein HKS mit der Single-Link-Methode konstruiert wird, bei einigen anderen Verfahren können auch nichtmonotone Indexfunktionen entstehen. Wir nehmen für die weiteren Überlegungen an, dass die Monotoniebedingung (6.8) erfüllt ist.

Ist nun ein HKS $\mathcal{H}$ mit einer Indexfunktion σ gegeben, kann ein Baum mit der Knotenmenge $\mathcal{H}$ gebildet werden, dessen Blätter den Elementen $C_1, \ldots, C_n$ entsprechen und dessen Wurzel dem Element C_q entspricht. Mit der Indexfunktion kann auch eine Abstandsfunktion für den Baum definiert werden. Zunächst wird eine Bewertungsfunktion v für die Kanten des Baums festgelegt. Wenn eine Kante l die Knoten C_i und C_j verbindet, kann man annehmen, dass C_j auf einem Pfad liegt, der von C_i zur Wurzel C_q führt; also liefert die Definition

$$v(l) := \sigma(C_j) - \sigma(C_i)$$

eine Bewertungsfunktion, die jeder Kante eine positive Bewertung zuordnet. Wie in Abschnitt 6.3-2 besprochen wurde, erhält man dann auch eine Abstandsfunktion, die je zwei Knoten C_i und C_j einen Abstand $d^v(C_i, C_j)$ zuordnet, der die Summe der Bewertungen der Kanten auf dem Pfad von C_i nach C_j angibt.

[29] Eine gute Darstellung dieses Ansatzes geben Barthélemy und Guénoche (1991).

[30] Die erste ausführliche Untersuchung stammt von Johnson (1967). Eine gute neuere Darstellung findet man bei Jain und Dubes (1988: 58ff.).

[31] Im Englischen: *indexed hierarchy*.

5. Darstellung durch Dendrogramme

Der Baum, der durch ein HKS entsteht, hat eine besondere Eigenschaft: Alle Blätter haben von der Wurzel den gleichen Abstand. Ein solcher Baum wird auch als ein *Dendrogramm* bezeichnet. Ein Dendrogramm kann auch dadurch charakterisiert werden, dass die durch seine Levelfunktion definierte Abstandsfunktion d^v folgende Bedingung erfüllt:

$$\text{Für alle } i, j, k\text{: } d^v_{ij} \leq \max\{d^v_{ik}, d^v_{jk}\} \tag{6.9}$$

Eine Abstandsfunktion, die diese Bedingung erfüllt, wird *ultrametrisch* genannt (sie impliziert die Dreiecksungleichung).

Bis auf die UPGMC-Methode liefern alle in Box 6.2-1 genannten SAHN-Algorithmen ultrametrische Abstandsfunktionen.[32] Als Beispiel kann man an die Konstruktion eines HKS für die Berufsstrukturdaten in Abschnitt 6.2-3 denken. Box 6.2-3 zeigt die Indexfunktion, Abbildung 6.2-1 zeigt das Dendrogramm. In diesem Beispiel hat der Baum $q = 15$ Knoten; die Knoten $C_1, \ldots, C_8$ entsprechen den Ländern, C_{15} ist die Gesamtmenge der Länder.

Eine bemerkenswerte Implikation ultrametrischer Abstandsfunktionen zeigt sich, wenn das Dendrogramm verwendet wird, um eine Zerlegung der Objektmenge in Cluster zu konstruieren (vgl. Abschnitt 6.2-6). Der maximale ultrametrische Abstand zwischen den Objekten innerhalb eines Clusters ist dann kleiner als der minimale ultrametrische Abstand zwischen Objekten, die verschiedenen Clustern angehören. Dies gilt zwar nur für die ultrametrischen Abstände; lieferten sie jedoch eine gute Approximation der ursprünglichen Abstände ($\mathbf{D}$), entstünden tatsächlich S-Cluster (im Sinne der in Abschnitt **??** gegebenen Definition).

6. Alternative Abstandsberechnung

Wenn das Baummodell durch ein Dendrogramm gegeben ist, kann alternativ zur Abstandsfunktion d^v_{ij} (Summe der Kantenbewertungen des Pfads von i nach j) auch eine Abstandsfunktion

$$d^h_{ij} := \text{ Index des kleinsten Clusters, das } i \text{ und } j \text{ enthält}$$

gebildet werden.[33] Schränkt man die Abstandsfunktion auf die ursprünglichen Objekte (die Blätter des Baums) ein, gilt offenbar: $d^v_{ij} = 2\, d^h_{ij}$. Tabelle 6.3-2 zeigt die so eingeschränkte Abstandsmatrix $\mathbf{D}^h = (d^h_{ij})$ für das Beispiel. Man erkennt, dass es wiederum nur sieben $(n-1)$ unterschiedliche Abstandswerte gibt.

[32] Vgl. Milligan (1979).

[33] Diese Abstandsdefinition wurde bereits von Johnson (1967: 244) vorgeschlagen, und sie wird auch meistens von Computerprogrammen für hierarchische Clusteranalysen verwendet; dies gilt auch für die hier verwendete TDA-Prozedur `hcls`. Vgl. auch die Hinweise bei Corter (1996: 15f.).

Tabelle 6.3-2 Aus dem mit der Single Link-Methode erzeugten Dendrogramm gewonnene Abstandsmatrix D^h für die acht Länder.

0.0000	0.1551	0.1652	0.1652	0.1652	0.1664	0.1652	0.1652
0.1551	0.0000	0.1652	0.1652	0.1652	0.1664	0.1652	0.1652
0.1652	0.1652	0.0000	0.1027	0.0520	0.1664	0.1027	0.1341
0.1652	0.1652	0.1027	0.0000	0.1027	0.1664	0.1002	0.1341
0.1652	0.1652	0.0520	0.1027	0.0000	0.1664	0.1027	0.1341
0.1664	0.1664	0.1664	0.1664	0.1664	0.0000	0.1664	0.1664
0.1652	0.1652	0.1027	0.1002	0.1027	0.1664	0.0000	0.1341
0.1652	0.1652	0.1341	0.1341	0.1341	0.1664	0.1341	0.0000

Tabelle 6.3-3 Vergleich der unterschiedlichen SAHN-Verfahren für die Berufsstrukturdaten.

Methode	$\|\mathbf{D}-\mathbf{D}^h\|$
Single Link	0.7580
Complete Link	0.8901
Weighted Average	0.4579
Weighted Centroid	0.7260
Group Average	0.4380
Unweighted Centroid	0.7059
Ward's Minimum Variance	1.4246

Analog zur Vorgehensweise in Abschnitt 6.3-2 können auch diese Abstandsmatrizen mit der vorgegebenen Abstandsmatrix $\mathbf{D}$ verglichen werden, um den Grad ihrer Repräsentation durch das hierarchische Klassifikationsschema zu erfassen. Tabelle 6.3-3 vergleicht die unterschiedlichen SAHN-Verfahren für die Berufsstrukturdaten.

7. Optimale ultrametrische Modelle

Hierarchische Klassifikationsverfahren liefern eine Abstandsmatrix $\mathbf{D}^h$, die dann durch $\|\mathbf{D}-\mathbf{D}^h\|$ mit der vorgegebenen Abstandsmatrix $\mathbf{D}$ verglichen werden kann. Offenbar gelangt man zu einem im Sinne dieses Kriteriums optimalen Modell, wenn man folgende Aufgabe löst: Finde eine ultrametrische Abstandsmatrix $\mathbf{D}^u$, die den Wert des Kriteriums $\|\mathbf{D}-\mathbf{D}^u\|$ minimal macht.

Für die Lösung dieser Aufgabe sind verschiedene Algorithmen vorgeschlagen worden.[34] Wir verwenden zur Illustration wiederum die Berufsstrukturdaten. Tabelle 6.3-4 zeigt für dieses Beispiel die im Sinne des LS-Kriterums optimale ultrametrische Abstandsmatrix $\mathbf{D}^u$.[35] Das Kriterium hat in diesem Fall den Wert $\|\mathbf{D}-\mathbf{D}^u\|=0.39$, offenbar niedriger als die in Tabelle 6.3-3 angegebenen Werte, die mit den SAHN-Algorithmen erzielt werden konnten.

[34] Vgl. De Soete (1984a, 1984b, 1988); De Soete, Carroll und De Sarbo, W. S. (1987); Sriram und Lewis (1993); Corter (1996: 26ff.).

[35] Für die Berechnung wurde die TDA-Prozedur `clu` verwendet, die auf einem von De Soete (1984a) vorgeschlagenen Algorithmus basiert. Das Skript ist `hca6.cf`.

Tabelle 6.3-4 Optimale ultrametrische Abstandsmatrix $\mathbf{D}^u$ für die Berufsstrukturdaten.

0.0000	0.3282	0.3282	0.3282	0.3282	0.3282	0.3282	0.3282
0.3282	0.0000	0.2184	0.2184	0.2184	0.2184	0.2184	0.2184
0.3282	0.2184	0.0000	0.1371	0.0517	0.1975	0.1371	0.1815
0.3282	0.2184	0.1371	0.0000	0.1371	0.1975	0.1004	0.1815
0.3282	0.2184	0.0517	0.1371	0.0000	0.1975	0.1371	0.1815
0.3282	0.2184	0.1975	0.1975	0.1975	0.0000	0.1975	0.1975
0.3282	0.2184	0.1371	0.1004	0.1371	0.1975	0.0000	0.1815
0.3282	0.2184	0.1815	0.1815	0.1815	0.1975	0.1815	0.0000

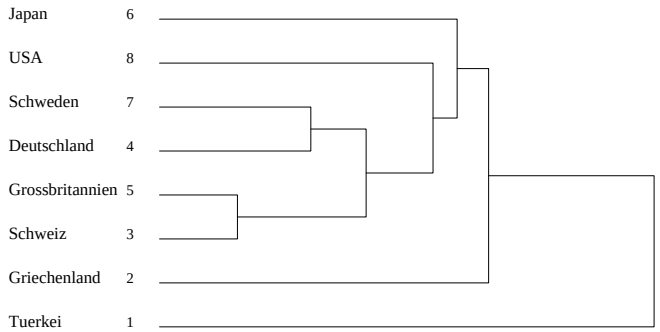


Abb. 6.3-3 Das der ultrametrischen Abstandsmatrix $\mathbf{D}^u$ in Tabelle 6.3-4 entsprechende Dendrogramm.

Da die Abstandsmatrix $\mathbf{D}^u$ ultrametrisch ist, kann sie auch durch ein Dendrogramm dargestellt werden. Das Dendrogramm kann beispielsweise dadurch erzeugt werden, dass man $\mathbf{D}^u$ als Ausgangsmatrix für ein hierarchisches Klassifikationsverfahren mit der Single-Link-Methode verwendet. Abbildung 6.3-3 zeigt das Ergebnis für das Beispiel.

8. Ultrametrische Modelle in R

In `R` können (optimale) ultrametrische Modelle mit dem Paket `clue` berechnet werden. Zur Illustration verwenden wir wieder die Berufsstrukturdaten, wobei davon ausgegangen wird, dass sich die Abstandsmatrix mit dem Dissimilaritätsindex zwischen den Ländern aus dem letzten Beispiel noch unter dem Namen `d` im Speicher befindet. Box 6.3-3 zeigt die

Box 6.3-3 R-Befehle für die Berechnung eines ultrametrischen Modells für die Berufsstrukturdaten.

```
> cl <- hclust(d,method="single")

> library(clue)
> um <- cl_ultrametric(cl)
> um

Dissimilarities using Ultrametric distances:
     [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
[1,] 0.15517386
[2,] 0.16529048 0.16529048
[3,] 0.16529048 0.16529048 0.10273233
[4,] 0.16529048 0.16529048 0.05213896 0.10273233
[5,] 0.16643096 0.16643096 0.16643096 0.16643096 0.16643096
[6,] 0.16529048 0.16529048 0.10273233 0.10016099 0.10273233 0.16643096
[7,] 0.16529048 0.16529048 0.13411477 0.13411477 0.13411477 0.16643096
     [,7]
[1,]
[2,]
[3,]
[4,]
[5,]
[6,]
[7,] 0.13411477

> sqrt(sum((as.matrix(d)-as.matrix(um))^2))
[1] 0.7582062

> oum <- ls_fit_ultrametric(d,control=list(nruns=200))
> oum

Dissimilarities using Ultrametric distances:
     [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
[1,] 0.32671483
[2,] 0.32671483 0.22002397
[3,] 0.32671483 0.22002397 0.10772872
[4,] 0.32671483 0.22002397 0.05213896 0.10772872
[5,] 0.32671483 0.22002397 0.19251463 0.19251463 0.19251463
[6,] 0.32671483 0.22002397 0.13422701 0.13422701 0.13422701 0.19251463
[7,] 0.32671483 0.22002397 0.17408222 0.17408222 0.17408222 0.19251463
     [,7]
[1,]
[2,]
[3,]
[4,]
[5,]
[6,]
[7,] 0.17408222

> sqrt(sum((as.matrix(d)-as.matrix(oum))^2))
[1] 0.3878994
```


Vorgehensweise.

Als erstes wird eine hierarchische Klassifikation mit der Single Link Methode durchgeführt. Das Ergebnis wird unter dem Namen `cl` abgespeichert und an den Befehl `cl_ultrametric` übergeben, nachdem das Paket `clue` geladen wurde. Ruft man das mit diesem Befehl erzeugte Ergebnis auf, erhält man Tabelle 6.3-2, also die durch das hierarchische Verfahren implizierte ultrametrische Abstandsmatrix. Durch den hierauf folgenden Befehl wird der in Tabelle 6.3-3 angegebene Abstand zwischen der ursprünglichen und der ultrametrischen Abstandsmatrix berechnet.

Um eine optimale ultrametrische Abstandsmatrix zu suchen, kann der Befehl `ls_fit_ultrametric` benutzt werden. Dieser setzt ein von De Soete (1984b) vorgeschlagenes Verfahren um. Bei diesem wird ausgehend von zufälligen gebildeten Abstandsmatrizen nach einer optimalen ultrametrischen Abstandsmatrix gesucht. Daher empfiehlt es sich, ausgehend von möglichst vielen unterschiedlichen Matrizen nach einer Lösung zu suchen. Dies wird in unserem Beispiel mit dem Argument `control=list(nruns=200)` erreicht, womit insgesamt 200 zufällige Matrizen verwendet werden. Ruft man das Ergebnis auf, erhält man von kleineren Abweichungen abgesehen Tabelle 6.3-4. Der Abstand $||\mathbf{D} - \mathbf{D}^u||$, der abschließend berechnet wird, beträgt wie in Abschnitt 6.3-6 etwa 0.39.