

# Seminar zum Thema Künstliche Intelligenz: Clusteranalyse

Wolfgang Ginolas

11.5.2005

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>4</b>
1.1	Ein einführendes Beispiel . . . . .	4
1.2	Definition der Clusteranalyse . . . . .	4
1.3	Allgemeines Vorgehen bei der Clusteranalyse . . . . .	6
1.4	Anwendungen der Clusteranalyse . . . . .	6
1.5	Ein einfacher Algorithmus . . . . .	6
<b>2</b>	<b>Distanzen</b>	<b>7</b>
2.1	Metrik . . . . .	7
2.2	Distanzen bei verschiedenen Datentypen . . . . .	7
2.2.1	Metrische Daten . . . . .	7
2.2.2	Ordinale Daten . . . . .	7
2.2.3	Nominale Daten . . . . .	8
2.3	Merkmalsvektoren . . . . .	8
2.3.1	Die Euklidische Distanz . . . . .	8
2.3.2	Häuserblockmetrik . . . . .	8
2.3.3	Maximum-Abstand . . . . .	8
2.3.4	Normalisieren . . . . .	9
2.3.5	Normalisierung der Distanzmatrix . . . . .	9
2.3.6	Gewichten . . . . .	9
2.3.7	Fehlwerte . . . . .	10
2.4	Abstände zwischen Clustern . . . . .	10
2.4.1	Zentroid Verfahren . . . . .	10
2.4.2	Average-Linkage Verfahren . . . . .	12
2.4.3	Single-Linkage Verfahren . . . . .	12
2.4.4	Complete-Linkage Verfahren . . . . .	12
<b>3</b>	<b>Verschiedene Algorithmen</b>	<b>15</b>
3.1	Hierarchische Klassifikation . . . . .	15
3.1.1	Agglomerativ . . . . .	15
3.1.2	Divisiv . . . . .	17
3.2	Disjunkte Klassifikation . . . . .	19
3.2.1	k-means . . . . .	19

3.3	Unscharfe Klassifikation . . . . .	19
3.4	Self-Organizing Maps . . . . .	19
3.4.1	Anwenden von Self-Organizing Maps . . . . .	21
3.4.2	Trainieren einer Self-Organizing Map . . . . .	21
3.4.3	Anwendung von Self-Organizing Maps in der Clusteranalyse .	22
3.4.4	Self-Organizing Maps zur Analyse von Musik . . . . .	22
<b>4</b>	<b>Quellen</b>	<b>22</b>

# 1 Einleitung

Die Clusteranalyse ist ein Verfahren, um Gruppen (Cluster) zusammengehöriger Objekte in einer Menge von numerisch beschriebenen Objekten zu ermitteln.

## 1.1 Ein einführendes Beispiel

Die Funktionsweise der Clusteranalyse soll an einem einführenden Beispiel demonstriert werden.

Bei 20 Patienten wurde der „Gehalt alkalischer Phosphate“ und der „Eisengehalt“ gemessen:

Pat. Nr.	AP (U/l)	Fe (mg/l)	Pat. Nr.	AP (U/l)	Fe (mg/l)
1	4.0	1.0	11	2.0	0.7
2	3.0	1.7	12	1.2	0.5
3	2.6	1.8	13	4.5	0.7
4	1.5	0.7	14	2.5	3.0
5	2.5	2.2	15	3.5	0.7
6	1.1	1.0	16	2.2	3.2
7	2.8	3.1	17	2.1	3.5
8	1.7	3.2	18	2.1	2.0
9	0.8	0.5	19	3.5	1.2
10	2.1	3.0	20	3.2	1.2

Die einzelnen Patienten sind *Objekte*, die einzelne *Merkmale* haben (hier der Gehalt alkalischer Phosphate und Eisen). Diese Merkmale eines Objektes können in einem *Merkmalsvektor* zusammengefasst werden. Die Merkmalsvektoren haben hier zwei Werte, so dass sie in ein Koordinatensystem (Abbildung 1) gezeichnet werden können.

Wie in Abbildung 2 zu sehen, können die 20 Patienten vier Gruppen (Clustern) zugeordnet werden. Die Entscheidung, welcher Patient zu welchem Cluster gehört, kann mittels Clusteranalyse getroffen werden.

Über die Bedeutung der einzelnen Cluster (hier „Hepatitis“, „Leberzirrhose“, „Normal“ und „Verschlussikterus“) macht die Clusteranalyse keine Aussage.

## 1.2 Definition der Clusteranalyse

**Gegeben:** Stichprobe von Objekten, die sich in eine noch unbekannt Anzahl von Gruppen ähnlicher Objekte unterteilen lässt.

**Gesucht:** Charakterisierung dieser potentiellen Gruppen und die Angabe, welches Objekt welcher Gruppe zugewiesen ist.

**Lösungsverfahren:** Clusteranalyse

Die verschiedenen Clusterverfahren kann man anhand der Ein- und Ausgabedaten der Algorithmen weiter unterteilen (siehe Abbildung 3).

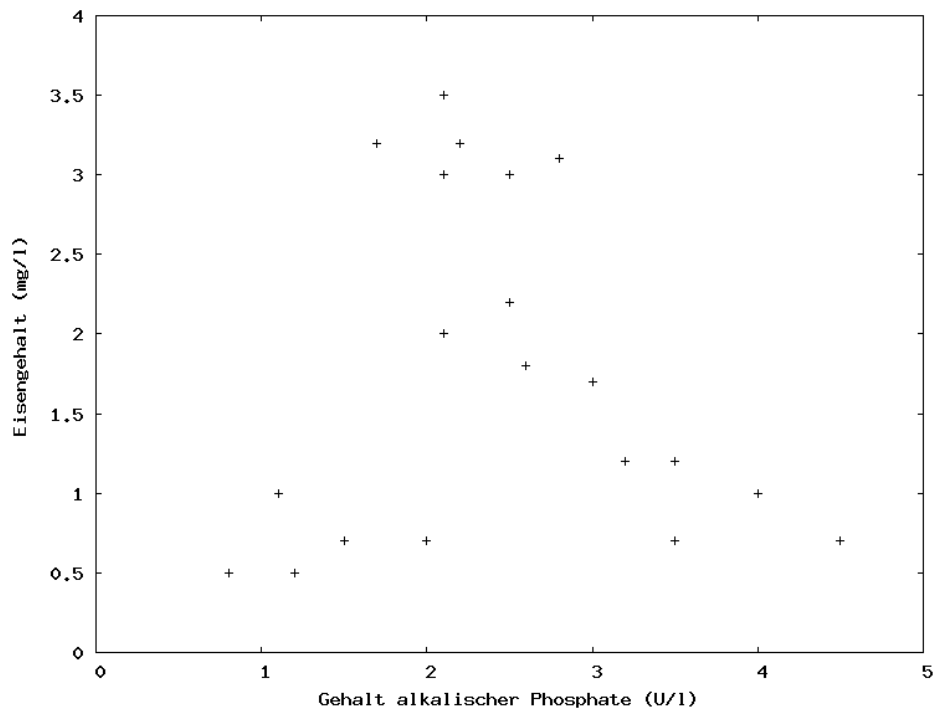


Abbildung 1: 20 Patienten

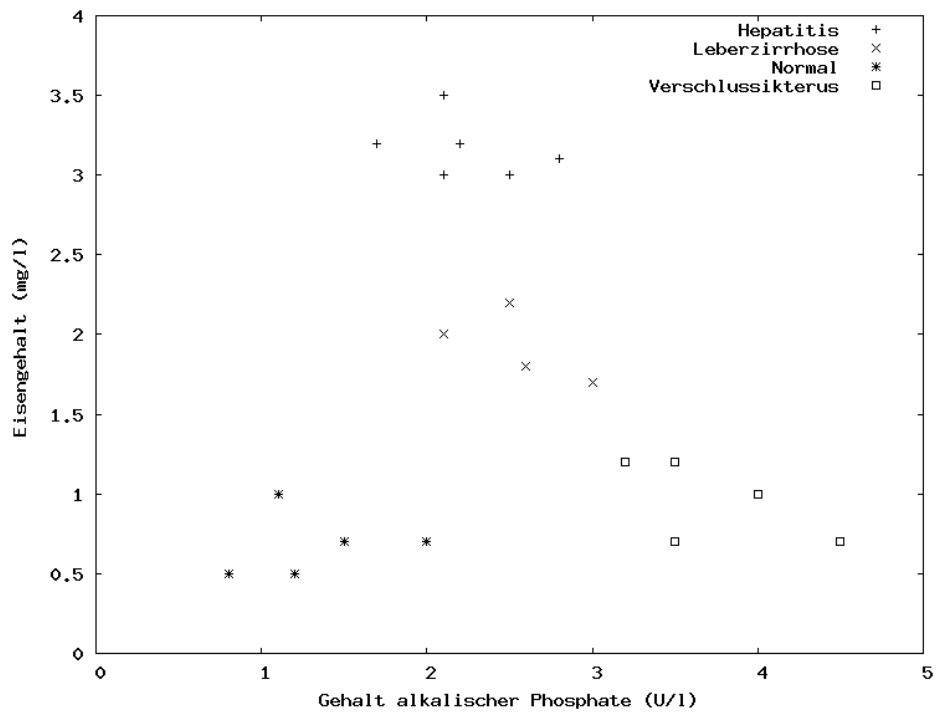


Abbildung 2: 20 Patienten Gruppen zugeordnet

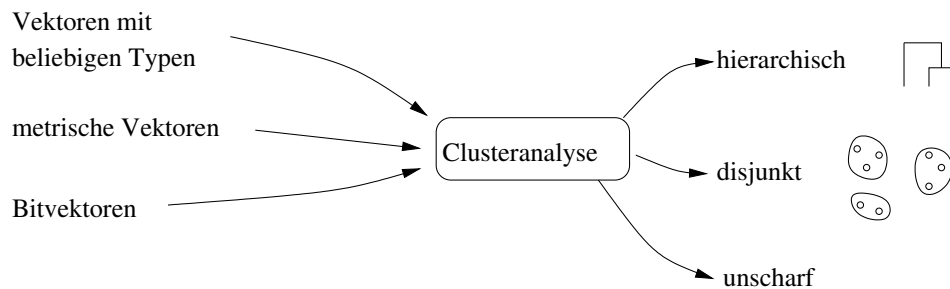


Abbildung 3: Ein- und Ausgabe bei der Clusteranalyse

Manche Algorithmen können Merkmalsvektoren mit beliebigen Datentypen verarbeiten. Andere allerdings brauchen reine metrische Vektoren oder Bitvektoren.

Die Ausgabe der Clusteranalyse, d.h. die Zuordnung von Objekten zu Clustern, kann hierarchisch, disjunkt oder unscharf sein.

### 1.3 Allgemeines Vorgehen bei der Clusteranalyse

**Definition der Objekte:** Zu allererst muss entschieden werden, welche Objekte mit der Clusteranalyse verarbeitet werden sollen.

**Auswahl und Aufbereitung der Merkmale:** Danach ist zu bestimmen, welche Merkmale der Objekte für die Clusteranalyse genutzt werden, um ein möglichst gutes Ergebnis zu erlangen. Außerdem ist unter Umständen eine Aufbereitung der Merkmale sinnvoll.

**Auswahl des Clusterverfahrens:** Bei der Wahl der Clusterverfahrens ist wichtig, welche Eingabedaten vorhanden sind und welche Ausgabedaten gewünscht sind. Außerdem gibt es bei allen Verfahren verschiedene Parameter (wie z.B. das Distanzmaß), die zu wählen sind.

**Die Interpretation der Ergebnisse:** Nach der Clusteranalyse sind die Ergebnisse zu interpretieren und eventuell Änderungen an dem Clusterverfahren oder den Merkmalen vorzunehmen, um das Ergebnis zu verbessern.

### 1.4 Anwendungen der Clusteranalyse

Die Clusteranalyse ist ein wichtiges Werkzeug beim Data-Mining. Außerdem wird sie z.B. oft in Bereichen wie der Medizin oder der Bildverarbeitung zur Mustererkennung verwendet. Zwei konkrete Beispiele aus der Computergrafik und der Analyse von Musik werden in den Abschnitten 3.1.2 und 3.4.4 beschrieben.

### 1.5 Ein einfacher Algorithmus

Um die verschiedenen Aspekte der Clusteranalyse im Folgenden besser erläutern zu können, wird hier ein einfacher Algorithmus zur Clusteranalyse vorgestellt.

1. Alle Objekte bilden jeweils einen eigenen Cluster.
2. Die beiden Cluster, die sich am nächsten sind, werden vereinigt.
3. Wiederhole 2, bis die gewünschte Clusteranzahl erreicht ist.

Es stellt sich nun die Frage, was „sich am nächsten“ konkret bedeutet. Dies wird im Folgenden Kapitel erläutert.

## 2 Distanzen

### 2.1 Metrik

Um Abstände zu bestimmen, muss eine Metrik definiert werden. Ein Distanzmaß  $d$  muss dafür folgende Eigenschaften erfüllen:

1.  $d(x, y) \geq 0$
2.  $d(x, y) = 0 \iff x = y$
3.  $d(x, y) = d(y, x)$
4.  $d(x, y) \leq d(x, z) + d(z, y)$

### 2.2 Distanzen bei verschiedenen Datentypen

Um später den Abstand zwischen Clustern bestimmen zu können, müssen sich zunächst die Distanzen bei Merkmalen ermitteln lassen. Das Distanzmaß hängt hierbei von dem Datentyp des Merkmals ab. Es lässt sich für praktisch jeden Datentyp ein Distanzmaß definieren. Exemplarisch wird hier auf die drei, wohl am häufigsten verwendeten, Typen eingegangen.

#### 2.2.1 Metrische Daten

Metrische oder Reelle Daten können in zwei Arten unterteilt werden:

**Intervallskaliert:** Intervallskalierte Daten haben keinen vorgegebenen Bezugs-/Nullpunkt (z.B. Temperatur, Datum).

**Verhältnisskaliert:** Verhältnisskalierte Daten haben einen vorgegebenen Nullpunkt (z.B. Länge, Gewicht).

Bei beiden Arten kann die Distanz mit  $d_m(x, y) = |x - y|$  definiert werden.

#### 2.2.2 Ordinale Daten

Um den Abstand bei Ordinalen Daten zu ermitteln, können sie in Metrische Daten umgewandelt werden. Z.B.:

$Dialup = 0$      $ISDN = 1$      $Broadband = 2$      $Cable = 3$

### 2.2.3 Nominale Daten

Bei nominalen Typen lässt sich der Abstand z.B. durch Gleichheit und Ungleichheit definieren:

$$d_n(x, y) = \begin{cases} 0 & \text{für } x = y \\ 1 & \text{für } x \neq y \end{cases}$$

## 2.3 Merkmalsvektoren

Fasst man mehrere Merkmale in einem Vektor zusammen, so erhält man einen *Merkmalsvektor*. Für die Clusteranalyse sollte man beachten, dass Merkmale, die voneinander abhängen (z.B. Körpergewicht und Körpergröße einer Person), ungeeignet für die Clusteranalyse sind, da dadurch langgestreckte Cluster entstehen. Solche Cluster, wie wir später noch sehen werden, sind schlecht zu erkennen.

Außerdem sollten Merkmale, die sich gegenseitig ausschließen (z.B. „Treiben Sie Sport“, „Spielen Sie Tennis“), vermieden werden.

Um den Abstand zwischen Merkmalsvektoren zu bestimmen, gibt es eine Vielzahl von Verfahren. Drei werden im Folgenden beschrieben.

### 2.3.1 Die Euklidische Distanz

Die Euklidische Distanz ist definiert durch:

$$d_e(x, y) = \sqrt{\sum_i d_i(x_i, y_i)^2}$$

$d_i$  ist das Distanzmaß des Merkmales  $i$ .  $x_i$  und  $y_i$  sind die Werte des Merkmals  $i$  in den Merkmalsvektoren  $x$  und  $y$ .

### 2.3.2 Häuserblockmetrik

Bei der Häuserblockmetrik werden die Abstände der Merkmale aufsummiert:

$$d_h(x, y) = \sum_i d_i(x_i, y_i)$$

### 2.3.3 Maximum-Abstand

Bei dem Maximum-Abstand entspricht die Distanz zwischen zwei Merkmalsvektoren der größten Distanz zwischen zwei Merkmalen:

$$d_m(x, y) = \max(d_1(x_1, y_1), d_2(x_2, y_2), \dots, d_n(x_n, y_n))$$

### 2.3.4 Normalisieren

Alle drei oben beschriebenen Verfahren sind nicht *skaleninvariant*, d.h. ändert man z.B. die Maßeinheit eines Merkmals (Gramm  $\Leftrightarrow$  Kilogramm), so verschieben sich auch die Distanzen. Außerdem beeinflussen Merkmale mit einem großen Wertebereich die Distanz mehr, als Merkmale mit einem kleinen Wertebereich. Um das zu verhindern, ist eine Normalisierung erforderlich, so dass alle Merkmale den gleichen Wertebereich haben. Eine Möglichkeit dies zu erreichen ist die *z-Transformation*.

Bei der z-Transformation wird von allen Werten  $x_i$  eines Merkmals der Mittelwert  $\bar{x}$  und die Standardabweichung  $\sigma x$  berechnet. Der neue Wert  $x'_i$  ergibt sich aus folgenden Formeln:

$$\begin{aligned}\bar{x} &= \frac{1}{N} \cdot \sum_{i=1}^N x_i \\ \sigma x &= \frac{1}{N} \cdot \sum_{i=1}^N (x_i - \bar{x})^2 \\ x'_i &= \frac{x_i - \bar{x}}{\sigma x}\end{aligned}$$

### 2.3.5 Normalisierung der Distanzmatrix

Die Normalisierung der Distanzmatrix ist im Gegensatz zur z-Transformation, die nur Metrische Daten normalisieren kann, bei allen Datentypen möglich. Zuerst muss für jedes Merkmal eine Distanzmatrix aufgestellt werden. Im Beispiel wird nur eine Matrix für das Merkmal M1 der Objekte 1-3 aufgestellt:

$d_{M1}$	Objekt 1	Objekt 2	Objekt 3
Objekt 1	0	1	2
Objekt 2	1	0	1
Objekt 3	2	1	0

Danach muss der größte Wert in der Matrix ermittelt werden (hier:  $d_{M1Max} = 2$ ). Nun erhält man die normalisierte Matrix, indem man die Distanzmatrix durch den größten Wert teilt:

$$\begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix} \cdot \frac{1}{2} = \begin{pmatrix} 0 & \frac{1}{2} & 1 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 1 & \frac{1}{2} & 0 \end{pmatrix}$$

Wie man sieht, ist die größte Distanz bei dem Merkmal M1 nun maximal 1. Führt man diese Normalisierung für alle Merkmale durch, so ist der Wertebereich der Distanzen aller Merkmale zwischen 0 und 1.

### 2.3.6 Gewichten

Manchmal weiß man, dass bestimmte Merkmale eine bessere Aussagefähigkeit haben als andere. In diesem Fall ist es möglich die Normalisierung teilweise rückgängig zu

machen, indem die Merkmale selber oder die Distanzmatrizen der Merkmale mit einem konstanten Faktor multipliziert werden.

Außerdem ist es möglich einzelne Objekte zu gewichten, falls diese z.B. eine besonders große oder kleine Aussagefähigkeit haben. Dies muss allerdings vom Clusterverfahren ermöglicht werden.

### 2.3.7 Fehlwerte

Es kann vorkommen, dass bestimmte Merkmale eines Objektes nicht bekannt sind, wenn sie außerhalb des Messbereiches liegen oder aus anderen Gründen nicht ermittelt werden konnten. Solche Fehlwerte können z.B. bei der Euklidischen Distanz berücksichtigt werden, indem das entsprechende Merkmal bei der Summenbildung weggelassen wird:

$$x = (1, 1) \quad y = (2, *) \quad z = (4, 5)$$

$$\begin{aligned} d(x, y) &= \sqrt{(1-2)^2} &= 1 \\ d(x, z) &= \sqrt{(1-4)^2 + (1-5)^2} &= 5 \\ d(y, z) &= \sqrt{(2-4)^2} &= 2 \end{aligned}$$

Es ist zu beachten, dass man nun keine Metrik mehr hat, da  $d(x, z) > d(x, y) + d(y, z)$ . Es gibt Distanzmaße, die Fehlwerte besser verarbeiten können.

## 2.4 Abstände zwischen Clustern

Wie bei den Abständen zwischen Vektoren gibt es auch hier eine Vielzahl von Möglichkeiten, den Abstand zwischen Clustern zu bestimmen. Vier einfache Verfahren werden hier beschrieben.

### 2.4.1 Zentroid Verfahren

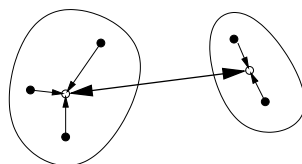


Abbildung 4: Zentroid Verfahren

Bei dem Zentroid Verfahren wird für jeden Cluster der Schwerpunkt berechnet und die Distanz zwischen den Clustern ermittelt (siehe Abbildung 4). Natürlich ist der Schwerpunkt nur bei metrischen Daten berechenbar.

Wie in Abbildung 5 zu sehen, werden die 3 Cluster richtig erkannt. Auch mit Störungen wie Ausreißern (Abbildung 6) oder einer Brücke (Abbildung 7) kommt das Zentroid Verfahren zurecht. Der Ring (Abbildung 8) allerdings wird nicht korrekt erkannt. Der Ring ist allgemein bei der Clusteranalyse ein Problem, wie wir noch sehen werden.

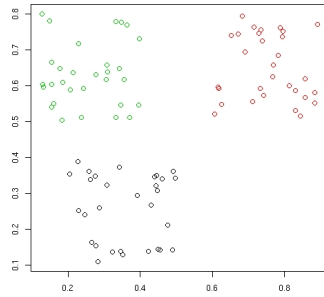


Abbildung 5: Zentroid Verfahren: 3 Cluster

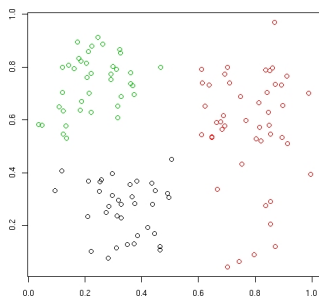


Abbildung 6: Zentroid Verfahren: 3 Cluster mit Ausreißern

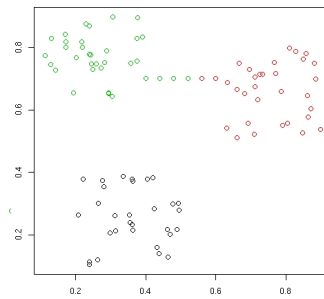


Abbildung 7: Zentroid Verfahren: 3 Cluster mit Brücke

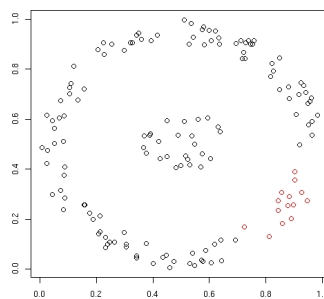


Abbildung 8: Zentroid Verfahren: Ring

### 2.4.2 Average-Linkage Verfahren

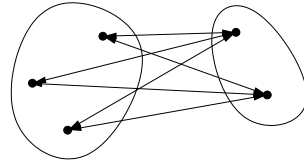


Abbildung 9: Average-Linkage Verfahren

Bei dem Average-Linkage Verfahren werden alle paarweisen Distanzen zwischen den Objekten der Cluster berechnet (Abbildung 9). Der Mittelwert dieser Distanzen bildet den Abstand der Cluster.

Auch hier ist das Erkennen von Clustern (Abbildung 10) mit Ausreißern (Abbildung 11) oder einer Brücke (Abbildung 12) kein Problem. Und auch hier wird der Ring (Abbildung 13) nicht richtig erkannt.

### 2.4.3 Single-Linkage Verfahren

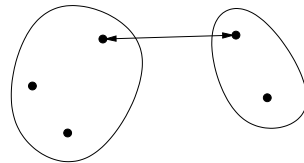


Abbildung 14: Single-Linkage Verfahren

Das Single-Linkage Verfahren nimmt die kürzeste aller paarweisen Distanzen zwischen den Objekten der Cluster als Abstand (Abbildung 14). Das Single-Linkage Verfahren ist das einzige hier vorgestellte, das den Ring (Abbildung 18) erkennen kann. Allerdings versagt es bei der Brücke (Abbildung 17) und den Ausreißern (Abbildung 16). Man kann sich vorstellen, dass der Ring auch nicht erkannt worden wäre, wenn es dort auch Ausreißer gegeben hätte.

### 2.4.4 Complete-Linkage Verfahren

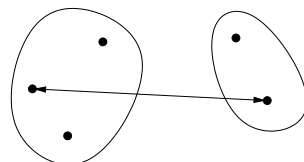


Abbildung 19: Complete-Linkage Verfahren

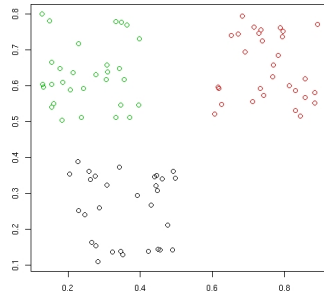


Abbildung 10: Average-Linkage Verfahren: 3 Cluster

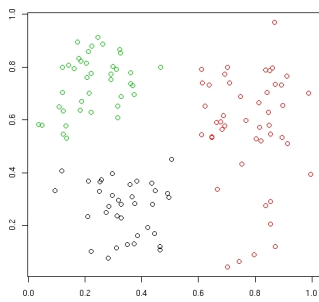


Abbildung 11: Average-Linkage Verfahren: 3 Cluster mit Ausreißern

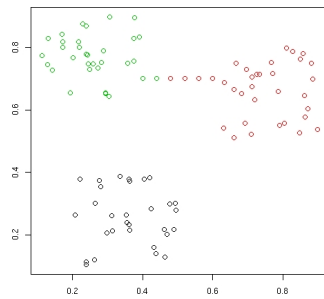


Abbildung 12: Average-Linkage Verfahren: 3 Cluster mit Brücke

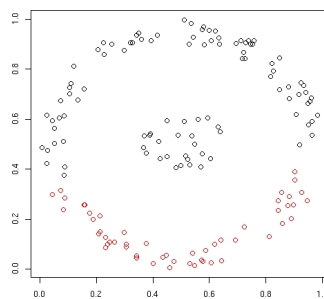


Abbildung 13: Average-Linkage Verfahren: Ring

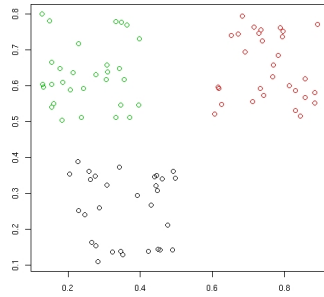


Abbildung 15: Single-Linkage Verfahren: 3 Cluster

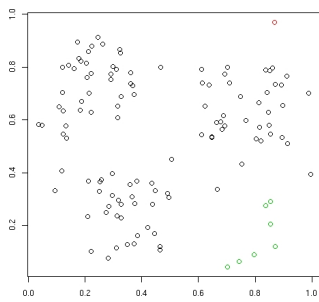


Abbildung 16: Single-Linkage Verfahren: 3 Cluster mit Ausreißern

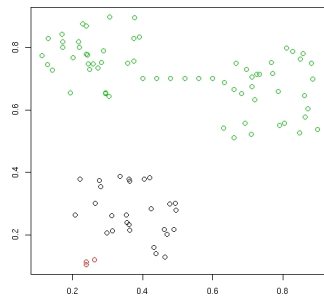


Abbildung 17: Single-Linkage Verfahren: 3 Cluster mit Brücke

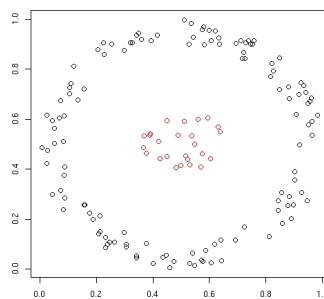


Abbildung 18: Single-Linkage Verfahren: Ring

Im Gegensatz zu dem Single-Linkage Verfahren, wird bei dem Complete-Linkage Verfahren nicht die kürzeste, sondern die längste Distanz als Abstand zwischen den Clustern genommen. Auch dieses Verfahren kann den Ring nicht erkennen (Abbildung 23). Außerdem kann man in Abbildung 21 sehen, dass in diesem Fall auch die Ausreißer Probleme machen.

## 3 Verschiedene Algorithmen

### 3.1 Hierarchische Klassifikation

Bei der Hierarchischen Klassifikation werden die Cluster in einer Hierarchie angeordnet. Diese Hierarchie kann man mit einer Art binären Baum beschreiben, den man hier *Dendrogramm* nennt. Die Blätter des Baumes sind die einzelnen Cluster. An der Höhe der Knoten lässt sich ablesen, wie weit die beiden Teilbäume (Teilcluster) voneinander entfernt sind.

In Abbildung 24 ist das Dendrogramm des einführenden Beispiels mit den 20 Patienten zu sehen. Man kann erkennen, dass z.B. die Patienten 10 und 16 sich sehr ähneln, weil ihr Knoten sehr niedrig (bei etwa 0.2) liegt. Trennt man das Dendrogramm auf einem Niveau von 1.75 auf, so entstehen 4 Teilbäume, die den 4 erkannten Clustern im einführenden Beispiel entsprechen.

Grundsätzlich gibt es bei der hierarchischen Klassifikation zwei Herangehensweisen.

#### 3.1.1 Agglomerativ

Die hierarchische Klassifikation nennt man *Agglomerativ*, wenn man die Objekte solange zu Clustern vereinigt, bis nur ein einziger Cluster übrigbleibt. Das Verfahren ähnelt sehr dem Beispiel in Abschnitt 1.5, nur dass hier ein Dendrogramm erzeugt wird:

1. Alle Objekte bilden jeweils einen eigenen Cluster.
2. Die beiden Cluster, die sich am nächsten sind, werden vereinigt.
3. Vereinigte Cluster und deren Distanz werden in das Dendrogramm eingetragen.
4. Wiederhole 2-3, bis nur noch ein Cluster übrig ist.

Wenn ein Cluster  $C$  durch Vereinigung von zwei Clustern  $C_u$  und  $C_v$  entsteht, so ist es nicht nötig alle Distanzen zu den restlichen Clustern  $C_i$  neu zu ermitteln. Sie können durch folgende Formel schneller berechnet werden<sup>1</sup>:

$$d(C, C_i) = \alpha_u d(C_u, C_i) + \alpha_v d(C_v, C_i) + \beta d(C_u, C_v) + \gamma |d(C_u, C_i) - d(C_v, C_i)|$$

Die Konstanten  $\alpha_u$ ,  $\alpha_v$ ,  $\beta$  und  $\gamma$  hängen dabei von dem Distanzmaß der Cluster<sup>2</sup> ab. Für das Single-Linkage Verfahren z.B. wählt man  $\alpha_u = \frac{1}{2}$ ,  $\alpha_v = \frac{1}{2}$ ,  $\beta = 0$  und  $\gamma = -\frac{1}{2}$ .

<sup>1</sup>Siehe: Nakhaeizadeh, S. 129ff

<sup>2</sup>Eine Abhängigkeit der Konstanten von dem Distanzmaß der Merkmalsvektoren oder Merkmale erwähnt Nakhaeizadeh nicht.

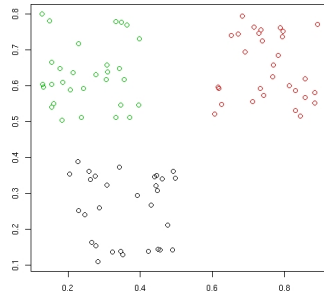


Abbildung 20: Complete-Linkage Verfahren: 3 Cluster

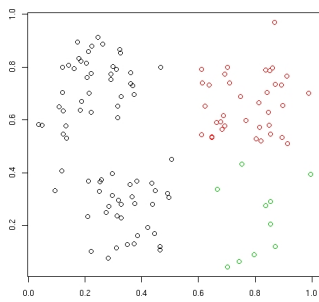


Abbildung 21: Complete-Linkage Verfahren: 3 Cluster mit Ausreißern

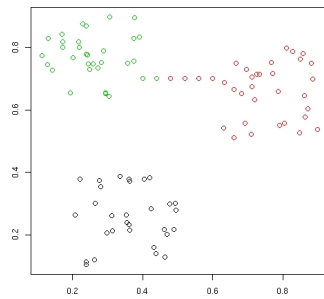


Abbildung 22: Complete-Linkage Verfahren: 3 Cluster mit Brücke

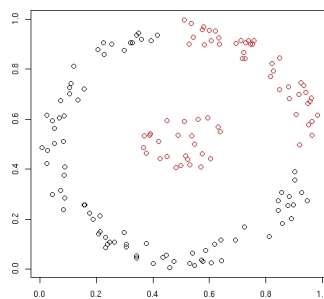


Abbildung 23: Complete-Linkage Verfahren: Ring

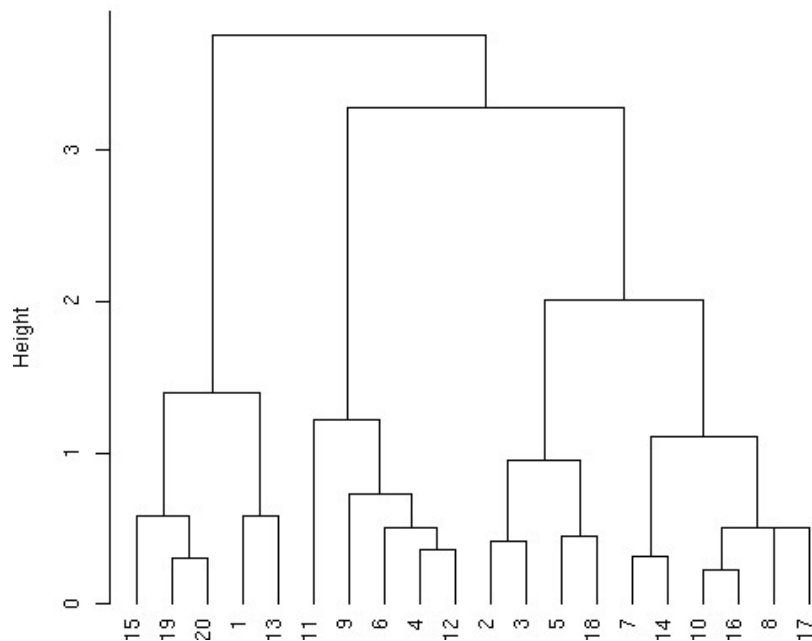


Abbildung 24: Dendrogramm der 20 Patienten

### 3.1.2 Divisiv

Bei der divisiven Klassifikation geht man genau andersherum vor, wie bei der agglomerativen. Ein Cluster der alle Objekte umfasst wird solange zerteilt, bis jeder Cluster nur noch ein Objekt beinhaltet. Die divisive Klassifikation wird in der Praxis kaum angewandt, weil sie langsamer ist als die Agglomerative. Aber z.B. in der Computergrafik wird sie verwendet um die Farben eines Bildes auf 16 oder 256 zu reduzieren um sie auf älterer Hardware darstellen zu können. In diesem Fall sind die Objekte die einzelnen Pixel und die Merkmale die Farbanteile Rot, Grün und Blau. Der Algorithmus sieht folgendermaßen aus (aus Gründen der Übersichtlichkeit wurden in dem Beispiel nur die Farbkomponenten Rot und Grün verwendet):

1. *Ausgangssituation*: Ein möglichst kleiner Quader (Cluster), der alle Objekte umfasst (Abbildung 25).
2. Den Quader mit dem größten Volumen ermitteln.
3. Diesen Quader senkrecht zu seiner längsten Kante so teilen, dass in den neuen Quadern etwa gleich viele Objekte sind (Abbildung 26).
4. Die beiden neuen Quader soweit verkleinern, bis alle Objekte gerade noch enthalten sind (Abbildung 27).
5. Wiederhole 2-4, bis die gewünschte Cluster-/Farbanzahl erreicht ist.
6. Die Schwerpunkte der Cluster sind die neuen Farben (Abbildung 28).

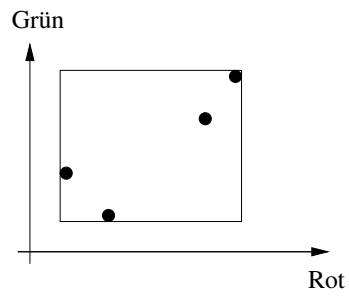


Abbildung 25: Ein Cluster umfasst alle Farben.

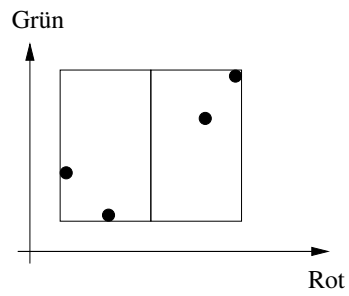


Abbildung 26: Der Cluster wurde an seiner „Rot“-Achse zerteilt.

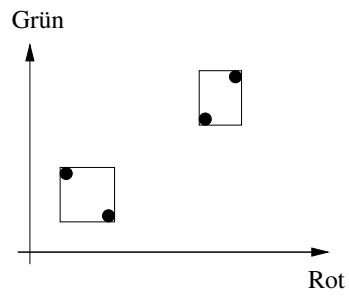


Abbildung 27: Die beiden Cluster werden reduziert.

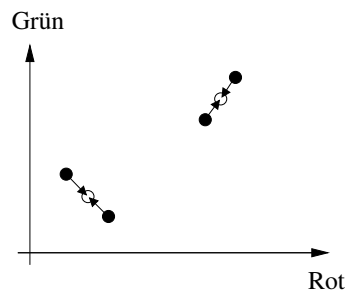


Abbildung 28: Die Schwerpunkte der Cluster.

## 3.2 Disjunkte Klassifikation

Bei der Disjunkten Klassifikation wird jedes Objekt genau einem Cluster zugeordnet. Es entsteht keine Hierarchie.

### 3.2.1 k-means

*k-means* ist ein mögliches Verfahren, um eine disjunkte Klassifikation durchzuführen. Um den Algorithmus zu starten, muss vorher bekannt sein, in wieviele Cluster die Objekte eingeordnet werden sollen.

1. *Ausgangssituation*: (Zufällige) Auswahl von  $k$  Clusterzentren.
2. Jedes Objekt wird dem nächsten Clusterzentrum zugeordnet.
3. Neuberechnung der Clusterzentren, so dass sie im Schwerpunkt der Cluster liegen.
4. Wiederhole 1-2 bis sich die Zuordnung der Objekte nicht mehr ändert.

Es ist zu beachten, dass *k-means* nicht zwingend konvergiert. Der Algorithmus sollte also nach einer bestimmten Anzahl an Schleifendurchläufen beendet werden.

Wie in den Abbildungen 29-32 zu sehen ist, erzeugt *k-means* ähnliche Cluster wie das Average-Linkage Verfahren. Es fällt allerdings auf, dass die Ausreißer, die Brücke und der Ring immer mittig zerteilt werden.

## 3.3 Unscharfe Klassifikation

Bei der unscharfen Klassifikation wird zu jedem Objekt die Wahrscheinlichkeit ermittelt, mit der es in einem bestimmten Cluster liegt. Der unscharfen Klassifikation geht eine disjunkte voraus. Die Wahrscheinlichkeit  $P$ , dass Objekt  $o$  in dem Cluster  $c_i$  liegt ist proportional zu  $e^{-d^2(o,c_i)}$ . Da die Wahrscheinlichkeiten in der Summe eins ergeben müssen, ist eine Normalisierung nötig. So erhält man folgende Formel:

$$P(c_i, o) = \frac{e^{-d^2(o,c_i)}}{\sum_j e^{-d^2(o,c_j)}}$$

## 3.4 Self-Organizing Maps

Self-Organizing Maps oder auch Kohonennetze sind Neuronale Netze mit zwei Schichten. Die Eingabeschicht hat so viele Neuronen, wie der Merkmalsvektor Merkmale hat. Die Ausgabeschicht bildet eine zweidimensionale Karte. Die beiden Schichten sind vollständig verbunden, d.h. jedes Neuron der Eingabeschicht ist mit jedem Neuron der Ausgabeschicht (auch Kohonen-Schicht) verbunden. Die Struktur eines Kohonennetzes ist in Abbildung 33 zu sehen. Es wurden nicht alle Verbindungen zwischen den Schichten gezeichnet damit die Abbildung nicht unübersichtlich wird. Die Linien zwischen

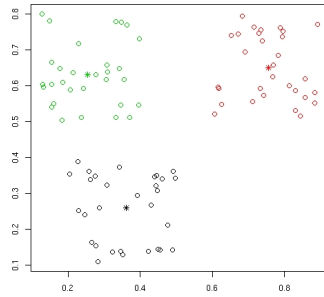


Abbildung 29: k-means: 3 Cluster.

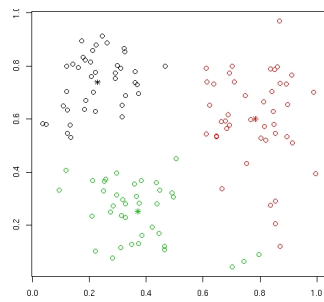


Abbildung 30: k-means: 3 Cluster mit Ausreißern.

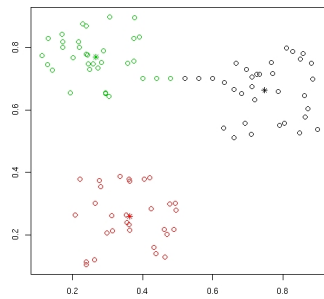


Abbildung 31: k-means: 3 Cluster mit Brücke.

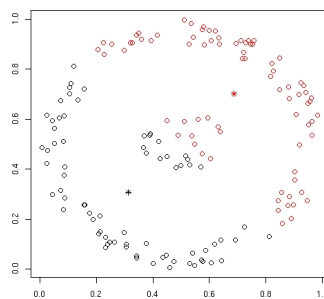


Abbildung 32: k-means: Ring.

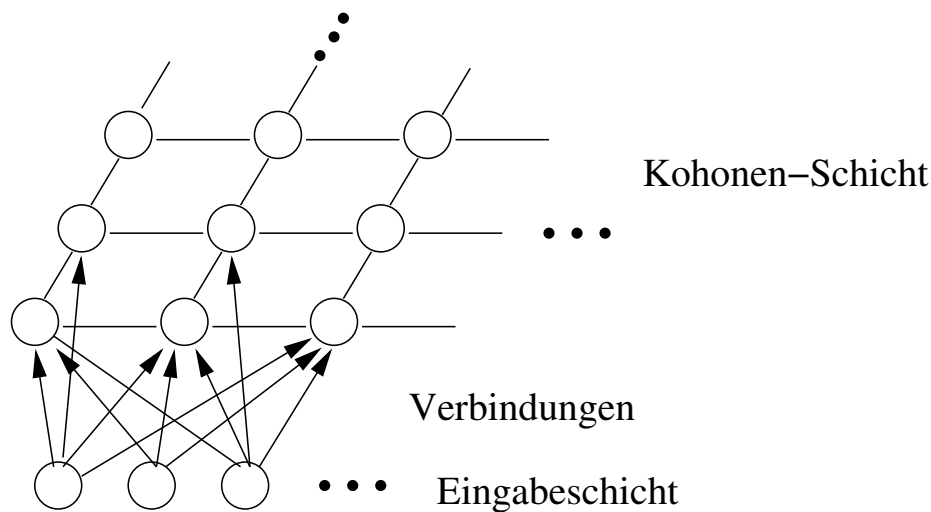


Abbildung 33: Ein Kohonennetz

den Neuronen der Ausgabeschicht sind keine Verbindungen im neuronalen Netz, sondern deuten die Struktur der Karte an.

Die besondere Eigenschaft von Self-Organizing Maps ist, dass sie den Merkmalsraum so auf die Karte abbilden, dass ähnliche Merkmalsvektoren auf der Karte benachbarte Neuronen aktivieren.

### 3.4.1 Anwenden von Self-Organizing Maps

Um ein fertig trainiertes Netz zu nutzen, legt man den Merkmalsvektor an die Eingabeschicht an. Nun sucht man auf der Karte das Neuron, dessen Gewichte dem Merkmalsvektor am ehesten entsprechen. Dieses Neuron ist das „Gewinner-Neuron“. Der Merkmalsvektor wird also auf das „Gewinner-Neuron“ abgebildet.

### 3.4.2 Trainieren einer Self-Organizing Map

Um das Neuronale Netz zu trainieren, werden die Merkmalsvektoren nacheinander, in zufälliger Reihenfolge, an das Netz angelegt und das Gewinner-Neuron wird bestimmt. Bei jedem Schritt werden die Gewichte des Gewinner-Neurons und seiner Nachbarschaft dem Merkmalsvektor angenähert. Die Nachbarschaft wird häufig durch eine Glockenkurve definiert.

Die Nachbarschaft und die Lernrate (der Grad der Annäherung pro Lernschritt) sind am Beginn des Lernens relativ groß und werden im Laufe des Trainings stetig verkleinert, bis das Netz stabil ist. Die Geschwindigkeit der Verkleinerung muss geschickt gewählt werden. Ist sie zu klein, dauert das Lernen unnötig lange und ist sie zu groß, so kann das Netz nicht fertiglernen.



Abbildung 34: Eine Kohonenkarte

### 3.4.3 Anwendung von Self-Organizing Maps in der Clusteranalyse

Bestimmt man nach dem Training für alle Merkmalsvektoren die Gewinner-Neuronen, so liegen diese gleichverteilt auf der Karte. Um die Grenzen zwischen den Clustern sichtbar zu machen, ist es möglich den Abstand der Gewichte der Neuronen auf der Kohonenkarte grafisch sichtbar zu machen. In der Abbildung 34 wurden die Abstände anhand von Höhenwerten einer Landschaft dargestellt. Die sechs Cluster sind deutlich zu erkennen.

### 3.4.4 Self-Organizing Maps zur Analyse von Musik

Self-Organizing Maps werden z.B. in der Software „MusicMiner“ zur Clusteranalyse verwendet, um Musik zu klassifizieren. Von jedem Musikstück werden Merkmale anhand von Rhythmus und Frequenzverteilung berechnet. Zu den Stücken kann so eine, wie oben beschriebene, Kohonenkarte erzeugt werden (Abbildung 35). Die Kästchen sind einzelne Musikstücke. Ähnliche Stücke liegen beieinander.

## 4 Quellen

- Deichsel, Kap. 1-5
- Nakhaeizadeh, S. 109-141
- Diplomarbeit „Identifikation und Analyse von Besucherprofilen auf Websites“ von Michael Fait



Abbildung 35: Eine Kohonenkarte mit Musik

- Scholl & Pfeiffer: „Natur als fraktale Grafik“, Markt&Technik
- <http://de.wikipedia.org/wiki/Clusteranalyse>
- [http://de.wikipedia.org/wiki/Self-Organizing\\_Maps](http://de.wikipedia.org/wiki/Self-Organizing_Maps)
- <http://www.mathematik.uni-marburg.de/~databionics/de//?q=esom>
- <http://musicminer.sourceforge.net/>
- <http://www.r-project.org/>