Data Mining

Friedhelm Schwenker

Institut für Neuroinformatik

Universität Ulm

16. Oktober 2012

Organisation

- Vorlesung (3h): Di 14-16 Uhr, Do 12-14 Uhr jeweils im Raum O27/123
- Übungen (1h): 14-tägig, donnerstags; 1.Übung am 8. November;
- Übungsaufgaben: schriftlich bearbeiten; 1. Übungsblatt am 23. Oktober; (Schein: 50% der erreichbaren Punkte und aktive Teilnahme in der Übungsstunde).
- Kernfächer: Mathematische/theoretische Methoden der Informatik und Praktische Informatik.
- Vertiefungsfach : Neuroinformatik.
- Kernmodul: Mathematische/theoretische Methoden der Informatik und Praktische Informatik.
- Vertiefungsmodul : Neuroinformatik und Mustererkennung.
- Projektmodul: Neuroinformatik

Inhalt

1. Data Mining Methoden – Ein kurzer Überblick
2. Grundlagen der beschreibenden Statistik
3. Clusteranalyse
4. Visualisierung und Merkmalsreduktion
5. Assoziationsanalyse
6. Klassifikation
7. Prognose
8. Anwendung: Text Mining, Web Mining, Bioinformatik

1. Data Mining Methoden - Überblick

- Einleitung
- Daten und Wissen
  - Kennzeichen/Unterschiede
  - Bewertungskriterien von Wissen
  - Beispiel: Tycho Brahe und Johannes Kepler
- KDD und Data Mining
  - Wie findet man Wissen
  - KDD–Prozess
  - Aufgabenbereiche für Data Mining Methoden
- Data Mining Methoden (einige Beispiele)
  Entscheidungsbäume, Neuronale Netze, Clusteranalyse

Einleitung

- Computer speichern in Unternehmen und Behörden Daten in großer Zahl
  - Kundendaten, Lieferantendaten, Personaldaten
  - Lagerverwaltung, Produktdaten
  - Vertriebsplanung, Produktionsprozessplanung

- Meist besteht eine enge Kopplung mit Datenbanksystemen. Viele Einzelinformationen sind abrufbar.

- **Regelhaftigkeiten, Strukturen und Muster** in den Daten bleiben aber meist verborgen !

Daten

Beispiele für Daten:

- *Kunde X hat Bier gekauft!*
- *QRS-Dauer des Patienten beträgt im Mittel 140 msec!*

Eigenschaften von Daten:

- beschreiben Einzelfälle (Personen, Zeitpunkte, Orte)
- sind vielfach in großer Zahl vorhanden
- sind oft leicht zu beschaffen (Internet, Scannerkassen, Rabattkarten)
- lassen meist keine Vorhersagen zu

Wissen

Beispiele von Wissen:

- Der *5-er-Bus* fährt im 10-Minuten-Takt
- Die Erdbeschleunigung beträgt etwa 9.81 m/s^2

Kennzeichen von Wissen:

- beschreibt allgemeine Muster, Strukturen, Gesetze und Prinzipien
- lässt Voraussagen zu
- soll aus möglichst wenigen und einfachen Aussagen bestehen
- ist i.a. schwer zu finden bzw. zu beschaffen

Bewertungskriterien für Wissen

Wissen muss bewertet werden, nicht jede allgemeine Aussage ist wichtig oder nutzbar.

Kriterien mit denen man Wissen bewerten kann:

- Korrektheit : *Wie wahrscheinlich ist die Regel?*
- Allgemeinheit : *Wann und unter welchen Bedingungen anwendbar?*
- Nutzbarkeit : *Welche Vorhersagekraft ist dadurch gegeben?*
- Verständlichkeit : *Liegt Wissen in übersichtlichen Regeln vor?*
- Neuheit : *Waren die Aussagen unbekannt bzw. so nicht erwartet worden?*

Tycho Brahe (1546-1601)

- dänischer Astronom; bedeutenster Astronom vor Erfindung des Fernrohrs (um 1610)

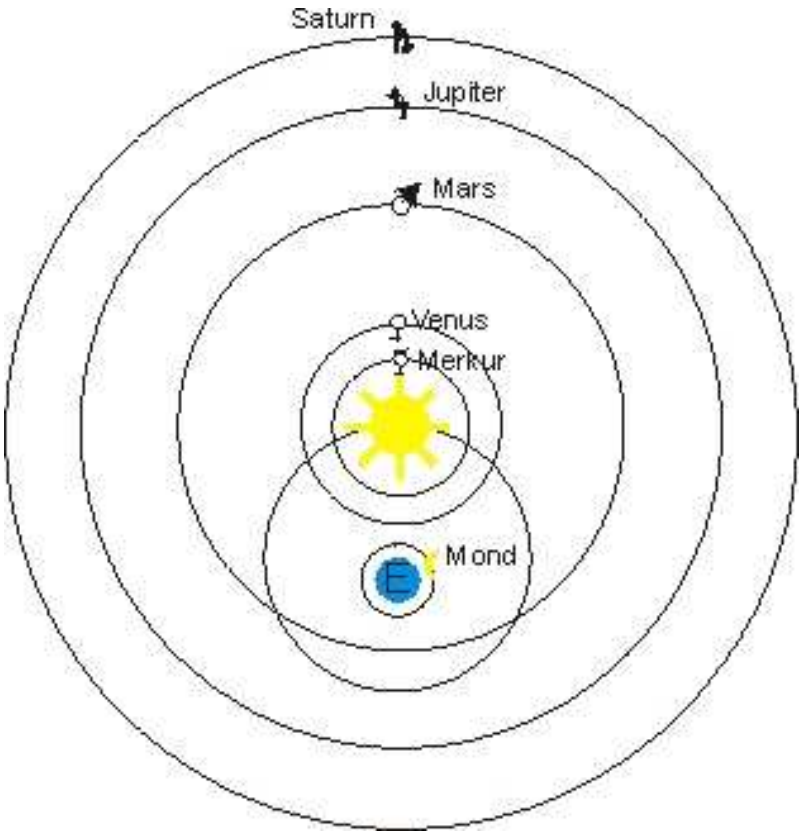
- ab 1582 erbaute Sternwarte Uranienborg auf der dänischen Ostseeinsel Hvan; ab 1599 Hofastronom von Rudolf II in Prag

- bestimmte Positionen der Sonne, des Mondes und der Planeten mit sehr hoher Präzision und zeichnete diese Daten über viele Jahre hinweg auf.

Brahes Problem:

- er konnte seine gesammelten Daten nicht in einem einheitlichen System zusammenfassen

- sein Modell bewährte sich nicht; Mischung aus dem ptolemmaiischen und kopernikanischen Modell für unser Planetensystem



Modell von Brahe für unser Planetensystem und Denkmal der Astronomen Tycho Brahe und Johannes Kepler in Prag.

Johannes Kepler (1571-1630)

- deutscher Astronom und Mathematiker
- ab 1600 Gehilfe von Tycho Brahe; ab 1601 dessen Nachfolger
- vertrat das Modell des kopernikanischen Planetensystems
- benutzte Brahes Datensammlung
  $\Longrightarrow$

Die Kepler'schen Gesetze (1609 und 1619)

1. Alle Planeten bewegen sich auf Ellipsen, in deren Brennpunkt die Sonne steht.
2. Eine von der Sonne zum Planeten gezogene Linie, überstreicht in gleichen Zeiten gleiche Flächen.
3. Die Quadrate der Umlaufzeiten zweier Planeten verhalten sich wie die Kuben der großen Ellipsenachsen ihrer Umlaufbahn.

Wie findet man Wissen ?

Es gibt natürlich keine universelle Methode um Wissen zu entdecken.

Probleme:

- Riesige Datenmengen in Datenbanken sind heute verfügbar. *Wir ertrinken in einem Meer von einzelnen Daten, aber wir hungern nach Wissen.*
- Manuelle Analysen sind kaum mehr durchführbar.
- Einfache Methoden (Diagramme,etc.) stoßen schnell an ihre Grenzen.

Lösungsversuche:

- Interaktive Datenanalyse-Programme
- Knowledge Discovery in Data Bases und Data Mining Methoden

KDD und Data Mining

Knowledge Discovery in Databases
Fayyad: KDD ist der nichttriviale Prozess der Identifizierung von gültigen, neuen, potenziell nützlichen und schließlich verständlichen Mustern in Daten.

Data Mining
Data Mining ist der Schritt des KDD-Prozesses, in dem nach interessanten Mustern in den Daten gesucht wird.

KDD-Prozess

Allgemeines

- Die einzelnen Stufen sind nicht strikt von einander getrennt.
- Der gesamte KDD-Prozess ist in seiner Gesamtheit und seinen Teilaspekten iterativ, d.h. mehrere Durchläufe sind erforderlich.

Vorstufen im Prozess

- Bestimmung des Nutzenpotenzials
- Anforderungs-/Durchführbarkeitsanalyse

Hauptstufen

- Sichtung des Datenbestandes.
- Datenvorverarbeitung!
  - Vereinheitlichung und Transformation der Daten in uniformes Format.
  - Datensäuberung: fehlerhafte/unvollständige Eingaben feststellen und ggf. solche Datensätze/Attribute aus dem Datensatz entfernen
  - Datenreduktion: Stichprobe, Attributauswahl, Beschränkung auf Prototypen
- Data Mining (mit verschiedenen Verfahren)
- Visualisierung der Resultate
- Interpretation, Analyse und Bewertung der erzielten Resultate.
- Anwendung und Dokumentation.

Data–Mining Aufgaben

- Klassifikation : *Wird der Kunde sein Darlehen zurückzahlen?*
- Prognose : *Wie entwickelt sich der Dollar-Kurs?*
- Abhängigkeitsanalysen : *Welche Produkte werden zusammen verkauft?*
- Konzeptbeschreibung : *Welche Lesegewohnheiten haben Leser von Data-Mining Büchern?*
- Segmentierung : *Welche QRS-Dauer ist typisch für Infarkt-Patienten?*
- Abweichungsanalyse : *Gibt es jahreszeitliche Umsatzschwankungen?*

Entscheidungsbaum - Das Prinzip

Idee: Einzelne Attribute werden getestet. In Abhängigkeit des Testresultats wird ein weiteres Attribut getestet. Dies wird solange durchgeführt, bis eine hinreichend präzise Entscheidung getroffen werden kann.

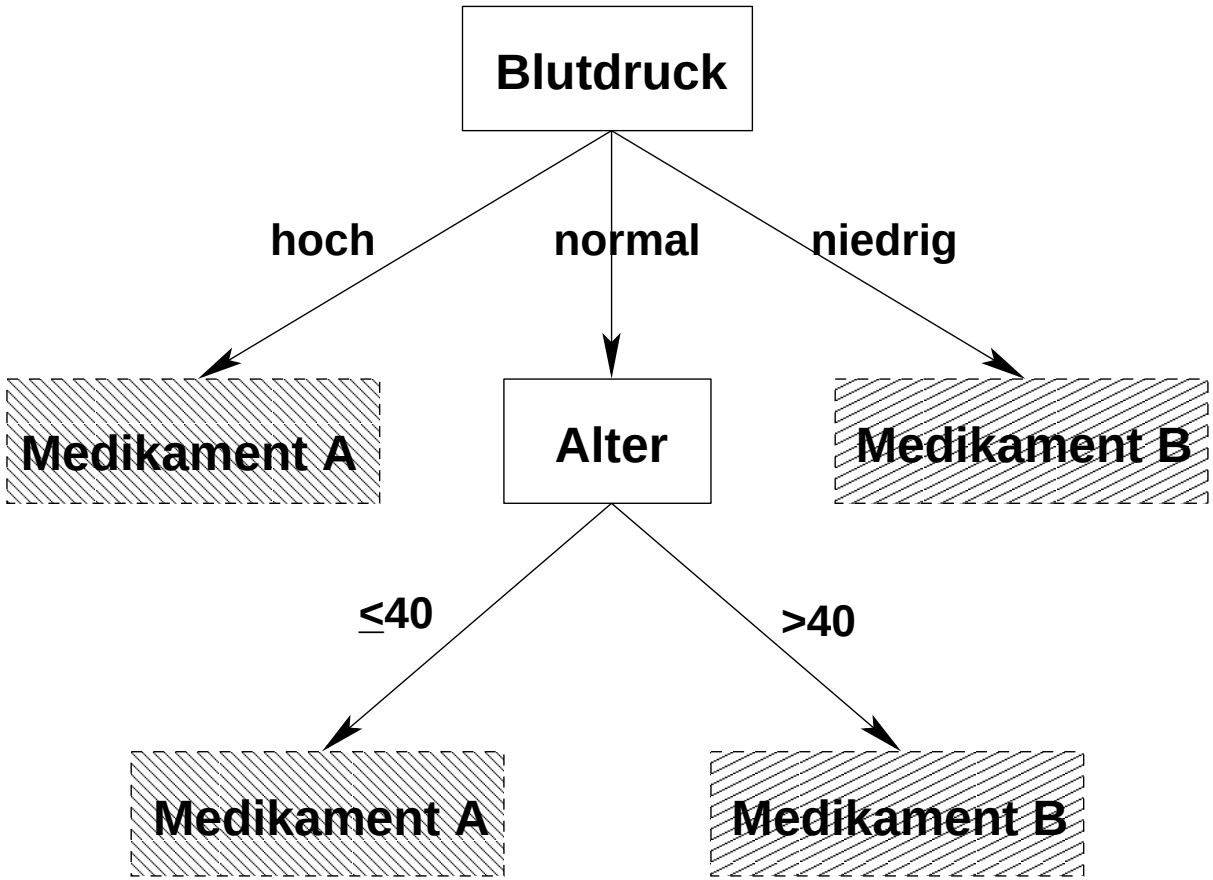
Beispiel: Vorgehensweise bei der Bestimmung des wirksamen Medikaments

1. Zuerst den Blutdruck messen.

2. Ist der Wert entweder **hoch** oder **niedrig**, so steht das richtige Medikament sofort fest.

3. Ist der Blutdruck normal, so muss das Alter des Patienten geprüft werden.

Entscheidungsbaum - Bild zum Beispiel



Entscheidungsbaum - Die Daten

Patientendaten mit dem wirksamen Medikament (bzgl. einer Krankheit).

Nr.	Geschlecht	Alter	Blutdruck	Medikament
1	m	20	normal	A
2	w	73	normal	B
3	w	37	hoch	A
4	m	33	niedrig	B
5	w	48	hoch	A
6	m	29	normal	A
7	w	52	normal	B
8	m	42	niedrig	B
9	m	61	normal	B
10	w	30	normal	A
11	w	26	niedrig	B
12	m	54	hoch	A

Entscheidungsbaum - Resultat

Aufteilung der Daten durch ein Entscheidungsbaumlernverfahren.

Nr.	Blutdruck	Alter	Medikament
3	hoch	37	A
5	hoch	48	A
12	hoch	54	A
1	normal	20	A
6	normal	29	A
10	normal	30	A
7	normal	52	B
9	normal	61	B
2	normal	73	B
11	niedrig	26	B
4	niedrig	33	B
8	niedrig	42	B

Blutdruck und Alter bestimmen das Medikament – Geschlecht nicht relevant.

Künstliche neuronale Netze

- Künstliche neuronale Netze (KNN) sind von biologischen neuronalen Netzen abstrahiert, zeigen meist nur sehr entfernte, schwache Analogie.

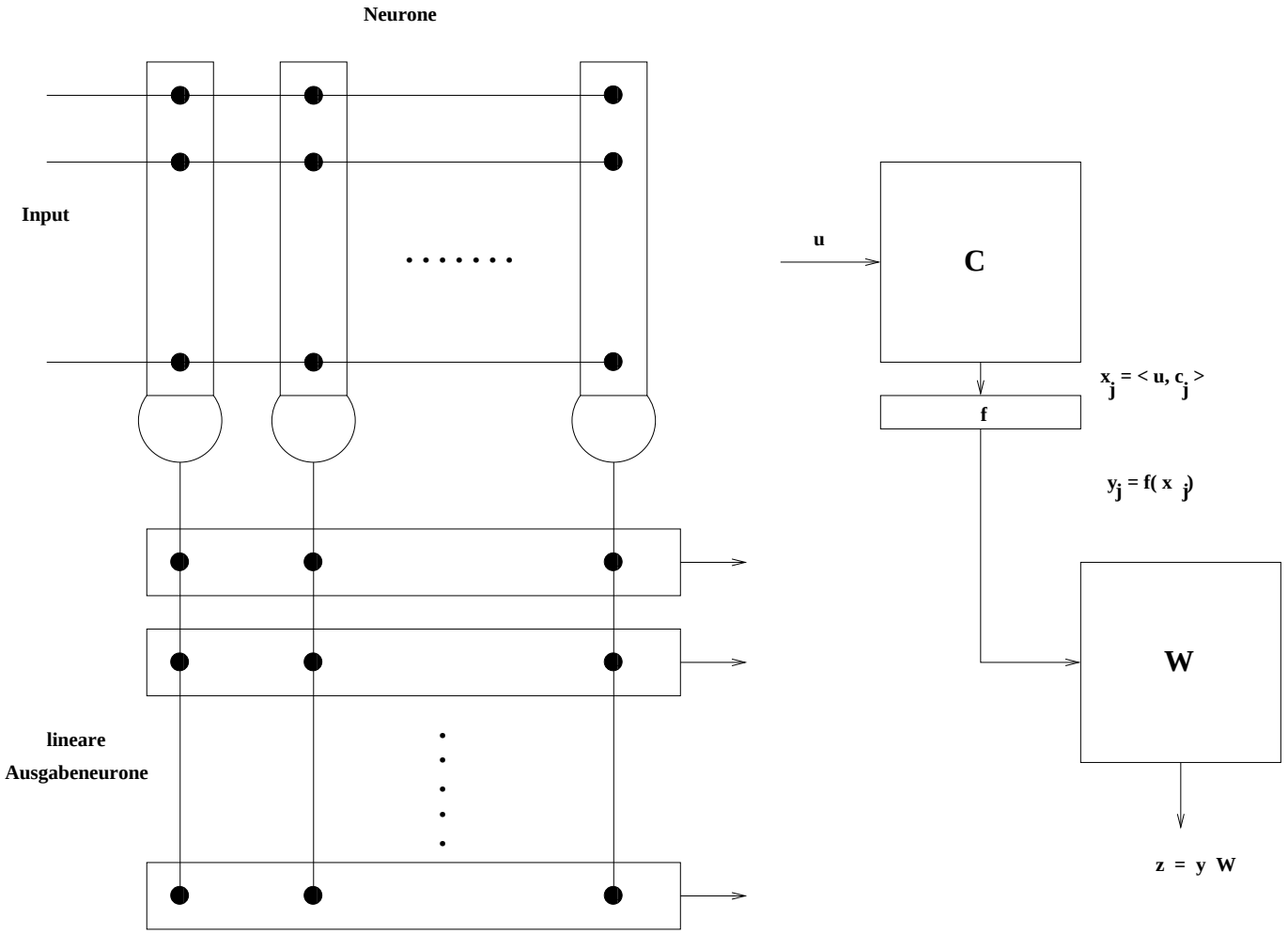
- **Idee**: Kleine, einfach strukturierte Einheiten (Neuronen) sind über gewichtete Verbindungen verschaltet, diese Verbindungsgewichte können durch Lernen adaptiert werden.

- **Anwendungsgebiete**: Klassifikation und Prognose

- **Problem**: Trainierte KNN sind schwer interpretierbar (*black box*).

- **Lösungsansatz**: Kombination von KNN und interpretierbaren Fuzzy–Systemen

- **Beispiele**: Multilayerperzeptrone, Kohonenkarten, etc.


Neurone
Input
lineare
Ausgabeneurone
u
C
$x_j = < u, c_j >$
f
$y_j = f(x_j)$
W
z = y W

Clusteranalyse

Idee

- Zusammenfassung von Einzelfällen zu Gruppen, sogenannten Clustern.
- Fälle innerhalb einer Gruppe sollen möglichst ähnlich sein.
- Fälle aus verschiedenen Gruppen sollen möglichst unterschiedlich sein.

Anwendungsgebiete

- Prototypbildung (Repräsentation einer Gruppe von Einzelfällen durch einen typischen Fall.)
- Konzeptbeschreibung (Bestimmung der Merkmale, die für die Unterscheidung in Cluster relevant sind.)

Clusteranalyse - Verfahren

Methoden

- hierarchische Clusterverfahren
- partionierende Clusterverfahren
- Fuzzy–Clusterverfahren
- Possibilistische Clusterverfahren
- Neuronale Methoden zur Clusteranalyse
- Clustervalidierung

Weitere DM-Ansätze

- statistische Verfahren
  - k-nearest neighbour
  - Zeitreihenanalyse
  - Hauptachsenanalse
  - Regressionsanalyse
  - Diskriminanzanalyse
- Maschinelles Lernen
  - instance based learning
  - induktive logische Programmierung
  - Bayes-Netze
- evolutionäre/genetische Algorithmen

Zusammenfassung

- Daten sind noch kein Wissen, aber in Daten kann Wissen, etwa in Form von Regeln, verborgen sein!
- Manuelle Analyse bei großen, hochdimensionalen Datenmengen ist undurchführbar, deshalb Unterstützung durch „intelligente“ Software.
- Software kann die Dateninspektion durch den Menschen nicht ersetzen, aber wertvolle Hilfe leisten.

2. Deskriptive Statistik

1. Aufgaben der deskriptiven (beschreibenden) Statistik
2. Merkmale und Skalen
3. Auswertung univariater (1-dimensionaler) Daten
4. Auswertung multivariater (mehrdimensionaler) Daten

Beispiel

Was kostet ein bestimmtes Konsumgut?

Stiftung-Warentest hat eine Waschmaschine getestet und will im Testbericht auch über den Preis informieren.

Testkäufe in verschiednen 10 Geschäften liefern das folgende Resultat:

Geschäft	1	2	3	4	5	6	7	8	9	10
Preis	398	379	458	398	368	379	394	379	458	398

Problem: Welche Preisinformation soll nun auf der Basis dieser gesammelten Daten im Bericht angegeben werden?

Mögliche Angaben wären:

- Der günstigste Preis – das **Minimum** ? (hier: 368 €)
- Der Preis der am häufigsten genannt wurde – der **Modalwert** ? (hier: 398 €)
- Der höchste Preis – das **Maximum** ? (hier: 458 €)
- Ein mittlerer Preis – das **arithmetisches Mittel** (hier: 400,90 €) oder der **Median** (394 €) ?

Interessant ist auch die Information über die Preisspanne, etwa zwischen dem Maximal- und Minimalpreis (hier: 90 €).

Aufgaben der deskriptiven Statistik

Daten unter bestimmten Aspekten beschreiben und die, in den Daten vorliegende Information, auf ihren wesentlichen Kern reduzieren.

- **Charakterisierung der Daten durch** einige wenige **Kennzahlen**.
  Häufig: **Mittlerer Wert** ergänzt durch ein **Streuungsmaß**.

- **Beobachtungskriterien festlegen:** Qualität der extrahierten Information wächst mit der Zahl der erhobenen Daten (mehr als 10 Testkäufe nötig?).

- **Erkennung und Elimination von Ausreißern:** Extreme/untypische Beobachtungen aus Stichprobe entfernen (Ist 458 € ein Ausreißer?).

- **Daten in Grafiken** übersichtlich und anschaulich **darstellen**.

Grundbegriffe

Es werden einige Grundbegriffe der beschreibenden Statistik eingeführt.

Die **Grundgesamtheit** (Bezeichnung: G) ist die Menge aller Einheiten, über die eine (statistische) Untersuchung etwas aussagen soll.

Es gilt immer: $|G| = n \in \mathbb{N}$ (G ist eine endliche Menge).

Beispiele:

- Personen mit deutscher Staatsangehörigkeit am 1.1.2004
- Geburten in Deutschland im Jahr 2003
- Studierende der Informatik an der Uni Ulm am 1.10.2004

Merkmal Merkmalsausprägungen

Unter einem **Merkmal** versteht man diejenige Eigenschaft, auf die sich die statistische Untersuchung bezieht.

Ein Merkmal hat verschiedene mögliche **Merkmalsausprägungen**.

Beispiele

- Studierende im WS 2004/05 an der Uni Ulm: *Geschlecht, Alter, Studienfach,*
- Private Haushalte in Ulm am 1.1.04: *verfügbares Einkommen, Zahl der Personen, Größe der Wohnung,*
- Betriebe in Ulm am 1.1.04: *Anzahl der Beschäftigten, Umsatz im letzten Quartal*

Qualitative/Quantitative Merkmale

Merkmale lassen sich nach verschiedenen Gesichtspunkten einteilen.
Eine mögliche Unterscheidung wäre:

- **Qualitative** Merkmale
- **Quantitative** Merkmale

Qualitative Merkmale sind durch verbale Ausdrücke der Merkmalsausprägung gegeben. Beispiele: Beruf, Geschlecht, Studienfach.

Quantitative Merkmale sind gegeben, falls Merkmalsausprägungen Zahlen sind. Beispiele: Alter, Einkommen, Klausurnoten (falls diese als Zahlen ausgedrückt sind).

Die Unterscheidung qualitativ-quantitativ ist wenig nützlich. (Umkodierung in Zahlenwerte ist immer möglich; Zahlen sind auch Namen.)

Diskrete/Stetige Merkmale

Eine weitere mögliche Unterscheidungsmöglichkeit wäre

- **Diskrete** Merkmale
- **Stetige** Merkmale

Merkmal heißt **diskret**, falls es nur endlich viele Ausprägungen besitzt.
Beispiele: Semesterzahl, Automarke, Beruf, Geschlecht.

Merkmal heißt **stetig** (kontinuierlich), falls die Menge der Merkmalsausprägungen Intervallen reeller Zahlen sind.
Beispiel: Einkommen, Temperatur, Blutdruck, Geschwindigkeit.

In der Praxis besitzt auch ein stetiges Merkmal nur endlich viele Ausprägungen (beschränkte Messgenauigkeit, digitale Zahlendarstellung).

Die Einteilung in diskret – kontinuierlich ist eher künstlich.

Skalen

- Merkmalsausprägungen lassen sich immer Zahlen zugeweisen, diese werden **Merkmalswerte** genannt.

- Diese Zuordnung (Abbildung) der Merkmalsausprägungen in Zahlen heißt eine **Skala**.

- Wir unterscheiden zwischen den folgenden Skalen:

  – Nominalskala
  – Ordinalskala
  – Intervallskala
  – Verhältnisskala

  Intervall- und Verhältnisskala werden zusammengefasst und dann als metrische Skala bezeichnet.

Nominalskala

Den Merkmalsausprägungen eines nominalskalierten Merkmals werden beliebige Zahlenwerte (eigentlich Kodes) zugeordnet.

Addieren und multiplizieren solcher Merkmalswerte ist nicht sinnvoll.

Jede bijektive Transformation der Merkmalsausprägungen in numerische Kodes ist verwendbar.

Beispiele nominalskalierter Merkmale:

- Geschlecht (0 = männlich, 1 = weiblich)
- Familienstand (0 = ledig, 1 = verheiratet, 2 = getrennt lebend, etc)
- Studienfach (0 = Jura, 1 = Medizin, 2 = Informatik, etc)

Ordinalskala

Zwischen den Merkmalsausprägungen besteht eine natürliche Anordnung.

Größe der Abstände hat keine Bedeutung.

Addition und Multiplikation der Merkmalswerte sind nicht sinnvoll.

Jede streng monoton wachsende Transformationen der Merkmalswerte ist möglich.

Beispiele ordinalskalierter Merkmale:

- Handelsklassen z.B. bei Lebensmitteln
- Windstärken nach Beaufort
  (windstill = 0, ..., Wirbelsturm = 12)
- Schwierigkeitsgrad von Klettertouren oder Skiabfahrten
  (schwarz = 0, rot = 1, blau = 2).

Intervallskala

- Merkmalswerte spiegeln nicht nur die Anordnung wider.

  Abstände zwischen den Merkmalswerten können verglichen werden.
- Absolute Größe der Merkmalswerte ist ohne Bedeutung.
- Affine, strikt monoton wachsende Transformation möglich:

$$x \to ax + b =: y \quad \text{wobei } a > 0 \text{ und } b \in \mathbb{R} \text{ ist}$$

- Maßeinheit a und Nullpunkt b kürzen sich heraus:

$$\frac{y_4 - y_3}{y_2 - y_1} = \frac{ax_4 + b - (ax_3 + b)}{ax_2 + b - (ax_1 + b)} = \frac{x_4 - x_3}{x_2 - x_1}$$

- **Beispiel**: Temperatur (in °Celsius bzw. °Fahrenheit)

  $y = 1.8x + 32$ (mit $y =$ Temperatur in °F und $x =$ Temperatur in °C).

Verhältnisskala/Ratioskala

- Verhältnisskala ist Intervallskala mit natürlichem Nullpunkt.
- Maßeinheit (Maßstab) ist allerdings nicht festgelegt.
- Verhältnisskalen sind eindeutig bis auf positive lineare Transformationen

$$x \to ax =: y \quad \text{hierbei ist } a > 0 \text{ ein Skalierungsfaktor.}$$

- Quotient zweier Merkmalswerte ist vom gewählten Maßtab unabhängig.
- **Beispiele** für verhältnisskalierte Merkmale
  Physikalische und ökonomische Größen (Länge, Gewicht, Zeit, Geschwindigkeit, Strom, Spannung, Einkommen, Vermögen, Geldmenge).

Auswertung univariater Daten

X sei das zu untersuchende Merkmal.

$G = \{e_1, \ldots, e_n\}$ die zu untersuchende Grundgesamtheit von n Objekten/Einheiten.

Mit $x_1, ..., x_n$ seien die Daten gegeben, die sogenannte **Urliste**.

x_i ist dabei die Ausprägung des Merkmals X für die Einheit e_i.

Statistische Auswertungsverfahren der Daten

$$x_1, .., x_n$$

werden nun in Abhängigkeit des Skalenniveaus (nominal, ordinal, metrisch (Intervall und Verhältnis)) vorgestellt.

Beliebig skaliertes Merkmal

Das Merkmal X habe J verschiedene mögliche Merkmalswerte, die wir mit $\zeta_1, \ldots, \zeta_J$ bezeichnen.

Absolute Häufigkeit von ζ_j ist n_j die Anzahl der Daten mit dem Wert ζ_j für $j = 1, \ldots, J$.

Relative Häufigkeit von ζ_j ist $f_j = \frac{n_j}{n}$ der Anteil der Daten mit dem Wert ζ_j für $j = 1, \ldots, J$

Es gilt dabei offenbar

$$\sum_{j=1}^{J} n_j = n \quad \text{und} \quad \sum_{j=1}^{J} f_j = 1$$

Diskrete Klassierung

Eine diskrete Klassierung der Urliste $x_1, ..., x_n$ ist gegeben durch:

$$(\zeta_1, n_1), (\zeta_2, n_2), \ldots, (\zeta_J, n_J)$$

oder durch hinzufügen der relativen Häufigkeiten $f_j = \frac{n_j}{n}$

$$(\zeta_1, n_1, f_1), (\zeta_2, n_2, f_2), \ldots, (\zeta_J, n_J, f_J)$$

Darstellung in Tabellenform:

Merkmalswert	Anzahl	relative Häufigkeit
ζ_1	n_1	$f_1 = \frac{n_1}{n}$
ζ_2	n_2	$f_2 = \frac{n_2}{n}$
.	.	.
.	.	.
ζ_J	n_J	$f_J = \frac{n_J}{n}$
$\sum$	n	1

Beispiel

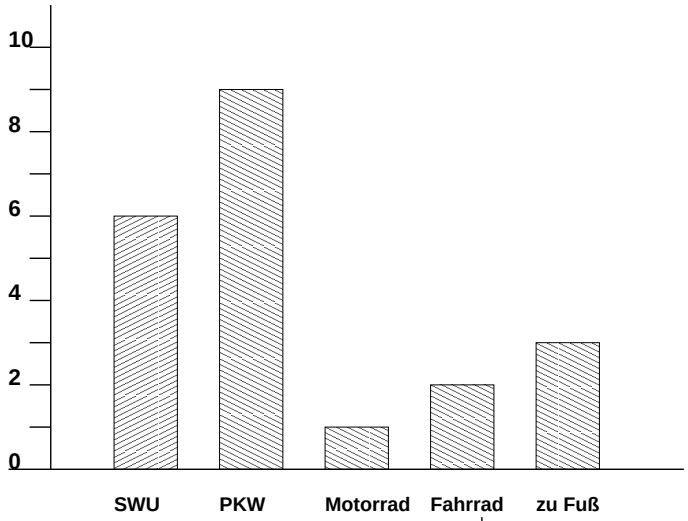
Grundgesamtheit: 20 Studenten.
Merkmal: Transportmittel für den Weg zur Uni.

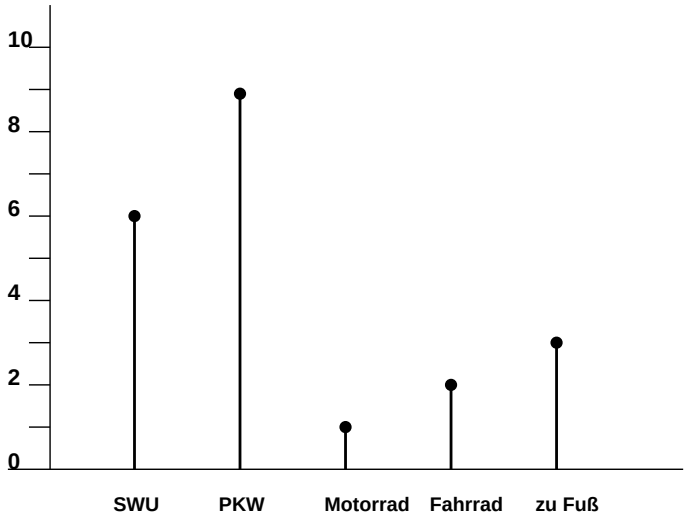
$\zeta_1 = 1$ (SWU)
$\zeta_2 = 2$ (PKW)
$\zeta_3 = 3$ (Motorrad)
$\zeta_4 = 4$ (Fahrrad)
$\zeta_5 = 5$ (zu Fuß)

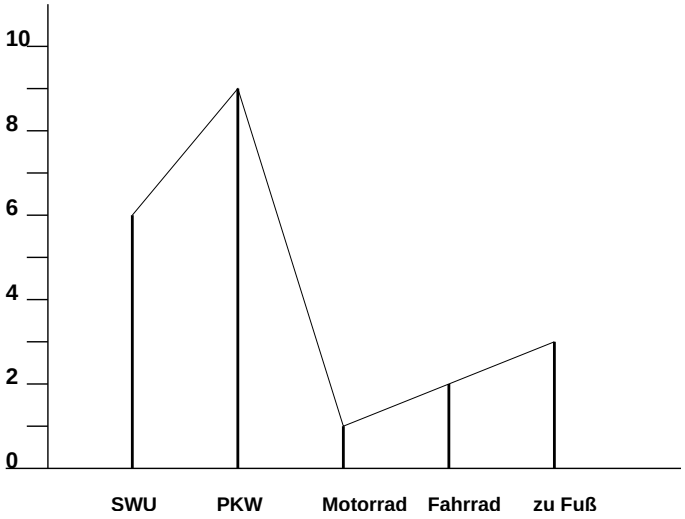
Daten der Urliste:
$1, 1, 2, 2, 2, 4, 3, 5, 2, 2, 5, 2, 4, 1, 1, 2, 2, 1, 2, 1$

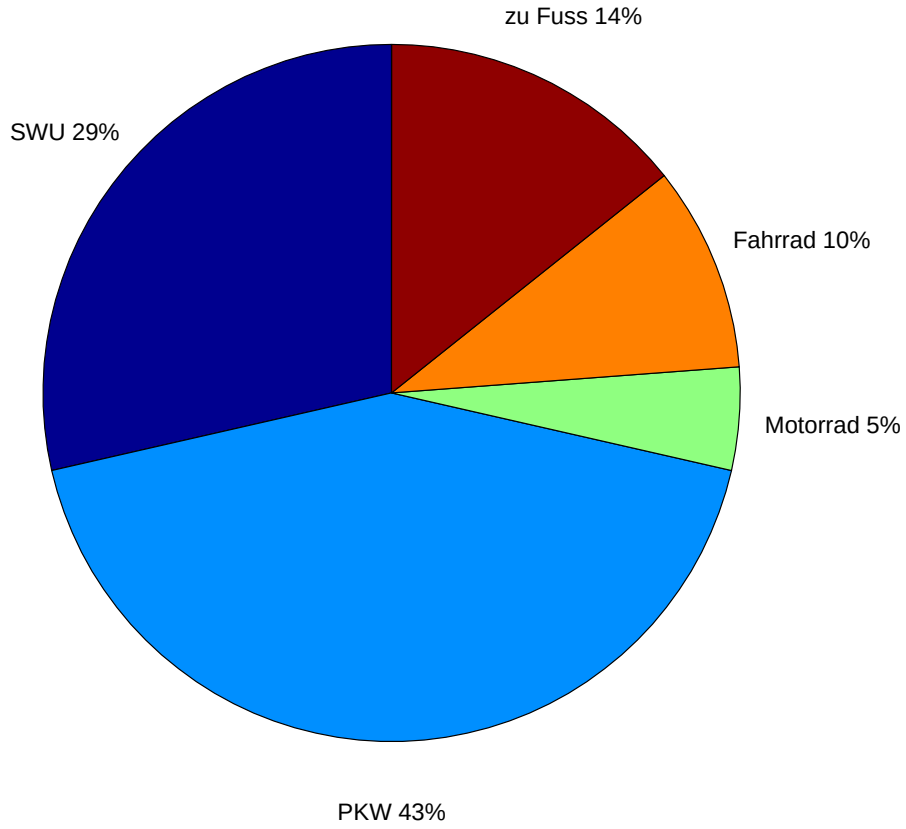
ζ_j	n_j	f_j
$1 =$ SWU	6	$6/20$
$2 =$ PKW	9	$9/20$
$3 =$ Motorad	1	$1/20$
$4 =$ Fahrrad	2	$2/20$
$5 =$ zu Fuß	2	$2/20$
	20	1.0

Balken-, Stab-, Kreisdiagramm und Polygonzug









Die Ausprägung ζ_j heißt **Modus** oder **Modalwert**, falls $n_j \geq n_k$ für alle $k = 1, \ldots, J$. Der Modus ist nicht eindeutig bestimmt, d.h. die Urliste kann mehrere Modi aufweisen.

Verteilungsfunktion

Das Merkmal X sei nun (mindestens) ordinalskaliert, d.h. es gibt eine natüliche Ordnung der Merkmalswerte (oBdA $\in \mathbb{R}$) und

$$x_1, \ldots, x_n$$

seien die Daten der Urliste.

Als **(empirische) Verteilungsfunktion** der Daten bezeichnet man die Funktion $F : \mathbb{R} \to [0, 1]$ definiert durch

$$F(x) = \frac{|\{e_i \ : \ x_i \leq x\}|}{n} = \sum_{j \in \{r \mid \zeta_r \leq x\}} f_j = \frac{1}{n} \sum_{j \in \{r \mid \zeta_r \leq x\}} n_j$$

$F(x)$ ist also der Anteil der Objekte e_i mit der Eigenschaft: $x_i \leq x$.

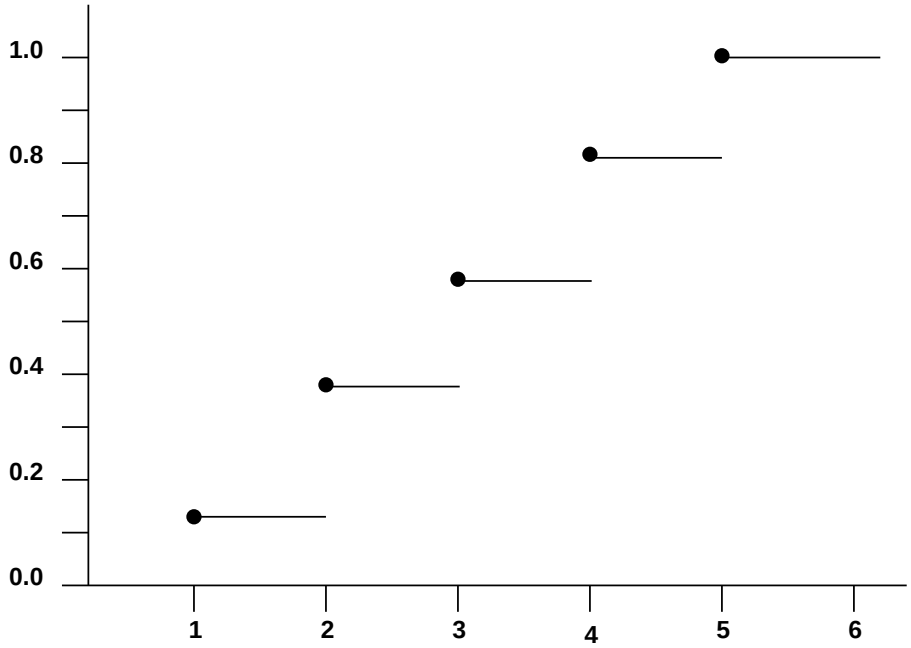
Die Verteilungsfunktion ergibt sich direkt aus den (relativen) Häufigkeiten.

Beispiel: Urliste der Klausurerebnisse für 16 Teilnehmern

$$3, 4, 2, 1, 2, 4, 5, 5, 2, 1, 4, 5, 3, 3, 2, 4$$

Hieraus ergibt sich die empirische Verteilungsfunktion dieser Daten:

ζ_j	n_j	f_j in %	$F(\zeta_j)$ in %
1	2	12,50	12,50
2	4	25,00	37,50
3	3	18,75	56,25
4	4	25,00	81,25
5	3	18,75	100,00



Eine Verteilungsfunktion ist eine monoton wachsende ($F(x_1) \leq F(x_2)$ für $x_1 < x_2$) und rechtsseitig stetige **Treppenfunktion**.

Quantile

Weiterer wichtiger Begriff zur Beschreibung von Daten ist der des **Quantils**.

Mit Hilfe der empirischen Verteilungsfunktion F definieren wir für $0 < p < 1$:

$$\tilde{x}_p = \min\{x \in \mathbb{R} \ : \ F(x) \geq p\}$$

$\tilde{x}_p$ ist der kleinste x-Wert mit der Eigenschaft: $F(x) \geq p$.

$\tilde{x}_p$ wird als p-Quantil bezeichnet.

$\tilde{x}_p$ ist der kleinste x-Wert, so dass $p * 100$ % der Daten $\leq x$ sind.

$Q : [0, 1] \rightarrow \mathbb{R}$ mit $p \rightarrow \tilde{x}_p$ heißt **Quantilfunktion**.

Quantil- und Verteilungsfunktion enthalten die gleiche Information über die Daten.

Quantile könne auch direkt aus den Daten berechnet werden, also ohne Bestimmung der empirischen Verteilungsfunktion.

Hierzu sollen die Daten in der Urliste bereits aufsteigend sortiert sein, also

$$x_1 \leq x_2 \leq x_3 \leq \cdots \leq x_n$$

Dann ist für $p \in (0, 1)$

$$\tilde{x}_p = \begin{cases} x_{np} & \text{falls } np \text{ ganzzahlig} \\ x_{[np]+1} & \text{sonst} \end{cases}$$

hierbei sei $[x]$ der ganzzahlige Anteil von x.

Einige Quantile haben besondere Namen:

Median	$\tilde{x}_{\frac{1}{2}}$
Quartile	$\tilde{x}_{\frac{1}{4}}, \tilde{x}_{\frac{2}{4}}, \tilde{x}_{\frac{3}{4}}$
Quintile	$\tilde{x}_{\frac{1}{5}}, \tilde{x}_{\frac{2}{5}}, \tilde{x}_{\frac{3}{5}}, \tilde{x}_{\frac{4}{5}}$
Dezile	$\tilde{x}_{\frac{1}{10}}, \tilde{x}_{\frac{2}{10}}, \ldots, \tilde{x}_{\frac{8}{10}}, \tilde{x}_{\frac{9}{10}}$
Perzentile	$\tilde{x}_{\frac{1}{100}}, \tilde{x}_{\frac{2}{100}}, \ldots, \tilde{x}_{\frac{98}{100}}, \tilde{x}_{\frac{99}{100}}$

Quantile sind offensichtlich gut zu interpretieren und nützlich um große Datenmengen mit vielen verschiedenen Werten zu charakterisieren.

- $\tilde{x}_{\frac{1}{2}}$ —der Median, ist der Wert der die unteren 50% von den oberen 50% der Daten trennt.

- $\tilde{x}_{\frac{1}{4}}, \tilde{x}_{\frac{1}{2}}, \tilde{x}_{\frac{3}{4}}$ —die Quartile, teilen die Daten in vier Blöcke, die jeweils 25 Prozent der Daten umfassen. Zwischen $\tilde{x}_{\frac{1}{4}}$ und $\tilde{x}_{\frac{3}{4}}$—dem unteren und oberen Quartil—liegen die mittleren 50% der Daten.

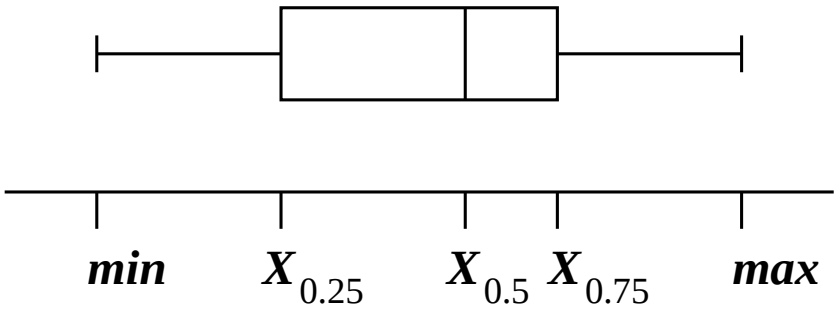
Analog sind Quintile, Dezile und Perzentile zu interpretieren.

Auswertung metrisch skalierter Daten

Wir gehen davon aus, dass das Merkmal X metrisch skaliert ist, d.h. mindestens intervallskaliert. Für bestimmte Mittelwerte müssen wir X sogar als verhältnisskaliertes Merkmal voraussetzen.

Alle Begriffe und Maßzahlen (Modalwert, Häufigkeiten, Verteilungsfunktion, Quantil), die für nominal oder ordinal Skalen definiert sind, gelten natürlich auch für metrische Daten .

Kastendiagramm/Boxplot



Minimum und Maximum.

Median innerhalb der Box.

1. und 3. Quartil definiert die Lage der Box.

Lagemaße

Für metrisch skalierte Daten $x_1, \ldots, x_n$ ist das **arithmetische Mittel** $\bar{x}$ definiert durch

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

das am weitesten verbreitete Lagemaß

Eigenschaften des arithmetischen Mittels $\bar{x}$:

1. $\sum_{i=1}^{n} x_i = n\bar{x}$: (multipliziere die Definitionsgleichung mit n).

2. $\min_i \ x_i \leq \bar{x} \leq \max_i \ x_i$.
   Denn es gilt offenbar $n \min_i \ x_i \leq \sum_{i=1}^{n} x_i \leq n \max_i \ x_i$

3. $\sum_{i=1}^{n} (x_i - \bar{x}) = \sum_{i=1}^{n} x_i - n\bar{x} = 0$. Dies folgt direkt aus 1.

4. $\sum_{i=1}^{n}(x_i - \bar{x})^2 = \min_{c\in\mathbb{R}} \sum_{i=1}^{n}(x_i - c)^2$.
   Die Ableitung der Funktion $F(c) = \sum_{i=1}^{n}(x_i - c)^2$ ist
   $F'(c) = -2\sum_{i=1}^{n}(x_i - c)$. Nullsetzen der Ableitung liefert: $0 = \sum_{i=1}^{n}(x_i - c)$
   oder $nc = \sum_{i=1}^{n} x_i$, also $c = \bar{x}$.
   $F''(c) = 2n > 0$ also liegt ein lokales Minimum vor für $c = \bar{x}$.

5. Affine Transformation und Mittelwertbildung sind vertauschbar, d.h. für eine affine Abbildung der Form $x \mapsto ax + b =: y$ mit $a, b \in \mathbb{R}$ gilt:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{1}{n}\sum_{i=1}^{n}(ax_i + b) = a\frac{1}{n}\sum_{i=1}^{n} x_i + \frac{1}{n}\sum_{i=1}^{n} b = a\bar{x} + b.$$

Median und **Modus/Modalwert** sind weitere wichtige Lagekennzahlen. Sie sind bereits für ordinal bzw. nominal skalierte Daten definiert worden.

Gewichteter Mittelwert

Verallgemeinerung des arithmetischen Mittels der Daten $x_1, \ldots, x_n$ durch Gewichtsvektor $w = (w_1, \ldots, w_n)$ mit $w_i \geq 0$ und $\sum_i w_i = 1$.

$$\bar{x}_w := \sum_{i=1}^{n} w_i x_i = \langle w, x \rangle = w \cdot x$$

$\bar{x}_w$ heißt das gewichtete Mittel zum Gewichtsvektor w.

Für $w = (1/n, \ldots, 1/n)$ ist natürlich $\bar{x}_w = \bar{x}$.

Beispiel: Verkaufspreise in den 10 Geschäften. Man bestimme G_i, $i = 1, \ldots, 10$ die Größe des Geschäftes (z.B. Kundenzahl, Umsatz,) und setze

$$w_k := \frac{G_k}{\sum_{i=1}^{n} G_i} \quad \text{für alle } k = 1, \ldots, 10.$$

Offenbar ist $w_i \geq 0$ und $\sum_i w_i = 1$.

Getrimmter Mittelwert

Arithmetische Mittelwerte sind empfindlich gegenüber Ausreißern:

Beispiel: Urliste $-27, 1, 4, 5, 10, 12, 14, 20, 25, 300$, dann ist $\bar{x} = 36, 4$.

Robusterer Mittelwert durch Trimmen der Daten, d.h. ein Teil (extremer) Werte wird bei der Mittelwertberechnung weggelassen.

Seien dazu $x_1 \leq x_2 \leq \cdots \leq x_n$ die Daten aufsteigend sortiert. Für $\alpha \in [0, 1/2)$ ist das α-getrimmte Mittel definiert durch:

$$\bar{x}_\alpha = \frac{1}{n - 2[n\alpha]} \sum_{i=[n\alpha]+1}^{n-[n\alpha]} x_i$$

Das α-getrimmte Mittel ist ein gewichtetes Mittel interpretierbar, nämlich mit $w_i = 1/(n - 2[n\alpha])$ für $i = [n\alpha] + 1, \ldots, n - [n\alpha]$ und sonst $w_i = 0$.

Beispiel: $\bar{x}_{\frac{1}{10}} = 11, 375$ für die obigen Daten.

Besondere Mittelwerte

Für verhältnisskalierte Daten mit $x_i > 0$ sind noch einige besondere Mittelwerte definiert, die aber im Folgenden keine Rolle spielen.

1. **Harmonisches Mittel** :

$$\bar{x}_H := \frac{1}{\frac{1}{n}\sum_{i=1}^{n}\frac{1}{x_i}}$$

2. **Geometrisches Mittel**:

$$\bar{x}_G := \left(x_1 \cdot \ldots \cdot x_n\right)^{\frac{1}{n}}$$

3. **p-Mittel** für $p \in \mathbb{R}_+$:

$$\bar{x}_p := \left(\frac{1}{n}\sum_{i=1}^{n} x_i^p\right)^{\frac{1}{p}}$$

Offenbar gilt

1.

$$\ln \bar{x}_G = \overline{\ln x_i}$$

2.

$$\bar{x}_H \leq \bar{x}_G \leq \bar{x}$$

und Gleichheit gilt genau dann wenn $x_1 = \cdots = x_n$.

Varianz/Standardabweichung

$x_1, \ldots, x_n$ seien metrisch skaliert, also mindestens intervallskaliert.

- **Varianz und Standardabweichung** sind am gebräuchlichsten

$$s^2 := \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \quad \text{(Varianz)}$$

durch Wurzelziehen ergibt sich

$$s := \sqrt{s^2} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2} \quad \text{(Standardabweichung)}$$

Gelegentlich findet man auch $\frac{1}{n}$ statt $\frac{1}{n-1}$ Normierung.
So wird die Varianz/Standardabweichung aber unterschätzt!

Eigenschaften der Varianz/Standardabweichung:

1. $s^2 \geq 0$ und $s \geq 0$. Es gilt: $s = 0 \Leftrightarrow s^2 = 0 \Leftrightarrow x_1 = \cdots = x_n = \bar{x}$
2. Durch Umformung erhält man leicht (bei $\frac{1}{n}$ Normierung): $s^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \bar{x}^2$
3. Nach affinen Transformation $y_i := ax_i + b$ der Daten x_i gilt: $s_y^2 = a^2 s_x^2$ bzw. $s_y = |a| s_x$

- **Mittlere absolute Abweichung vom Median**

$$d := \frac{1}{n}\sum_{i=1}^{n} |x_i - \tilde{x}_{1/2}|$$

Es gilt für d und $\tilde{x}_{1/2}$ die folgende Extremaleigenschaft:

$$d = \frac{1}{n}\sum_{i=1}^{n} |x_i - \tilde{x}_{1/2}| = \min_{a \in \mathbb{R}} \frac{1}{n}\sum_{i=1}^{n} |x_i - a|$$

- **Mittlere Differenz**

$$\Delta := \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|$$

d und Δ sind in geringerem Maße von Ausreißern betroffen als s^2.
Denn es gehen nicht die quadrierten, sondern nur die gewöhnlichen Abstände in das Maß ein.

- **Quartilabstand**

$$Q := \tilde{x}_{\frac{3}{4}} - \tilde{x}_{\frac{1}{4}}$$

Q ist die Spanne in der die mittleren 50% der Daten liegen (siehe Boxplot).
Q ist besonders robust gegenüber Ausreißern.

- **Spannweite/Range**

$$R := \max_i x_i - \min_i x_i$$

R ist besonders empfindlich gegenüber Ausreißern.

Schiefemaße

Neben der Lage und der Streuung der Daten sind ggf. weitere Aspekte ihrer Verteilung von Interesse. Hier betrachten wir Maßzahlen, die die **Abweichung von einer symmetrischen Verteilung** beschreiben.

Die Daten seien aufsteigend geordnet, also $x_1 \leq \cdots \leq x_n$

Zentraler Punkt der Daten ist definiert durch:

$$x_{\text{zentr}} = \begin{cases} x_{(n+1)/2} & \text{falls } n \text{ ungerade} \\ \frac{1}{2}(x_{n/2} + x_{n/2+1}) & \text{falls } n \text{ gerade} \end{cases}$$

Verteilung heißt symmetrisch, falls: $x_{\text{zentr}} - x_i = x_{n-i+1} - x_{\text{zentr}}$ für alle $i = 1, \ldots, n$ gilt.

Empirische Daten sind (fast) nie symmetrisch.

Schiefe

Die **Schiefe** der Daten $x_1, \ldots, x_n$ mit Standardabweichung s ist definiert

$$g = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{s}\right)^3$$

Das Vorzeichen der Schiefe lässt sich interpretieren:

- $g > 0 \Leftrightarrow$ Summanden vom Typ $(x_i - \bar{x})^3 > 0$ überwiegen
- $g < 0 \Leftrightarrow$ Summanden vom Typ $(x_i - \bar{x})^3 < 0$ überwiegen
- $g > 0$ heißt **rechtsschiefe** Verteilung
- $g < 0$ heißt **linksschiefe** Verteilung
- Für eine symmetrische Verteilung der Daten gilt $g = 0$ (Umkehrung gilt nicht)

Nachteile der Schiefe:

- nicht normiert
- sensitiv gegenüber Ausreißern

Die **Quartilschiefe** ist definiert durch

$$g_Q = \frac{(\tilde{x}_{\frac{3}{4}} - x_{\text{zentr}}) - (x_{\text{zentr}} - \tilde{x}_{\frac{1}{4}})}{\tilde{x}_{\frac{3}{4}} - \tilde{x}_{\frac{1}{4}}}$$

Für symmetrisch verteilte Daten gilt: $g_Q = 0$.

Die Quartilschiefe g_Q ist

- normiert, genauer gilt $g_Q \in [-1, 1]$.
- weniger sensitiv gegenüber Ausreißern als die Schiefe g.

Histogramme

Angenommen es liegen sehr viele Daten eines metrisch skalierten Merkmals vor.
$x_1, x_2, \ldots, x_n$ und n sehr groß.

Komprimierte Darstellung der Daten durch **Histogramme** an.

Histogramm-Darstellung

- Merkmalswerte werden in Intervalle („Klassen“) K_j zusammengefaßt.
- Einzeldaten x_i kommen nicht mehr vor.
- Nur noch Anzahl n_j der Daten je Klasse K_j werden angegeben.

Mittels der Klassengrenzen

$$a_1 < b_1 = a_2 < b_2 = a_3 < \cdots = b_{J-1} = a_J < b_J$$

werden J Klassen festgelegt durch

$$K_j := [a_j, b_j) \text{ für } j = 1, \ldots, J-1 \quad \text{und} \quad K_J = [a_J, b_J]$$

Aus den eigentlichen Daten

$$x_1, \ldots, x_n$$

werden nun Klassen K_j mit Häufigkeiten n_j gebildet, wobei gilt

$$n_j = K_j \cap \{x_1, \ldots, x_n\}.$$

Beispiel: 5000 Studierende werden nach dem monatlich verfügbaren Einkommen befragt.

j	K_j	n_j	f_j
1	$[0, 500)$	300	$0,06$
2	$[500, 1000)$	1000	$0,2$
3	$[1000, 1500)$	2000	$0,4$
4	$[1500, 2000)$	1000	$0,2$
5	$[2000, \infty)$	700	$0,14$
$\sum$		5000	$1,0$

Probleme, die bei der Histogrammbildung aufkommen:

- Wie viele Klassen sind für die vorliegenden Daten erforderlich ? Ist im Beispiel $J = 5$ ausreichend?

- Eine sehr grobe Faustregel für die Klassenzahl lautet:

$$J \approx \begin{cases} 10 \log_{10} n & \text{falls } n > 1000 \\ \sqrt{n} & \text{sonst} \end{cases}$$

- Sollen die Klassen (Intervalle) jeweils gleich lang sein?

- Kann man sich auf endliche Unter- und Obergrenzen der Klassen beschränken?

Annahme: Innerhalb der Klassen K_j gilt Gleichverteilung der Daten.

Die Daten liegen umso dichter,

- je größer die relative Häufigkeit f_j ist.
- je kleiner die Klassenbreite ist.

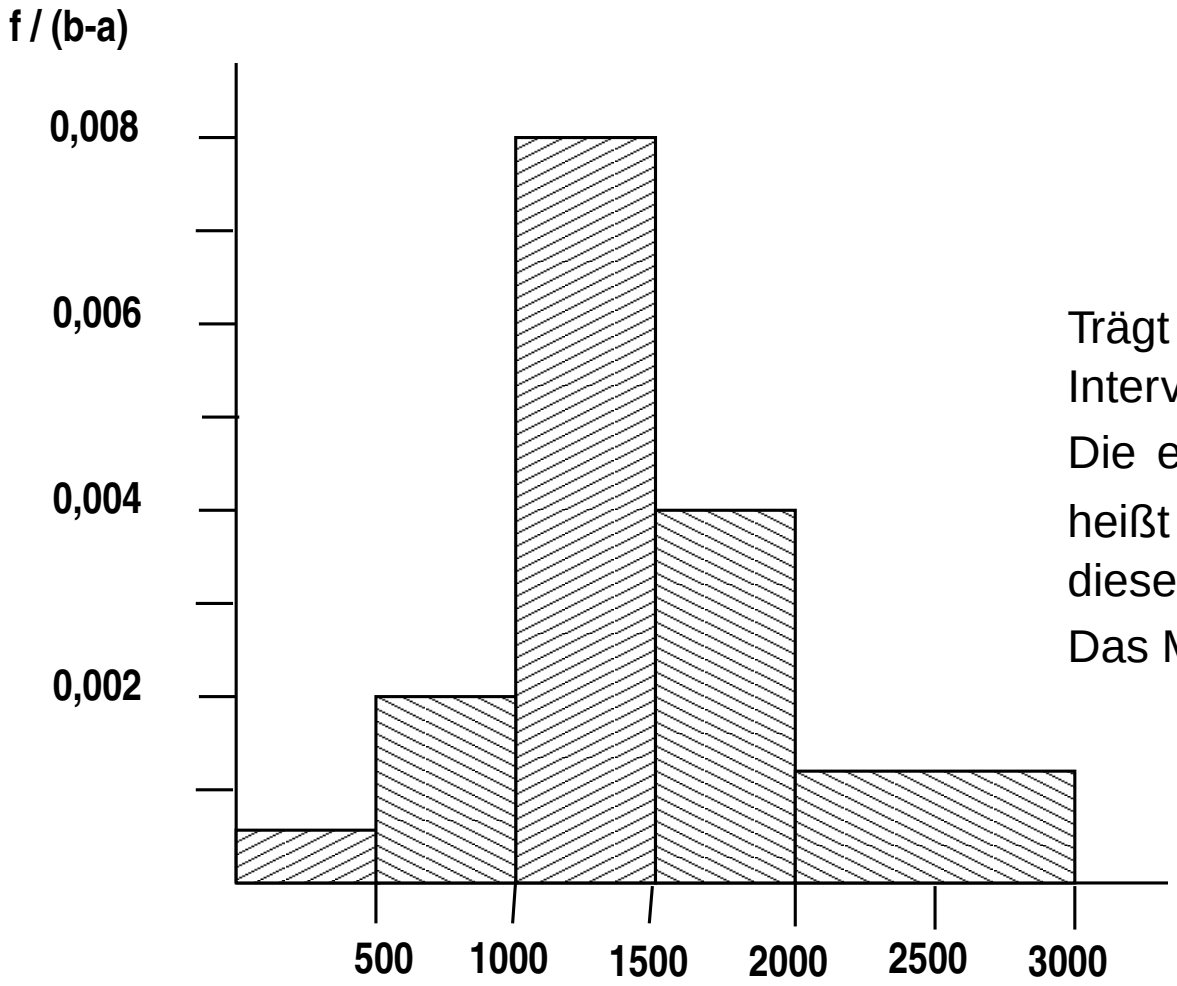
Den Quotienten

$$\frac{n_j}{n(b_j - a_j)} = \frac{f_j}{b_j - a_j} \quad j = 1, \ldots, J$$

bezeichnen wir als **empirische Dichte** der Daten in der Klasse K_j.

Problem bei der Berechung der empirischen Dichte:

Untere und obere Klassengrenze a_j und b_j müssen beschränkt sein. Häufig kann man $a_1 = 0$ annehmen; die Wahl der oberen Grenze b_J ist schwieriger.



Trägt man die empirischen Dichten über den Intervallen auf, so entsteht ein Histogramm.

Die empirische Dichte $\frac{f_j}{b_j - a_j}$, $j = 1, \ldots, J$ heißt **unimodal**, falls es genau Maximum in dieser Zahlenfolge gibt.

Das Maximum heißt **Modus/Modalwert**.

- Die einzelnen Rechtecksflächen über den Intervallen sind

$$(b_j - a_j) \cdot \frac{f_j}{b_j - a_j} = f_j$$

  gleich ihrer relativen Häufigkeiten.

- Die Gesamtfläche unter der empirischen Dichtefunktion ($=$ Summe der Rechteckflächen) ist $= 1$

- Die relevanten Größen in einem Histogrmm sind die **Rechteckflächen** über den Intervallen.

Empirische Verteilungsfunktion

Die empirische Verteilungsfunktion $F : \mathbb{R} \to [0, 1]$ an der Stelle $x \in \mathbb{R}$ definiert $F(x) =$ Anteil der Daten mit $x_i \leq x$.

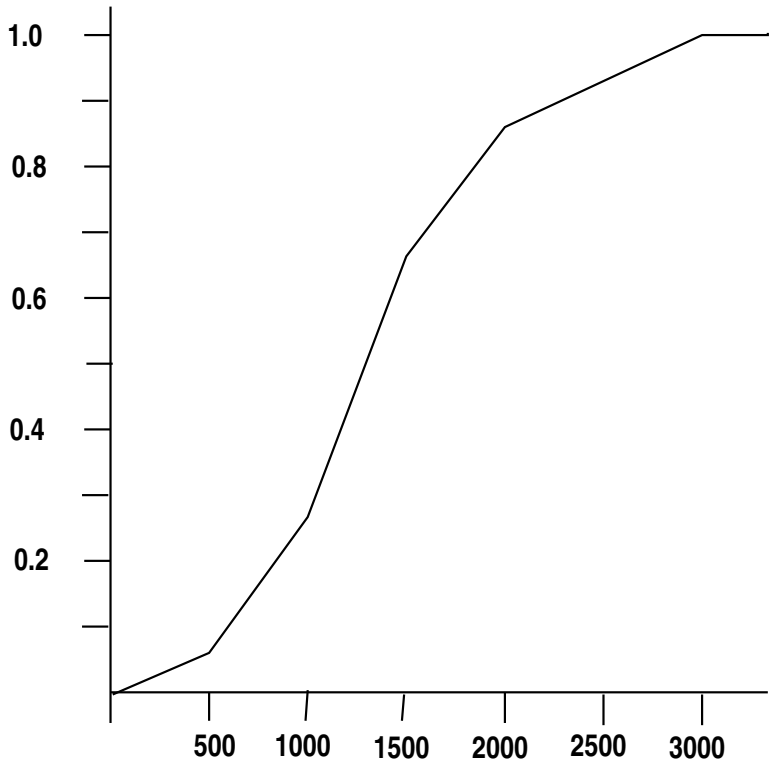
An den oberen Grenzen der Intervalle $b_1, ..., b_J$ ist F exakt nämlich

$$F(b_j) = \sum_{r=1}^{j} f_r \quad j = 1, \ldots, J$$

Ferner $F(x) = 0$ für $x \leq a_1$ und $F(x) = 1$ für $x \geq b_J$

Innerhalb der Klassen wird linear interpoliert: Für $x \in K_j = [a_j, b_j)$ gilt

$$F(x) \approx F(a_j) + \frac{f_j}{b_j - a_j}(x - a_j) = \sum_{r=1}^{j-1} f_r + f_j \frac{x - a_j}{b_j - a_j}$$



Mit Hilfe der (interpolierten) empirischen Verteilungsfunktion lassen sich nun auch die **p-Quantile** der Daten approximativ ermitteln.
Für das arithmetische Mittel (oder andere Größen) greift man auf die Mittelpunkte der Intervalle

$$\zeta_j = (a_j + b_j)/2$$

zurück und berechnet:

$$\bar{x} \approx \frac{1}{J} \sum_{j=1}^{J} \zeta_j n_j$$

Beispiel:

$$\bar{x} \approx \frac{3}{50} \cdot 250 + \frac{10}{50} \cdot 750 + \frac{20}{50} \cdot 1250 + \frac{10}{50} \cdot 1750 + \frac{7}{50} \cdot 2500 = 1365$$

Auswertung multivariater Daten

- Univariate Daten: Objekte der Grundgesamtheit sind durch ein einzelnes Merkmal beschrieben.
- Multivariate Daten: Objekte sind durch $p \geq 2$ Merkmale beschreiben.
- Vielfach stehen zweidimensionale Daten (bivariat) im Vordergrund.

p Merkmale: $X_1, X_2, \ldots, X_p$.
Dann ist in einer Grundgesamtheit $G = \{e_1, \ldots, e_n\}$ und der Vektor

$$(x_{i1}, \ldots, x_{ip})$$

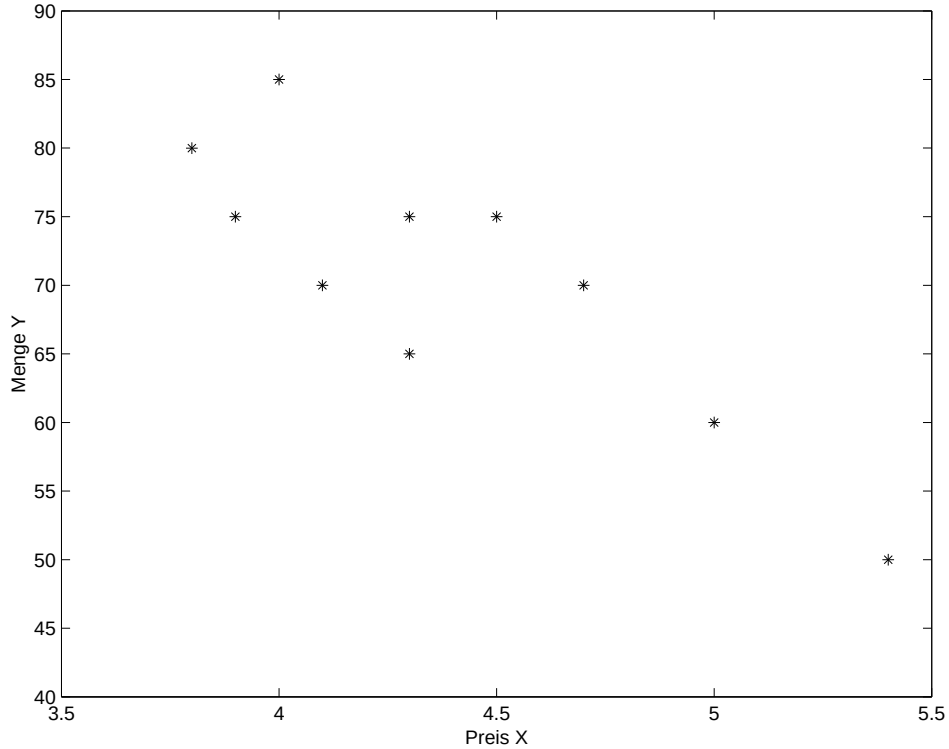
die Ausprägung von $X_1, \ldots, X_p$ für die Einheit e_i.

Die Urliste hat die Form einer $n \times p$ Datenmatrix D:

$$D = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}$$

Preis €/Liter (X)	Menge in Liter (Y)
4,70	70
4,30	75
3,80	80
4,50	75
5,40	50
5,00	60
4,10	70
4,30	65
3,90	75
4,00	85

Sind zwei Merkmale X und Y metrisch skaliert, veranschaulicht man sich die Daten in einem **Streudiagramm** (scatterplot)



Wie hängen X und Y von einander ab?

Vermutung: *Höhere Preise entsprechen geringeren Mengen*.

Kovarianz

Zur Herleitung einer **Zusammenhangsmaßzahl** für zwei Merkmale X und Y bilden wir die arithmetischen Mittel

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{und} \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

und die zugehörigen Varianzen

$$s_X^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \quad \text{und} \quad s_Y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2$$

Wir definieren die **Kovarianz** s_{XY} durch:

$$s_{XY} := \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

Eigenschaften der Kovarianz

- Die Kovarianz s_{XY} kann negativ sein! (Varianz s_X^2 ist stets ≥ 0)
- Es gilt $s_{XY} = s_{YX}$
- Durch einen Punkt $(x_i, y_i) \in \mathbb{R}^2$ und den Schwerpunkt $(\bar{x}, \bar{y}) \in \mathbb{R}^2$ wird offenbar ein Rechteck aufgespannt, dessen Flächeninhalt ist

$$F_i = |(x_i - \bar{x}) \cdot (y_i - \bar{y})|$$

- Ist $(x_i - \bar{x}) \cdot (y_i - \bar{y}) > 0$, so liegt der Punkt (x_i, y_i) im 1. oder 3. Quadranten. (bzgl. des Datenschwerpunktes $(\bar{x}, \bar{y})$)
- Ist $(x_i - \bar{x}) \cdot (y_i - \bar{y}) < 0$, so liegt (x_i, y_i) im 2. oder 4. Quadranten.

- $s_{XY} > 0$, so haben X und Y die gleiche Tendenz.
- $s_{XY} < 0$, so haben X und Y die entgegengesetzte Tendenz.
- Die Kovarianz ist lage-invariant und linear. Für die Transformation

$$(x_i, y_i) \mapsto (\hat{x}_i, \hat{y}_i)$$

$$\hat{x}_i := ax_i + b, \quad \text{und} \quad \hat{y}_i = cx_i + d \quad \text{mit} \quad a, b, c, d \in \mathbb{R}$$

gilt dann $s_{\hat{X}\hat{Y}} = acs_{XY}$, denn

$$s_{\hat{X}\hat{Y}} = \frac{1}{n-1} \sum_{i=1}^{n} (ax_i + b - (a\bar{x} + b))(cy_i + d - (c\bar{y} + d)) = acs_{XY}$$

- Offenbar ist s_{XY} nicht normiert und kann beliebige Werte annehmen.

Korrelationskoeffizienten

Normierung der Kovarianz durch die Standardabweichungen geben den **Korrelationskoeffizienten**:

$$r_{XY} := \frac{s_{XY}}{s_X s_Y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

1. Es gilt $r_{XY} = r_{YX}$, da $s_{XY} = s_{YX}$

2. Für $\hat{x}_i = ax_i + b$ und $\hat{y}_i = cy_i + d$ mit $a, b, c, d \in \mathbb{R}$ und $a \cdot c \neq 0$ gilt:

$$r_{\hat{X}\hat{Y}} = \frac{ac}{|a||c|} r_{XY}$$

   Dies folgt aus den bekannten Eigenschaften der Kovarianz und der Standardabweichung.

Es ist offensichtlich, dass sich $r_{\hat{X}\hat{Y}}$ und r_{XY} nur um das Vorzeichen des Produkts der Skalierungskonstanten a und c unterscheiden. Es gilt:

- $ac > 0$, dann ist $r_{\hat{X}\hat{Y}} = r_{XY}$
- $ac < 0$, dann ist $r_{\hat{X}\hat{Y}} = -r_{XY}$

3. Es gilt $r_{XY} \in [-1, 1]$.

4. $|r_{XY}| = 1$, gdw. $y_i = ax_i + b$ für alle $i = 1, \ldots, n$ gilt, d.h. ein exakter linearer Zusammenhang zwischen den Merkmalen X und Y besteht.

   - $r_{XY} = 1$, gdw. $a > 0$ und $b \in \mathbb{R}$ mit $y_i = ax_i + b$ für alle $i = 1, \ldots, n$.
   - $r_{XY} = -1$, gdw. $a < 0$ und $b \in \mathbb{R}$ mit $y_i = ax_i + b$ für alle $i = 1, \ldots, n$.

Häufigkeits- und Kontingenztafeln

Zwei Merkmale X und Y seien beliebig skaliert. Für X seien die möglichen Merkmalsausprägungen $\xi_1,\ldots,\xi_J$ und für Y heißen sie $\eta_1,\ldots,\eta_K$.

Sei nun die Urliste in Form einer $n \times 2$ Datenmatrix gegeben, so lässt sich hieraus eine sogenannte **Häufigkeitstabelle** erstellen.

Es sei n_{jk} die Anzahl der Datenpaare (x_i, y_i) mit $x_i = \xi_j$ und $y_i = \eta_k$.

$$n_{j\cdot} = \sum_{k=1}^{K} n_{jk} \quad \text{und} \quad n_{\cdot k} = \sum_{j=1}^{J} n_{jk}$$

sind die **absoluten Randhäufigkeiten** von ξ_j bzw. η_k.

		Y				
		η_1	η_2	$\cdots$	η_K	$\sum$
X	ξ_1	n_{11}	n_{12}	$\cdots$	n_{1K}	$n_{1.}$
	ξ_2	n_{21}	n_{22}	$\cdots$	n_{2K}	$n_{2.}$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	ξ_J	n_{J1}	n_{J2}	$\cdots$	n_{JK}	$n_{J.}$
	$\sum$	$n_{.1}$	$n_{.2}$	$\cdots$	$n_{.K}$	n

$$\sum_{j=1}^{J}\sum_{k=1}^{K} n_{jk} = \sum_{j=1}^{J} n_{j.} = \sum_{k=1}^{K} n_{.k} = n$$

Die Randhäufigkeiten $n_{.1}, \ldots, n_{.K}$ beziehen sich offenbar nur auf das Merkmal Y und die Randhäufigkeiten $n_{1.}, \ldots, n_{J.}$ nur auf das Merkmal X.

Kontingenztafel mit relativen Häufigkeiten:

		Y				
		η_1	η_2	$\cdots$	η_K	$\sum$
	ξ_1	f_{11}	f_{12}	$\cdots$	f_{1K}	$f_{1\cdot}$
X	ξ_2	f_{21}	f_{22}	$\cdots$	f_{2K}	$f_{2\cdot}$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	ξ_J	f_{J1}	f_{J2}	$\cdots$	f_{JK}	$f_{J\cdot}$
	$\sum$	$f_{\cdot 1}$	$f_{\cdot 2}$	$\cdots$	$f_{\cdot K}$	1

$f_{1\cdot}, \ldots, f_{J\cdot}$ heißt Randverteilung von X

$f_{\cdot 1}, \ldots, f_{\cdot K}$ heißt Randverteilung von Y

Bedingte Verteilungen

Von den gemeinsamen relativen Häufigkeiten zu unterscheiden sind die sogenannten **bedingten relativen Häufigkeiten**

Für festes $k \in 1, \ldots, K$ und ist für $j \in \{1, \ldots, J\}$

$$f_{j|Y=\eta_k} = \frac{f_{jk}}{f_{\cdot k}} = \frac{n_{jk}}{n_{\cdot k}}$$

die **bedingte relative Häufigkeit** von ξ_j unter der Bedingung $Y = \eta_k$.

Sie stellt die relative Häufigkeit des Wertes ξ_j in der Teilmenge der Objekte dar, die in der Variablen Y den Wert η_k haben.

$$f_{1|Y=\eta_k}, \ldots, f_{J|Y=\eta_k}$$

heißt die **bedingte Verteilung** von X unter der Bedingung $Y = \eta_k$.

Analog ist

$$f_{k|X=\xi_j} = \frac{f_{jk}}{f_{j\cdot}} = \frac{n_{jk}}{n_{j\cdot}}$$

die **bedingte relative Häufigkeit** von η_k unter der Bedingung $X = \xi_j$.

Die **bedingte Verteilung** von Y unter der Bedingung $X = \xi_j$ ist gegeben durch:

$$f_{1|X=\xi_j}, \ldots, f_{K|X=\xi_j}$$

Aus den bedingten relativen Häufigkeiten für Y unter der Bedingung $X = \xi_j$ und den absoluten Randhäufigkeiten von X können die gemeinsamen absoluten Häufigkeiten η_{jk} bestimmt werden, es gilt:

$$n_{jk} = f_{k|X=\xi_j}\, n_{j\cdot} = \frac{n_{jk}}{n_{j\cdot}} n_{j\cdot} \quad \text{und} \quad n_{jk} = f_{j|Y=\eta_k}\, n_{\cdot k} = \frac{n_{jk}}{n_{\cdot k}} n_{\cdot k}$$

Unabhängigkeit

Variable (Merkmale) X und Y heißen **deskriptiv unabhängig**, wenn gilt:

$$n_{jk} = \frac{n_{j\cdot} n_{\cdot k}}{n} \quad \text{für alle} \quad j = 1, \ldots, J \text{ und } k = 1, \ldots, K$$

Folgende Aussagen sind dazu äquivalent:

1. $f_{jk} = f_{j\cdot} f_{\cdot k}$ für alle $j = 1, \ldots, J,$ und $k = 1, \ldots, K$
2. $f_{j\cdot} = f_{j|Y=\eta_1} = f_{j|Y=\eta_2} = \cdots = f_{j|Y=\eta_K}$ für alle $j = 1, \ldots, J$
3. $f_{k\cdot} = f_{k|X=\xi_1} = f_{k|X=\xi_2} = \cdots = f_{k|X=\xi_J}$ für alle $k = 1, \ldots, K$

Beispiel:

	A	B	C	$\sum$
M	20	12	8	40
W	30	**18**	12	**60**
$\sum$	50	**30**	20	100

$18 = \frac{18}{60} \cdot 60$ sowie $18 = \frac{18}{30} \cdot 30$

	A	B	C	$\sum$
M	0,2	0,12	0, 08	0,4
W	0,3	0,18	0,12	0,6
$\sum$	0,5	0,3	0,2	1,0

Unabhängigkeit und Korrelation

X und Y seien deskriptiv unabhängige Variablen, dann ist $s_{XY} = 0$ und wir sagen X und Y sind unkorreliert, denn es gilt:

$$\begin{aligned} s_{XY} &= \frac{1}{n-1}\sum_{j=1}^{J}\sum_{k=1}^{K} n_{jk}(\xi_j - \bar{x})(\eta_k - \bar{y}) \\ &= \frac{1}{n-1}\sum_{j=1}^{J}\sum_{k=1}^{K} \frac{n_{j.}n_{.k}}{n}(\xi_j - \bar{x})(\eta_k - \bar{y}) \\ &= \frac{1}{n(n-1)}\sum_{j=1}^{J} n_{j.}(\xi_j - \bar{x}) \cdot \sum_{k=1}^{K}(\eta_k - \bar{y})n_{.k} = 0 \end{aligned} \tag{1}$$

Damit ist auch $r_{XY} = 0$. Die Umkehrung ist aber falsch!

Zusammenhangsmaß für ordinal skalierte Daten

Für ordinal skalierte Merkmale sind $\bar{x}, \bar{y}, {s_X}^2, s_Y^2$ und s_{XY} nicht sinnvoll berechenbar.

Idee: Die Daten x_i der Variablen X werden durch **Ränge** $R_X(x_i)$ ersetzt und der Korrelationskoeffizient für die Ränge bestimmt.

Daten $x_1, x_2, \ldots, x_n$ seien paarweise verschieden. Dann ist $R_X(x_i) = r$ der Rang von gleich $r \in \{1, \ldots, n\}$, wenn x_i in der aufsteigend sortierten Folge der x-Werte an der r-ten Position steht.

Der **Rangordnungskoeffizient von Spearman** ist nun definiert durch:

$$r_{Sp} = \frac{\sum_{i=1}^{n}(R_X(x_i) - \bar{R}_X)(R_Y(y_i) - \bar{R}_Y)}{\left(\sum_{i=1}^{n}(R_X(x_i) - \bar{R}_X)^2\right)^{\frac{1}{2}} \left(\sum_{i=1}^{n}(R_Y(y_i) - \bar{R}_Y)^2\right)^{\frac{1}{2}}}$$

Hierbei sind $\bar{R}_X$ und $\bar{R}_Y$ die mittleren Ränge der x- bzw. y-Werte. Es gilt natürlich $\bar{R}_X = \bar{R}_Y = \frac{1}{n}\sum_{k=1}^{n} k = (n+1)/2$.

Sind die x_i und die y_i jeweils paarweise verschieden, so gilt die vereinfachte Formel:

$$r_{Sp} = 1 - \frac{6 \sum_{i=1}^{n} (R_X(x_i) - R_Y(y_i))^2}{n(n^2-1)}$$

Hierbei verwendet man:

- $\bar{R}_X = \bar{R}_Y = \frac{1}{n} \sum_{k=1}^{n} k = (n+1)/2$
- $\sum_{i=1}^{n} (R_X(x_i) - \bar{R}_X)^2 = \sum_{k=1}^{n} (k - (n+1)/2)^2 = n(n-1)(n+1)/12$ insbesondere die Summenformel $\sum_{k=1}^{n} k^2 = n(n+1)(2n+1)/6$.

Kommen Datenwerte mehrfach vor (sog. Bindungen), so werden Durchschnittsränge gebildet:

$x_1 = 3.7$, $x_2 = 3.9$, $x_3 = 3.1$ und $x_4 = 3.7$
Dann ist $R_X(x_2) = 4$, $R_X(x_3) = 1$ und wegen $x_1 = x_4 = 3.7$ entfallen die Ränge 2 und 3 und werden durch einen Durchschnittsrang realisiert
$R_X(x_1) = R_X(x_4) = 2.5$

Eigenschaften des Rangordnungskoeffizienten r_{Sp}

1. $r_{Sp}(X, Y) = r_{Sp}(Y, X)$

2. r_{Sp} ist invariant gegenüber streng monoton wachsender Transformation der Daten x_i bzw. y_i.

3. $-1 \leq r_{Sp} \leq 1$

   - $r_{Sp} = 1$, gdw. $R_X(x_i) = R_Y(y_i)$ für alle $i = 1, \ldots, n$ (Ränge gleich).
   - $r_{Sp} = -1$, gdw. $R_X(x_i) = n - R_Y(y_i) + 1$ für alle $i, \ldots, n$ (Ränge entgegengesetzt).

r_{Sp} ist ein Maß, das einen monotonen Zusammenhang der Merkmale X und Y anzeigt.

Zusammenhangsmaß für nominal skalierte Daten

Daten in Kontingenztafel gegeben. Variablen sind deskriptiv unabhängig, falls

$$n_{jk} = \frac{n_{j.} n_{.k}}{n} \quad \text{für alle} \quad j = 1, \ldots, J \text{ und } k = 1, \ldots, K$$

Maß für die Abweichung von der Unabhängigkeit ist

$$\chi^2 = \sum_{j=1}^{J} \sum_{k=1}^{K} \frac{\left(n_{jk} - \frac{n_{j.} n_{.k}}{n}\right)^2}{\frac{n_{j.} n_{.k}}{n}} = n \sum_{j=1}^{J} \sum_{k=1}^{K} \frac{n_{jk}^2}{n_{j.} n_{.k}} - n$$

χ^2 ist nicht normiert! Statt χ^2 verwendet man den **Kontingenzkoeffizienten**

$$C = \sqrt{\frac{\chi^2}{\chi^2 + n} \cdot \frac{\min\{J, K\}}{\min\{J, K\} - 1}} \in [0, 1]$$

$C = 0$ gdw. $\chi^2 = 0$ gdw. X und Y deskriptiv unabhängig.

3. Clusteranalyse

1. *Womit befasst sich Clusteranalyse?*
2. Datenrepräsentation, Skalen (im Kap. 2), Distanz- und Ähnlichkeitsmaße
3. Hierarchische Clusteranalyse
4. Partitionierende Clusteranalyse
5. Fuzzy Clusteranalyse
6. Clusteranalyse mit neuronalen Netzen
7. Validation von Clusterungen

3.1 Was ist Clusteranalyse?

- **Clusteranalyse** ist eine Teildisziplin der multivariaten Statistik.

- In der Clusteranalyse werden Methoden und Algorithmen untersucht, die es erlauben Objekte der Grundgesamtheit in Gruppen einzuteilen.

- Die Objekte der Grundgesamtheit sind meist gegeben als

  – Distanzen bzw. Ähnlichkeiten zwischen Objektpaaren, in Form einer **Distanzmatrix** bzw. **Ähnlichkeitsmatrix**.
  – **Datenmatrix**, in der jedes Objekt als ein Zeilenvektor von d Merkmalsausprägungen repräsentiert ist.

- Im Gegensatz zur Musterklassifikation enthalten die Objekte keine Klasseninformation, etwa in Form eines Klassenmerkmals.

- Clusteranalyseverfahren sind den unüberwachten maschinellen Lernverfahren zuzuordnen.

- Eine Vielzahl von Clusteranalyseverfahren ist seit ca. 1960 entwickelt worden, dabei sind viele Verfahren erst mit zunehmender Hardwareleistung effizient anwendbar geworden.

- Ziel der Clusteranalyse: Die Objekte in Cluster aufteilen. Wobei ein Cluster eine Teilmenge der Grundgesamtheit ist, die auf der Basis eines festgelegten Distanz- oder Ähnlichkeitsmaßes, aus ähnlichen Objekten zusammengesetzt ist.

- Anwendungen: Quantisierung/Kompression multidimensionaler Daten (Vektoren)

Definitionsversuche

Everitt: *Cluster Analysis*, John Wiley, 1974.

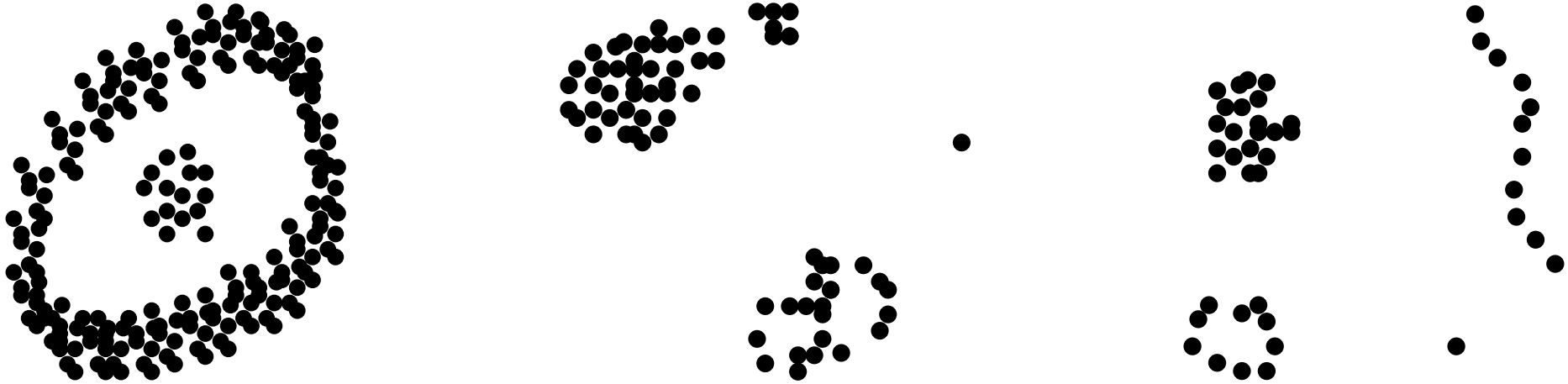
Hier findet man folgende Definitionsversuche für den Begriff des **Clusters**.

1. A cluster is a set of entities which are **alike**, and entities from different clusters are not alike.

2. A cluster is an aggregation of points in the test space such that the **distance** between any two points in the cluster is less than the distance between any point in the cluster and any point not in it.

3. Clusters may be described as connected regions of a multidimensional space containing a relatively **high density** of points, separated from other such regions by a region containing a relatively low density of points.

Cluster



- Form der Cluster variiert
- Anzahl er Objekte pro Cluster varriiert
- Anzahl der Cluster ist schwer zu bestimmen.

Typisches Clusterproblem

Gegeben seien n Objekte $G = \{e_1, \ldots, e_n\}$ durch ihre Merkmalsvektoren

$$x_1, \ldots, x_n \in \mathbb{R}^d.$$

Gesucht ist nun eine **Clusterung** mit $k \in \{1, \ldots, n\}$ Clustern

$$\mathcal{C} = \{C_1, \ldots, C_k\}$$

mit $C_j \subset G$ nichtleer, $C_i \cap C_j = \emptyset$ für $i \neq j$ und $G = C_1 \cup \ldots \cup C_k$.

Ferner soll gelten

- Objekte innerhalb eines Clusters C_j möglichst ähnlich
- Objekte aus verschiedenen Clusters C_j und C_i möglichst unähnlich

Hierfür benötigt man ein Distanz- oder Ähnlichkeitsmaß.

Bewertungsfunktion

Um die Güte einer Clusterung $\mathcal{C} = \{C_1, \ldots, C_k\}$ zu quantifizieren, ist eine (zu optimierende) Bewertungsfunktion $D(\mathcal{C})$ festzulegen. In $D(\mathcal{C})$ geht das gewählte Distanz- oder Ähnlichkeitsmaß p ein.

Formal etwa so:

$$\begin{aligned} D_p : P(k, G) &\rightarrow \mathbb{R}^+ \\ \mathcal{C} &\rightarrow D_p(\mathcal{C}) \end{aligned}$$

hierbei sei $P(k, G)$ ($=: P(k, n)$) die Menge der möglichen Clusterungen der Grundgesamtheit G mit n Objekten in k Cluster.

Falls k nicht durch die Anwendung spezifiziert ist, müssen Clusteranalysen mit verschiedenen Werten für k.

Theoretischer Algorithmus

1. Wähle eine Distanz- oder Ähnlichkeitsmaß p und Bewertungsfunktion D_p.

2. Setze k.

3. Berechne die optimale Clusterung $\mathcal{C}^{opt}$ durch

$$\mathcal{C}^{opt} := \text{argmin}\{D_p(\mathcal{C}) \mid \mathcal{C} \in P(k, G)\}$$

Für beliebig große n und k ist der Algorithmus nicht realisierbar, da die Zahl der möglichen Clusterungen rasch anwächst.

Es sei $s(k, n) := |P(k, n)|$ die Anzahl der Elemente aus $P(k, n)$, dann gilt

$$s(k, n) = \frac{1}{k!} \sum_{j=1}^{k} (-1)^{k-j} \binom{k}{j} j^n$$

Das Optimum kann nur für kleine n und k durch vollständige Aufzählung ermittelt werden.

Beispiel:

- $n = 20$ und $k = 4$, dann ist $s(k, n) = 45232115901$.
- $n = 100$ und $k = 5$, dann ist $s(k, n) > 10^{68}$.

Satz: Clusteranalyse ist NP-vollständig.

Gezeigt von P. Brucker (1974). Siehe dazu etwa:
Garey and Johnson: Computers and Intractability—A Guide to the Theory of NP-Completeness

3.2 Datenrepräsentation

Wir unterscheiden:

1. **Datenmatrix**-Darstellung
   Die Menge G wird die Form einer $n \times d$ Datenmatrix X dargestellt (n Objekte mit je d Merkmalen) (siehe Abschnitt 2):

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{pmatrix}$$

2. **Distanzmatrix-/Ähnlichkeitsmatrix**-Darstellung in Form einer $n \times n$ Matrix $P = (p_{ij})$, wobei p_{ij} ein Distanz- oder Ähnlichkeitswert für das Objektpaar $(e_i, e_j) \in G \times G$ ist.

Distanz- und Ähnlichkeitsmaße

Es sei X eine nichtleere Menge. Dann heißt $p : X \times X \to \mathbb{R}_+$ eine Distanz- bzw. Ähnlichkeitsfunktion auf X wenn gilt:

1. $p(x, y) \geq 0$ für alle $x, y \in X$.

2. $p(x, y) = p(y, x)$ für alle $x, y \in X$.

3. $p(x, x) = 0$ für alle $x \in X$. (Distanzfunktion)
   $p(x, x) \geq \max_y p(x, y)$ für alle $x \in X$. (Ähnlichkeitsfunktion)

Falls p eine Distanzfunktion ist, so heißt p eine **Metrik** auf X und (X, p) ein **metrischer Raum** falls außerdem gilt:

4. $p(x, y) = 0$, gdw. $x = y$.

5. $p(x, z) \leq p(x, y) + p(y, z)$ (Dreiecksungleichung).

Beispiele

Statt p (für proximity) wird d (für distance) oder s (für similarity) verwendet.

1. Normen auf $\mathbb{R}^n$ induzieren Distanzen $d(x, y)$:
   Für $r \in [1, \infty)$ ist definiert:

$$d_r(x, y) := \|x - y\|_r := \left(\sum_{i=1}^{n} |x_i - y_i|^r \right)^{\frac{1}{r}}$$

   Ausserdem für $r = \infty$

$$d_\infty(x, y) := \|x - y\|_\infty := \max_{i=1}^{n} |x_i - y_i|$$

   d_r sind Metriken (**Minkowski-Metriken**)!

   Die geläufigsten Minkowski-Metriken sind:

- $r = 1$ Manhattan- oder City-Block-Metrik
- $r = 2$ Euklidische Metrik
- $r = \infty$ Supremum- oder Maximum Metrik

2. d_1 ist auch auf $\mathbb{B} := \{0, 1\}^n$ eine Metrik die sogenannte **Hamming**-Metrik.

3. Das Skalarprodukt ist eine Ähnlichkeitsfunktion auf $\mathbb{B} := \{0, 1\}^n$:

$$s(x, y) := \langle x, y \rangle := \sum_{i=1}^{n} x_i y_i$$

Wegen der Symmetrie braucht die obere Dreiecksmatrix von P betrachtet werden. Ist P durch eine Distanzfunktion d festgelegt, ist $p_{ii} = 0$ für alle i und die Diagonale wird auch nicht betrachtet.

Matching-Koeffizienten

Seien nun die **Merkmale binär**. Erweiterung auf nominalskalierte Merkmale ist einfach möglich.

Merkmalsausprägungen sind $\xi_1 = 0$ und $\xi_2 = 1$.

Ähnlichkeitsmaß zwischen $x, y \in \mathbb{B}^n$ sind durch sogenannte Vierfeldertafeln oder Kontingenztafeln definiert:

	0	1
0	n_{00}	n_{01}
1	n_{10}	n_{11}

Dann gilt: $n = n_{00} + n_{01} + n_{10} + n_{11}$.

Beispiel:
$x = (0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0)$
$y = (0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0)$
Dann ist $n_{00} = 7$, $n_{11} = 8$, $n_{10} = 1$ und $n_{01} = 4$.

1. Simple-matching-coefficient (SMC)

$$s(x,y) = \frac{n_{00} + n_{11}}{n_{00} + n_{01} + n_{10} + n_{11}} = \frac{n_{00} + n_{11}}{n}$$

2. Jaccard-coefficient (JC)

$$s(x,y) = \frac{n_{11}}{n_{01} + n_{10} + n_{11}} = \frac{n_{11}}{n - n_{00}}$$

3. Rao-Russel-coefficient (RRC)

$$s(x,y) = \frac{n_{11}}{n}$$

Beispiel:
$n_{00} = 7$, $n_{11} = 8$, $n_{10} = 1$, $n_{01} = 4$ und
$s_{\text{SMC}} = \frac{15}{20}$, $s_{\text{JC}} = \frac{8}{13}$ und $s_{\text{RRC}} = \frac{8}{20}$.

Gemischte Merkmale

In der Praxis sind die Objekte häufig durch metrische und nominal skalierte Merkmale beschrieben, also $x = (x_m, x_n)$.

Idee: Berechne Distanz-/Ähnlichkeitsmaß auf den metrischen und den nominal skalierten Merkmale getrennt.

$p_m(x_m, y_m)$ sei das Distanz-/Ähnlichkeitsmaß zwischen den metrisch skalierten Teilvektoren x_m und y_m von x bzw. y.

$p_n(x_n, y_n)$ sei das Distanz-/Ähnlichkeitsmaß zwischen den nominal skalierten Teilvektoren x_n und y_n von x bzw. y.

Definiere das Gesamt-Distanz-/Ähnlichkeitsmaß für $\alpha > 0$ durch

$$p(x, y) = \alpha p_m(x_m, y_m) + (1 - \alpha) p_n(x_n, y_n)$$

Unvollständige Daten

In der Praxis sind die Daten gelegentlcih unvollständig erhobenen:

$$x\mu = (x_1\mu, x_2\mu, ?, x_4\mu, ?, x_6\mu, ?)$$

Es fehle nun $x_{\mu j}$.

1. Streiche Objekt μ

2. Streiche Merkmale j

3. Seien x^μ und x^ν unvollständig, dann lässt sich etwa die Euklidische Distanz bestimmen durch:

$$d_j := \begin{cases} 0 & x_j^\mu =? \text{ oder } x_j^\nu =? \\ (x_j^\mu - x_j^\nu) & \text{sonst} \end{cases}$$

und dann etwa

$$d_2(x_\mu, x_\nu) = \frac{n}{n - n_0} \sum_{j=1}^{n} d_j^2.$$

n_0 ist die Anzahl der gestrichenen Merkmale.

4. Wähle ein $K \geq 1$. Finde die nächsten K Nachbarn (ohne Berücksichtigung des j-ten Merkmals) in der Datenmatrix und ersetze $x_{\mu j}$ durch den arithmetischen Mittelwert oder Median oder Modus der j-ten Merkmalswerte dieser K nächsten Nachbarn.

3.3 Hierarchische Clusterverfahren

1. Einleitung
2. Agglomerative und divisive hierarchische Verfahren
3. Allgemeines agglomeratives Clusterverfahren
4. Spezielle agglomerative Clusterverfahren
5. Single-Linkage-Verfahren mit dem *minimal spanning tree*-Algorithmus

Einführung

Gegeben seien n Objekte $G = \{e_1, \ldots, e_n\}$ mit ihrer **Distanzmatrix**

$$D = (d_{\mu\nu})_{1 \leq \mu, \nu \leq n}$$

wobei $d_{\mu\nu}$ die Distanz zwischen den Objekten e_μ und e_ν (z.B. Distanz der zugehörigen Merkmalsvektoren)

$$d_{\mu\nu} := d(e_\mu, e_\nu) = d(x^\mu, x^\nu)$$

mit Distanzfunktion d. d braucht keine Metrik zu sein (d.h. Dreiecksungleiung muss nicht gelten, ferner sind Nulleinträge ausserhalb der Diagonalen zulässig (aus $d(x, y) = 0$ muss nicht $x = y$ folgen)).

Die vorgestellten Verfahren können leicht für Ähnlichkeitsfunktionen formuliert werden.

Eine Partition $\mathcal{C}$ auf G ist eine Menge $\mathcal{C} = \{C_1, \ldots, C_k\}$ mit

1. $C_i \neq \emptyset$ für alle i
2. $C_i \cap C_j = \emptyset$ für alle i und j mit $i \neq j$.
3. $C_1 \cup \cdots \cup C_k = G$

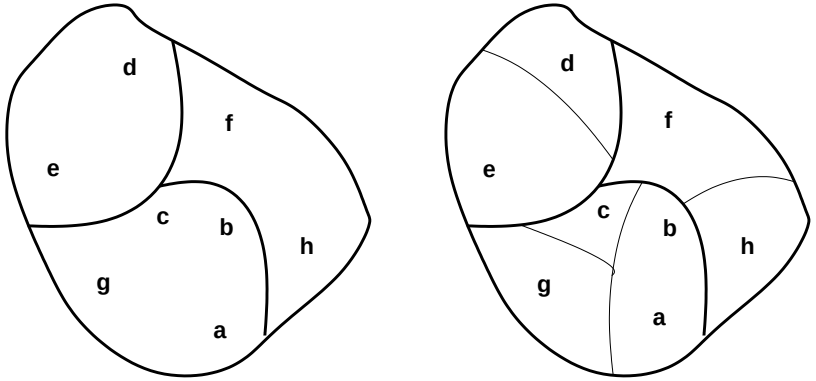
Eine Partition $\mathcal{B}$ heißt eine **Verfeinerung** einer Partition $\mathcal{C}$ falls jede Menge $B_i \in \mathcal{B}$ Teilmenge genau einer Menge $C_j \in \mathcal{C}$ ist. Notation: $\mathcal{B} \subset \mathcal{C}$.

Beispiel:

$$\mathcal{C} = \{\{a, b, c, g\}, \{d, e\}, \{f, h\}\}$$

$\mathcal{B} = \{\{a, b\}, \{c\}, \{g\}, \{d\}, \{e\}, \{f\}, \{h\}\}$

Dann ist $\mathcal{B} \subset \mathcal{C}$.



Agglomerative und divisive Verfahren

Man unterscheidet bei den hierarchischen Clusterverfahren zwischen

- **agglomerative (aufbauende) Clusterverfahren** (Partitionen werden im Verlauf der Clusteranalyse gröber)
- **divisive (teilende) Clusterverfahren** (Partitionen werden im Verlauf der Clusteranalyse feiner)
- Agglomerative Clusterverfahren starten mit der Anfangsclusterung

$$\mathcal{C}_1 = \{\{e_1\}, \{e_2\}, \ldots, \{e_n\}\}$$

dies ist offenbar die feinste Partition von G und terminieren mit

$$\mathcal{C}_n = \{\{e_1, \ldots, e_n\}\}$$

der gröbsten Partition von G.

Im Verlauf der Clusteranalyse werden jeweils zwei Cluster C_i und C_j zu einem Fusions-Cluster C_f vereinigt. So entsteht eine Folge von Clusterungen $(\mathcal{C}_i)_{i=1}^n$ mit der Eigenschaft: $\mathcal{C}_{i-1} \subset \mathcal{C}_i$.

- Die divisiven Clusterverfahren gehen genau anders vor. Anfangsclusterung ist

$$\mathcal{C}_1 = \{\{e_1, \ldots, e_n\}\}$$

In jedem Iterationsschritt i wird ein Cluster $C \in \mathcal{C}_i$ ausgewählt, dass in 2 Cluster C_i und C_j aufgeteilt wird, also $C_i \cap C_j = \emptyset$, C_i und C_j nichtleer und $C = C_i \cup C_j$. Sie terminieren mit

$$\mathcal{C}_n = \{\{e_1\}, \{e_2\}, \ldots, \{e_n\}\}$$

Das gibt eine Folge von Clusterungen $(\mathcal{C}_i)_{i=1}^n$ mit $\mathcal{C}_i \subset \mathcal{C}_{i-1}$.

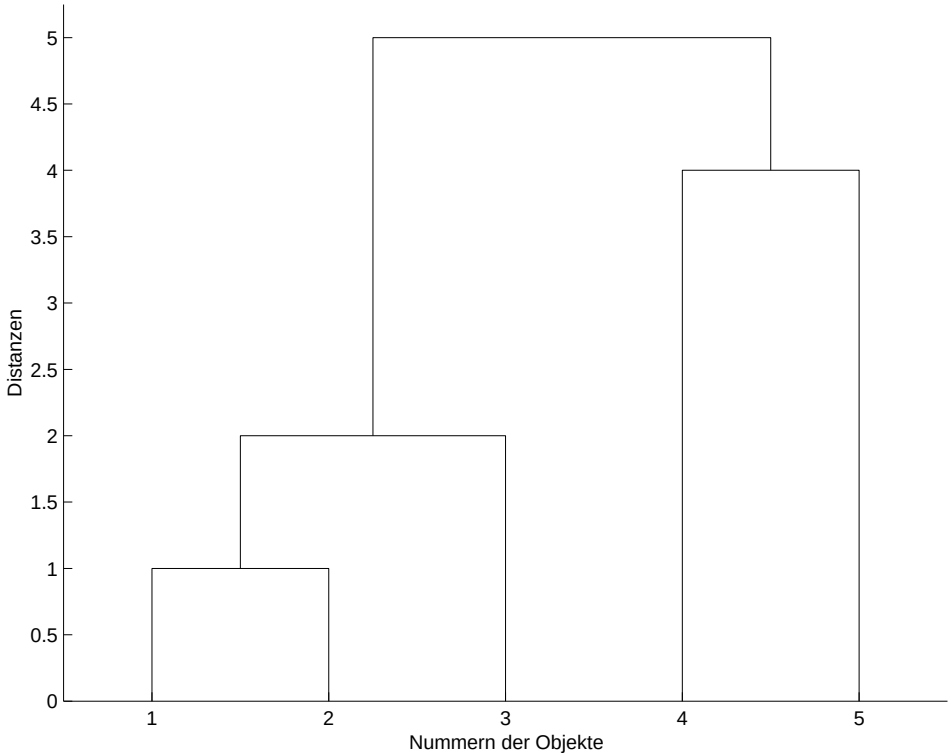
Dendrogramme

Graphische Darstellung von Folgen hierarchischer Clusterungen:

Clusterungen

1. $\{\{1,2,3,4,5\}\}$
2. $\{\{1,2,3\},\{4,5\}\}$
3. $\{\{1,2,3\},\{4\},\{5\}\}$
4. $\{\{1,2\},\{3\},\{4\},\{5\}\}$
5. $\{\{1\},\{2\},\{3\},\{4\},\{5\}\}$

Dendrogramm



Agglomerative Clusteranalyse

1. Eingabe ist eine $n \times n$ Distanzmatrix $D = (d_{ij})$ der Objekte $G = \{e_1, \ldots, e_n\}$.
2. Resultat ist eine Folge von Partitionen.
3. Start mit der feinsten Partition $\mathcal{C}_1 = \{\{e_1\}, \ldots, \{e_n\}\}$.
4. Im Verlauf der Clusteranalyse werden in jedem Verarbeitungsschritt jeweils die beiden Cluster fusioniert, die die geringste Distanz haben.
5. Distanzfunktion d die für G definiert ist, wird zu einer Distanzfunktion d_c auf der Potenzmenge von G fortgesetzt, dabei ist $d_c(\{e_i\}, \{e_j\}) := d(e_i, e_j)$.
6. Fortsetzung der Distanzfunktion durch verschiedene Ansätze möglich.

Agglomerativer Basisalgorithmus

1. Input: $n \times n$ Distanzmatrix $D = (d_{ij})$
2. Bestimme die beiden Cluster C_{i^*} und C_{j^*} mit der geringsten Distanz:

$$d_c(C_{i^*}, C_{j^*}) = \min_{(i,j)} d_c(C_i, C_j)$$

3. Fusioniere Cluster C_{i^*} und C_{j^*}, dh. Streiche C_{i^*} und C_{j^*} und nehme dafür $C_F := C_{i^*} \cup C_{j^*}$ in die bisherige Clusterung auf.
4. Aktualisiere die Distanzmatrix D
   (a) Streiche die zu C_{i^*} und C_{j^*} gehörenden Zeilen und Spalten.
   (b) **Berechne Distanzen** zwischen C_F und den verbleibenden Clustern C_r.
5. Gehe zu **2.** falls gröbste Clusterung $\mathcal{C}_n = \{\{e_1, \ldots, e_n\}\}$ nicht erreicht, sonst fertig;

Inter-Cluster-Distanzen

Es sind verschiedenen hierarchische, agglomerative Verfahren definiert, die sich durch die Berechnung der sogenannten **Inter-Cluster-Distanzen** zwischen dem Fusionscluster $C_F = C_{i^*} \cup C_{j^*}$ und den verbleibenden Clustern C_r unterscheiden.

Die Algorithmen lassen sich durch folgende Rekursionsformel beschreiben:

$$\begin{aligned} d_c(C_F, C_r) \quad := \quad & \alpha_{i^*} d_c(C_{i^*}, C_r) + \alpha_{j^*} d_c(C_{j^*}, C_r) + \\ & \beta d_c(C_{i^*}, C_{j^*}) + \gamma |d_c(C_{i^*}, C_r) - d_c(C_{j^*}, C_r)| \end{aligned}$$

in der die Parameter α_{i^*}, α_{j^*}, β und γ das jeweilige Clusterverfahren charakterisieren.

Single-Linkage-Clusteranalyse (SLC)

Hierbei gilt: $\alpha_{i^*} = \alpha_{j^*} = 1/2$, $\beta = 0$ und $\gamma = -1/2$, also

$$d_c(C_F, C_r) = \frac{1}{2} d_c(C_{i^*}, C_r) + \frac{1}{2} d_c(C_{j^*}, C_r) - \frac{1}{2} |d_c(C_{i^*}, C_r) - d_c(C_{j^*}, C_r)|$$

Wegen $\frac{x+y}{2} - \frac{|x-y|}{2} = \min\{x, y\}$ gilt dann einfach

$$d_c(C_F, C_r) = \min\{d_c(C_{i^*}, C_r), d_c(C_{j^*}, C_r)\}$$

- In jeder Fusionsstufe werden die beiden Cluster vereinigt, die die zueinander am nächsten liegenden Nachbarobjekte haben.
- SLC auch als *nearest neighbour* Clusterverfahren bekannt (nicht mit K-nearst-neighbour-Klassifikator zu verwechseln).
- SLC gehört zu den ältesten Verfahren (Sneath 1957).

Beispiel: SLC

Es sei eine Distanzmatrix D für $n = 5$ Objekte $G = \{e_1, \ldots, e_5\}$ gegeben:

$$D = D_0 = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 & e_5 \\ 0 & 1 & 2 & 9 & 13 \\ 1 & 0 & 5 & 10 & 10 \\ 2 & 5 & 0 & 5 & 13 \\ 9 & 10 & 5 & 0 & 4 \\ 13 & 10 & 13 & 4 & 0 \end{bmatrix}$$

Im ersten Fusionschritt werden die Cluster $\{e_1\}$ und $\{e_2\}$ vereinigt zu $C_F = \{e_1, e_2\}$, da sie die geringste Distanz haben ($d = 1$).

Distanzen zum Fusionscluster $\{e_1, e_2\}$ sind

$$\begin{aligned} d_c(\{e_1, e_2\}, \{e_3\}) &= \min\{d_c(\{e_1\}, \{e_3\}), d_c(\{e_2\}, \{e_3\})\} = 2 \\ d_c(\{e_1, e_2\}, \{e_4\}) &= \min\{d_c(\{e_1\}, \{e_4\}), d_c(\{e_2\}, \{e_4\})\} = 9 \\ d_c(\{e_1, e_2\}, \{e_5\}) &= \min\{d_c(\{e_1\}, \{e_5\}), d_c(\{e_2\}, \{e_5\})\} = 10 \end{aligned}$$

Damit ergibt sich die neue Distanzmatrix

$$D_1 = \begin{bmatrix} \{e_1, e_2\} & e_3 & e_4 & e_5 \\ 0 & \mathbf{2} & \mathbf{9} & \mathbf{10} \\ & 0 & 5 & 13 \\ & & 0 & 4 \\ & & & 0 \end{bmatrix}$$

Im zweiten Fusionschritt werden die Cluster $\{e_1, e_2\}$ und $\{e_3\}$ fusioniert ($d = 2$), dass Fusionscluster ist $C_F = \{e_1, e_2, e_3\}$.

Distanzen zum Fusionscluster sind

$$
\begin{aligned}
d_c(\{e_1, e_2, e_3\}, \{e_4\}) &= \min\{d_c(\{e_1, e_2\}, \{e_4\}), d_c(\{e_3\}, \{e_4\})\} = 5 \\
d_c(\{e_1, e_2, e_3\}, \{e_5\}) &= \min\{d_c(\{e_1, e_2\}, \{e_5\}), d_c(\{e_3\}, \{e_5\})\} = 10
\end{aligned}
$$

Damit ergibt ist die neue Distanzmatrix

$$
D_2 = \begin{bmatrix}
\{e_1, e_2, e_3\} & \{e_4\} & \{e_5\} \\
0 & \mathbf{5} & \mathbf{10} \\
 & 0 & 4 \\
 & & 0
\end{bmatrix}
$$

Nun haben die Cluster $\{e_4\}$ und $\{e_5\}$ die geringste Distanz und werden fusioniert zu $C_F = \{e_4, e_5\}$ (Distanz ist $d = 4$);

Distanz zum Fusionscluster

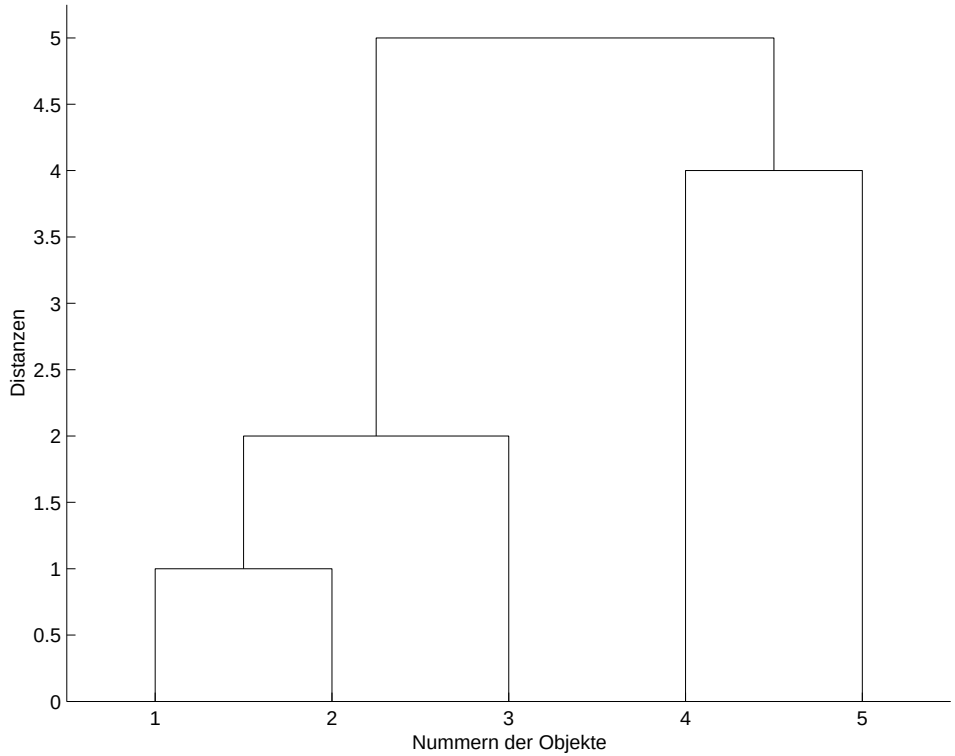
$$
d_c(\{e_1, e_2, e_3\}, \{e_4, e_5\}) = \min\{d_c(\{e_1, e_2, e_3\}, \{e_4\}), d_c(\{e_1, e_2, e_3\}, \{e_5\})\} = 5
$$

Somit ist die Distanzmatrix

$$D_3 = \begin{bmatrix} \{e_1, e_2, e_3\} & \{e_4, e_5\} \\ 0 & \mathbf{5} \\ & 0 \end{bmatrix}$$

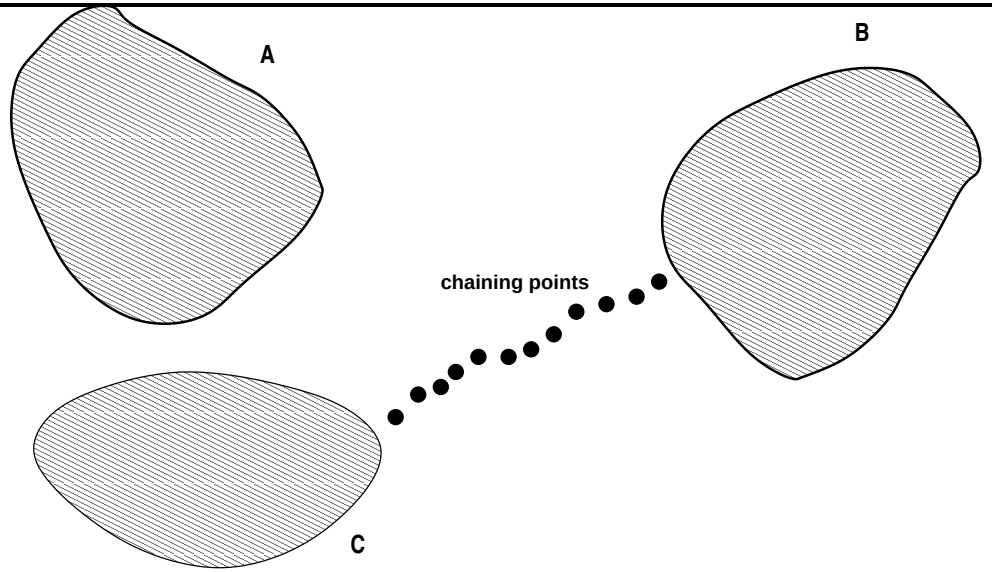
Schließlich werden $\{e_1, e_2, e_3\}$ und $\{e_4, e_5\}$ fusioniert zum Fusionscluster $C_F = \{G\}$ (Distanz ist $d = 5$)

Der Clusterprozess ist nun abgeschlossen und lässt sich übersichtlich in Form eines Dendrogramms darstellen:



Chaining-Effekt bei SLC

Die Situation sei wie folgt: 3 Bereiche mit vielen Datenpunkten, die bereits zu Clustern A, B, C zusammengefasst wurden. Außerdem eine Kette kleinerer Cluster (oder Punkte) die zwischen B und C verläuft.



Mit dem SLC-Verfahren werden nicht die Cluster **A** und **C** fusioniert, sondern die visuell separierten Cluster **C** und **B**, denn durch die Minimumbildung bei der Berechnung der Distanz zwischen dem Fusionszentrum und den verbleibenden Restclustern, diese durch sogenannte *chaining points* verbunden.

Der Chaining-Effekt führt dazu, dass Distanzen zwischen Objekte eines Clusters häufig größer sind, als Distanzen zwischen Objekten verschiedener Cluster.

Complete-Linkage-Clustering (CLC)

Es ist $\alpha_{i^*} = \alpha_{j^*} = 1/2$, $\beta = 0$ und $\gamma = 1/2$.

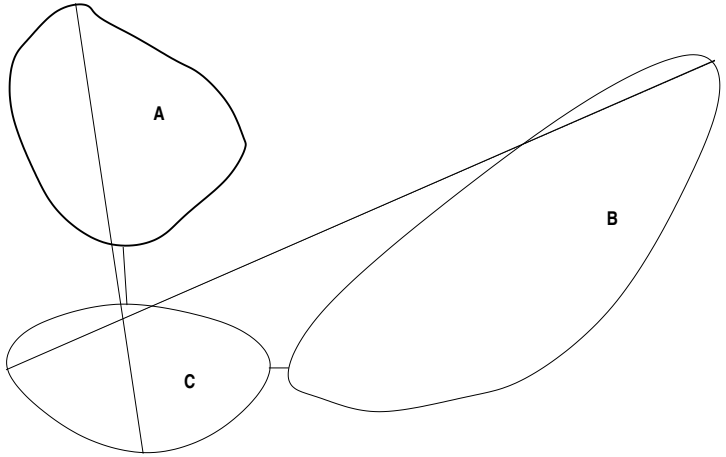
$$d_c(C_F, C_r) = \frac{1}{2} d_c(C_{i^*}, C_r) + \frac{1}{2} d_c(C_{j^*}, C_r) + \frac{1}{2} |d_c(C_{i^*}, C_r) - d_c(C_{j^*}, C_r)|$$

Wegen $\frac{x+y}{2} + \frac{|x-y|}{2} = \max\{x, y\}$ erhalten wir:

$$d_c(C_F, C_r) = \max\{d_c(C_{i^*}, C_r), d_c(C_{j^*}, C_r)\}$$

- Auf jeder Fusionsstufe werden die beiden Cluster fusioniert, die über die minimale Maximaldistanz von zwei Objekten verfügen.
- Deshalb unter dem Namen *minimal furthest neighbours*-Verfahren bekannt (Mac Naughton-Smith 1965)
- CLC ist altes Clusteranalyseverfahren (Soerensen 1948)

CLC vs SLC



Fusion

- SLC: Cluster **C** und Cluster **B**
- CLC: Cluster **C** und Cluster **A**

- Beim Complete-Linkage werden zwei Cluster nicht auf der Basis einer einzelnen kleinen Distanz zwischen Objektpaaren (*single link*) fusioniert, sondern die Distanzen aller Objektpaare werden betrachtet (*complete link*).

- Chaining tritt beim Complete-Linkage nicht auf.

- Complete-Linkage und Single-Linkage sind gewissermaßen die beiden Extreme bei der Verrechnung der Fusionsdistanzen.

Group-Average-Verfahren

$\alpha_{i^*} = \frac{m_{i^*}}{m_{i^*}+m_{j^*}}$, $\alpha_{j^*} = \frac{m_{j^*}}{m_{i^*}+m_{j^*}}$ und $\beta = \gamma = 0$.

Dann hat die Rekursionsformel die folgende Form:

$$\begin{aligned} d_c(C_F, C_r) &= \frac{1}{m_{i^*} + m_{j^*}} \Big(m_{i^*} d_c(C_{i^*}, C_r) + m_{j^*} d_c(C_{j^*}, C_r) \Big) \\ &= \frac{1}{(m_{i^*} + m_{j^*}) m_r} \sum_{e_i \in C_F} \sum_{e_j \in C_r} d(e_i, e_j) \end{aligned}$$

Die Distanz zwischen den beiden Cluster C_F und C_r wird durch das arithmetische Mittel der Distanzen aller Objektpaare der beiden beteiligten Cluster C_F und C_r definiert.

Bei einem Fusionschritt werden somit die beiden Cluster fusioniert, für die das arithmetische Mittel aller Objektdistanzen minimal ist.

Centroid-Verfahren

$\alpha_{i^*} = \frac{m_{i^*}}{m_{i^*}+m_{j^*}}$, $\alpha_{j^*} = \frac{m_{j^*}}{m_{i^*}+m_{j^*}}$ und ferner sind $\beta = -\frac{m_{i^*}m_{j^*}}{m_{i^*}+m_{j^*}}$ und $\gamma = 0$.

Dann hat die Rekursionsformel die folgende Form (mit $m_F := m_{i^*} + m_{j^*}$):

$$d_c(C_F, C_r) = \frac{m_{i^*}}{m_F} d_c(C_{i^*}, C_r) + \frac{m_{j^*}}{m_F} d_c(C_{j^*}, C_r) - \frac{m_{i^*}m_{j^*}}{m_F^2} d_c(C_{i^*}, C_{j^*})$$

Idee beim Centroid-Verfahren:

- Die Objekte e_μ sind durch Vektoren $x^\mu \in \mathbb{R}^n$ beschrieben und die Euklidische Metrik wird als Abstandsfunktion verwendet.
- Die Cluster C_i werden repräsentiert durch **Prototypen/Clusterzentren**

$$c_i = \frac{1}{m_i} \sum_{e_\mu \in C_i} x^\mu \in \mathbb{R}^n.$$

Median-Verfahren

$\alpha_{i^*} = \frac{1}{2}$, $\alpha_{j^*} = \frac{1}{2}$, ferner $\beta = -\frac{1}{4}$ und $\gamma = 0$.

Das Median-Verfahren ist eine Näherung an das Centroid-Verfahren: Beide Cluster bei der Fusion werden hier gleich gewichtet. Man nimmt an $m_{i^*} = m_{j^*} = m$ und gewinnt so:

$$d_c(C_F, C_r) = d^2(c_F, c_r) = \frac{1}{2}\big(d^2(c_{i^*}, c_r) + d^2(c_{j^*}, c_r)\big) - \frac{1}{4}d^2(c_{j^*}, c_{i^*})$$

Für die quadratische Euklidische Metrik als Abstandsfunktion lässt sich noch eine anschauliche Interpretation geben:
Prototyp c_F des Fusionsclusters C_F wird durch den Mittelpunkt ($=$ Median) der Verbindungslinie von c_{i^*} und c_{j^*} festgelegt.

Unweighted-Average-Verfahren

$\alpha_{i^*} = \alpha_{j^*} = \frac{1}{2}$ und $\beta = \gamma = 0$.

Somit hat die Rekursionsformel die Gestalt

$$d_c(C_F, C_r) = \frac{1}{2}\Big(d_c(C_{i^*}, C_r) + d_c(C_{j^*}, C_r)\Big)$$

Geometrisch ist das Unweighted-average-Verfahren nicht interpretierbar.

Das Unweighted-Average-Verfahren ergibt sich als Näherung aus dem Group-Average-Verfahren durch $m_{i^*} = m_{j^*} = m$.

Ward's Verfahren

$\alpha_{i^*} = \frac{m_{i^*}+m_r}{m_{i^*}+m_{j^*}+m_r}$, $\alpha_{j^*} = \frac{m_{j^*}+m_r}{m_{i^*}+m_{j^*}+m_r}$, $\beta = -\frac{m_r}{m_{i^*}+m_{j^*}+m_r}$ und $\gamma = 0$.

Dann hat die Rekursionsformel die folgende Form (mit $M := m_{i^*} + m_{j^*} + m_r$):

$$d_c(C_F, C_r) = \frac{m_r + m_{i^*}}{M} d_c(C_{i^*}, C_r) + \frac{m_r + m_{j^*}}{M} d_c(C_{j^*}, C_r) - \frac{m_r}{M} d(C_{i^*}, C_{j^*})$$

- Verfahren wurde von Ward 1963 entwickelt.
- Ursrpünglich ein allgemeines agglomeratives Verfahren, das in jedem Fusionschritt eine beliebige Zielfunktion zu optimieren versucht.
- Ward benutzte die Summe der quadrierten Euklidischen Abstände der Datenvektoren zum Clusterschwerpunkt als Distanzfunktion.
- Wishart (1969) zeigt: Ward'sche Clusteranalyse genügt der Rekursionsformel für die quadrierte Euklidische Distanz als Abstandsmaß.

Die Summe der quadratischen Abstände der Datenpunkte $x^\mu \in C_r$ zum Clusterzentrum c_r ist

$$E_r = \sum_{x_\mu \in C_r} \|x_\mu - c_r\|_2^2$$

Die Quadratsumme für eine Clusterung mit k Clustern ist somit: $E = \sum_{j=1}^{k} E_j$

ΔE bezeichne die Zunahme der Quadratsumme bei der Fusion von C_{i^*} und C_{j^*} zu C_F, also

$$\Delta E = E_F - E_{i^*} - E_{j^*}$$

Man kann zeigen, dass ΔE nur von c_{j^*} und c_{i^*} abhängt, nämlich:

$$\Delta E = \frac{m_{i^*} m_{j^*}}{m_{i^*} + m_{j^*}} \|c_{i^*} - c_{j^*}\|_2^2$$

Ward's Methode minimiert ΔE in jedem Fusionschritt.

Überblick

Verfahren	α_{i^*}	α_{j^*}	β	γ
single-linkage	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$
complete-linkage	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
unweighted average	$\frac{1}{2}$	$\frac{1}{2}$	0	0
Median	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{4}$	0
group average	$\frac{m_{i^*}}{m_{i^*}+m_{j^*}}$	$\frac{m_{j^*}}{m_{i^*}+m_{j^*}}$	0	0
centroid	$\frac{m_{i^*}}{m_{i^*}+m_{j^*}}$	$\frac{m_{j^*}}{m_{i^*}+m_{j^*}}$	$-\frac{m_{i^*}m_{j^*}}{(m_{i^*}+m_{j^*})^2}$	0
Ward's Verfahren	$\frac{m_{i^*}+m_r}{m_{i^*}+m_{j^*}+m_r}$	$\frac{m_{j^*}+m_r}{m_{i^*}+m_{j^*}+m_r}$	$-\frac{m_r}{m_{i^*}+m_{j^*}+m_r}$	0

m_k bezeichnet die Anzahl der Objekte im Cluster C_k.

Minimal Spanning Tree

- Graph $G = (V, E)$, V Knotenmenge und $E \subset V \times V$ Kantenmenge.
- G heisst zusammenhängend, falls für alle $v_i, v_j \in V$ ein $l \geq 2$ und Pfad $(v_i, \ldots, v_j) \in E^l$ existiert. Pfad $(v_i, \ldots, v_j)$ heisst geschlossen, falls $v_i = v_j$ gilt.
- Ein Baum ist ein zusammenhängender Graph ohne geschlossene Pfade.
- $G(V, E)$ ein Baum, dann ist $|E| = |V| - 1$.
- Ein (auf-)spannender Baum eines Graphen, ist ein Baum, der sämtliche Knoten enthält.
- Sind die Kanten eines Graphen gewichtet, so ist ein minimaler spannender Baum (minimal spanning tree), ein Baum mit minimalem Kantensumme, also Summe aller Kantengewichte des spannenden Baumes

MST-Algorithmus nach Kruskal

- Input: Graph $G = (V, E)$, Kantengewichte $w : E \to \mathbb{R}$ mit $w(v_i, v_j) = w_{ij}$
- Output: $T_{\min} = (V_T, E_T)$ mit $V_T = V$ und mit

$$\sum_{(v_i,v_j)\in E_T} w_{ij} = \min_{T\subset G, T\text{Baum}} \sum_{(v_i,v_j)\in T} w_{ij}$$

1. Wähle $v_i \in V$ und setze $V_T = \{v_i\}$ und $E_T = \emptyset$.
2. Bestimme $v_{i^*} \in V_T$ und $v_{j^*} \in V \setminus V_T$ mit $(v_{i^*}, v_{j^*}) \in E$ und mit

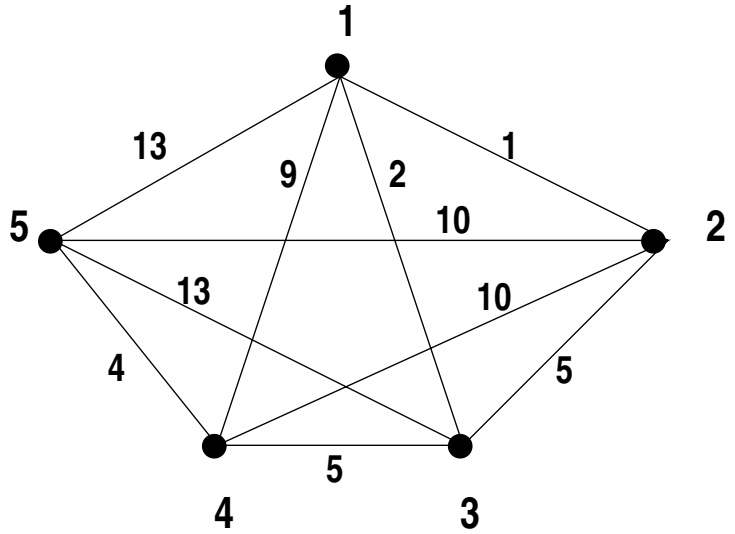
$$w_{i^*j^*} = \min\{w_{ij} \ : \ v_i \in V_T,\ v_j \in V \setminus V_T,\ (v_i, v_j) \in E\}$$

   Setze dann $V_T = V_T \cup \{v_{j^*}\}$ und $E_T = E_T \cup (v_{i^*}, v_{j^*})$.
3. If $V_T = V$ Then Stop Else Goto 2.

Beispiel

Es sei $V = \{1, 2, 3, 4, 5\}$ und $E = V \times V \setminus \{(i, i) \ : \ i = 1, \ldots, 5\}$.
Die Kantengewichtung ist gegeben durch folgende Matrix

$$W = \begin{vmatrix} 0 & 1 & 2 & 9 & 13 \\ 1 & 0 & 5 & 10 & 10 \\ 2 & 5 & 0 & 5 & 13 \\ 9 & 10 & 5 & 0 & 4 \\ 13 & 10 & 13 & 4 & 0 \end{vmatrix}$$



1. Wähle nun $i = 4$, dann ist $V_T = \{4\}$ und $E_T = \emptyset$.

2. Gemäß Schritt 2 werden nun nacheinander folgende Kanten ausgewählt:

   (a) $(4, 5)$ mit Kantenbewertung $w_{45} = 4$.

(b) $(4, 3)$ mit Kantenbewertung $w_{43} = 5$.
(c) $(3, 1)$ mit Kantenbewertung $w_{31} = 2$.
(d) $(1, 2)$ mit Kantenbewertung $w_{12} = 1$.

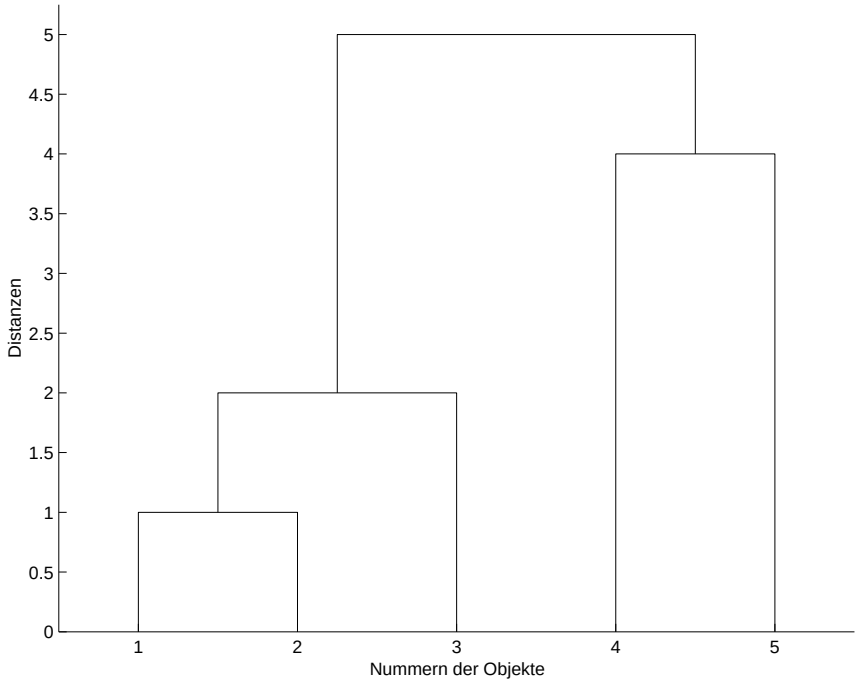
3. Gesamtbewertung : $\sum_{(i,j)\in E_T} = 12$

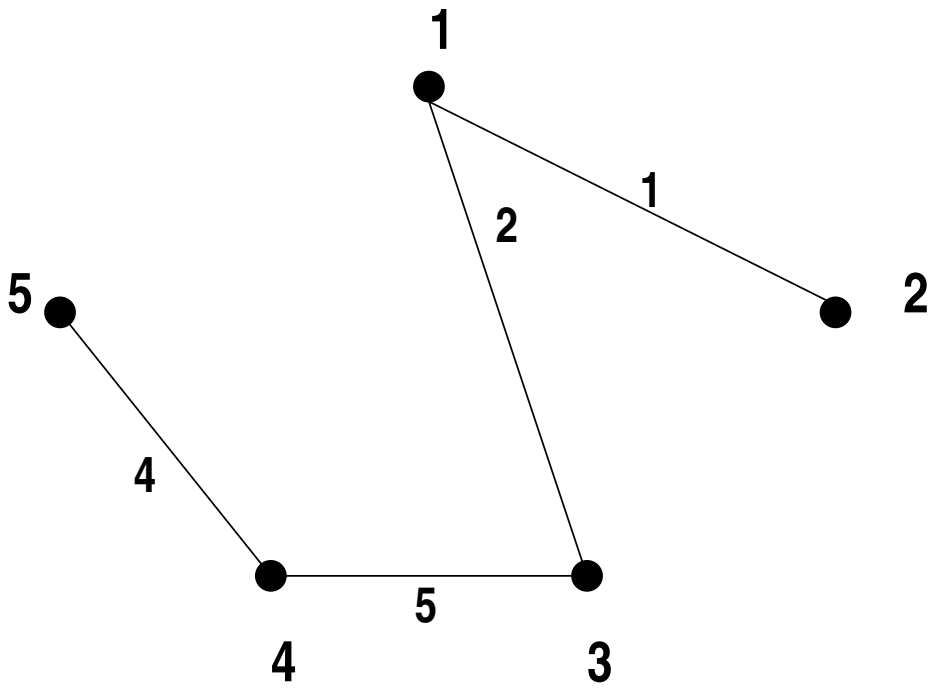
4. Laufzeit: $O(|V| + |E| \log_2 |E|)$ speziell für $E = V \times V$ folgt $O(|V|^2 \log_2 |V|)$.

Eine Komponente auf dem Distanzniveau d ist eine Komponente in der jeder Knoten mit jedem anderen Konten der Komponenten über einen Pfad mit Kanten mit jeweils Kantengewicht $\leq d$ verbunden ist.

Die Clusterfolge beim SLC ergibt sich nun als Folge von Zusammenhangskomponenten für Distanzniveaus $d_1, d_2, \ldots, d_{n-1}$ wobei die d_i die minimalen Distanzen der zu fusionierenden Cluster sind (Schritt 2 des allgemeinen agglomerativen Verfahrens).

MST und Dendrogramm





3.4 Partitionierende Clusteranalyse

- Die hierarchische Clusteranalyse liefert eine Folge von Partitionen (Verfeinerungen bzw. Vergröberungen) von Clustern.

- In vielen Anwendungen ist man nur an einer einzelnen Partition interessiert, etwa bei **Datenkompression** wenn Repräsentanten von Datenpunktmenge gesucht werden.

- Die **partitionierende** Clusteranalyse bestimmt genau eine Partition.

- Ausgangspunkt der partitionierenden Clusteranalyseverfahren sind Datenmatrizen.

- Voraussetzung: metrisch skalierte Merkmale und Euklidische Metrik.

Allgemeines Problem

Gegeben n Objekte durch Merkmalsvektoren $x_\mu = (x_1^\mu, \ldots, x_p^\mu) \in \mathbb{R}^p$ in einer $n \times p$ Datenmatrix X.

Bestimme nun eine Partition/Clusterung $\mathcal{C} = \{C_1, \ldots, C_k\}$ der Länge $k \in \mathbb{N}$.

Für die Bewertung einer Clusterung wird eine Zielfunktion/Bewertungsfunktion $D(\mathcal{C})$ definiert:

$$\begin{aligned} D : P(k, G) &\rightarrow \mathbb{R}^+ \\ \mathcal{C} &\rightarrow D(\mathcal{C}) \end{aligned}$$

$P(k, G)$ Menge der möglichen Clusterungen der Grundgesamtheit G in k Cluster.

Die Cluster C_i sind durch Prototypen/Clusterzentren $c_i \in \mathbb{R}^p$ repräsentiert

Idee: Jeder Datenpunkt x^μ wird dem Cluster C_{i^*} zugeordend, dessen Zentrum c_{i^*} zu x_μ am nächsten liegt.

Varianzkriterium

Sehr viele Bewertungsfunktionen für Clusterungen $\mathcal{C} = \{C_1, \ldots, C_k\}$ basieren auf varianzanalytische Überlegungen.

Das wichtigste ist wohl das **Varianz-/Fehlerquadratsummenkriterium**

$$D_{\text{Var}}(\mathcal{C}) = D_{\text{Var}}(\{C_1, \ldots, C_k\}) := \sum_{j=1}^{k} \sum_{x^\mu \in C_j} \sum_{i=1}^{p} (x_i^\mu - c_{ji})^2 \to \min$$

Hierbei ist $c_j \in \mathbb{R}^p$ der Schwerpunkt der Datenpunkte des Clusters C_j, also

$$c_j = \frac{1}{|C_j|} \sum_{x^\mu \in C_j} x^\mu.$$

Streuungszerlegung

Daten $x^\mu \in \mathbb{R}^p$ innerhalb eines Cluster C_j seien repräsentiert durch ihren Schwerpunkt $c_j \in \mathbb{R}^p$. (Wir identifizieren die Objekte e_μ und die Daten x^μ).

Dann gilt offenbar für für $x^\mu \in C_j$ und $y \in \mathbb{R}^p$

$$x^\mu - y = (x^\mu - c_j) + (c_j - y)$$

daraus folgt dann:

$$\begin{aligned}(x^\mu - y)(x^\mu - y)^T &= (x^\mu - c_j)(x^\mu - c_j)^T + (c_j - y)(x^\mu - c_j)^T \\ &\quad +(x^\mu - c_j)(c_j - y)^T + (c_j - y)(c_j - y)^T\end{aligned}$$

Summation über alle Datenpunkte x^μ des Clusters C_j liefert dann

$$\begin{aligned}\sum_{x^\mu \in C_j} (x^\mu - y)(x^\mu - y)^T &= \sum_{x^\mu \in C_j} (x^\mu - c_j)(x^\mu - c_j)^T + \underbrace{\sum_{x^\mu \in C_j} (x^\mu - c_j)(c_j - y)^T}_{=0} \\ &\quad + \underbrace{\sum_{x^\mu \in C_j} (c_j - y)(x^\mu - c_j)^T}_{=0} + |C_j|(c_j - y)(c_j - y)^T \\ &= \sum_{x^\mu \in C_j} (x^\mu - c_j)(x^\mu - c_j)^T + |C_j|(c_j - y)(c_j - y)^T\end{aligned}$$

Setzen wir nun $y = \bar{x} = \frac{1}{M}\sum_{\mu=1}^{M} x^\mu$ (Schwerpunkt aller Datenpunkte x^μ) dann folgt für jedes Cluster C_j

$$\sum_{x^\mu \in C_j} (x^\mu - \bar{x})(x^\mu - \bar{x})^T = \underbrace{\sum_{x^\mu \in C_j} (x^\mu - c_j)(x^\mu - c_j)^T}_{W_j} + \underbrace{|C_j|(c_j - \bar{x})(c_j - \bar{x})^T}_{B_j}$$

Summation über alle Cluster $C_1, \ldots, C_k$ liefert nun

$$T := \sum_{x^\mu \in X} (x^\mu - \bar{x})(x^\mu - \bar{x})^T = \underbrace{\sum_{j=1}^{k} W_j}_{W} + \underbrace{\sum_{j=1}^{k} B_j}_{B}$$

Dies ergibt die sogenannte *Streuungszerlegung*:

$$\mathbf{T} = \mathbf{W} + \mathbf{B}.$$

- T ist die totale Streuungsmatrix der Datenmenge
- W die Streuungsmatrix innerhalb der Cluster
- B die Streuungsmatrix außerhalb der Cluster.
- $\frac{1}{|C_j|-1} W_j$ heißt ist Kovarianzmatrix des j-ten Clusters
- W_j nennen wir die Streuungsmatrix vom j-ten Cluster.

Satz: Das Varianzkriterium lässt sich umformulieren

$$D_{\text{Var}}(\{C_1, \ldots, C_k\}) = \text{tr}(W) \to \min$$

hierbei heisst $\text{tr}(W) = \sum_{i=1}^{p} w_{ii}$ die **Spur** der Matrix $W \in \mathbb{R}^{p^2}$.

Eigenschaften der **Spurabbildung**:

1. $\text{tr}(A + B) = \text{tr}(A) + \text{tr}(B)$ und $\text{tr}(\alpha A) = \alpha \text{tr}(A)$ für $A, B \in \mathbb{R}^{p^2}$ und $\alpha \in \mathbb{R}$.
2. $\text{tr}(AB) = \text{tr}(BA)$ für $A, B \in \mathbb{R}^{p^2}$.
3. $\text{tr}(A) = \sum_{i=1}^{p} \lambda_i$ wobei $\lambda_1, \ldots, \lambda_n \in \mathbb{C}$ die Eigenwerte von A sind.
4. $\text{tr}(yy^T) = \|y\|_2^2$ mit $y^T = (y_1, \ldots, y_p)$ und $yy^T \in \mathbb{R}^{p^2}$.

Für die Spur von W gilt:

$$\begin{aligned}
\mathsf{tr}(W) &= \sum_{j=1}^{k} \mathsf{tr}(W_j) \\
&= \sum_{j=1}^{k} \sum_{x_\mu \in C_j} \mathsf{tr}\big((x^\mu - c_j)(x^\mu - c_j)^T\big) \\
&= \sum_{j=1}^{k} \sum_{x_\mu \in C_j} \|x^\mu - c_j\|_2^2
\end{aligned}$$

Varianzkriterium kann also auch als Spurkriterium (für W) aufgefasst werden:

$$D_{\mathsf{Var}}(\{C_1, \ldots, C_k\}) = \mathsf{tr}(W) \rightarrow \min$$

- Das Varianzkriterium minimiert also $\mathsf{tr}(W)$.

- Wegen $T = W + B$ gilt $\text{tr}(T) = \text{tr}(W) + \text{tr}(B)$.

- Es $\text{tr}(T) = \text{const}$ von der Clusterung unabhängig. Minimiert man die Spur von W, so maximiert die Spur von B.

- Es gilt

$$\text{tr}(B) = \sum_{j=1}^{k} |C_j| \text{tr}\big((c_j - \bar{x})(c_j - \bar{x})^T\big) = \sum_{j=1}^{k} |C_j| \|c_j - \bar{x}\|_2^2$$

  Die Summe der quadrierten Euklidischen Abstände zwischen den Clusterzentren c_j und dem Datenschwerpunkt $\bar{x}$ (gewichtet mit $|C_j|$) wird durch das Varianzkriterium maximiert.

K-Means-Clusteranalyse

Input : n Datenpunkte $x^\mu \in \mathbb{R}^p$ repräsentiert als $n \times p$ Datenmatrix X.

1. Wähle Clusteranzahl $k \in \{1, \ldots, n\}$ und maximale Iterationszeit t_{max}.
2. Setze Iterationszeit $t = 0$ und bestimme Anfangspartition von X.

$$\mathcal{C}(t) := \{C_1(t), \ldots, C_k(t)\}$$

3. Bestimme Schwerpunkte der k Cluster $C_j(t)$ (*k-means*)

$$c_j(t) := \frac{1}{|C_j(t)|} \sum_{x^\mu \in C_j(t)} x^\mu$$

4. Bestimme die sogenannte **Minimaldistanzpartition** $\mathcal{C}(t+1)$ durch

$$C_j(t+1) = \{x^\mu \ : \ \|x^\mu - c_j(t)\| = \min_{i=1,\ldots,k} \|x^\mu - c_i(t)\|\} \qquad j = 1, \ldots, k$$

(Falls das Minimum nicht eindeutig ist, wähle j zufällig.)

5. Falls $t < t_{\text{max}}$ dann $t := t + 1$ und Goto 3.

- Andere Bezeichnungen: Batch-Modus-K-means, Minimaldistanzverfahren,
- Verfahren vermutlich zuerst von Mac Queen 1963 vorgeschlagen (und später viele Male wiederentdeckt).
- Intitalisierung: Zufällige Partition von k Teilmengen aus der Grundmenge.
- Im Verlauf der Iteration können leere Cluster entstehen!

K-Means minimiert Varianzkriterium

Satz: Es sei $\mathcal{C}(t)$, $t = 0, 1, \ldots$ eine Folge von Partition die durch den K-means-Algorithmus entsteht. Dann ist $D_{\text{var}}(\mathcal{C}(t))$ monoton fallend.

$$
\begin{aligned}
D_{\text{var}}(\mathcal{C}(t)) &= \sum_{j=1}^{k} \sum_{x^\mu \in C_j(t)} \|x_\mu - c_j(t)\|^2 \\
&\geq \sum_{j=1}^{k} \sum_{x^\mu \in C_j(t)} \min_i \|x^\mu - c_i(t)\|^2 \\
&= \sum_{j=1}^{k} \sum_{x^\mu \in C_j(t+1)} \|x^\mu - c_j(t)\|^2 \\
&\geq \sum_{j=1}^{k} \sum_{x^\mu \in C_j(t+1)} \|x^\mu - c_j(t+1)\|^2 = D_{\text{var}}(\mathcal{C}(t+1))
\end{aligned}
$$

Austauschverfahren

Input : n Datenpunkte $x_\mu \in \mathbb{R}^p$ repräsentiert in einer Datenmatrix X.

1. Wähle Clusteranzahl k mit $1 \leq k \leq n$, maximale Iterationszeit t_{max}, minimale Anzahl Datenpunkte pro Cluster n_{min}
2. Setze $t = 0$ und bestimme eine Anfangspartition von X mit $|C_j(t)| > n_{\text{min}}$.

$$\mathcal{C}(t) = \{C_1(t), \ldots, C_k(t)\}$$

3. Berechne die k Schwerpunkte $c_j(t)$ und Abweichungssummen

$$e(C_j(t)) := \sum_{x^\mu \in C_j(t)} \|x^\mu - c_j(t)\|^2$$

4. Wähle einen Index μ. Dann sei $x^\mu \in C_p(t)$.

5. Falls $C_p(t) = n_{\text{min}}$ dann Goto 10
6. Transportierte nun x_μ versuchsweise die Cluster $C_j(t)$ mit $j \neq p$.
7. Berechne Zielfunktionsänderung

$$f_{\mu,j} := D_{\text{Var}}(\mathcal{C}(t)) - D_{\text{Var}}(\mathcal{C}^j(t))$$

Hierbei ist $\mathcal{C}^j(t)$ die Austauschclusterung

$$\mathcal{C}^j(t) = \{C_1(t), \ldots, C_p(t) \setminus \{x_\mu\}, \ldots, C_j(t) \cup \{x_\mu\}, \ldots, C_k(t)\}$$

8. Bestimme die Clusterund $\mathcal{C}_q$ mit der Eigenschaft

$$f_{\mu,q} = \max_j f_{\mu,j}$$

(Eindeutigkeit von q durch zufällige Wahl)

9. Falls $f_{\mu,q} > 0$, also $D_{\text{Var}}(\mathcal{C}(t)) > D_{\text{Var}}(\mathcal{C}^j(t))$, dann ordne den Datenpunkt x^μ aus dem Cluster $C_p(t)$ dem Cluster $C_q(t)$ zu.
   Also:
   $$\mathcal{C}(t+1) = \mathcal{C}_q(t)$$
   und bestimme die Schwerpunkte $c_q(t+1)$ und $c_p(t+1)$..

10. Falls $t < t_{\text{max}}$ dann $t := t + 1$ und Goto 4.

Inkrementelle Updating Formeln

Im Austauschverfahren muss berechnet werden:

$$f_{\mu,j} = D_{\text{Var}}(\mathcal{C}(t)) - D_{\text{Var}}(\mathcal{C}^j(t))$$

Durch einen (versuchsweisen) Austausch eines Punktes sind offenbar nur die Cluster C_j und C_p betroffen. Deshalb beweisen wir nun

$$f_{\mu,j} = \frac{|C_p(t)|}{|C_p(t)| - 1}\|x^\mu - c_p(t)\|^2 - \frac{|C_j(t)|}{|C_j(t)| + 1}\|x^\mu - c_j(t)\|^2$$

Satz: Für $n \in \mathbb{N}$ sei nun $C_n = \{x_1, \ldots, x_n\} \subset \mathbb{R}^p$ und

- $c_n := \frac{1}{n} \sum_{i=1}^{n} x_i$ der Schwerpunkt von C_n
- $W_n := \sum_{i=1}^{n} (x_i - c_n)(x_i - c_n)^T$ die Streuungsmatrix von C_n.
- $e_n := \sum_{i=1}^{n} \|x_i - c_n\|^2 = \text{tr } W_n$ die Abweichungsquadratsumme von C_n.

Dann gelten die folgenden inkrementellen Auf-/Abdatierungsformeln

1. $c_{n+1} = c_n + \frac{1}{n+1}(x_{n+1} - c_n)$
2. $W_{n+1} = W_n + \frac{n}{n+1}(x_{n+1} - c_n)(x_{n+1} - c_n)^T$
3. $e_{n+1} = e_n + \frac{n}{n+1}\|x_{n+1} - c_n\|^2$
4. $c_{n-1} = c_n - \frac{1}{n-1}(x_n - c_n)$
5. $W_{n-1} = W_n - \frac{n}{n-1}(x_n - c_n)(x_n - c_n)^T$
6. $e_{n-1} = e_n - \frac{n}{n-1}\|x_n - c_n\|^2$

Beweis der Updating Fromeln

Beweisen die Aufdatierungsformeln:

1. für den Schwerpunkt:

$$
\begin{aligned}
c_{n+1} &= \frac{1}{n+1}\sum_{i=1}^{n+1} x_i \\
&= \frac{1}{n+1}\sum_{i=1}^{n} x_i + \frac{1}{n+1}x_{n+1} \\
&= \frac{n}{n+1}c_n + \frac{1}{n+1}x_{n+1} \\
&= c_n + \frac{1}{n+1}(x_{n+1} - c_n)
\end{aligned}
$$

2. für die Streuungsmatrix:

$$
\begin{aligned}
W_{n+1} &= \sum_{i=1}^{n+1}(x_i - c_{n+1})(x_i - c_{n+1})^T \\
&= \sum_{i=1}^{n+1}(x_i - c_n - \frac{1}{n+1}(x_{n+1} - c_n))(x_i - c_n - \frac{1}{n+1}(x_{n+1} - c_n))^T \\
&= \sum_{i=1}^{n+1}(x_i - c_n)(x_i - c_n)^T - \frac{1}{n+1}\sum_{i=1}^{n+1}(x_i - c_n)(x_{n+1} - c_n)^T \\
&\quad -\frac{1}{n+1}\sum_{i=1}^{n+1}(x_{n+1} - c_n)(x_i - c_n)^T \\
&\quad +\frac{1}{(n+1)^2}\sum_{i=1}^{n+1}(x_{n+1} - c_n)(x_{n+1} - c_n)^T \\
&= W_n + (1 - \frac{2}{n+1} + \frac{1}{n+1})(x_{n+1} - c_n)(x_{n+1} - c_n)^T \\
&= W_n + \frac{n}{n+1}(x_{n+1} - c_n)(x_{n+1} - c_n)^T
\end{aligned}
$$

3. für die Abweichungsquadratsummen:

$$
\begin{aligned}
e_{n+1} &= \text{tr } W_{n+1} \\
&= \text{tr } W_n + \text{tr } \left(\frac{n}{n+1}(x_{n+1}-c_n)(x_{n+1}-c_n)^T\right) \\
&= e_n + \frac{n}{n+1}\text{tr } \left((x_{n+1}-c_n)(x_{n+1}-c_n)^T\right) \\
&= e_n + \frac{n}{n+1}\|x_{n+1}-c_n\|^2
\end{aligned}
$$

Die Abdatierungsformeln folgen analog.

Minimaldistanzclusterung

Definition: Eine Clusterung $\{C_1, \ldots, C_k\}$ heisst eine Minimaldistanzclusterung bzgl. der Norm $\|\cdot\|$, gdw. für alle datenpunkte x^μ gilt

$$x_\mu \in C_j \quad \Longleftrightarrow \quad \|x_\mu - c_j\| = \min_i \|x_\mu - c_i\|$$

Für das Austauschverfahren gilt dann der folgende

Satz: Ohne Begrenzung der Iterationszeit t_{max} liefert das Austauschverfahren eine Minimaldistanzpartition Resultat.

Angenommen, es lässt sich der Datenpunkt $x_\mu \in C_p$ in kein anderes Cluster transportieren, d.h. es gilt

$$f_{\mu,q} \leq 0 \quad \text{für alle } \mu$$

Dann muss gelten

$$\frac{|C_p|}{|C_p|-1}\|x^\mu - c_p\|^2 \leq \frac{|C_j|}{|C_j|+1}\|x^\mu - c_j\|^2 \quad \text{für alle } j \neq p.$$

Offenbar ist

$$1 < \frac{|C_p|}{|C_p|-1} \quad \text{und} \quad \frac{|C_j|}{|C_j+1|} < 1$$

folgt dann

$$\|x^\mu - c_p\| \leq \frac{|C_p|}{|C_p|-1}\|x^\mu - c_p\|^2 \leq \frac{|C_j|}{|C_j|+1}\|x^\mu - c_j\|^2 \leq \|x^\mu - c_j\|.$$

Also ist c_p das nächste Clusterzentrum zum Datenpunkt x_μ. Da x_μ beliebig ausgewählt war, gilt die Behauptung für alle Datenpunkte. Das Austauschverfahren liefert (bei unbeschränkter Laufzeit) eine Minimaldistanzpartition!

Invarianzeigenschaften

- Aus der linearen Algebra ist bekannt, dass die Euklidische Distanz $\|x-y\|_2$ zweier Vektoren $x, y \in \mathbb{R}^p$ gegenüber Translation und Rotation invariant ist.

- Translations-/Rotationsabbildungen T sind definiert durch Matrix $A \in \mathbb{R}^{p^2}$ und Vektor $b \in \mathbb{R}^p$ und haben die Form: $Tx = Ax + b$.

- Es gilt nun

$$\|x - y\| = \|Tx - Ty\| = \|(Ax + b) - (Ay + b)\| = \|A(x - y)\|$$

  gdw. also $\|Az\| = \|z\|$ für alle $z \in \mathbb{R}^p$ gilt, also A eine orthogonale Matrix ist, d.h. wenn $A^T A = I$. Insbesondere gilt dann $A^{-1} = A^T$.

- Offenbar ist das Varianzkriterium und die resultierenden Clusterungen invariant gegenüber Transformationen $Tx = Ax + b$ der Datenpunkte wenn A eine orthogonale Matrix ist.

Skalentransformationen

- Eine **Skalentransformation** ist gegeben durch eine Transformationen $Sx = Hx + b$ mit $b \in \mathbb{R}^p$ und einer nichtsingulären Diagonalmatrix H, dass heisst mit

$$H = \text{diag}(h_1, \ldots, h_p) \quad \text{mit } h_j \neq 0 \; j = 1, \ldots, p$$

- Eine orthogonale Skalentransformation liegt vor wenn

$$I = H^T H = \text{diag}(h_1^2, \ldots, h_p^2)$$

  D.h. falls $h_j = \pm 1$ für alle $j = 1, \ldots, p$.

- Das Varianzkriterium ist gegenüber Skalentransformationen $Sx = Hx + b$ invariant, wenn $h_j = \pm 1$ für alle $j = 1, \ldots, p$.

- Beim Varianzkriterium hängt als das Ergebnis einer Clusteranalyse von den gewählten Maßeinheiten ab (keine schöne Eigenschaft!)

- Ausweg in der Praxis: Standardskalierung der Daten für die einzelnen Merkmale durch

$$x_i^\mu \to \frac{x_i^\mu - \bar{x}_i}{s_i} \quad \text{für alle } \mu = 1, \ldots, n \quad \text{und} \quad i = 1, \ldots, p$$

mit

$$\bar{x} = \frac{1}{n} \sum_{\mu=1}^{n} x^\mu \quad \text{und} \quad s_i = \frac{1}{n-1} \sum_{\mu=1}^{n} (x_i^\mu - \bar{x}_i)^2$$

- Die transformierten Merkmale haben Mittelwert 0 und Varianz 1.

3.5 Fuzzy Clusteranalyse

- Einleitung : *Was ist Fuzzy Clustering?*
- Varianzkriterium mit *fuzzy membership*
- Optimierung des verallgemeinerten Varianzkriteriums
- Fuzzy-k-means Clusteranalyse

Hard-Clustering

Gegeben n Datenpunkte in einer Menge $X = \{x^1, \ldots, x^n\} \subset \mathbb{R}^p$ oder aufgefasst als $n \times p$ Matrix.

Ferner gegeben sei $k \in \mathbb{N}$, die Anzahl der Cluster.

Gesucht ist eine Partition bzw. Clusterung $\mathcal{C} = \{C_1, \ldots, C_k\}$ der Menge X, so dass

- $C_i \neq \emptyset$ für alle $i = 1, \ldots, k$.
- $C_i \cap C_j = \emptyset$ für alle $i \neq j$.
- $X = C_1 \cup \cdots \cup C_k$.

$\Rightarrow$ Jeder Datenpunkt x^μ liegt in genau einem Cluster C_j

Fuzzy-Clustering

Datenmenge $X = \{x^1, \ldots, x^n\} \subset \mathbb{R}^p$ und $k \in \mathbb{N}$ (Clusterzahl) gegeben.

Forderung, das jeder Datenpunkt in genau einem Cluster liegt wird zu:

Jeder Datenpunkt x^μ gehört zu einem Grad $f_{\mu,j} \in [0,1]$ zum Cluster C_j.

Der Wert $f_{\mu,j}$ heisst auch **fuzzy membership** bzw. Zugehörigkeit des Datenpunktes x^μ zum Cluster C_j.

Für jeden Datenpunkte x^μ, $\mu = 1, \ldots, n$ gilt: $\sum_{j=1}^{k} f_{\mu,j} = 1$.

Fuzzy membership als (Zugehörigkeits-)Wahrscheinlichkeit interpretierbar.

Hard-Clustering ist Spezialfall von Fuzzy-Clustering, falls nämlich

$$f_\mu = e_l \quad \text{für einen Einheitsvektor ist.}$$

Varianzkriterium mit Fuzzy-Membership

Fehlerfunktion als Verallgemeinerung des Varianzkriteriums:

$$D_(\mathcal{C}) := \sum_{\mu=1}^{n} \sum_{j=1}^{k} f_{\mu,j}^{b} \|x^{\mu} - c_j\|^2 \rightarrow \min$$

- $\|\cdot\|$ sei die Euklidische Norm im $\mathbb{R}^p$.
- c_j Repräsentanten der Clusters C_j für $j = 1, \ldots, k$.
- $F = (f_{\mu,j})_{\substack{\mu=1,\ldots,n \\ j=1,\ldots,k}}$ die Fuzzy-Membership-Matrix, $f_{\mu j} \in [0, 1]$ **fuzzy membership** für Datenpunkt x^{μ} zum Cluster C_j.
- $b > 1$ ein Gewichtungsexponent (*fuzzifier*). Häufig ist $b = 2$.

Optimierung von $D_F(\mathcal{C})$

Optimierung des verallgemeinerten Varianzkriteriums

$$D(\mathcal{C})_F := \sum_{\mu=1}^{n} \sum_{j=1}^{k} f_{\mu,j}^b \|x^\mu - c_j\|^2 \to \min$$

ist offenbar nur unter zusätzlichen Nebenbedingungen an F sinnvoll.

Optimierung unter probabilistischen (Fuzzy) Nebenbedingungen:

- $$f_{\mu,j} \in [0,1] \quad \text{für alle } \mu = 1,\ldots,n \text{ und alle } j = 1,\ldots,k$$
- $$\sum_{j=1}^{k} f_{\mu,j} = 1 \quad \text{für alle } \mu = 1,\ldots,n.$$

Satz: Bezüglich einer festen Zugehörigkeitsmatrix F ist $D_F(\mathcal{C})$ minimal für

$$c_j = \frac{1}{\sum_{\mu=1}^{n} f_{\mu,j}^b} \sum_{\mu=1}^{n} f_{\mu,j}^b x^\mu \quad \text{für alle } j = 1, \ldots, k$$

Beweis: Nullsetzen der partiellen Ableitungen $\frac{\partial}{\partial c_{rs}} D(\mathcal{C})_F$ liefert

$$0 = \frac{\partial}{\partial c_{rs}} \sum_{\mu=1}^{n} \sum_{j=1}^{k} f_{\mu,j}^b \sum_{i=1}^{p} (x_i^\mu - c_{ji})^2 = (-2) \sum_{\mu=1}^{n} f_{\mu,r}^b (x_S^\mu - c_{rs})$$

Damit folgt

$$\sum_{\mu=1}^{n} f_{\mu,r}^b x_s^\mu = \sum_{\mu=1}^{n} f_{\mu,r}^b c_{rs}$$

und schließlich

$$c_{rs} = \frac{1}{\sum_{\mu=1}^{n} f_{\mu,r}^{b}} \sum_{\mu=1}^{n} f_{\mu,r}^{b} x_s^{\mu}$$

die Behauptung des Satzes.

Mit etwas mehr Aufwand (Optimierung unter Nebenbedingungen nach der Lagrange-Methode) lässt sich nun die folgende Aussage beweisen.

Satz: Die Cluster-Bewertungsfunktion $D_F(\mathcal{C})$ ist minimal falls für die Cluster-Memberships $f_{\mu,j}$ gilt:

$$f_{\mu,j} = \frac{1}{\sum_{i=1}^{k} \left(\frac{\|x^{\mu} - c_j\|^2}{\|x^{\mu} - c_i\|^2} \right)^{\frac{1}{b-1}}} \quad \text{für alle } \mu = 1, \ldots, n \text{ und alle } j = 1, \ldots, k$$

Beweis: Hierzu minimiert man für jeden Datenpunkt x die Teilsumme:

$$s_x := \sum_{j=1}^{k} f_j^b \|x - c_j\|^2$$

unter der Bedingung $\sum_{j=1}^{k} f_j = 1$.

Mit Lagrange Multiplikator λ führt dies auf folgendes Funktional $L : \mathbb{R}^{k+1} \to \mathbb{R}$

$$L(f, \lambda) := \sum_{j=1}^{k} f_j^b \|x - c_j\|^2 - \lambda \Big(\sum_{j=1}^{k} f_j - 1 \Big) \to \min$$

Berechnung der partiellen Ableitungen:

$$\frac{\partial}{\partial \lambda} L(f, \lambda) = \sum_{j=1}^{k} f_j - 1$$

$$\frac{\partial}{\partial f_j} L(f, \lambda) = b \cdot f_j^{b-1} \cdot \|x - c_j\|^2 - \lambda$$

Letzte Gleichung auf Nullsetzen liefert dann

$$f_j = \Big(\frac{\lambda}{b \cdot \|x - c_j\|^2}\Big)^{\frac{1}{b-1}} = (\frac{\lambda}{b})^{\frac{1}{b-1}} (\frac{1}{\|x - c_j\|^2})^{\frac{1}{b-1}}$$

Einsetzen in die Gleichung $\frac{\partial}{\partial \lambda} L(f, \lambda) = 0$ so folgt:

$$1 = \sum_{j=1}^{k} f_j = \sum_{j=1}^{k} \Big(\frac{\lambda}{b \cdot \|x - c_j\|^2}\Big)^{\frac{1}{b-1}} = \Big(\frac{\lambda}{b}\Big)^{\frac{1}{b-1}} \sum_{j=1}^{k} \Big(\frac{1}{\|x - c_j\|^2}\Big)^{\frac{1}{b-1}}$$

Also folgt:

$$\Big(\frac{\lambda}{b}\Big)^{\frac{1}{b-1}} = \frac{1}{\sum_{j=1}^{k} \Big(\frac{1}{\|x - c_j\|^2}\Big)^{\frac{1}{b-1}}}$$

Nun alles zusammenführen liefert die behauptete Eigenschaft für f_j :

$$f_j = \frac{1}{\sum_{i=1}^{k} \left(\frac{1}{\|x - c_i\|^2} \right)^{\frac{1}{b-1}}} \cdot \left(\frac{1}{\|x - c_j\|^2} \right)^{\frac{1}{b-1}}$$

Damit liegen die Bedingungen der freien Parameter $f_{\mu,j}$ und c_j fest nämlich:

1.

$$c_j = \frac{1}{\sum_{\mu=1}^{n} f_{\mu,j}^b} \cdot \sum_{\mu=1}^{n} f_{\mu,j}^b x^\mu$$

2.

$$f_{\mu,j} = \frac{1}{\sum_{i=1}^{k} \left(\frac{\|x^\mu - c_j\|^2}{\|x^\mu - c_i\|^2} \right)^{\frac{1}{b-1}}}$$

Falls $f_{\mu,j} \in \{0, 1\}$, so ist c_j der geläufige Schwerpunkt der Daten aus C_j.

Fuzzy-k-means Algorithmus

Input: $X = \{x^1, \ldots, x^n\} \subset \mathbb{R}^p$, als Datenmatrix.

1. Wähle Clusterzahl $k \in \mathbb{N}$, den Fuzzifier $b > 1$ und die Toleranz $\epsilon > 0$.

2. Initialisiere die fuzzy membership matrix F gemäß Nebenbedingung.

3. **repeat**

$$c_j = \frac{1}{\sum_{\mu=1}^{n} f_{\mu,j}^b} \sum_{\mu=1}^{n} f_{\mu,j}^b x_\mu, \quad j = 1, \ldots, k$$

$$f_{\mu,j} = \frac{1}{\sum_{i=1}^{k} \left(\frac{\|x^\mu - c_j\|^2}{\|x^\mu - c_i\|^2} \right)^{\frac{1}{b-1}}}, \quad \mu = 1, \ldots, n \quad j = 1, \ldots, k$$

4. **until** $\|\Delta F\| < \epsilon$

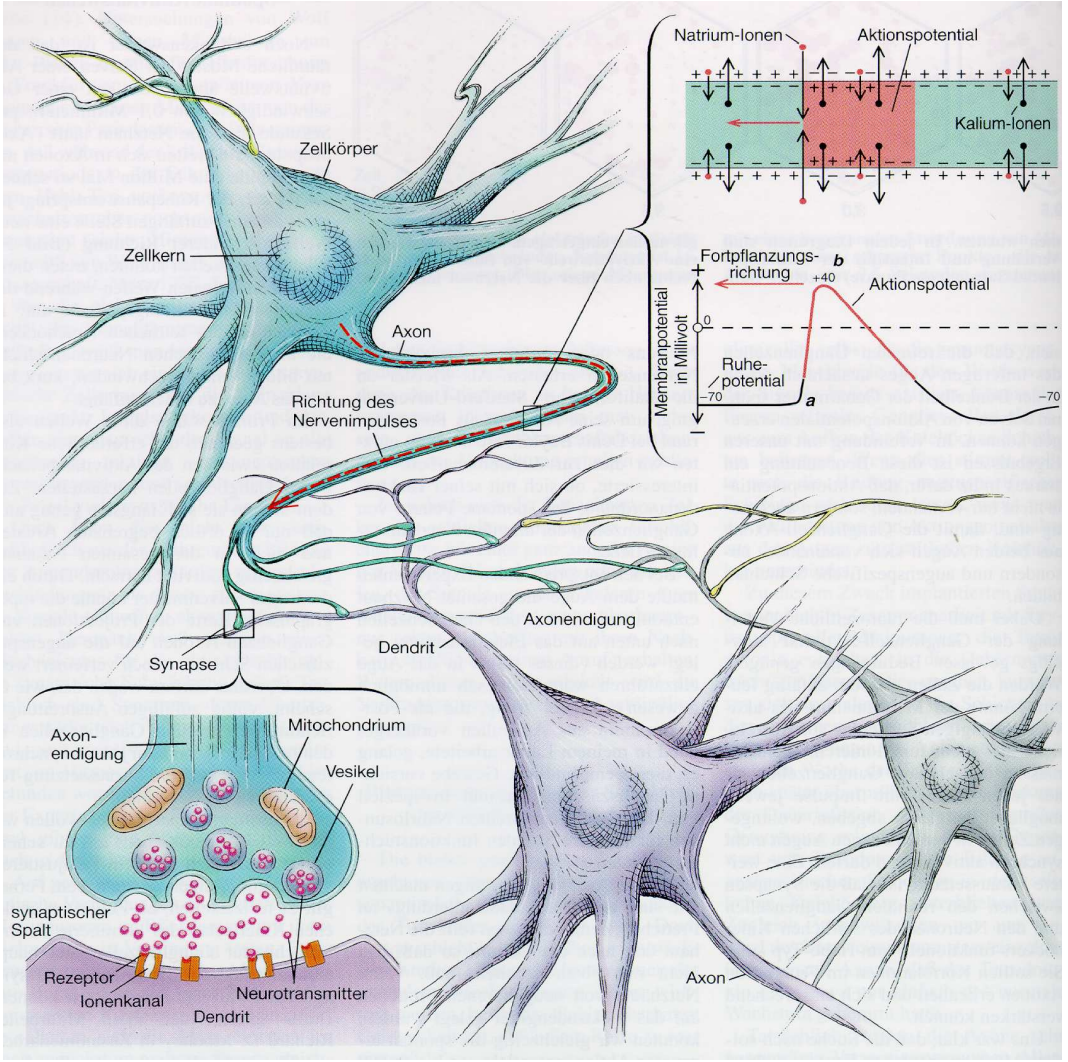
3.6 Neuronale Clusteranalyse

- Einleitung : *Was sind neuronale Netze?*, Neuronenmodelle
- Kompetitive Netze
- Inkrementelles K-means Clusteranalyseverfahren
- Kohonen’s selbstorganisierende Karten (SOM)
- Verwandte Verfahren
- Beispiele

Einleitung

- Künstliche neuronale Netz sind Modelle biologischer neuronaler Netze.
- Biologische neuronale Netze sind einzelnen **Neuronen** aufgebaut.
- Die Neuronen sind einfache Berechnungseinheiten.
- Bestandteile: Dendrit (Eingabe), Zellkörper (Verarbeitung), Axon (Ausgabe).
- Menschliches Gehirn: $10^{10} - 10^{11}$ Neuronen. Neuronen sind hochgradig unter einander verknüpft.
- Kontaktstellen zwischen zwei Neuronen sind die **Synapsen**. Jedes Neuron hat $10^3 - 10^5$ Synapsen. Synapsen sind gerichtete Verbindungen!

Biologisches neuronales Netz



Neuronale Netze

1. Neuronale Netze bestehen aus vielen Einzelbausteinen – den **Neuronen**, die untereinander über **Synapsen** verbunden sind.

2. Neuronen senden über ihr Axon sogenannte **Aktionspotentiale** oder **Spikes** aus.

3. Neuronen sammeln über ihren **Dendriten(baum)** die über die Synapsen eingehenden Signale (EPSPs und IPSPs) auf. Man spricht von einer **räumlich-zeitlichen Integration**.

4. Überschreitet die am Dendriten integrierte Aktivität einen **Schwellwert**, so erzeugt das Neuron ein Aktionspotential (Spike).

5. Bleibt die am Dendriten integrierte Aktivität diesen **Schwellwert**, so erzeugt das Neuron kein Aktionspotential (Spike).

Neuronenmodelle

- Grundmodell (einfaches nichtlineares Modell)

$$\begin{aligned} \tau \dot{u}_j(t) &= -u_j(t) + \underbrace{x_j(t) + \sum_{i=1}^{n} c_{ij} y_i(t - \Delta_{ij})}_{=:e_j(t)} \\ y_j(t) &= f(u_j(t)) \end{aligned}$$

- Grundmodell in diskreter Zeit

$$\begin{aligned} \frac{\tau}{\Delta t}(u_j(t + \Delta t) - u_j(t)) &= -u_j(t) + e_j(t) \\ u_j(t + \Delta t) &= (1 - \varrho) \cdot u_j(t) + \varrho \cdot e_j(t), \\ y_j(t) &= f(u_j(t)) \end{aligned}$$

$\varrho := \frac{\Delta t}{\tau}, \;\; 0 < \varrho \leq 1$

Transferfunktionen I

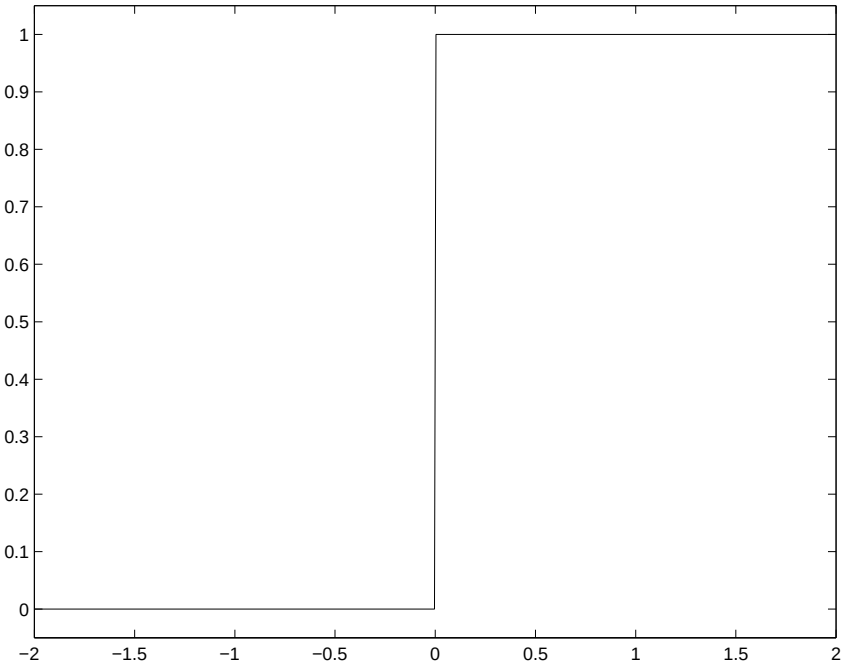
1. Die Funktion $f(u) := H(u - \theta)$ mit der *Heaviside-Funktion* H. Die Heaviside-Funktion H nimmt für $u \geq 0$ den Wert $H(u) = 1$ und für $u < 0$ den Wert $H(u) = 0$ an.

2. Die beschränkte stückweise lineare Funktionen:

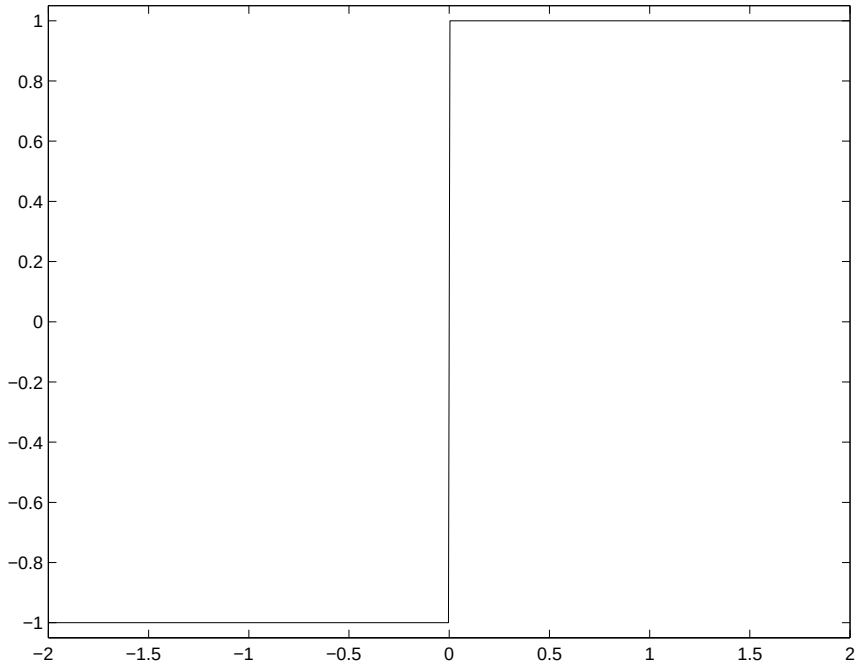
$$f(u) := \begin{cases} 0 & u < 0 \\ u & u \in [0, 1] \\ 1 & u > 1 \end{cases} \tag{2}$$

3. Die Funktion $f(u) := F_\beta(u)$ mit $F_\beta(u) = 1/(1 + \exp(-\beta u))$, wobei $\beta > 0$ ist. Die Funktion F ist die aus der statistischen Mechanik bekannte *Fermi-Funktion*.

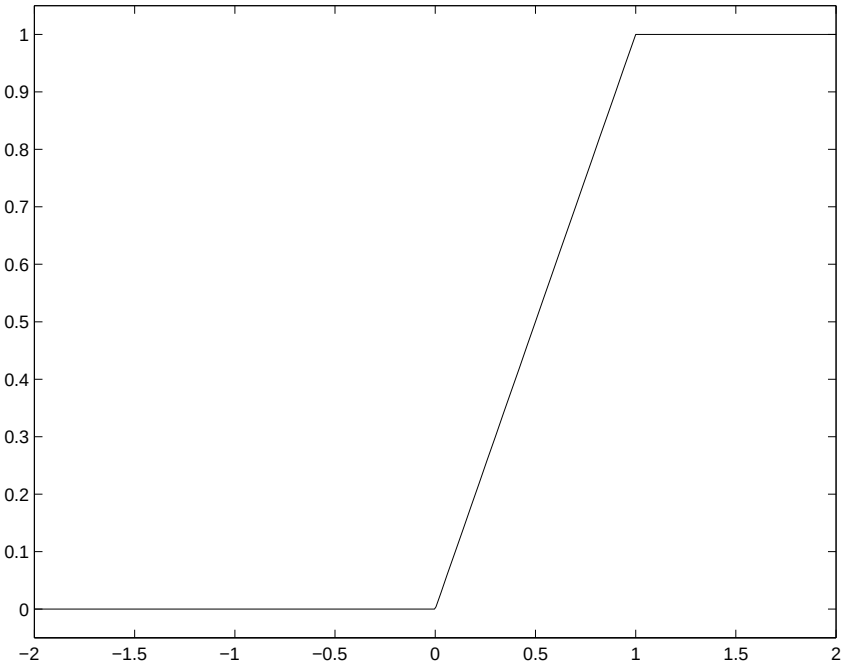
4. Die lineare Funktion $f(u) := u$. Ein Neuron mit dieser Transferfunktion heißt *lineares Neuron*.



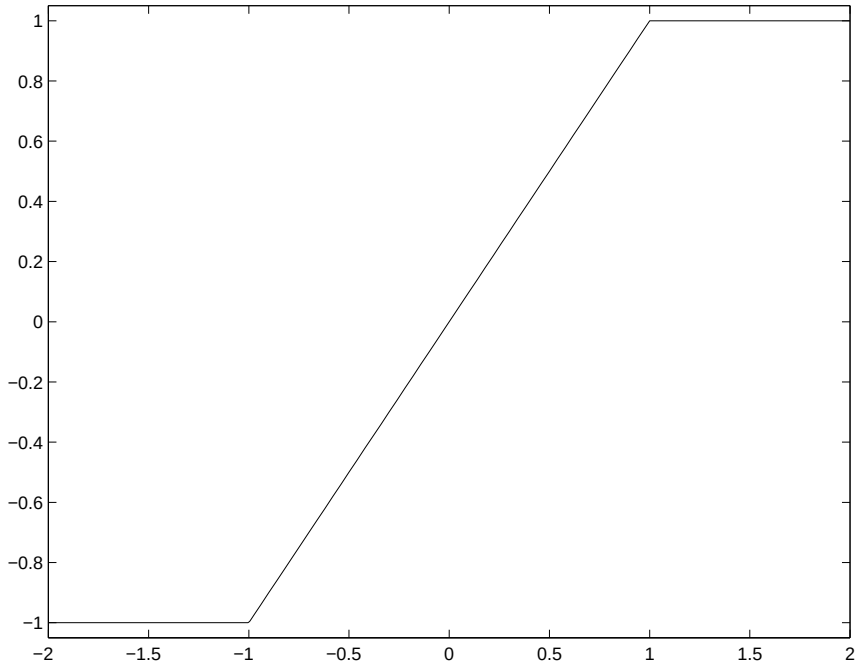
Heaviside-Funktion.



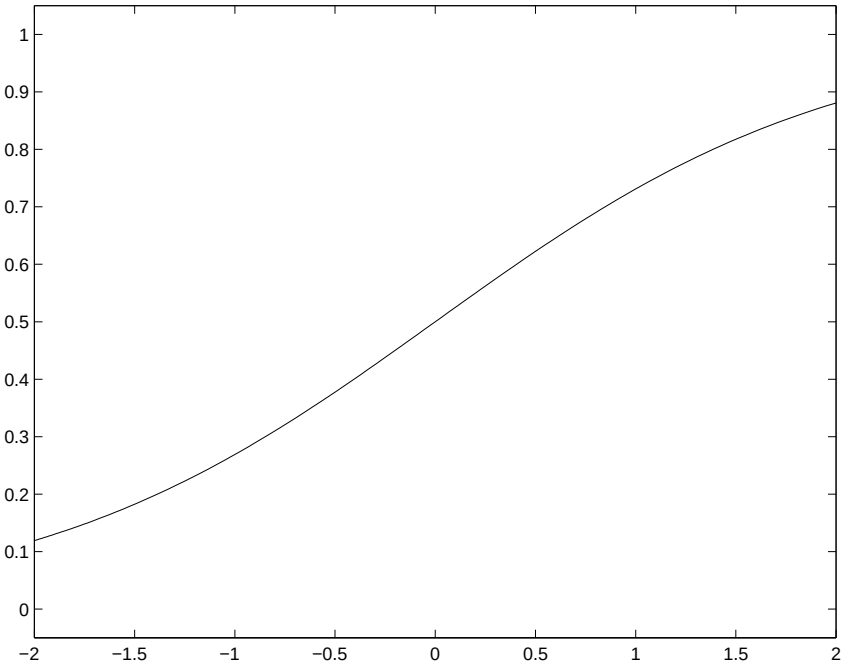
Signum-Funktion.



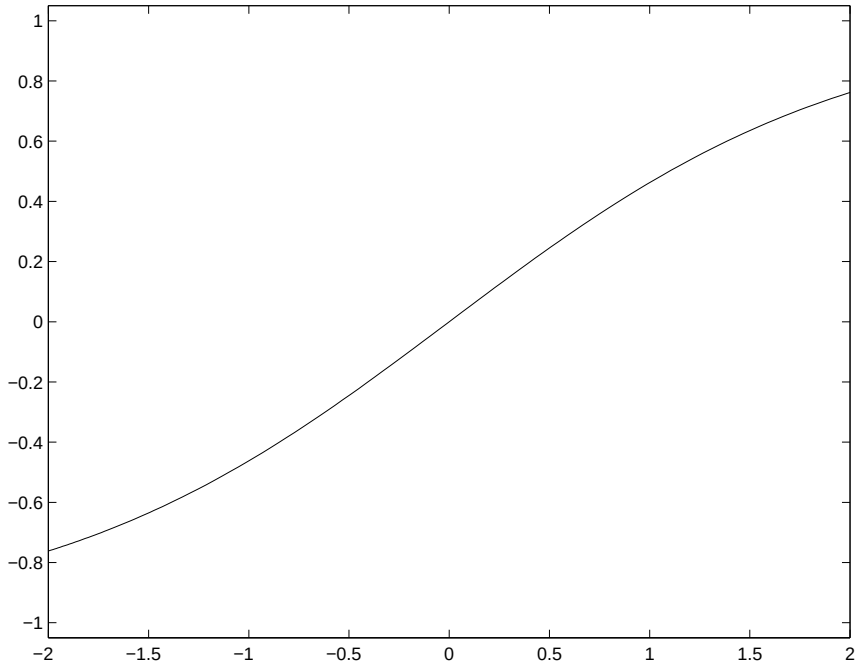
Begrenzte lineare Funktion.



Sym. begrenzte lineare Funktion.

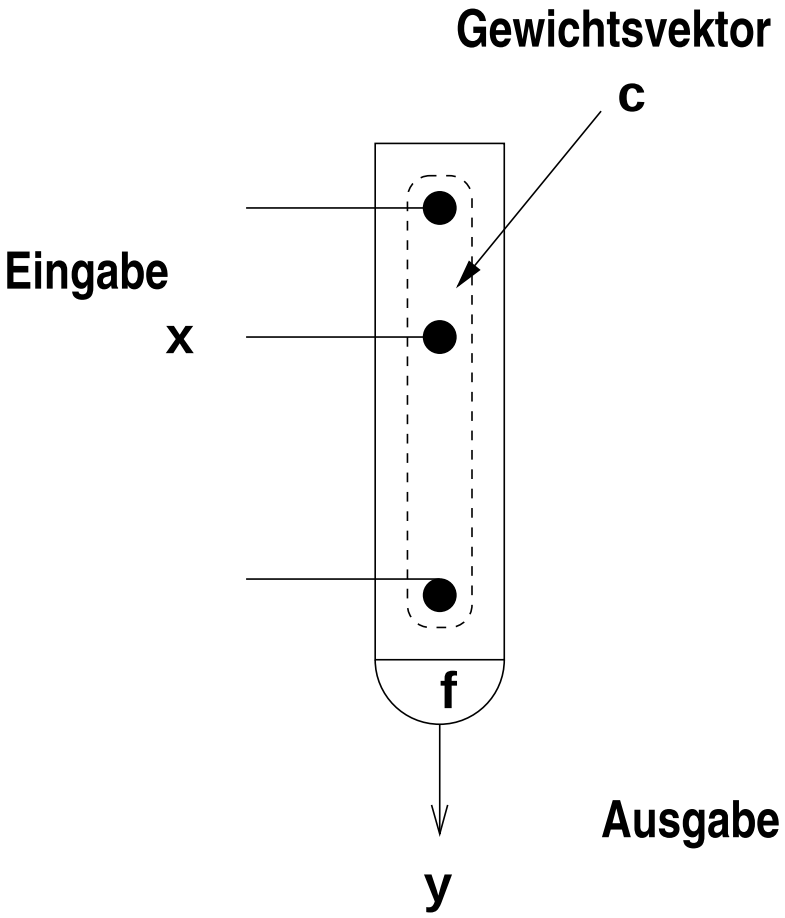


Fermi-Funktion, $\beta = 1$.



Hyperbolischer Tangens, $\gamma = 1$.

Neuronenmodell



Vereinfachtes Neuonenmodell

- Kein Gedächtnis, d.h. $\varrho = 1$.
- Laufzeit $\Delta_{ij} = 0$

Beispiele I

- **Lineares Neuron**

$$y = f(\langle x, c \rangle + \theta)$$

- **Schwellwertneuron**

$$y = f(x) = \begin{cases} 1 & \langle x, c \rangle \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

- **Kontinuierliches nichtlineares Neuron**

$$y = f(\langle x, c \rangle + \theta), \quad f(s) = \frac{1}{1 + \exp(-s)}$$

Beispiele II

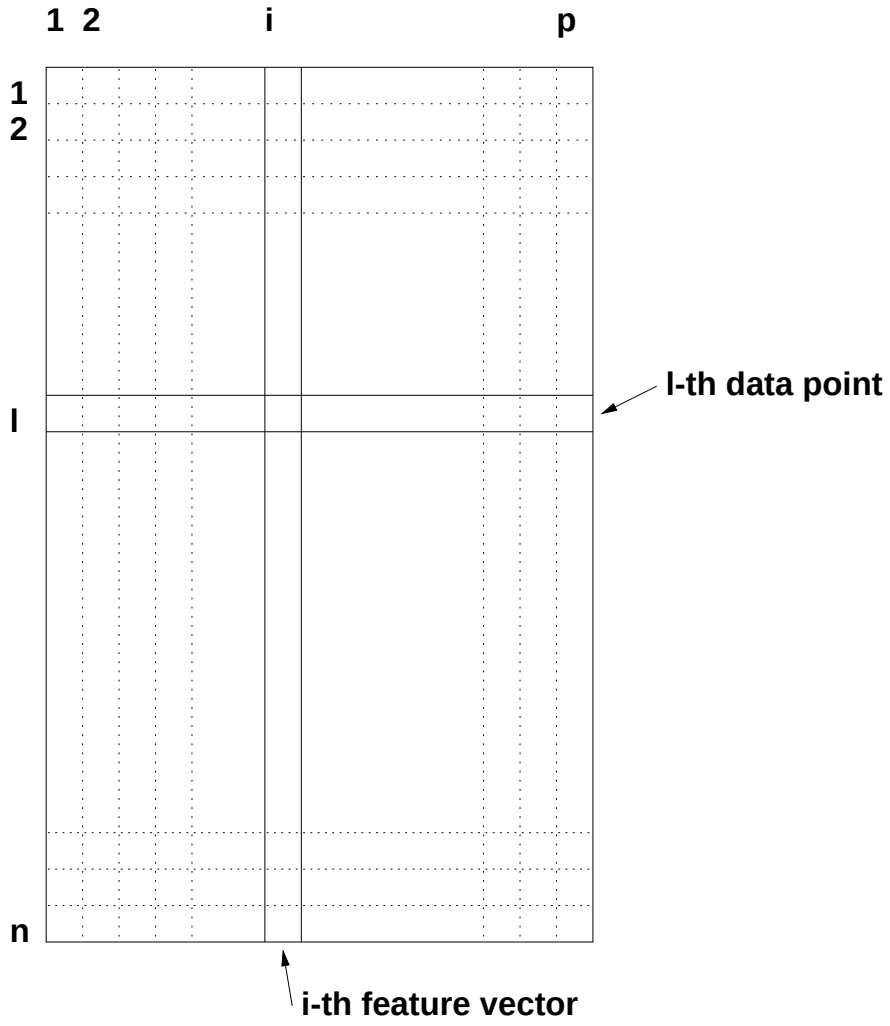
- **Distanzberechnendes Neuron**

$$y = f(\|x - c\|)$$

- **Radial symmetrisches Neuron**

$$y = f(\|x - c\|), \quad f(r) = \exp(-\frac{r^2}{2\sigma^2})$$

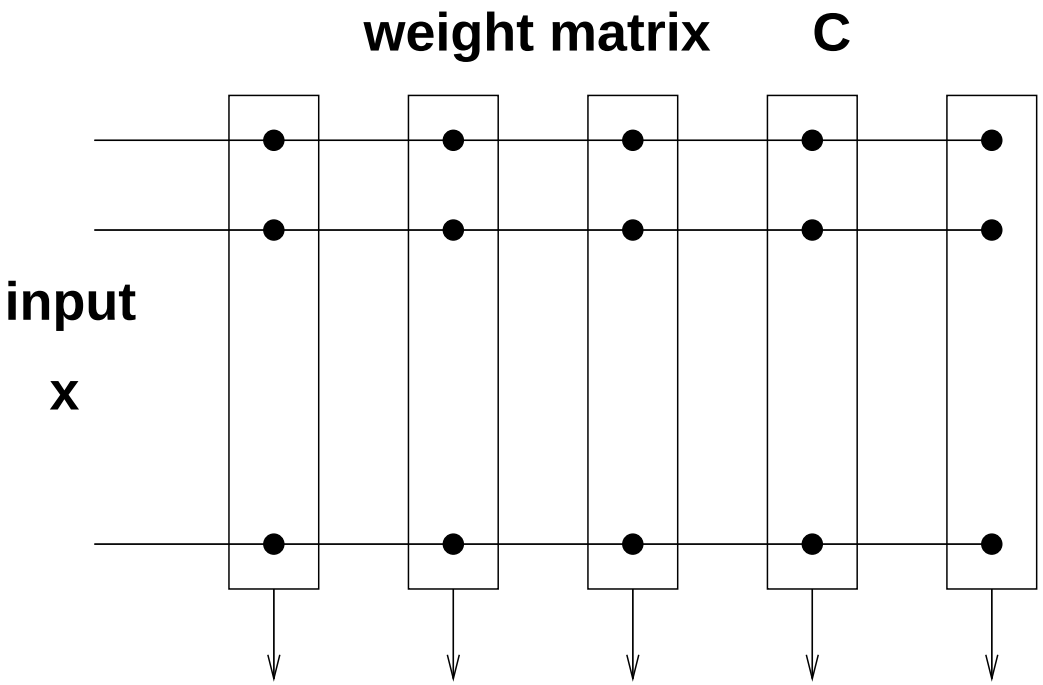
Datenanalyse Problem

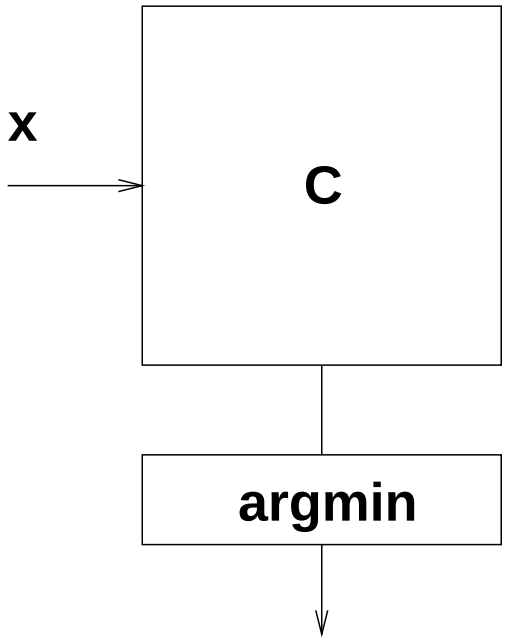


Probleme

- Viele Datenpunkte
- Hohe Dimension des Merkmalsraumes
- Datenpunkte zu Beginn der Analyse möglicherweise nicht vollständig bekannt.

Neuronales Netz mit Wettbewerb





Distanz und Skalarprodukt

Die Euklidische Norm ist bekanntermaßen durch das Skalarprodukt im $\mathbb{R}^p$ definiert:

$$\|x\|_2 = \sqrt{\langle x, x \rangle}$$

Für den Abstand zweier Punkte $x, y \in \mathbb{R}^p$ gilt demnach:

$$\|x - y\|_2^2 = \langle x - y, x - y \rangle = \langle x, x \rangle - 2\langle x, y \rangle + \langle y, y \rangle = \|x\|_2^2 - 2\langle x, y \rangle + \|y\|_2^2$$

Die Gewinnersuche für Datum x unter den Prototypen $c_1, \ldots, c_k$ und $\|c_i\|_2 = 1$ mit $i = 1, \ldots, k$, ist für die beiden folgenden Verfahren äquivalent

- argmax$_i \langle x, c_i \rangle$
- argmin$_i \|x - c_i\|_2$

Kompetitives Lernen (Skalarprodukt)

Input: $X = \{x^1, \ldots, x^n\} \subset \mathbb{R}^p$

1. Wähle Clusterzahl $k \in \mathbb{N}$, eine Lernrate $l > 0$, N, $\epsilon > 0$

2. Initialisiere Prototypen $c_1, \ldots, c_k \in \mathbb{R}^p$ ($k \times p$ Matrix C) mit $\|c_i\| = 1$

3. **repeat**

   Wähle $x \in X$
   $j = \text{argmax}_i \langle x, c_i \rangle$ (winner detection)
   $c_j = c_j + lx$ (winner update)
   $c_j = c_j / \|c_j\|_2$ (normalization)

4. **until** $\|\Delta C\| < \epsilon$ über N Punktpräsentationen

Kompetitives Lernen (Euklidische Distanz)

Input: $X = \{x^1, \ldots, x^n\} \subset \mathbb{R}^p$

1. Wähle Clusterzahl $k \in \mathbb{N}$, eine Lernrate $l > 0$, N, $\epsilon > 0$

2. Initialisiere Prototypen $c_1, \ldots, c_k \in \mathbb{R}^p$ ($k \times p$ Matrix C) mit $\|c_i\| = 1$

3. **repeat**

   Wähle $x \in X$
   $j = \text{argmin}_i \|x - c_i\|$ (winner detection)
   $c_j = c_j + l(x - c_j)$ (winner update)

4. **until** $\|\Delta C\| < \epsilon$ über N Punktpräsentationen

Inkrementelles k-means Lernen

Datenpunkt $x \in \mathbb{R}^p$ wird dem nächsten Clusterzentrum c_{j^*} zugeordnet:

$$j = \text{argmin}_i \|x - c_i\|.$$

Anpassung des Clusterzentrums:

$$\Delta c_j = \frac{1}{|C_j| + 1}(x - c_j)$$

Zu Vergleich: **Kompetitves Lernen**

$$\Delta c_j = l_t(x - c_j)$$

$l_t > 0$ eine Folge von Lernraten mit $\sum_t l_t = \infty$ und $\sum_t l_t^2 < \infty$

Inkrementeller K-means Algorithmus

Input: $X = \{x^1, \ldots, x^n\} \subset \mathbb{R}^p$

1. Wähle Clusterzahl $k \in \mathbb{N}$, ferner N und $\epsilon > 0$

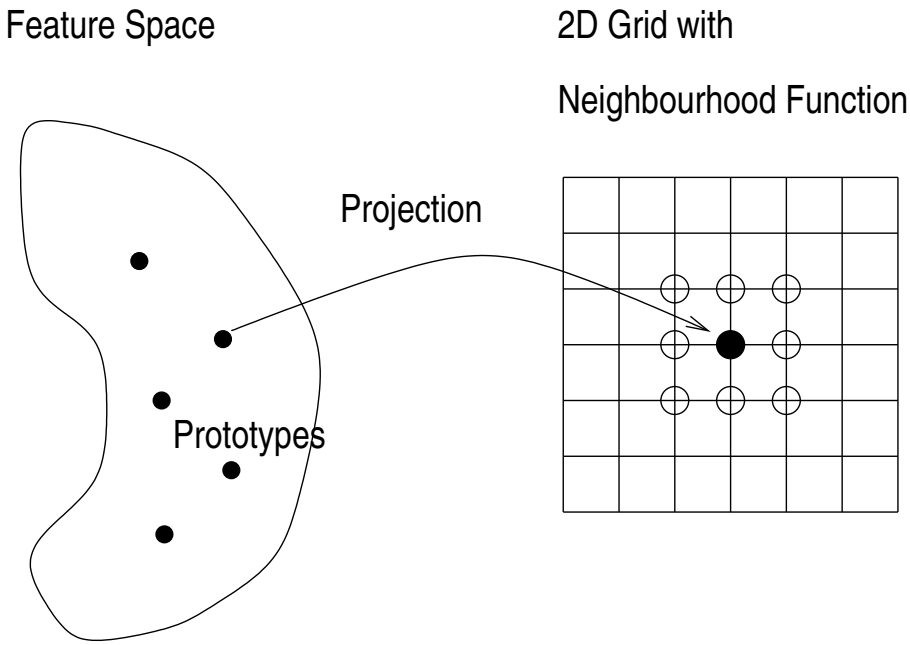
2. Initialisiere Prototypen $c_1, \ldots, c_k \in \mathbb{R}^p$ ($k \times p$ Matrix C) und $n_i = 0$.

3. **repeat**

   Wähle $x \in X$
   $j = \text{argmin}_i \|x - c_i\|$ (winner detection)
   $n_j = n_j + 1$
   $c_j = c_j + \frac{1}{n_j}(x - c_j)$ (winner update)

4. **until** $\|\Delta C\| < \epsilon$ über N Punktpräsentationen

Kohonen's Selbstorganisierende Karte



Kohonen Lernregel: $\Delta c_j = l_t \cdot \mathcal{N}(g_j, g_{j^*}) \cdot (x - c_j)$

Gewinner: j^* und Nachbarschaftsfunktion: $\mathcal{N}(j, j^*)$

Beispiel: $\mathcal{N}_{\sigma_t}(j, j^*) = exp(-\|p(j) - p(j^*)\|^2 / 2\sigma_t^2)$, hierbei ist $p(j)$ die Gitterposition des j-ten Neurons; $\sigma_t \to 0$ und $l_t \to 0$

Kohonen Lernalgorithmus

Input: $X = \{x^1, \ldots, x^n\} \subset \mathbb{R}^p$

1. Wähle $r, s \in \mathbb{N}$, eine Clusterzahl $k = rs \in \mathbb{N}$, eine Lernrate $l > 0$, eine Nachbarschaftsfunktion $\mathcal{N}$, ferner N und $\epsilon > 0$

2. Initialisiere Prototypen $c_1, \ldots, c_k \in \mathbb{R}^p$ ($k \times p$ Matrix C)

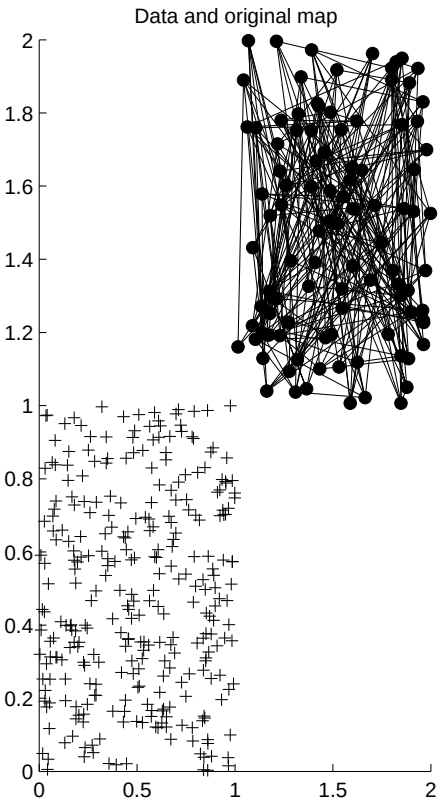
3. Jeder Prototypen c_i auf eine Gitterposition $g_i \in \{1, \ldots, r\} \times \{1, \ldots, s\}$.

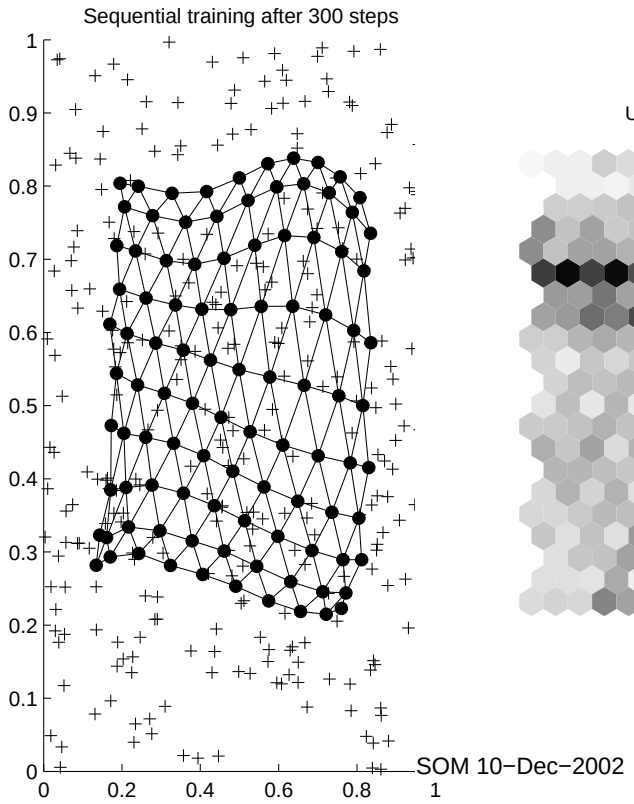
4. **repeat**

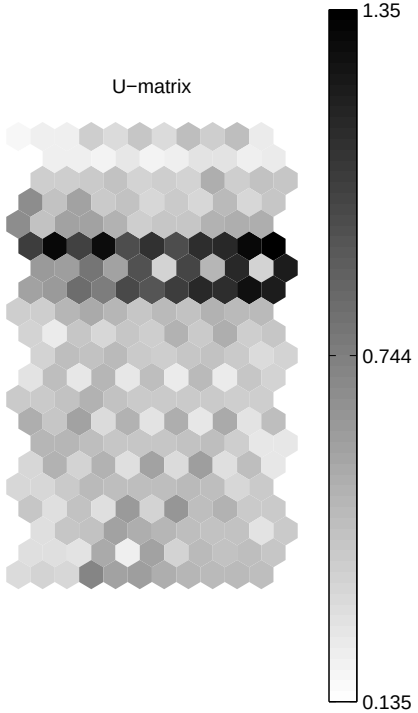
   Wähle $x \in X$
   $j^* = \text{argmin}_i \|x - c_i\|$ (winner detection)
   for $j = 1, \ldots, n$
   $c_j = c_j + l\mathcal{N}(j, j^*)(x - c_j)$ (update)

5. **until** $\|\Delta C\| < \epsilon$ über N Punktpräsentationen

SOM-Beispiele







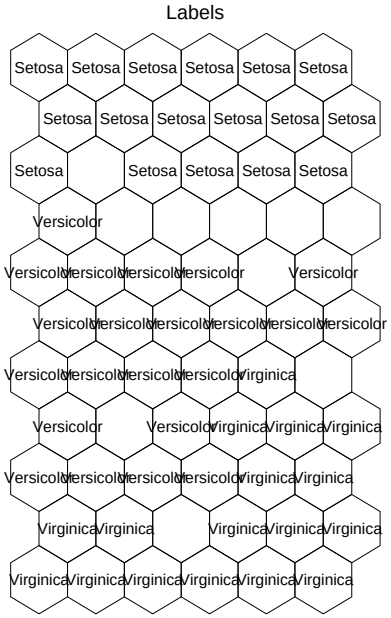


Table 3.4. Animal names and their attributes

		dove	hen	duck	goose	owl	hawk	eagle	fox	dog	wolf	cat	tiger	lion	horse	zebra	cow
is	small	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0
	medium	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	big	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
has	2 legs	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	4 legs	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hair	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hooves	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	mane	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0
	feathers	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
likes to	hunt	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
	run	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
	fly	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	swim	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

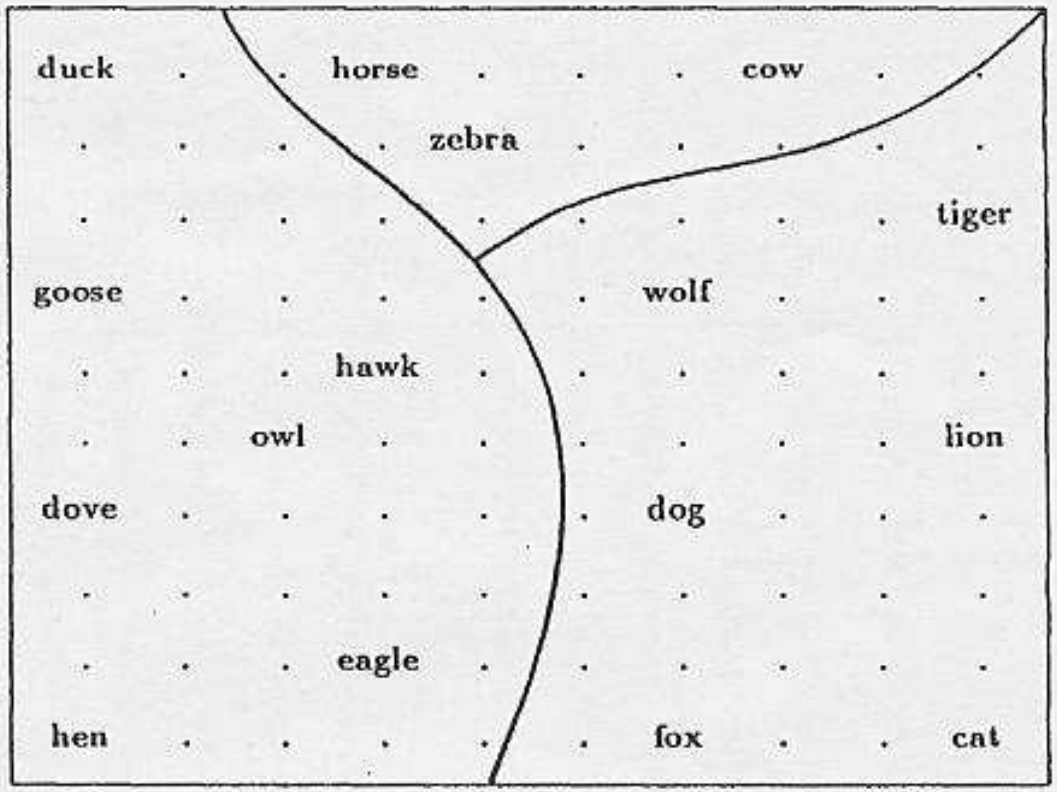
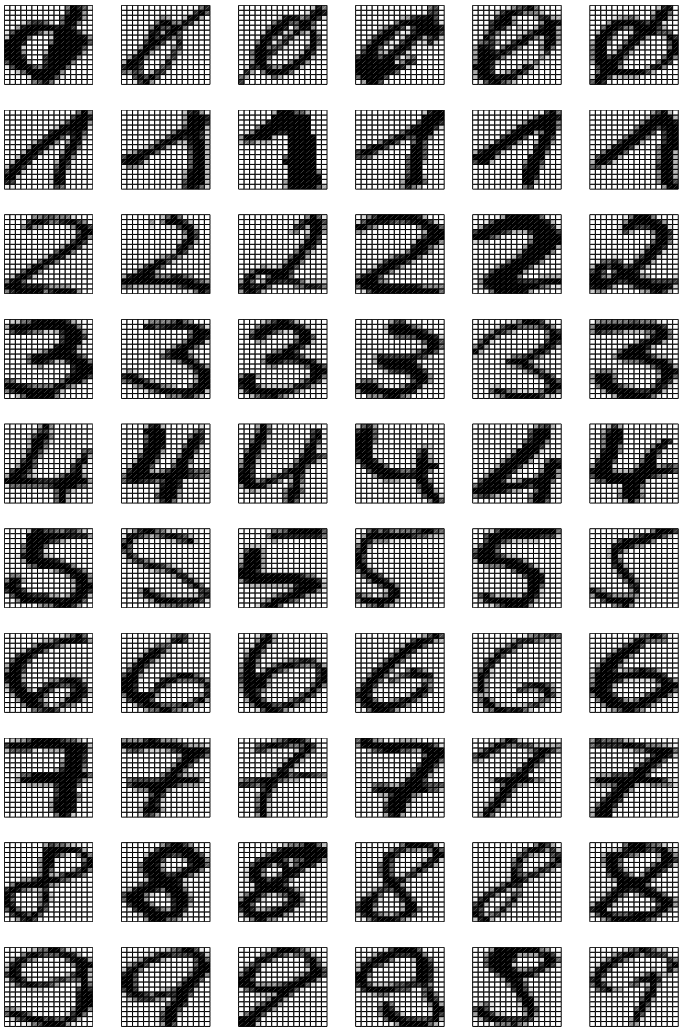
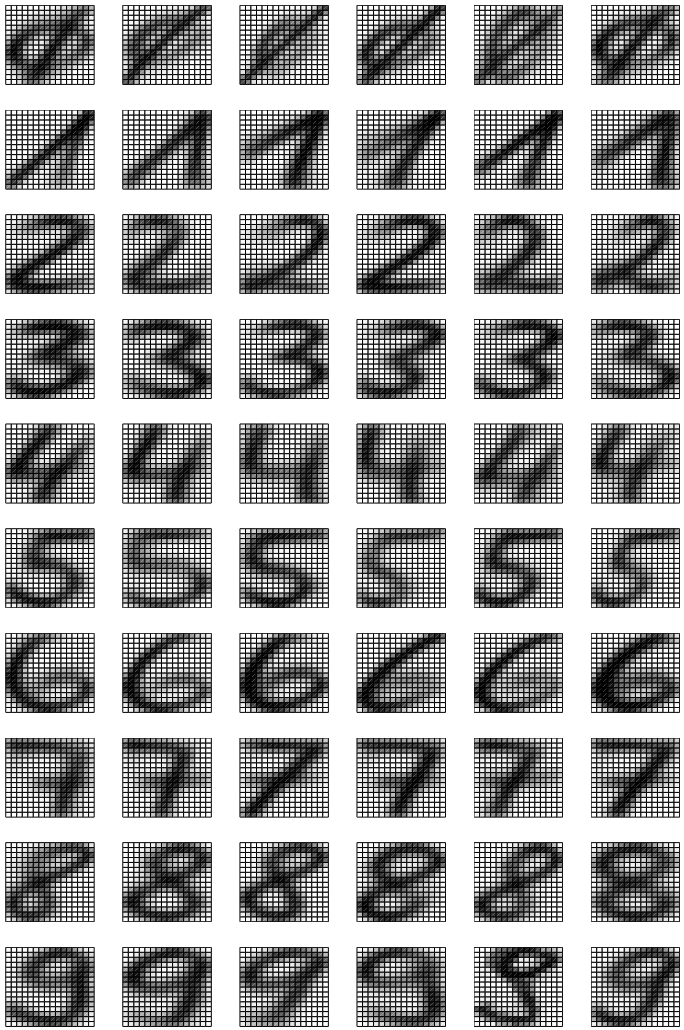


Fig. 3.22. After the network had been trained with inputs describing attribute sets from Table 3.4, the map was calibrated by the columns of Table 3.4 and labeled correspondingly. A grouping according to similarity has emerged

Daten und Clusterzentren





Verwandte Verfahren: ART-Netze

- Adaptive-Resonanz-Theorie (ART) entwickelt von Stephen Grossberg und Gail Carpenter
- Hier nur **ART1** Architektur
- Erweiterungen: ART 2, ART 3, FuzzyART, ARTMAP
- ART-Netze sind Netze mit Kompetition
- Speziell bei ART1: Binäre Inputdaten und Gewichtsvektoren (Prototypen)
- Besonderheit: ART-Netze sind wachsende Netze, d.h. die Anzahl der Neuronen ist während des Trainings nicht fest; allerdings nach oben beschränkt.

ART1 Lernen : Idee

- Inputvektoren und Prototypen binär.
- Es wird höchstens der Gewichtsvektor des Gewinnerneurons c_{j^*} adaptiert.
- Ist die Ähnlichkeit zwischen Inputvektor x und c_{j^*} zu gering, so definiert x einen neuen Prototypen und c_{j^*} bleibt unverändert.
- Die Ähnlichkeit wird durch das Skalarprodukt $x \cdot c_j = \langle x, c_j \rangle$ gemessen.
- Schranke für die Mindestähnlichkeit wird durch den sogenannten **Vigilanz-parameter** gemessen.
- Ist die Ähnlichkeit groß genug, so wird c_{j^*} durch komponentenweises **AND** von x und c_{j^*} adaptiert.
- Maximale Zahl von Neuronen wird vorgegeben. Aus dieser Grundidee lassen sich viele mögliche Algorithmen ableiten.

ART1 : Bezeichnungen

- $x_\mu \in \{0,1\}^p$ die Eingabevektoren $\mu = 1, \ldots, n$
- $c_i \in \{0,1\}^p$ die Gewichtsvektoren der Neuronen (Prototypen)
- $\mathbf{1} = (1,1,\ldots,1) \in \{0,1\}^p$ der Eins-Vektor mit p Einsen.
- k Anzahl der maximal möglichen Neuronen.
- $\|x\|_1 = \sum_{i=1}^{p} x_i$ die l_1-Norm ($=$ Anzahl der Einsen).
- $\varrho \in [0,1]$ der Vigilanzparameter.

ART1 : Algorithmus

1. Wähle $k \in \mathbb{N}$ und $\varrho \in [0, 1]$.
2. Setze $c_i = \mathbf{1}$ für alle $i = 1, \ldots, k$.
3. **WHILE** noch ein Muster x vorhanden **DO**

   Lies x und setze $I := \{1, \ldots, k\}$

   REPEAT

   $j^* = \text{argmax}_{j \in I} \langle x, c_j \rangle / \|c_j\|_1$ (winner detection)

   $I = I \setminus \{j^*\}$

   UNTIL $I = \emptyset \vee \langle x, c_{j^*} \rangle \geq \varrho \|x\|_1$

   IF $\langle x, c_{j^*} \rangle \geq \varrho \|x\|_1$

   THEN $c_{j^*} = x \wedge c_{j^*}$ (winner update)

   ELSE keine Bearbeitung von x
4. **END**

Verwandte Verfahren : LVQ

- Lernende Vektorquantisierung (LVQ) sind **überwachte Lernverfahren** zur Musterklassifikation.
- LVQ-Verfahren wurden von Teuvo Kohonen entwickelt.
- LVQ-Verfahren sind Kompetitive Lernmethoden.
- Euklidische Distanz zur Berechnung der Ähnlichkeit/Gewinnerermittlung.
- Es wird nur **LVQ1** vorgestellt.
- Erweiterungen: LVQ2 und LVQ3 (ggf. Adaptation des 2. Gewinners); OLVQ-Verfahren (neuronenspezifische Lernraten).
- Heuristisches Verfahren

LVQ1 : Algorithmus

Input: $X = \{(x^1, y^1), \ldots, (x^n, y^n)\} \subset \mathbb{R}^p \times \Omega$ hierbei ist
$\Omega = \{1, \ldots, L\}$ eine endliche Menge von L Klassen(-Labels).

1. Wähle Prototypenzahl $k \in \mathbb{N}$, eine Lernrate $l_t > 0$ und N. Setze $t = 0$.

2. Initialisiere Prototypen $c_1, \ldots, c_k \in \mathbb{R}^p$.

3. Bestimme für alle c_i eine Klasse $\omega_i \in \Omega$.

4. **repeat**

   Wähle Paar $(x, y) \in X$ und $t = t + 1$
   $j^* = \text{argmin}_i \|x - c_i\|$ (winner detection + class from nearest neighbor)
   if $\omega_{j^*} \neq y$ then $\Delta = -1$ else $\Delta = 1$ (correct classification result?)
   $c_{j^*} = c_{j^*} + l_t \Delta (x - c_{j^*})$ (winner update)

5. **until** $t \geq N$

3.7 Bewertung von Clusterungen

- Hintergrund
- Statistisches Testen
- Zufallshypothesen
- Definition eines statistischen Tests
- Γ-Index nach Hubert
- Goodman-Kruskal γ Statistik
- Monte-Carlo-Analyse
- Beispiel: Bewertung hierarchischer Clusterungen

Hintergrund

- Ziel ist die objektive quantitative Bewertung von Resultaten einer Clusteranalyse.
- Hierarchien, Clusterungen und Cluster werden meist durch Inspektion von Experten der Anwendungsdomäne, und meistens auch nur qualitativ, beurteilt.
- Das Problem der quantitativen Bewertung von Clusterergebnissen ist ein **statistisches** Problem, genauer ein Problem der **beurteilenden/schließenden Statistik**.
- Eine Clusterung ist dabei valide, wenn sie in irgendeinem Sinne eine unwahrscheinliche Anordnung der Daten ist.
- Statistische Verfahren zum Testen von Hypothesen müssen dazu entwickelt werden.
- Entwicklung von Bewertungsmaßen für Clusterungen ist noch relativ einfach; schwierig ist es Grenzen für diese Maße zu definieren die dann valide Clusterungen festlegen.

Statistisches Testen

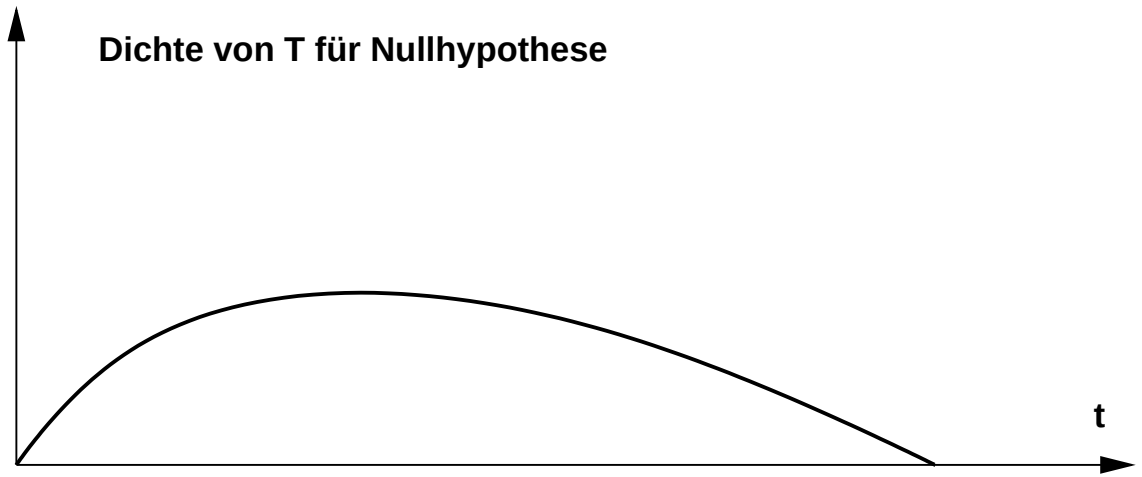
- Eine Statistik T ist eine Funktion der Daten aus der Information über die Güte der Clusterung gewonnen werden soll.
- Beispiele sind der Diskretisierungsfehler (Varianzkriterium), Tiefe der Hierarchie in einer Clusterung, ein Kompaktheitsmaß eines Clusters, usw.
- T ist also eine Zufallsvariable. Ihre Verteilung beschreibt die relative Häufigkeit mit der bestimmte Werte von T unter gewissen Hypothesen vorkommen.
- Eine Hypothese ist eine Aussage über die relativen Häufigkeiten von Ereignissen in der Grundmenge aller möglichen Ereignisse.
- Beispiel: *Daten sind zufällig* oder *Daten sind geclustert*
- Mit einer Hypothese wird die Beobachtung der Größe T getestet und an Hand der Verteilung von T entschieden, ob die Beobachtung, basierend auf der Verteilung von T, wahrscheinlich ist oder nicht.

Zufälligkeitshypothesen

- Eine sogenannte **Nullhypothese** H_0 im Bereich der Clustervalidierung ist von der Form: *Es ist keine Struktur in den Daten vorhanden!*
- **Zufallsgraph-Hypothese**
  H_0: *Alle $n \times n$ Ähnlichkeits-/Abstandsmatrizen haben die gleiche Wahrscheinlichkeit.*
- **Zufallslabel-Hypothese**
  H_0: *Alle Permutationen der Klassenlabel auf die n Objekte haben die gleiche Wahrscheinlichkeit.*
- **Zufallspositionen-Hypothese**
  H_0: *Alle Mengen mit n Positionen in einer bestimmten Region des $\mathbb{R}^d$ haben die gleiche Wahrscheinlichkeit.*

Idee des statistischen Testens

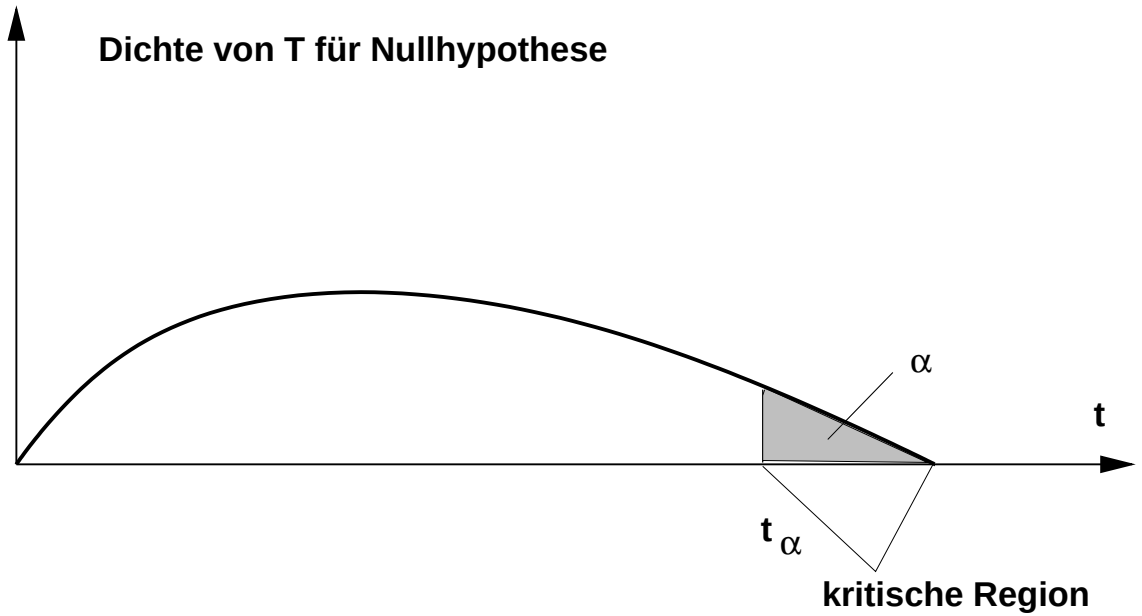
- Test T und Nullhypothese H_0 seien festgelegt. Weiterhin sei die Verteilung von T unter der Hypothese H_0 gegeben. (Die ist baer leider in vielen Fällen nur sehr schwer auszurechnen.)



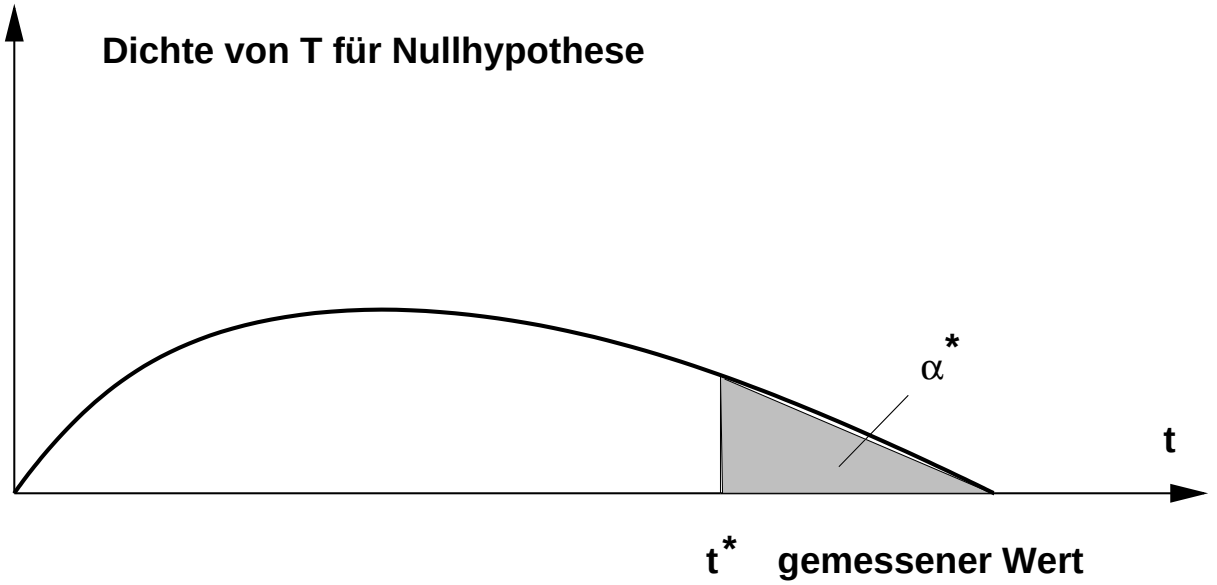
- Wie soll man nun testen, ob die Hypothese die vorliegenden Daten gut beschreibt?
- Es sei $P(B|H_0)$ die Wahrscheinlichkeit des Ergnisses B bei gegebener Hypothese H_0.

- B kann beispielsweise sein: $T \leq t$ oder $T \leq t$ für eine Schranke t.
- Es sei $\alpha > 0$ eine kleine Zahl, etwa $\alpha = 0.05$ oder 0.01.
  α heißt das Signifikanzniveau des Tests.
- Angenommen große Werte von T zeigen, dass H_0 abgelehnt werden sollte, dann können wir eine Grenze t_α für T festlegen durch lösen der Gleichung:

$$P(T \geq t_\alpha | H_0) = \alpha$$



- Es sei nun der Wert t^* für die Zufallsvariable T in dem Experiment gemessen worden.
- Falls $t^* \geq t_\alpha$ dann weist man H_0 auf dem Niveau α zurück.



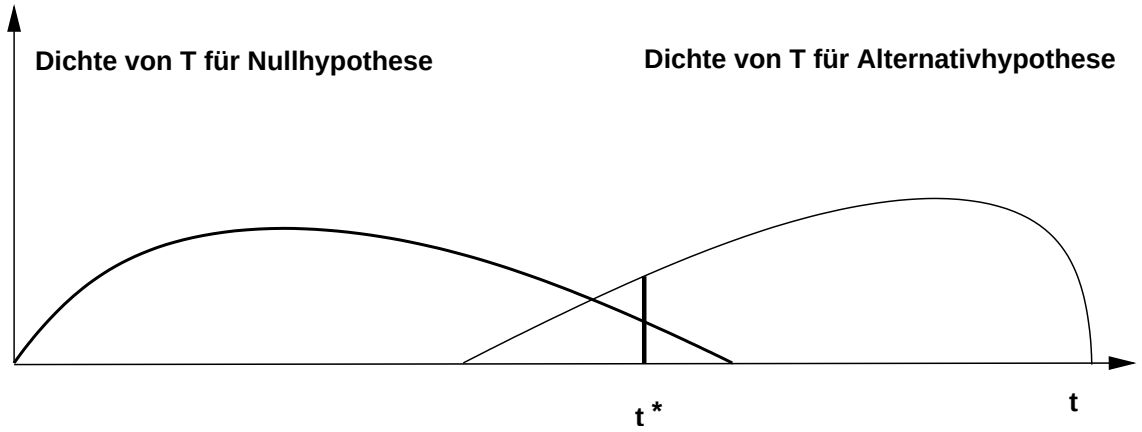
- Alternative:
  Das Niveau α^* aus der folgenden Gleichung bestimmen:

$$P(T \geq t^* | H_0) = \alpha^*$$

- Das Testen von H_0 ist nur ein Teil des Problems. Es fehlt eine alternative Hypothese H_1, die man mit H_0 vergleichen kann und die eine Hypothese bzgl. der Struktur der Daten beinhaltet, also z.B. *Die Daten enthalten 3 Cluster*.
- Sei $\{t|t \geq t_\alpha\}$ der kritische Bereich des H_0 Tests (die Menge der Werte von T für die H_0 zurückgewiesen werden muss). Dann ist

$$power = P(T \geq t_\alpha | H_1)$$

die *Macht* des Test, also die Wahrscheinlichkeit von $T \geq t_\alpha$ wenn H_1 gilt.



Vorgehensweise bei der Clustervalidierung

- Nullhypothese H_0 muss definiert werden. Sie soll für das vorliegende Szenario ausdrücken, dass keinerlei Struktur in den Daten vorhanden ist.
- Eine Statistik (Bewertungsmaß, Index) T soll festgelegt werden, diese soll sensitiv auf Struktur in den Daten sein.
- Die Verteilung von T unter der Nullhypothese muss vorhanden sein.
- Bestimmung von Grenzen t_α, welche festlegen wann die Werte der Zufallsvariablen groß (klein) sind.
- Mit t_α kann dann ein Test durchgeführt werden.

Dieses allgemeine Vorgehen wird im Folgenden an Beispielen genauer studiert.

Γ-Statistik nach Hubert

- Gegeben seinen n Objekte einer Grundgesamtheit $G = \{e_1, \ldots, e_n\}$.
- X und Y seien 2 verschiedene $n \times n$ Ähnlichkeitsmatrizen dieser n Objekte
- $X_{i,j}$ und $Y_{i,j}$ beschreiben also Ähnlichkeiten oder Distanzen zwischen den beiden Objekten e_i und e_j (Beispiel kommt gleich).
- Hubert's Γ-Statistik ist dann definiert durch die Korrelation von X und Y, also:

$$\Gamma_{raw} = \sum_{i=1}^{n} \sum_{j=1}^{n} X_{i,j} Y_{i,j} \in \mathbb{R}$$

- Γ_{raw} misst den Grad der linearen Abhängigkeit zwischen X und Y.
- Problem: Γ_{raw} ist nicht normalisiert, kann jede Zahl annehmen und ist abhängig von den gewählten Skalen der Ähnlichkeitswerte.

- Hubert's Γ-Statistik in normalisierter Form

$$\Gamma = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \left(X_{i,j} - \bar{X}\right)\left(Y_{i,j} - \bar{Y}\right)}{S_X \cdot S_Y} \in [-1, 1]$$

  hierbei sind $\bar{X}$ und $\bar{Y}$ die Mittelwerte von X und Y und S_X und S_Y die Standardabweichungen von X und Y.

- Γ misst den Grad der linearen Abhängigkeit zwischen X und Y
- Γ nimmt nur Werte zwischen -1 und 1 an.

Anwendung der Γ-Statistik

- Häufigste Anwendung der Γ-Statistik ist der Test auf Zufallsklassenlabel, bzgl. einer externen Klassenzugehörigkeit.
- Voraussetzung: Für jedes Objekt ist ein Klassenattribut vorhanden.
- Verfahren bewertet die Klassenzugehörigkeit des Paares (e_i, e_j) im Vergleich seiner Ähnlichkeit $s(e_i, e_j)$:

$$Y_{i,j} = \begin{cases} 1 & e_i \text{ und } e_j \text{in der gleichen Klasse} \\ 0 & \text{sonst} \end{cases}$$

und $X_{i,j} = s(e_i, e_j)$ ein Ähnlichkeitsmaß s auf der Grundmenge.

Vergleich mit Clusterzugehörigkeiten ist auch möglich, also:

$$X_{i,j} = \begin{cases} 1 & e_i \text{ und } e_j \text{im gleichen Cluster } C_l \\ 0 & \text{sonst} \end{cases}$$

Beispiel für Γ-Statistik

X und Y seien gegeben druch:

$$X = \begin{bmatrix} 0 & 1.2 & 0.6 & 0.2 \\ - & 0 & 0.3 & 0.4 \\ - & - & 0 & 0.1 \\ - & - & - & 0 \end{bmatrix} \qquad Y = \begin{bmatrix} 0 & 1 & 0 & 1 \\ - & 0 & 1 & 0 \\ - & - & 0 & 0 \\ - & - & - & 0 \end{bmatrix}$$

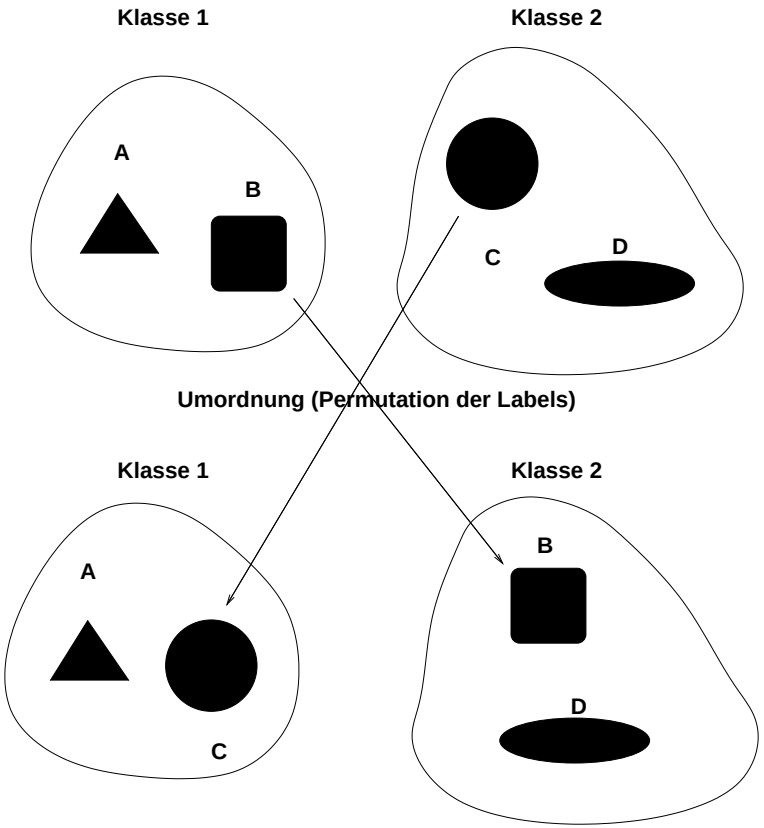
Hierbei ist $X_{i,j} = d(e_i, e_j)$ die Distanz zwischen den Objekten e_i und e_j

$$Y_{i,j} = \begin{cases} 1 & e_i \text{ und } e_j \text{in verschiedenen Klassen} \\ 0 & e_i \text{ und } e_j \text{in derselben Klasse} \end{cases}$$

e_1 und e_3 liegen in einer Klasse und e_2 und e_4 in einer anderen.

Zufallsklassenhypothese H_0: Alle Permutationen der Zeilen (und Spalten) in Y sind gleich wahrscheinlich (entspricht einer Umordnung der Objekte zu Klassen).

Hier sind die Objekte A, B, C, D in 2 Klassen aufgeteilt. Vertauschung von Zeilen/Spalten entspricht dem Vertauschen von Objekten der Klassenzugehörigkeiten:



Für die 4 Objekte ergibt dies also $4! = 24$ möglicher Permutationen. Um die Verteilung von Γ unter dieser Hypothese bestimmen zu können, sind die Γ bzw. Γ_{raw} Indices für sämtliche Permutationen g zu bestimmen:

$$\Gamma_{raw}(g) := \sum_i \sum_j X_{i,j} Y_{g(i),g(j)} \in \mathbb{R}$$

Für die Permutation g definiert durch $(1,2,3,4) \rightarrow (3,1,4,2)$ ergibt sich die Matrix

$$Y_{g(i),g(j)} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ - & 0 & 1 & 1 \\ - & - & 0 & 0 \\ - & - & - & 0 \end{bmatrix}$$

Die Verteilung von Γ ist für die 24 Permutationen:

Γ_{raw}	1.5	1.8	2.3
Häufigkeit	8	8	8

- Hohe γ-Werte sind nicht unbedingt unwahrscheinlich.
- Vollständige Berechnung erfordert die Berechnung von $n!$ Permutation g mit zugehörigem $\Gamma(g)$.
- Monte-Carlo Analysen auf kleineren Zufallsstichproben sind notwendig.
- Weitere Möglichkeit ist die Berechnung des Mittelwertes und der Varianz von $\Gamma(g)$ unter der Normalverteilungsannahme von $\Gamma(g)$. Es gibt Hinweise, dass Nährerung

$$\gamma' = \frac{\Gamma - E_0(\Gamma)}{SD_o(\Gamma)}$$

asymptotisch ($n \to \infty$) normal verteilt ist. E_0 und SD_0 sind Mittelwert und Standardabweichung von Γ und H_0.

Kruskal γ Statistik

- Allgemeine Problemstellung ist formuliert für zwei Folgen X und Y mit je m Elementen:

$$X = (x_1, x_2, \ldots, x_m) \qquad Y = (y_1, y_2, \ldots, y_m)$$

- Das Paar $\{(x_i, x_j), (y_i, y_j)\}$ heißt *konkordant* falls

$$x_i < x_j \text{ und } y_i < y_j \qquad \text{oder} \qquad x_i > x_j \text{ und } y_i > y_j$$

- Das Paar $\{(x_i, x_j), (y_i, y_j)\}$ heißt *diskordant* falls

$$x_i < x_j \text{ und } y_i > y_j \qquad \text{oder} \qquad x_i > x_j \text{ und } y_i < y_j$$

- Das Paar $\{(x_i, x_j), (y_i, y_j)\}$ ist weder konkordant noch diskordant falls

$$a_i = a_j \qquad \text{oder} \qquad b_i = b_j$$

- S_+ ist die Menge der konkordanten Paare.
- S_- ist die menge der diskordanten Paare.
- Der Kruskal γ Index ist dann definiert durch

$$\gamma = \frac{|S_+| - |S_-|}{|S_+| + |S_-|} \in [-1, 1]$$

- γ bei 1, dann sind X und Y beide wachsend oder beide fallend.
- γ bei -1, dann ist eine Folge wachsend und die andere fallend.

Beispiel

Gegeben seien die beiden folgenden Sequenzen:

i	1	2	3	4	5	6
x_i	3	5	2	2	4	6
y_i	2	3	1	6	4	5

die Brechnung der γ Statistik ist einfacher, wenn eine der Folgen aufsteigend sortiert wird.

i	4	3	1	5	2	6
x_i	2	2	3	4	5	6
y_i	6	1	2	4	3	5

Nun die Ränge und die Menge der konkordanten und diskordanten Paare bestimmen:

(i,j)	X	Y	Zustand	(i,j)	X	Y	Zustand
$(1,2)$	$(3,5)$	$(2,3)$	$+$	$(2,6)$	$(5,6)$	$(3,5)$	$+$
$(1,3)$	$(3,2)$	$(2,1)$	$+$	$(3,4)$	$(2,2)$	$(1,6)$	$*$
$(1,4)$	$(3,2)$	$(2,6)$	$-$	$(3,5)$	$(2,4)$	$(1,4)$	$+$
$(1,5)$	$(3,4)$	$(2,4)$	$+$	$(3,6)$	$(2,6)$	$(1,5)$	$+$
$(1,6)$	$(3,6)$	$(2,5)$	$+$	$(4,5)$	$(2,4)$	$(6,4)$	$-$
$(2,3)$	$(5,2)$	$(3,1)$	$+$	$(4,6)$	$(2,6)$	$(6,5)$	$-$
$(2,4)$	$(5,2)$	$(3,6)$	$-$	$(5,6)$	$(4,6)$	$(4,5)$	$+$
$(2,5)$	$(5,4)$	$(3,4)$	$-$				

Damit erhalten wir

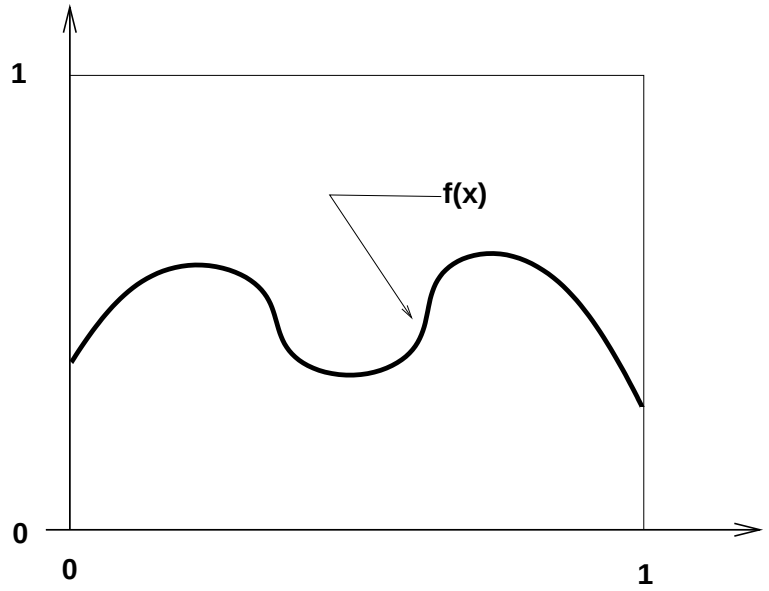
$$|S_+| = 9 \qquad |S_-| = 5 \qquad \gamma = \frac{4}{14}$$

Monte-Carlo Analyse

- Monte-Carlo-Analyse ist eine Method zur Schätzung von Parametern und Wahrscheinlichkeiten durch Computersimulationen, wenn diese Größen nicht oder nur schwer direkt berechenbar sind.
- Verteilung von vielen Indizes (im Bereich der Clusteranalyse) sind nur so zu approximieren.
- Beispiel zu Illustration der Monte-Carlo-Methoden ist die Berechnung eines Integrals

$$Q = \int_0^1 f(x)dx$$

  für eine bekannte (und berechenbare) Funktion $f : [0, 1] \rightarrow [0, 1]$.
  Die Stammfunktion von f sei nicht direkt berechnenbar.
- Berechnung von Q kann durch Schätzung der Fläche unter der Funktion f bestimmt werden.



- Zwei Beispiele werden unabhängig gemäß der Gleichverteilung auf dem Intervall $[0, 1]$ gezogen.
- Diese beiden Zahlen markieren eine Position (x, y) im Quadrat $[0, 1]^2$.
- **Erfolg** falls $y \leq f(x)$. Die relative Häufigkeit für das Ereignis **Erfolg** ist eine Schätzung für Q verwendbar.

Monte-Carlo-Schätzung mit Binomial Sampling

- Zufallsexperiment mit Computer wobei ein Ereignis (2 Ausgänge: Erfolg Misserfolg) bei jedem Experiment beobachtet wird.
- Ereignis muss in Beziehung zur zu schätzenden Größe stehen.
- Das Experiment muss sehr häufig wiederholt werden.
- Experiment mit Ausgang **Erfolg** werden gezählt.
- Relative Häufigkeit für des Ereignisses **Erfolg** ergibt die Approximation für die zu schätzende Größe.

- X_i eine Zufallsvariable, die den Ausgang des i-ten Experiments beschreibt (0 $=$ Misserfolg, 1 $=$ Erfolg).
- Monte-Carlo-Simulation ergibt Werte einer binomialverteilten Zufallsvariablen Y

$$Y = \sum_{i=1}^{m} X_i$$

m ist die Anzahl der Monte-Carlo-Experimente und

$$\{X_1, X_2, \ldots, X_m\}$$

unabhängig und identisch verteilt (i.i.d.), denn es wird angenommen, dass die Monte-Carlo-Simulationen unabhängig voneinander ausgeführt werden.

- Für die zu schätzende Größe Q ist der Erwartungswert und Varianz gilt dann

$$\mu_Q = Q \qquad \sigma_Q^2 = \frac{Q(1-Q)}{m}$$

- Konfidenzintervalle können um die Schätzung Y gelegt werden. Ein 95% Konfidenzintervall ist von der Form

$$[Y - c_m, Y + c_m]$$

wobei c_m so zu setzen ist, dass mit 95% Wahrscheinlichkeit Q in diesem Intervall liegt. Für c_m (als Funktion von m) existieren Tabellen.

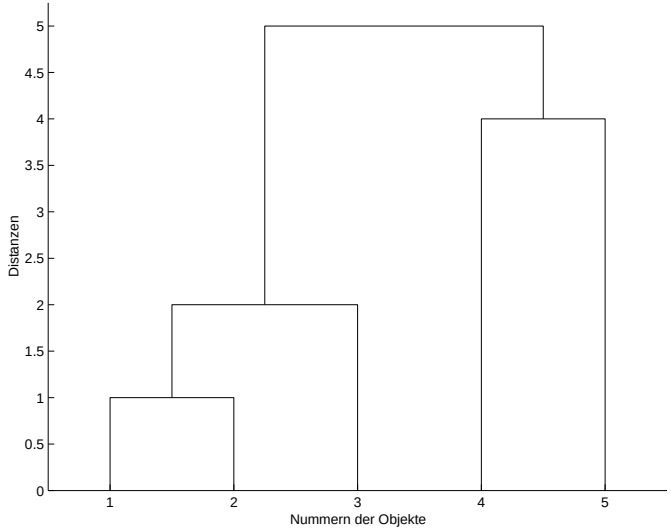
Bewertung hierarchischer Clusterungen

Mögliche Fragen in diesem Zusammenhang sind

1. Passen die errechnete hierarchische Clusterung und eine bekannte (externen) Clusterung zusammen?
   (externe Bewertung)
2. Spiegelt die errechnete hierarchische Clusterung die Ähnlichkeiten der Daten wider?
   (interne Bewertung)
3. Welche von (zwei) Clusterungen passt besser zu den Daten?
   (relative Bewertung)

Wesentliche Idee ist die Festlegung von Distanzen $d_C(e_i, e_j)$ für 2 Objekte e_i und e_j auf der Basis einer Clusterhierarchie (eigentlich ja ein Baum).

$$D = \begin{vmatrix} 0 & 1 & 2 & 9 & 13 \\ 1 & 0 & 5 & 10 & 10 \\ 2 & 5 & 0 & 5 & 13 \\ 9 & 10 & 5 & 0 & 4 \\ 13 & 10 & 13 & 4 & 0 \end{vmatrix}$$



Aus dem Dendrogramm lassen sich nun auch Distanzen bestimmen und zwar so, dass $d_C(e_i, e_j) = d$ ist wobei d das minimale Distanzniveau ist, für dass e_i und e_j erstmals in einem Cluster liegen, also hier:

$$D_C = d_C(e_i, e_j) = \begin{vmatrix} 0 & 1 & 2 & 5 & 5 \\ & 0 & 2 & 5 & 5 \\ & & 0 & 5 & 5 \\ & & & 0 & 4 \\ & & & & 0 \end{vmatrix}$$

1. **Externe Bewertung:** Hierzu braucht man eine a priori bekannte Hierarchie der n Objekte. Ist meist nicht gegeben.
2. **Interne Bewertung:** Vergleich von der Matrizen D und D_C.
3. **Relative Bewertung:** Vergleich von (zwei) Dendrogrammen (z.B. aus *single Linkage* und *complete Linkage*). Hieraus lassen sind die Matrizen D_{C_1} und D_{C_2} bestimmen, diese lassen sich vergleichen mit D.

Es führt somit jeweils auf einen Vergleich von zwei Distanzmatrizen etwa D und D_C durch einen Bewertungsindex Γ-Index oder γ-Index oder *Kendall's* τ-Index

$$\tau = \frac{|S_+| - |S_-|}{n(n-1)/2}$$

$n(n-1)$ ist die Zahl der Einträge der oberen $n \times n$ Dreiecksmatrix.

Problem: Die Verteilung dieser Indizes ist von sehr vielen Parametern abhängig,so dass eine Monte-Carlo-Analyse benutzt werden muss, um die Verteilung unter einer angenommenen Nullhypothese zu approximieren.

Interne Bewertung von n Daten aus $\mathbb{R}^d$ (genauer $[a_1, b_1] \times \cdots \times [a_d, b_d]$). Hierfür wurde eine Clusterung (etwa nach *single linkage*) und der Γ-Index für die Ausgangsdistanzmatrix D und der Matrix D_C bestimmt.
Frage : Ist der errechnete Γ_0-Wert groß genug?

Bestimme erstmal die Verteilung von Γ unter der Zufallspositionshypothese durch Monte-Carlo.

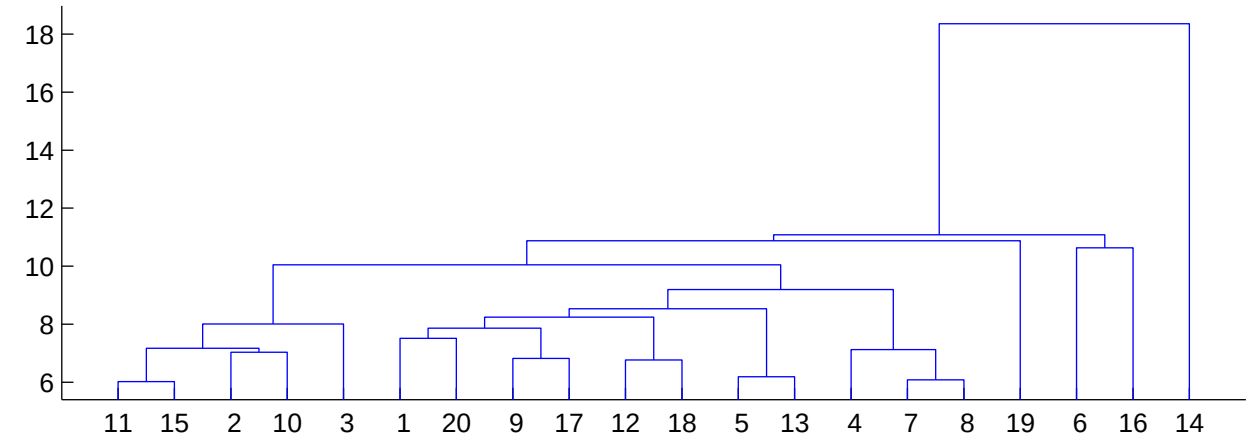
1. n Datenpunkte gemäß einer Gleichverteilung erzeugen (d Werte unabhängig gemäß der Gleichverteilung auf $[a_i, b_i]$).
2. Gemäß der gewählten Distanz die Distanzmatrix D bestimmen.
3. Das ausgewählte Clusterverfahren durchführen (z.B. *single linkage*).
4. Aus dem Dendrogramm nun die Matrix D_C ermitteln.
5. Γ-Index für D und D_C ermitteln und in ein Histogramm H_Γ eintragen
6. 1.-5. m-mal wiederholen ($m = 1000$).
7. Γ_0 mit H_Γ vergleichen.

Beispiel

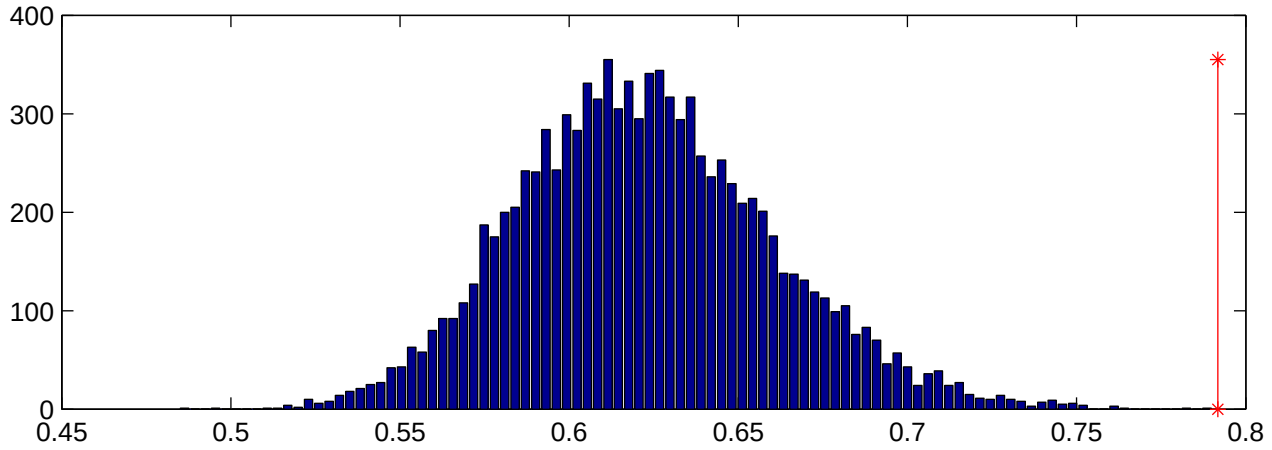
- Datensatz mit 45 Datenpunkten
- Datum: 8 Integerzahlen zwischen 0 und 20
- Monte-Carlo-Simulation mit Random-Positions-Hypothese
- Γ Statistik für $m = 10000$ Versuche
- Vergleich errechnetem Γ-Wert für die vorhandenen Daten
- Average, Complete und Single Linkage wurden angewendet mit Euklidischem Abstand.

```
 7 13  5  5  6 13  2  3
 5 13  6  4  6 13  3 13
 9 10  6  6  8 10  2  3
 7  7  6  6  8  7  2  3
 8  7  6  6  8  7  2  0
 7  7  6  7  7  7  1  1
 6 10  7  8  8  9  4  4
       .....
       .....
 8  7  5  4  6 10  1  0
 6 10  5  2  6  8  1  2
 7 10  5  5  8  7  1 20
 7 12  8  6  9 11  9  1
11  8  7 10 11 10  6  9
 9  5  6  7 10  9  7  5
10  5  6  4  9  9  6 11
 0  5 11  6  9 11  5  9
```

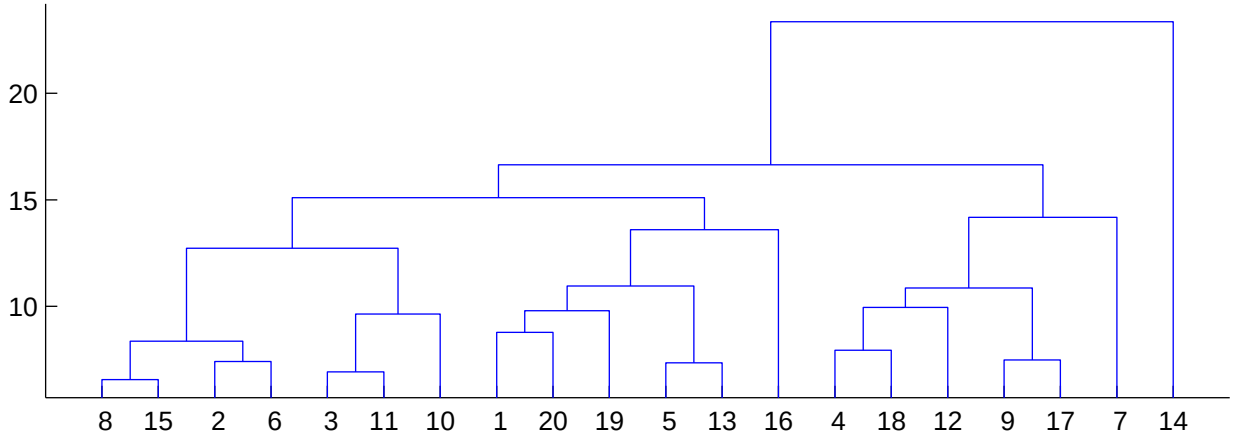
Resultate für Average Linkage



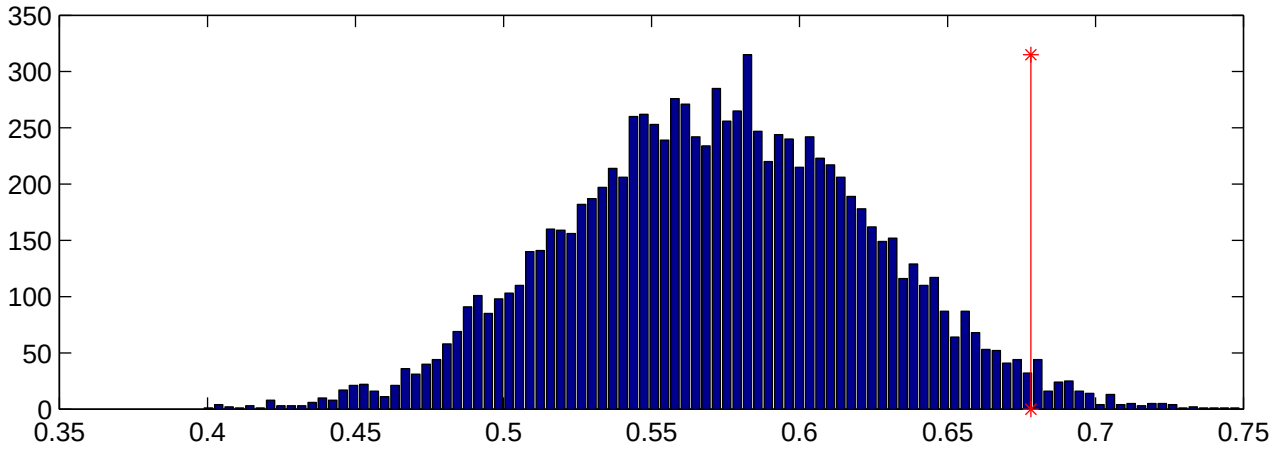
Γ_0 = 0.79177 und Verteilung Γ mit Fehler α= 0



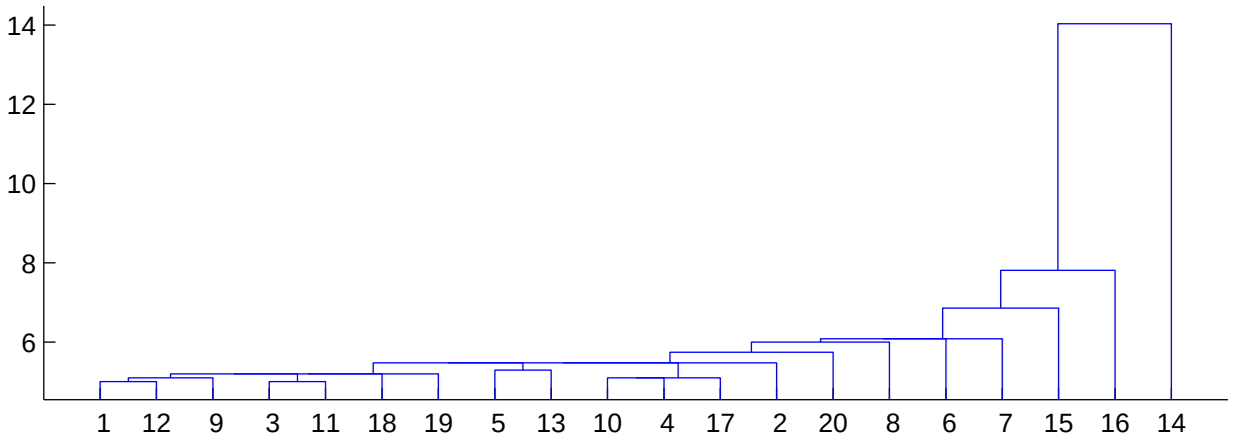
Resultate für Complete Linkage



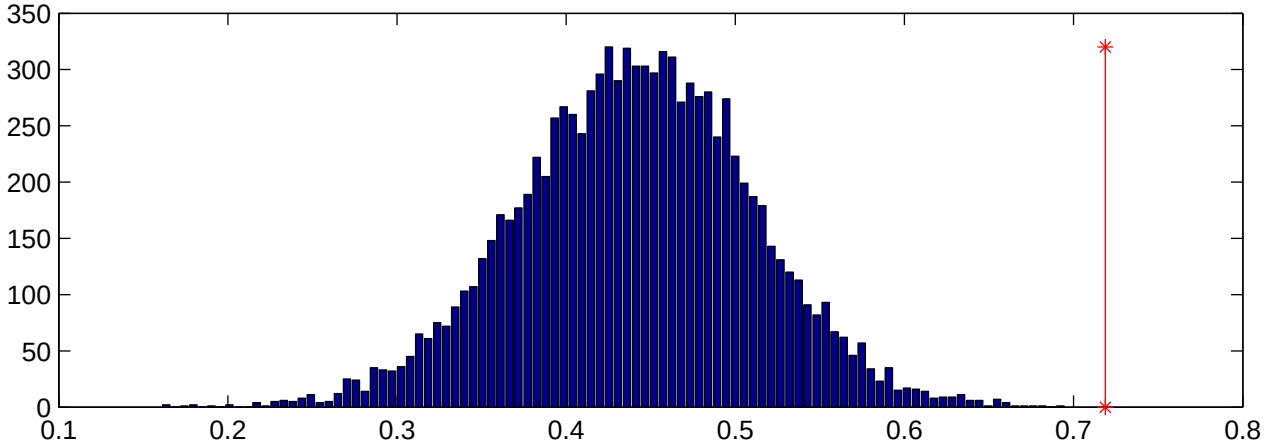
Γ_0 = 0.67811 und Verteilung Γ mit Fehler α= 0.0192



Resultate für Single Linkage



Γ_0 = 0.71881 und Verteilung Γ mit Fehler α= 0



Bewertung von Partitionen

Potenzielle Fragestellungen in diesem Zusammenhang:

- Wie wiele Cluster sind denn in den Daten vorhanden?
- Passt die gefundene Partition mit der Partitionierung gemäß eines externen Klassenlabels?
- An welcher Stelle sollte das Dendrogramm abgeschnitten werden?
- Welche von zwei Partitionen passt besser zu den daten?

Es handelt sich dabei wiederum um

- externe
- interne
- relative

Bewertungen von Clusterungen.

Externe Bewertung

- n Objekte liegen vor, $X = \{x_1, \ldots, x_n\}$, beispielsweise Vektoren aus dem $\mathbb{R}^d$.
- 2 Partitionen $\mathcal{C}$ und $\mathcal{L}$ sollen verglichen werden, eine durch ein Clusteranalyseverfahren berechnet, die andere gegeben, beispielsweise als eine Einteilung der Daten in L Kategorien/Klassen.

$$\mathcal{C} = \{C_1, \ldots, C_k\} \qquad \mathcal{L} = \{L_1, \ldots, L_l\}$$

  also C_i das i-te von k Clustern und L_j das j-te von l Klassenmengen.
- Es sein nun n_{ij} die Anzahl von $x \in X$ die in C_i und L_j liegen, also

$$n_{ij} = |\{x \in X \ : \ x \in C_i, x \in L_j\}|$$

  Dies ergibt eine $k \times l$ Kontingenztafel:

$$\begin{vmatrix} n_{11} & n_{12} & \dots & n_{1l} \\ n_{21} & n_{22} & \dots & n_{2l} \\ & & \dots & \\ n_{k1} & n_{k2} & \dots & n_{kl} \end{vmatrix}$$

- Eine Reihe von Indices (etwa Γ-Index) basieren auf dieser Kontingenztafel und lassen sich durch die folgenden beiden Indikatorfunktionen ausdrücken.

$$I_{\mathcal{C}}(i,j) = \begin{cases} 1 & x_i \in C_r \text{ und } x_j \in C_r \text{ für ein} r \\ 0 & \text{sonst} \end{cases}$$

$$I_{\mathcal{L}}(i,j) = \begin{cases} 1 & x_i \in L_s \text{ und } x_j \in L_s \text{ für ein} s \\ 0 & \text{sonst} \end{cases}$$

Dies führt in eine 2×2 Häufigkeitstabelle

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

a ist die Anzahl der Objektpaare, die in beiden Partitionen in gleichen Mengen liegen, d ist die Anzahl der Objektpaare, die in beiden Partitionen in verschiedenen Mengen liegen, b und c die Anzahl der Objektpaare, die in einer Partionen in der gleichen Menge, in der anderen Partition aber in verschiedenen Mengen liegen.

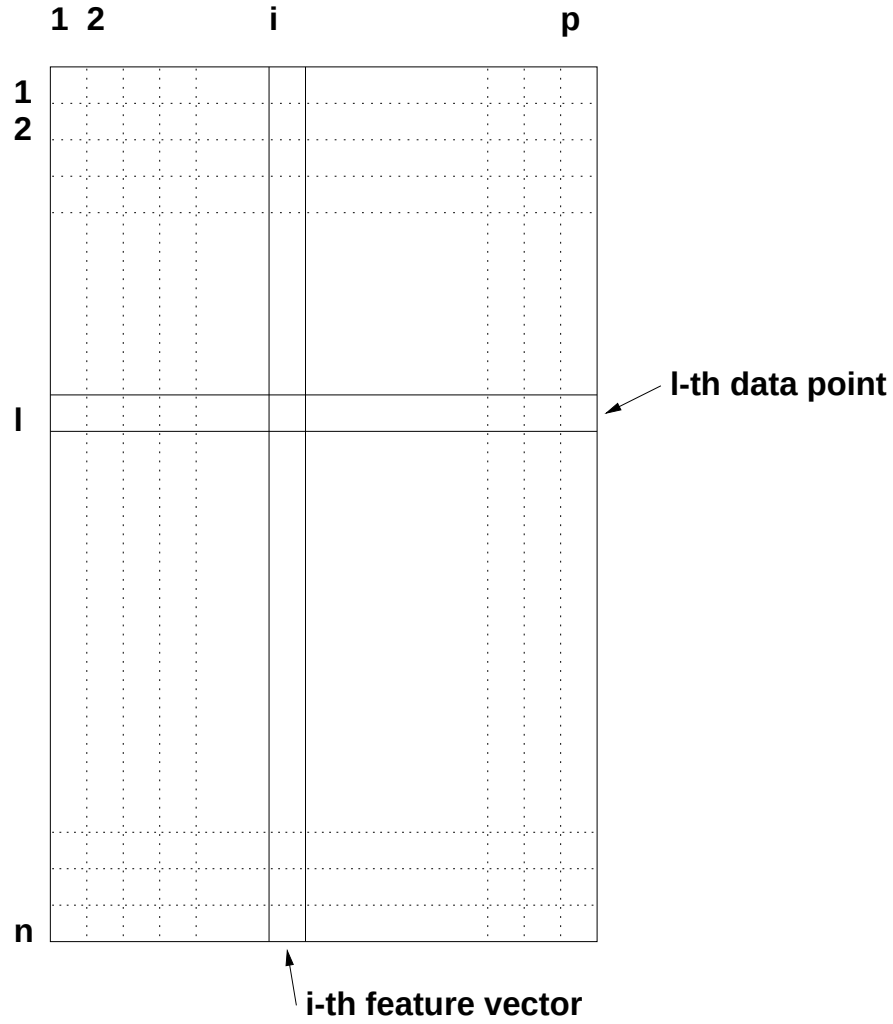
Es gilt insgesamt $a + b + c + d = n(n-1)/2 =: M$, ferner sei $m_1 := a + b$ und $m_2 := a + c$ die Anzahl der Objekte in den beiden Partitionen. Dann ergibt sich der Γ-Index zu

$$\Gamma = \frac{(Ma - m_1 m_2)}{\sqrt{m_1 m_2 (M - m_1)(M - m_2)}}$$

4. Visualisierung und Merkmalsreduktion

1. Zielsetzung und elementare Verfahren
2. Hauptachsentransformation
3. Multidimensionale Skalierung
4. Neuronale Karten (SOM) (siehe Kapitel 3.6 Neuronale Clusteranalyse)
5. Bewertung

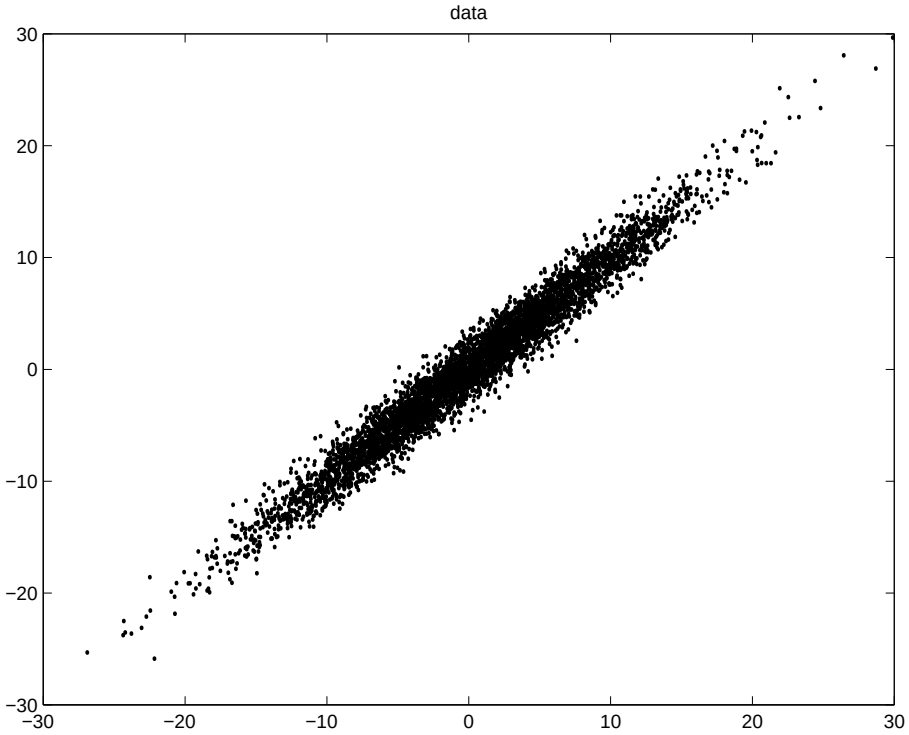
Zielsetzung



- Clusteranalyse (partionierende Verfahren und Fuzzy Verfahren) führt eine Reduktion der Datenpunkte auf einige wenige repräsentative Prototypen.
- Kohonen's SOM: Reduktion der Datenpunkte auf Prototypen und gleichzeitig Visualisierung der Prototypen durch nachbarschaftserhaltende Projektion auf ein 2D-Gitter
- Nun gesucht Reduktion der Datenpunkte auf repräsentative Merkmale, so dass die Datenmenge visualisiert werden kann.

Elementare Verfahren

- Varianzanalyse auf den Einzelmerkmalen und Reduktion auf Merkmale mit großer Varianz.
- Korrelationsanalyse von Merkmalspaaren und Reduktion auf unkorrelierte Merkmalspaare.
- Scatterplots, Histogramme (1D und 2D)



- Variation der Daten in Richtung der beiden vorgegeben Merkmale ist gleich.

- In Richtung des Vektors $(1, 1)$ ist die Variation der Daten groß; in Richtung $(1, -1)$ dagegen gering.

- Offensichtlich sind Merkmale in denen die Merkmalsausprägungen überhaupt nicht variieren bedeutungslos.

- Datenreduktion in hochdimensionalen Merkmalsräumen durch Auffinden von Richtungsvektor mit großer Variation (die sogenannten **Hauptachsen**).

- Die Hauptachsen lassen sich anordnen:
  - $1.$ Hauptachse beschreibt den Vektor $v_1 \in \mathbb{R}^d$ mit der größten Variation der Daten;
  - $2.$ Hauptachse ist der Vektor $v_2 \in \mathbb{R}^d$ der senkrecht auf v_1 steht Vektoren und in dessen Richtung die Datenpunkte am stärksten variieren.
  - $l.$ Hauptachse ist der Vektor $v_l \in \mathbb{R}^d$ der senkrecht auf $V_{l-1} := lin\{v_1, \ldots, v_{l-1}\}$ steht und in dessen Richtung die Datenpunkte am stärksten variieren

Hauptachsentransformation

- Gegeben sei ein Datensatz mit n Punkten $x^\mu \in \mathbb{R}^p$, zusammengefasst als Datenmatrix X.

- Die einzelnen Merkmale ($=$ Spaltenvektoren in der Datenmatrix X) haben den Mittelwert $= 0$. Sonst Mittelwertbereinigung durchführen.

- Für einen Vektor $v \in \mathbb{R}^p$ und $x^\mu \in X$ ist $\langle v, x^\mu \rangle = \sum_{i=1}^{p} v_i \cdot x_i^\mu$ die Projektion von x^μ auf v.

- Für alle Datenpunkte X ist Xv der Vektor mit den Einzelprojektionen.

- Die Gesamt-Varianz in Richtung v ist dann

$$\sigma_v^2 = (Xv)^t(Xv) = v^t X^t X v = v^t C v$$

  mit $C = X^t X$.

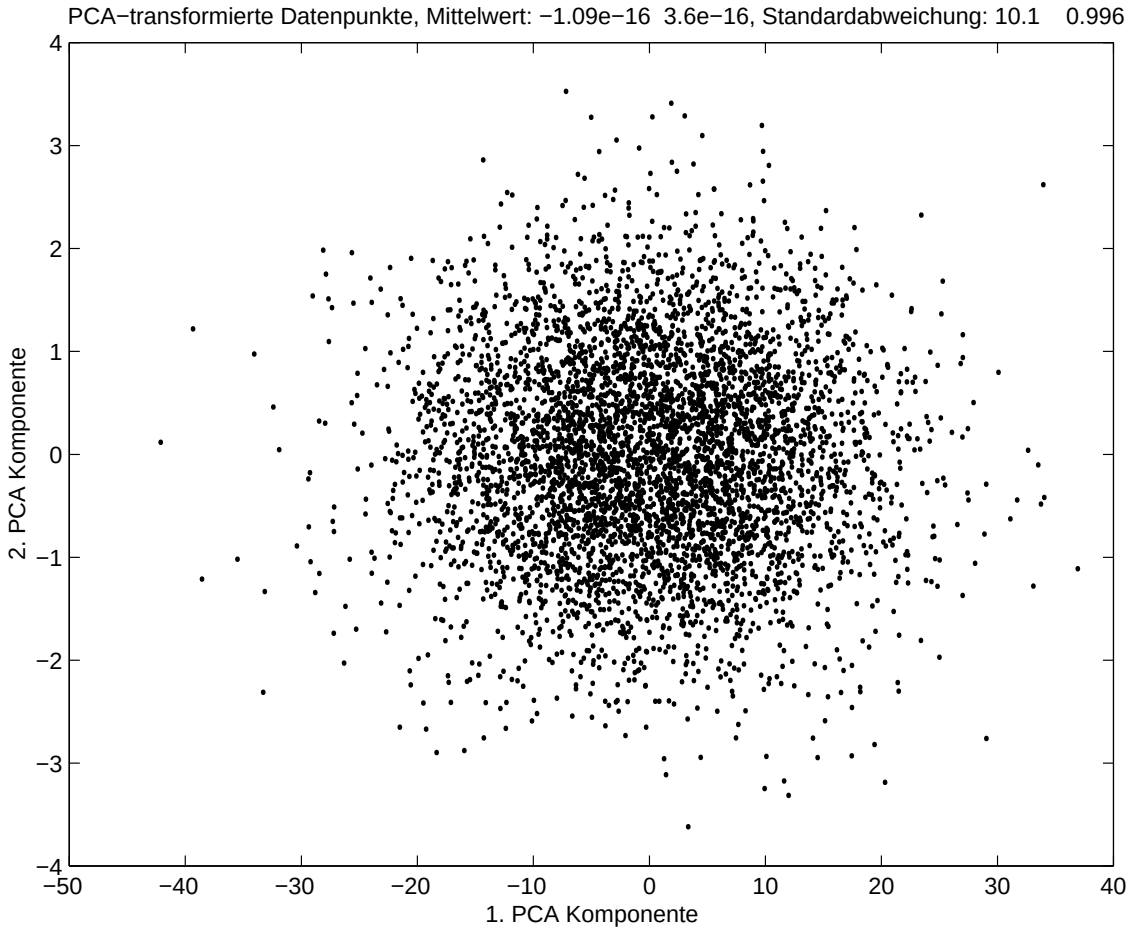
- Bezüglich der Matrix C soll nun σ_v^2 maximiert werden.
- Ohne Randbedingungen an v ist eine Maximierung nicht möglich.
- Normierung als Bedingung: $v^t v = \|v\|^2 = 1$
- Maximierung unter Nebenbedingungen führt auf die Maximierung der Funktion.
$$\varphi(v) = v^t C v - \lambda(v^t v - 1)$$
mit dem **Lagrange Multiplikator** $\lambda \in \mathbb{R}$.
- Differenzieren von φ nach v und Nullsetzen liefert:
$$\frac{\partial \varphi}{\partial v} = 2Cv - 2\lambda v = 0$$
- Dies führt direkt auf die Matrixgleichung in Eigenvektorform
$$Cv = \lambda v$$

- C hat nur Eigenwerte $\lambda_i \geq 0$, da C symmetrisch und nichtnegativ definit ist, OBdA. $\lambda_1 \geq \cdots \lambda_p \geq 0$

- Der Eigenvektor v_l zum größten Eigenwert λ_l ist dann die l. Hauptachse.

- Vorgehensweise in der Praxis:
  - Merkmale auf Mittelwert $= 0$ transformieren;
  - Kovarianzmatrix $C = X^t X$ berechnen
  - Eigenwerte und Eigenvektoren (die Hauptachsen) von C bestimmen
  - Daten X auf die $p' \leq p$ Hauptachsen transformieren.
  - Dies ergibt eine Datenmatrix X' mit n Zeilen (Anzahl der Datenpunkte) und p' Merkmalen.

Hauptachsentransformierte Beispieldaten



Multidimensionale Skalierung

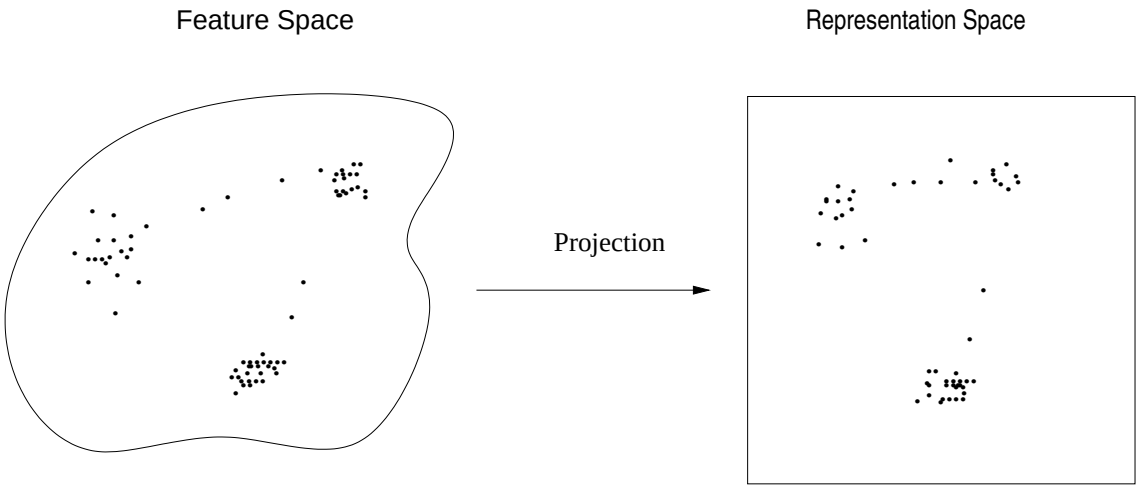
- Gegeben n Objekte $\{x^1, \ldots, x^n\} \subset \mathbb{R}^p$

- d^X sei eine (symmetrische) Abstandsfunktion in X und $d^X_{ij} := d^X(x^i, x^j)$ seien gegeben.

- Y sei nun eine weitere Menge (der *Visualisierungsraum*) mit einer Abstandsfunktion d^Y. Y ist meist eine Teilmenge des $\mathbb{R}^2$.

- Gesucht ist nun eine abstandserhaltende Abbildung $\mathcal{P} : X \to Y$ derart, dass für die Distanzen $D^X := (d^X(x^i, x^j))_{1 \leq i,j \leq M}$ in X und $D^Y := (d^Y(\mathcal{P}(x^i), \mathcal{P}(x^j)))_{1 \leq i,j \leq M}$ in Y gilt:

$$D^X \approx D^Y.$$



- Die Abweichung zwischen D^X und D^Y kann man durch sogenannte *Stressfunktionale* messen:

$$S = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\Phi[d^X(x^i, x^j)] - \Phi[d^Y(\mathcal{P}(x^i), \mathcal{P}(x^j))] \right)^2$$

  $\Phi : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ ist dabei eine streng monoton wachsende Funktion, z.B. $\Phi(s) = \log(s+1)$ oder $\Phi(s) = s$ oder $\Phi(s) = s^2$.

- Setzen nun $y^j := \mathcal{P}(x^j)$ und gehen davon aus, dass $Y = \mathbb{R}^r$ mit der Eu-

klidischen Abstandsfunktion d ausgestattet ist, dann sind die Positionen y^j für $j = 1, 2, \ldots, n$ gesucht.

- Für das Stressfunktional gilt dann:

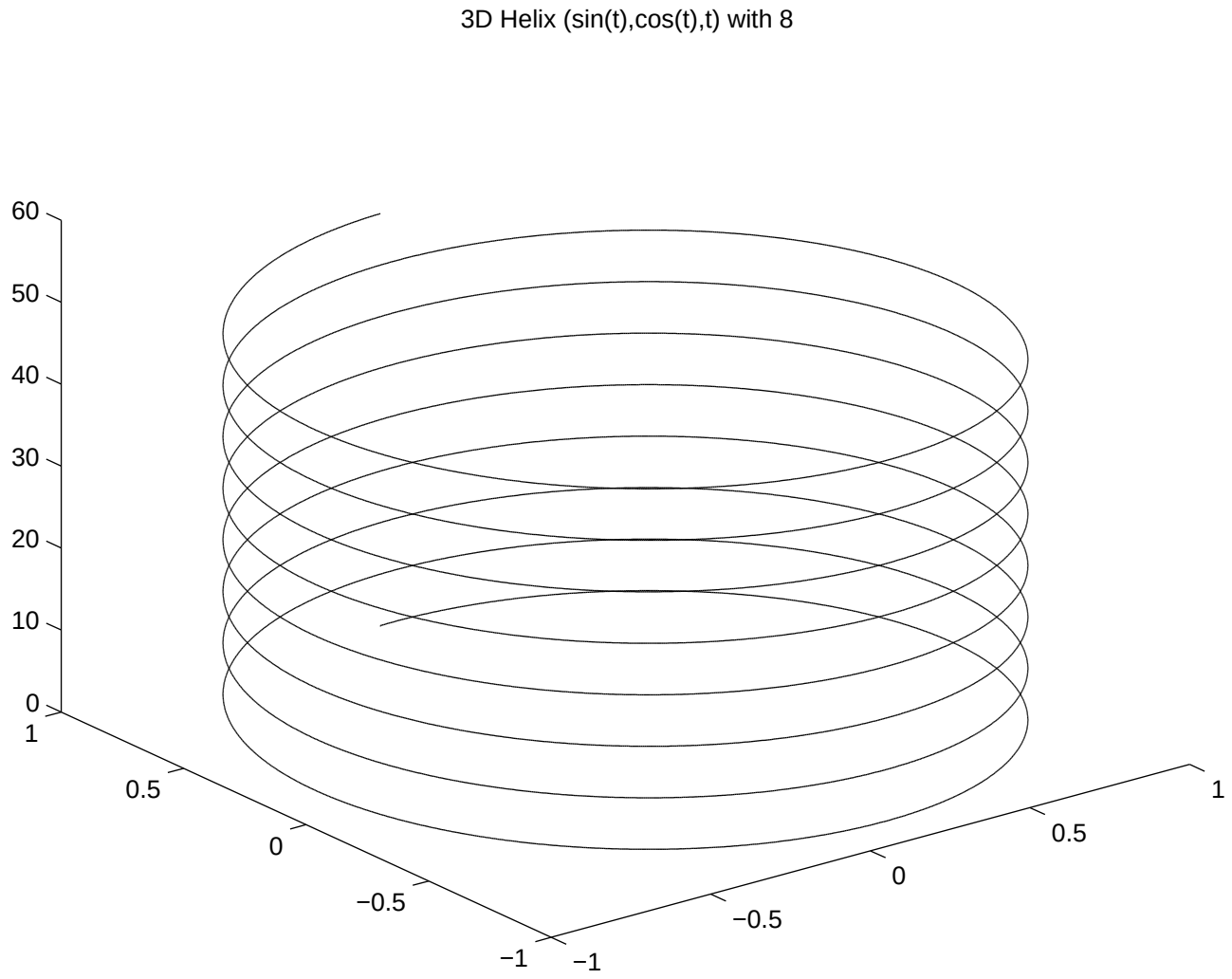
$$S(y^1, \ldots, y^n) = \sum_{j,i=1}^{n} \Big(\Phi[d^X(x_i, x_j)] - \Phi[d(y^i, y^j)] \Big)^2$$

- Das Stressfunktional S lässt sich durch Gradientenverfahren minimieren.

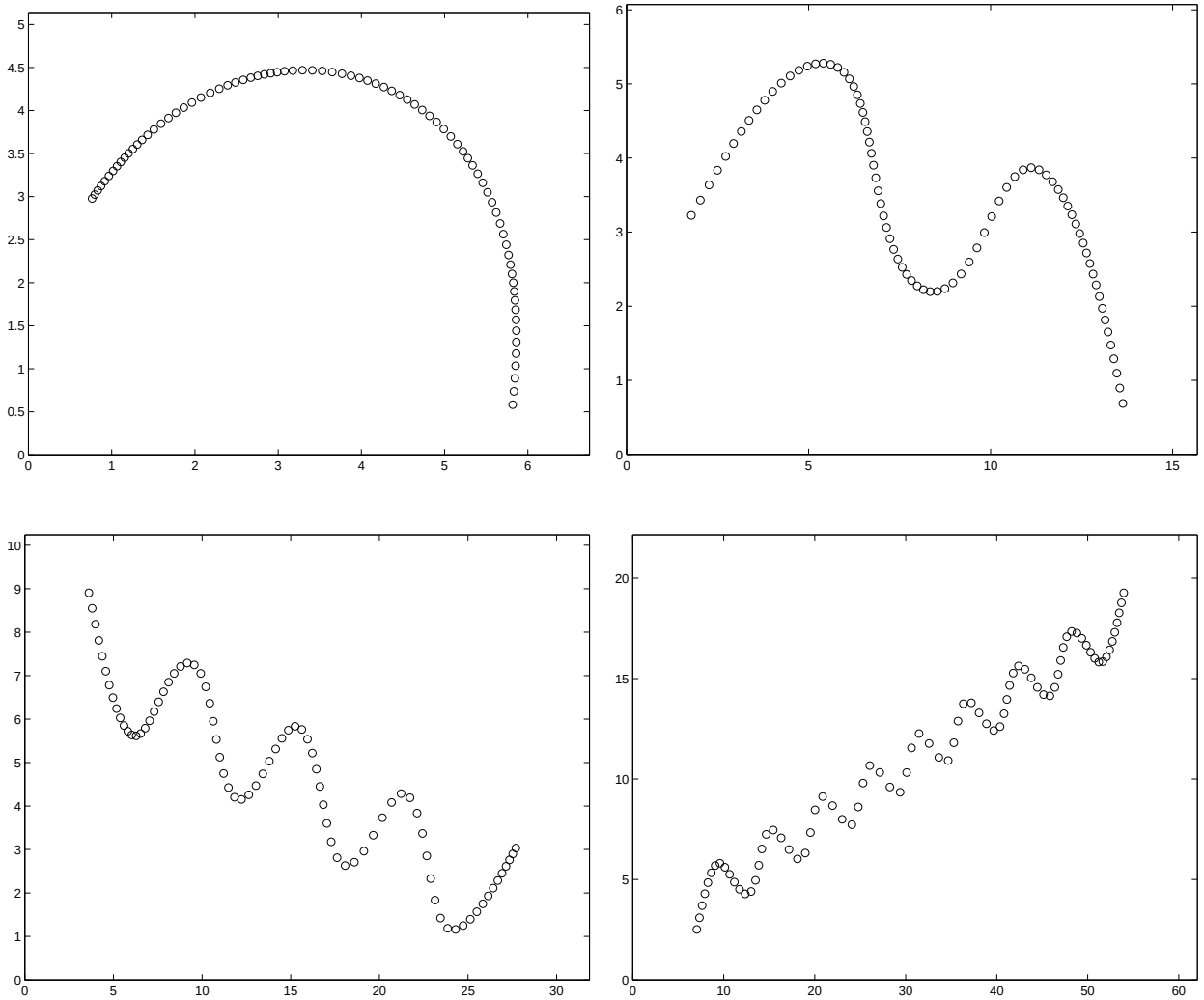
- Hierfür ergibt sich die folgende inkrementelle *Adaptationsregel* ($l > 0$ Lernrate) für die $y^j \in \mathbb{R}^r$, $j = 1, \ldots, n$.

$$\Delta y^j = l \sum_{i=1}^{n} \Phi'[d^2(y^i, y^j)] \Big(\Phi[d^X(x^i, x^j)] - \Phi[d^2(y^i, y^j)] \Big) \Big(y^i - y^j \Big)$$

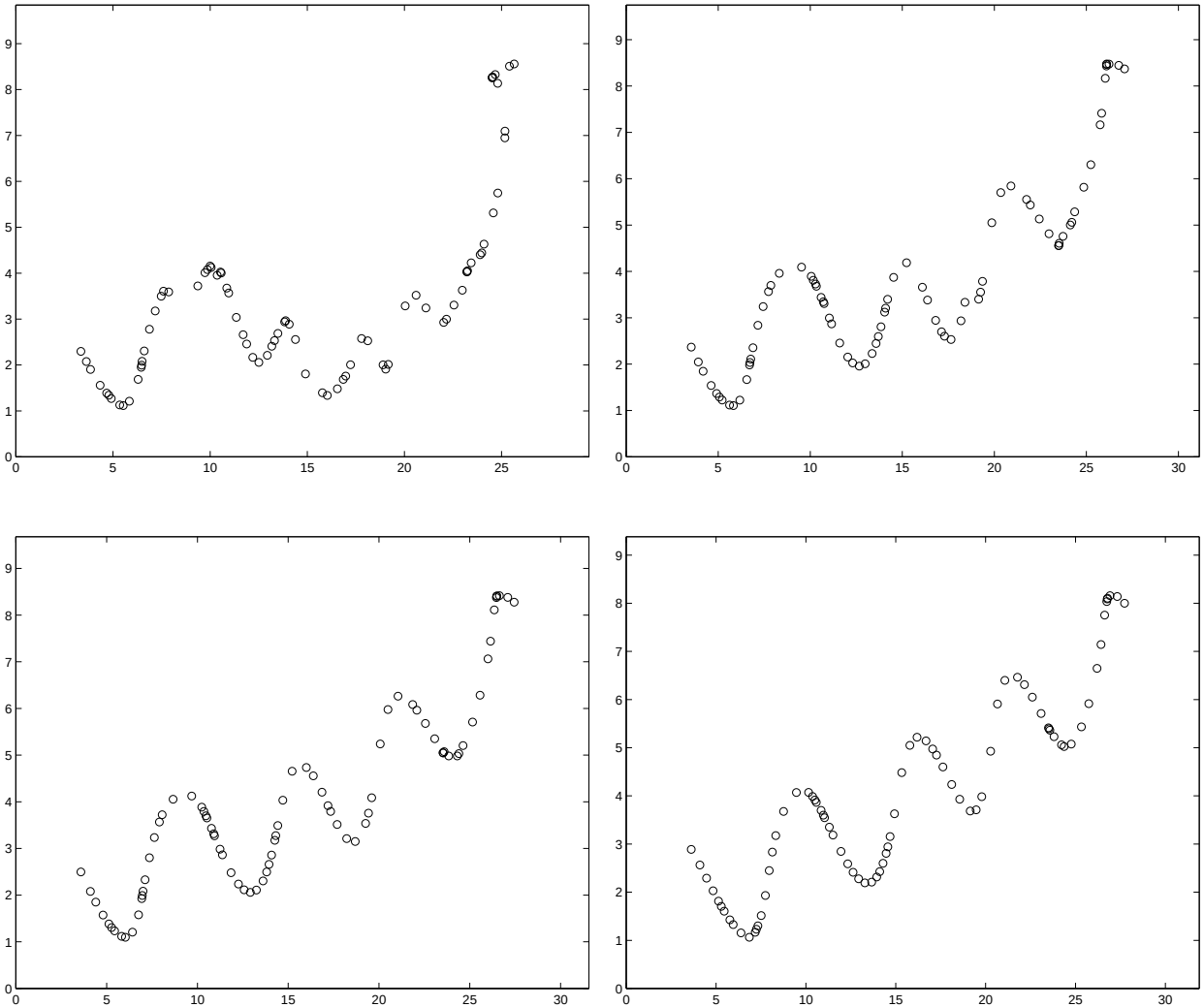
Beispiel: 3D-Helix mit 8 Schleifen



Resultate der MDS für 3D-Helix



Entwicklung der 2D-Projektion für 4 Schleifen



Kombination von Clusterung und MDS

- MDS berechnet für jeden der n Datenpunkte ein Projektionspunkt.
- Berechnungsaufwand ist mindestens $O(n^2)$.
- MDS ist ein glabales Verfahren (alle $n(n-1)/2)$ Abstände sollen erhalten bleiben). Ist für große n nicht mehr realisierierbar.
- Ausweg: Erst Clusteranalyse durchführen, genauer eine kleine Menge repräsentaitver Prototypen ($c_1 \ldots, c_k$) berechnen und dann MDS auf die Prototypen anwenden.
- Alternative: Protoypen $c_1, \ldots, c_k$ und zugehörige MDS-Projektionen $p_1, \ldots, p_k$ inkrementell bestimmen.

- Kombination von 2 Zielen ähnlich wie beim Kohonen-Verfahren: Clusteranalyse und distanzerhaltende Projektion durch MDS.

$$E(c_i, p_i) := \sum_{j=1}^{k} \sum_{x^\mu \in C_j} \|x^\mu - c_j\|^2 + \frac{\lambda}{2} \sum_{i=1}^{k} \sum_{j=1}^{k} \Big(\Phi[\|c_i - c_j\|^2] - \Phi[\|p_i - p_j\|^2] \Big)^2$$

$\lambda > 0$ ein Gewichtungsparameter.

- Lernregeln für die Projektionen $p_1, \ldots, p_k$ sind dann wie beim Standard MDS-Verfahren.

- Lernregeln der $c_1, \ldots, c_k$ ähnlich wie beim k-means-Verfahren:

$$\Delta c_{j^*} = \frac{1}{C_{j^*} + 1}(x^\mu - c_{j^*}) + l\lambda \sum_{i=1}^{k} \delta_{ij^*} \Big(c_i - c_{j^*} \Big)$$

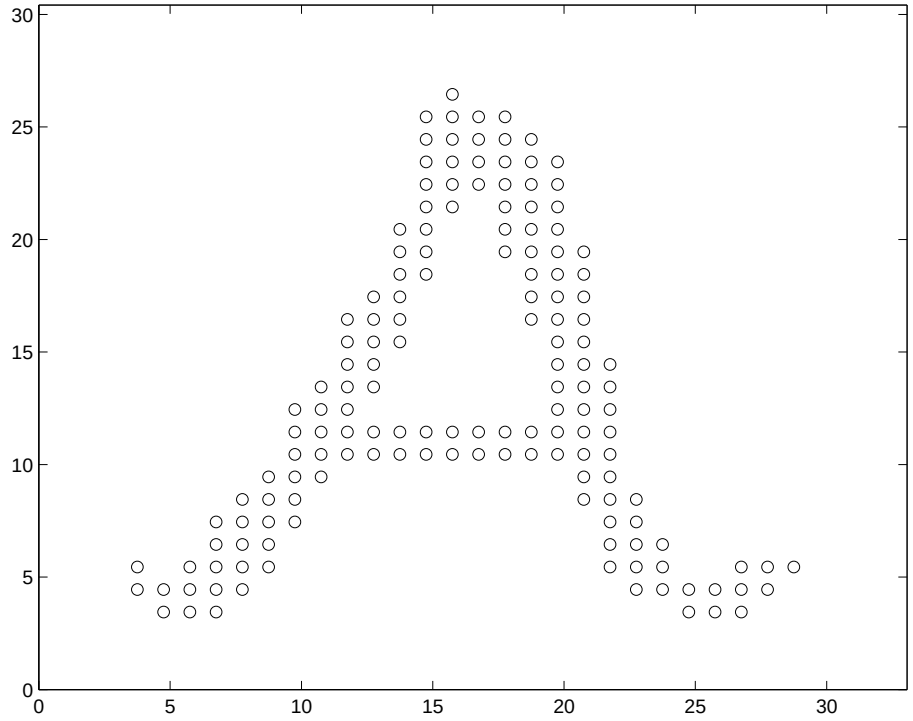
$$\delta_{ij^*} = \Phi'[\|c_i - c_{j^*}\|^2] \Big(\Phi[\|c_i - c_{j^*}\|^2] - \Phi[\|p_i - p_{j^*}\|^2] \Big)$$

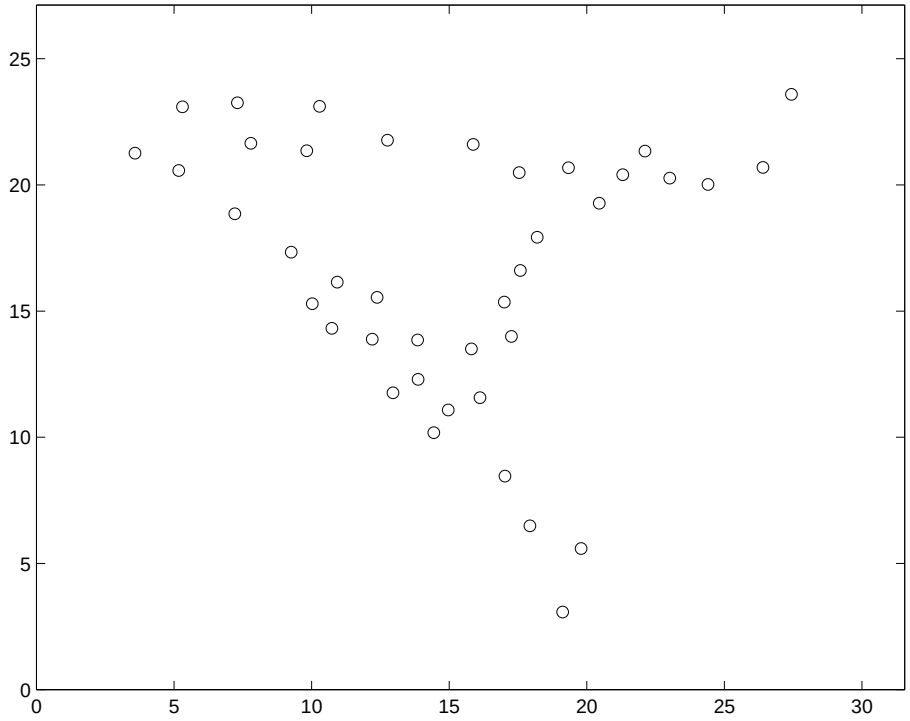
ACMMDS Algorithm

ACMMDS = **A**daptive **c**-**m**eans and **M**ulti-**D**imensional **s**caling

estimate thresholds θ_{new}
set $k = 0$ (no prototypes)
  choose a data point $x \in X$
  calculate $d_j = d(x, c_j), j = 0, \ldots, k$
  detect the winner $j^* = \text{argmin}_j \; d_j$
  if $(d_{j^*} > \theta_{new})$ or $k = 0$
    $c_k := x$ and adapt p_k
    $k := k + 1$
  else
    adapt c_{j^*} and p_{j^*}
 goto: choose data point

Beispiele



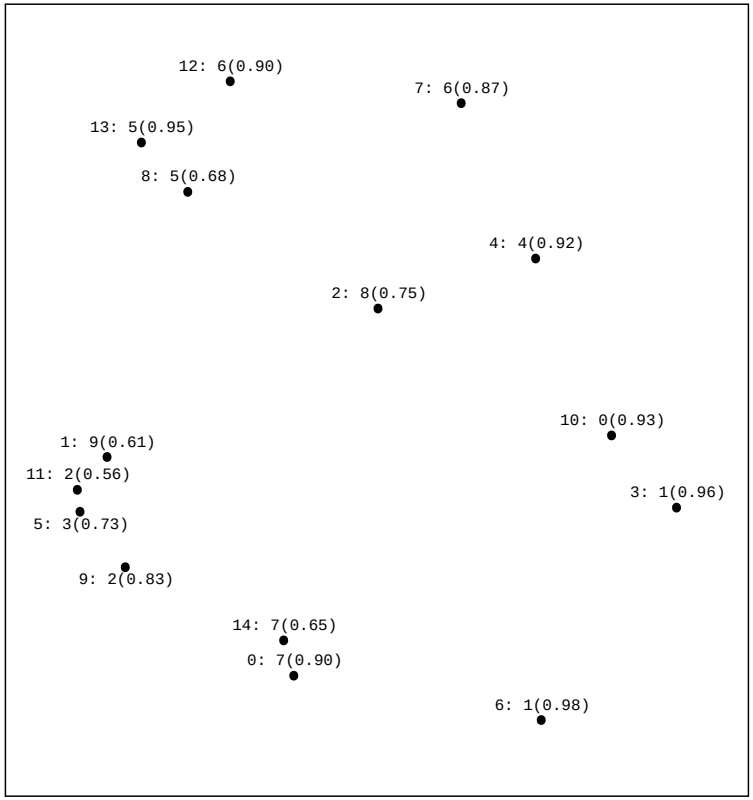


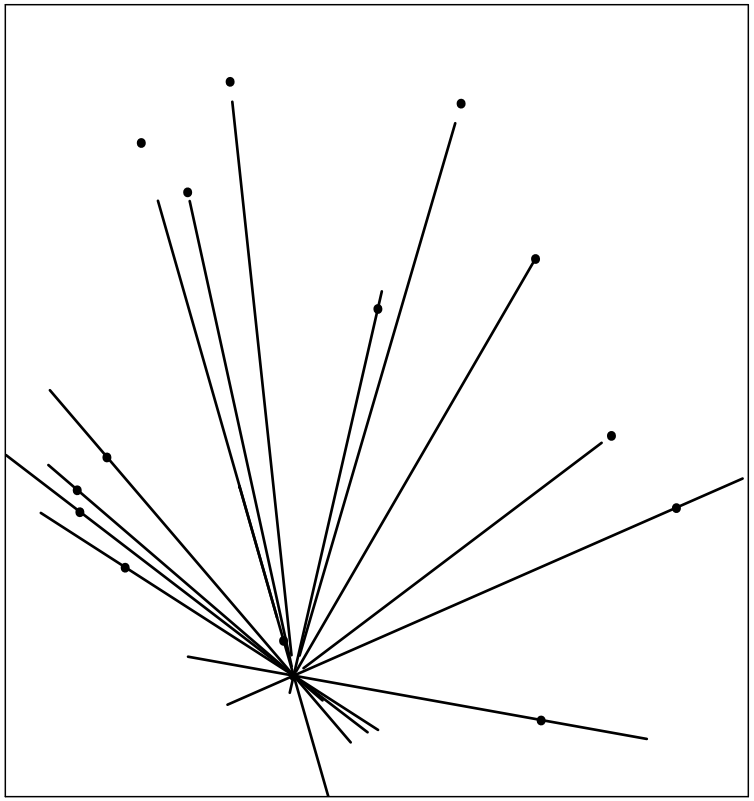
153 Punkte werden in den $\mathbb{R}^{128}$ eingebettet. Dann ACMMDS Reduktion auf 2D und 40 Prototypen.

Beispiel: Ziffern

10000 handgeschriebene Ziffern reduziert auf 15 Clusterzentren

Projektionen mit Stress





Bewertungen von Projektionen

- Stressfunktionale für x_i und $p(x_i)$

$$S = \sum_{j,i=1}^{n} \Big(\Phi[d^X(x^i, x^j)] - \Phi[d^Y(\mathcal{P}(x^i), \mathcal{P}(x^j))] \Big)^2$$

- Ränge für die Distanzen $d^X(x^i, x^j)$ in X und $d^Y(\mathcal{P}(x^i), \mathcal{P}(x^j))$ in Y bilden und über Rangordungskorrelation auswerten. Siehe dazu **Rangordnungskoeffizient von Spearman** in dem Kapitel 2 (multivariate Statistik).

5. Lernen von Assoziationsregeln

1. Zielsetzung
2. Support und Konfidenz
3. Items und Assoziationsregeln
4. Der A-priori-Algorithmus
5. Klassifikationsregeln

Zielsetzung

- Verfahren zum Entdecken von Assoziationsregeln sind typische Data Mining Methoden.
- Assoziationsregeln beschreiben Zusammenhängen zwischen gemeinsam auftretenden Merkmalsausprägungen.
- Algorithmen zur Bestimmung von Assoziationsregeln sind unüberwachte Lernverfahren, d.h. ohne Lehrersignale.
- Typisches Anwendungsfeld: Zusammenhänge beim Einkauf, die sogenannte Warenkorbanalyse.
  Bei 60% der Einkäufe, in denen Bier gekauft wird, werden auch Kartoffel-Chips gekauft. Beide Produkte kommen in 2% der Einkäufe vor.
- Viele Daten sind zu analysieren (Scannerkassen im Supermarkt, log-files im Internet, Rabattkarten, etc.).
- Merkmalskombinationen kommen spärlich vor.

Support und Konfidenz

Kenngrößen von Assoziationsregeln sind

- **Support**: relative Häufigkeit der Beispiele, in denen die Regel anwendbar ist. *.... kommen in 2% der Einkäufe vor.*

- **Konfidenz**: relative Häufigkeit der Beispiele, in denen die Regel richtig ist. *Bei 60% der Einkäufe, ...*

Algorithmen sind so zu entwerfen, dass alle gefundenen Assoziationsregeln a priori definierte **Mindestkonfidenz** und **Mindestsupport** erfüllen sollen.

Diese Verfahren sollen dabei keine Annahmen über die zu analysierenden Merkmale benötigen (wäre z.B. in einem Versandhandel mit vielen Tausend verschiedenen Artikeln auch nicht durchführbar).

Items

Ausgangspunkt ist eine Datenmatrix X mit p **nichtnumerischen** Merkmalen (nominal oder ordianal skaliert) und mit n Beispielen (in den genannten Anwendungsfeldern auch als **Transaktionen** bezeichnet)

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
overcast	cool	normal	true	yes
sunny	cool	normal	false	yes
rainy	mild	high	true	no

Ein **Item** ist ein (Merkmal,Ausprägung)-Paar: $(\texttt{temperature}, \texttt{cool})$, $(\texttt{windy}, \texttt{true}))$.

In typischen Anwendungen (wie Warenkorb-Analysen) mit binären Merkmalsausprägungen (aber mit vielen Merkmalen), sind nur einige wenige Merkmalsausprägung $= 1$.

Transaktionen sind effizienter als Mengen von Items speicherbar: z.B. $x^k = \{(\texttt{Bier}, 1), (\texttt{Chips}, 1)\}$ oder intuitiver $x^k = \{\texttt{Bier}, \texttt{Chips}\}$.

Assoziationsregeln

Eine **Assoziationsregel** $Y \rightarrow Z$ besteht aus einem

- **Regelrumpf** Y
- **Regelkopf** Z

wobei Y und Z zwei disjunkte Item-Mengen sind.

Ein Beispiel/Transaktion x^k aus der Datenmatrix/Transaktionenmenge erfüllt die Assoziationsregel $Y \rightarrow Z$, gdw. $Y \cup Z \subset x^k$.

Beispiel: Die Regel $(\texttt{temperature}, \texttt{cool}) \rightarrow (\texttt{humidity}, \texttt{normal})$ wird von den Beispielen/Transaktionen x^2 und x^3 erfüllt.

Definition von Support und Konfidenz

Für eine Item-Menge Y ist der *Support* definiert als

$$\text{support}(Y) := \frac{|\{x^k \in X \ : \ Y \subset x^k\}|}{n}$$

(n Anzahl der Beispiele in der Datenmatrix X).

Der *Support* für eine Assoziationsregel $Y \rightarrow Z$ zweier disjunke Item-Mengen Y und Z ist definiert durch:

$$\text{support}(Y \rightarrow Z) := \text{support}(Y \cup Z)$$

Die *Konfidenz* für eine Assoziationsregel $Y \rightarrow Z$ zweier disjunke Item-Mengen Y und Z ist definiert durch:

$$\text{konfidenz}(Y \rightarrow Z) := \frac{\text{support}(Y \rightarrow Z)}{\text{support}(Y)}$$

Beispieldatensatz

outlook	temparature	humidity	windy	play
sunny	85	85	false	no
sunny	80	90	true	no
overcast	83	86	false	yes
rainy	70	96	false	yes
rainy	68	80	false	yes
rainy	65	70	true	no
overcast	64	65	true	yes
sunny	72	95	false	no
sunny	69	70	false	yes
rainy	75	80	false	yes
sunny	75	70	true	yes
overcast	72	90	true	yes
overcast	81	75	false	yes
rainy	71	91	true	no

Problem hier: Merkmale `temperature` und `humidity` sind numerisch skaliert!

Eine Quantisierung (Klassenbildung) der numerischen Merkmale (`temperature` und `humidity`) in ordinal skalierte Merkmale ist notwendig. Etwa wie folgt:

- `temperature` $\in \{\text{hot}, \text{normal}, \text{mild}, \text{cool}\}$
- `humidity` $\in \{\text{high}, \text{normal}\}$

Außerdem

- `outlook` $\in \{\text{sunny}, \text{overcast}, \text{rainy}\}$
- `windy` $\in \{\text{false}, \text{true}\}$
- `play` $\in \{\text{yes}, \text{no}\}$ (Klassenattribut)

Insgesamt gibt es 96 verschiedene Kombinationsmöglichkeiten, davon sind 14 Beispiele (Transaktionen) gegeben.

Datensatz nach Klassenbildung

outlook	temparature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Beispiele:

support$(\{(windy, true)\}) = 6/14$

support$(\{(temperature, cool), (humidity, normal)\}) = 4/14$

support$(\{(temperature, hot), (windy, false), (play, yes)\}) = 2/14$

support$(\{(temperature, cool)\} \rightarrow \{(humidity, normal)\}) = 4/14$

konfidenz$(\{(temperature, cool)\} \rightarrow \{(humidity, normal)\}) = 1$

konfidenz$(\{\{(humidity, normal)\} \rightarrow \{(temperature, cool)\}) = 4/7$

support$(\{(humidity, normal), (windy, false)\} \rightarrow \{(play, yes)\}) = 4/14$

konfidenz$(\{(humidity, normal), (windy, false)\} \rightarrow \{(play, yes)\}) = 1$

Es können aber noch viel mehr Assoziationsregeln gefunden werden mit: konfidenz $= 1$

Finden von Assoziationsregeln

Es seien gegeben

- Datenmatrix/Transaktionsmenge. Transaktionen x_k als Menge von Items.
- $s_{\text{min}} \in (0, 1)$ Wert für den minimalen Support.
- $k_{\text{min}} \in (0, 1)$ Wert für die minimale Konfidenz.

Gesucht sind alle Assoziationsregeln $Y \rightarrow Z$ mit

- support$(Y \rightarrow Z) \geq s_{\text{min}}$
- konfidenz$(Y \rightarrow Z) \geq k_{\text{min}}$

Apriori Algorithmus : Idee

Zerlegung des Gesamtproblems in die beiden folgenden Teilprobleme:

1. Bestimme **alle** Item-Mengen X für die gilt:

$$\text{support}(X) \geq s_{\text{min}}$$

   Item-Mengen X mit $\text{support}(X) \geq s_{\text{min}}$ sind die **häufigen** Item-Mengen.

2. Berechne **alle** Assoziationsregeln $X \rightarrow Y$ aus den häufigen Item-Mengen für die gilt:

$$\text{konfidenz}(X \rightarrow Y) \geq k_{\text{min}}$$

Bestimmung der häufigen Item-Mengen

Bei der Suche nach häufigen Item-Mengen macht man sich die folgende Eigenschaft von Item-Mengen zu Nutze:

Für zwei Item-Mengen mit $Y \subset Z$ gilt offenbar

$$\text{support}(Z) \leq \text{support}(Y)$$

Das heißt

- Teilmengen einer häufigen Item-Menge sind häufige Item-Menge.
- Obermengen einer nicht häufigen Item-Menge sind nicht häufige Item-Mengen.

Die häufigen Item-Mengen lassen sich iterativ (bzgl. ihrer Länge) bestimmen.

Definiere dazu für $l \geq 1$:

$$I_l := \{Y \ : \ \text{support}(Y) \geq s_{\text{min}}, |Y| = l\}$$

Die Berechnung von I_{l+1} aus I_l erfolgt in zwei Schritten:

1. Die l-elmentigen häufigen Item-Mengen aus I_l werden systematisch zu $l+1$ elementigen Item-Mengen erweitert.
   Für eine binärer Datenmatrix/Transaktionen gehen aus einer einzelnen l-elmentigen Item-Menge $n - l$ Item-Mengen der Kardinalität $l + 1$ hervor.
   Dies sind zunächst Kanditaten für häufige $l + 1$ elementige Item-Mengen.

2. Jede dieser $l + 1$ elementigen Item-Mengen Y muss anschließend geprüft werden, ob sie häufig ist, also ob $\text{support}(Y) \geq s_{\text{min}}$ gilt

A priori Algorithmus

1. Wähle $s_{\text{min}} \in (0,1)$; Setze $n = 1$ und $I = \emptyset$ und
   $H_1 = \{Y \ : \ Y$ ist 1-elementige Item-Menge$\}$

2. Bestimme nun den support(Y) für alle $Y \in H_n$
   (ein Lauf durch die Transaktionsmenge!)

3. $I_n = \{Y \in H_n \ : \ \text{support}(Y) \geq s_{\text{min}}\}$

4. Falls $I_n = \emptyset$ return I sonst $I = I \cup I_n$

5. $H_{n+1} = \{Y \cup Y' \ : \ Y \in I_n$ und $Y' \not\subset Y$ mit $|Y'| = 1\}$

6. $n = n + 1$; Goto 2.

Bestimmung der Assoziationsregeln

Aus einer n-elementigen Item-Menge lassen sich 2^{n-1} verschiedene Assoziationsregeln bilden, die allerdings möglicherweise nicht alle die vorgegebene Mindestkonfidenz haben.

Beispiel: Aus der 3-elementigen Item-Menge

$$X = \{(temperature, cool), (humidity, normal), (play, yes)\}$$

lassen sich die folgenden Assoziationsregeln bilden:

1. $\{(temperature, cool), (humidity, normal)\} \rightarrow \{(play, yes)\}$
2. $\{(temperature, cool), (play, yes)\} \rightarrow \{(humidity, normal)\}$
3. $\{(play, yes), (humidity, normal)\} \rightarrow \{(temperature, cool)\}$
4. $\{(temperature, cool)\} \rightarrow \{(humidity, normal), (play, yes)\}$
5. $\{(humidity, normal)\} \rightarrow \{(temperature, cool), (play, yes)\}$
6. $\{(play, yes)\} \rightarrow \{(temperature, cool), (humidity, normal)\}$

- Aus den häufigen Item-Mengen werden nun die Assoziationsregeln mit einer Konfidenz $\geq k_{\text{min}}$ erzeugt.

- Für zwei Item-Mengen X und Y mit $Y \subset X$ gilt offenbar, falls

$$konfidenz((X \setminus Y) \rightarrow Y) \geq k_{\text{min}}$$

  so gilt für alle $Y' \subset Y$ ebenfalls

$$konfidenz((X \setminus Y') \rightarrow Y') \geq k_{\text{min}}$$

- Zur Regelgenerierung nutzt man die Umkehrung. D.h. man beginnt mit einer möglichst kleinen Item-Menge Y' und schließt dann alle Item-Mengen Y mit $Y' \subset Y$ aus, falls schon gilt:

$$konfidenz((X \setminus Y') \rightarrow Y') < k_{\text{min}}$$

- Man erzeugt aus einer häufigen Item-Menge X alle Regeln mit einer 1-elementigen rechten Seite. Also Regeln der Form $(X \setminus Y) \rightarrow Y$ mit $|Y| = 1$.

- Prüfe von diesen Regeln ob die Konfidenz $\geq k_{\text{min}}$. Diese Regeln werden ausgegeben.

- Sei H_l die Menge der Rechten Seiten von häufigen Item-Mengen mit l Elementen. Erzeuge aus H_l nun $l+1$ elementige Item-Mengen H_{l+1}.

- Für alle Rechten Seiten $h \in H_{l+1}$ prüfe

$$konfidenz((X \setminus h) \rightarrow h) \geq k_{\text{min}}$$

  Falls ja, dann gib die Regel aus, sonst h aus H_{l+1} entfernen.

Klassifikationsregeln

Sonderfall von Assoziationsregeln: Klassifikationsregeln, hier ist ein Merkmal besonders ausgezeichnet (das sogenannte Klassifikationsmerkmal). Nur dieses Merkmal kommt auf der rechten Seite der Regel, d.h. im Regelkopf, vor; im Regelrumpf soll es nicht vorkommen.

- Es soll die Entscheidung getroffen werden. Im Beispiel: Soll ein bestimmtes – nicht näher spezifiziertes – Spiel gespielt werden oder nicht?
- Um die Entscheidung zu automatisieren, sollen möglichst einfache Regeln gefunden werden.
- Es liegen hierzu Beispiele vor, aus denen diese Regeln herleitbar sind.
- Es sollen wieder `IF ... THEN`-Regeln sein, mit den oben genannten Einschränkungen für Position des Klassenattributs.

Klassifikationsregeln aus dem Beispiel

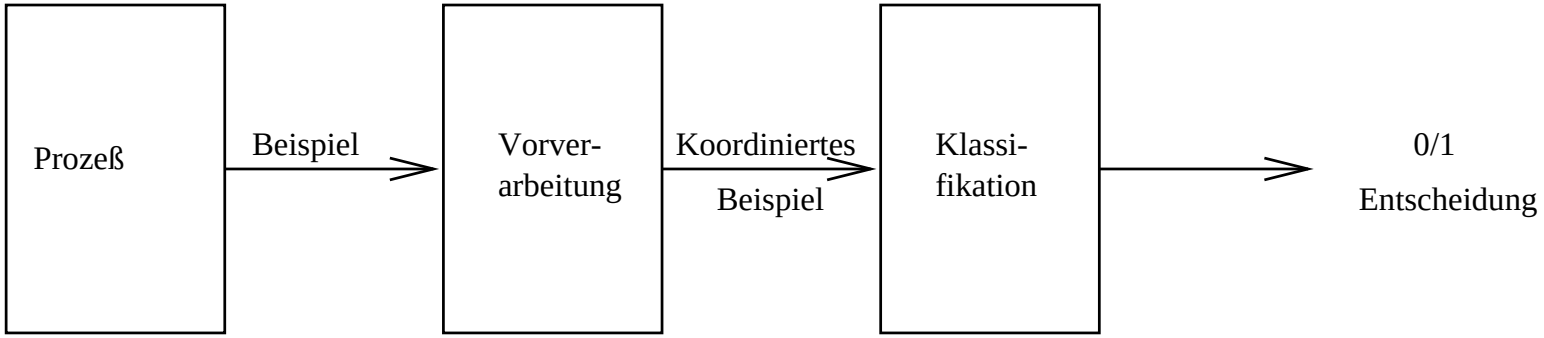
1. `if` outlook $=$ sunny `and` humiditiy $=$ high `then` play $=$ no
2. `if` outlook $=$ rainy `and` windy $=$ true `then` play $=$ no
3. `if` outlook $=$ overcast `then` play $=$ yes
4. `if` humidity $=$ normal `then` play $=$ yes
5. `if` none of the above `then` play $=$ yes

Bei diesem Beispiel gilt sogar: Werden die Regeln sind in der angegebenen Reihenfolge anzuwenden, dann 100% korrekte Entscheidung Die Anwendung einzelne Regeln kann allerdings zu Fehlern führen!

6. Klassifikation

1. Zielsetzung
2. Entscheidungsbäume
3. Prototypbasierte Klassifikation
4. Lineare Klassifikation
5. Bewertung von Klassifikatoren

6.1 Zielsetzung



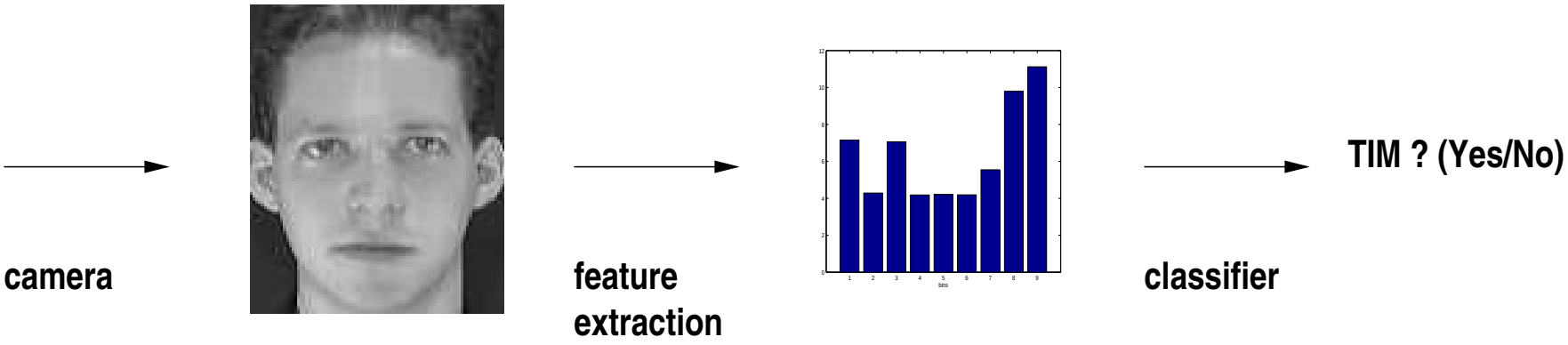
- Wir betrachten nur den Klassifikationsmodul.

- Die gesuchte (aber unbekannte) Klassifikationsabbildung ist von der Form $c : X \to Y$. X Eingabemenge, Y Ausgabemenge.

- Hier ist die Ausgabemenge endlich (nominal) $Y = \{0, 1, \ldots, L\}$, die Namen der Klassen (klassenlabel). Wir beschränken uns häufig auf binäre (d.h. 2-Klassen) Probleme mit $Y = \{0, 1\}$ oder $Y = \{-1, 1\}$.

- Die Klassifikatoreingaben sind reelle oder binäre Vektoren oder Mischformen, d.h. $X \subset \{0,1\}^p$ oder $X \subset \mathbb{R}^p$.

- Beispiel: Verifikation einer Person durch Gesichtserkennung.



- **Überwachtes Lernen der Klassifikationsabbildung**
  Gegeben (endliche) Stichprobe (Trainingsmenge) von Eingabe-Ausgabe-Paaren (x^μ, y^μ) (wobei $y^\mu = c(x^\mu)$) mit dem Ziel eine Klassifikatorabbildung f zu konstruieren, die für jede Eingabe x einen Funktionswert $f(x) = y$ bestimmt, der möglichst gleich $c(x)$ ist.

6.2 Entscheidungsbäume

1. Zielsetzung
2. Breimann’sche Anforderungen für Homogenitätsmaße
3. Beipiele für Homogenitätsmaße
4. Merkmalsauswahl durch Homogenitätsmaximierung
5. Pruning in Entscheidungsbäumen

Zielsetzung

- Entscheidungsbäume sind weit verbreitete Methoden zur Klassifikation von Mustern.
- Entscheidungsbäume sind rekursive Verfahren zur Bestimmung der Klassenzugehörigkeit eines Merkmalsvektors.
- Entscheidungsbaum-Verfahren sind überwachte Lernverfahren.
- **Idee**: Einzelne Merkmale werden getestet. In Abhängigkeit des Testresultats wird ein weiteres Merkmal getestet. Dies wird solange durchgeführt, bis eine hinreichend genaue Klassifikation (hinsichtlich des Klassenmerkmals) getroffen werden kann.

Ausgangslage

- Gegeben n Objekte durch Merkmalsvektoren $x^\mu \in \mathbb{R}^p$ mit zugehörigem Klassenlabel $y^\mu \in \Omega$ gespeichert etwa in $n \times (p+1)$ Datenmatrix X.

- Hierbei sind $x_\mu = (x_{\mu 1}, \ldots, x_{\mu d})$ die eigentliche Merkmale mit denen klassifiert wird; $y_\mu \in \Omega = \{1, 2, \ldots, L\}$ das zugehörige Klassenlabel.

- Die d Merkmale können nominal, ordinal aber auch metrisch skaliert (auch gemischt) sein, da die Merkmale separat behandelt werden.

- **Ziel:** Aufteilung nach einzelnen Merkmalen in möglichst Teilmengen, die möglichst homogen bzgl. des Klassenmerkmals sind

Entscheidungsbaum - Beispiel

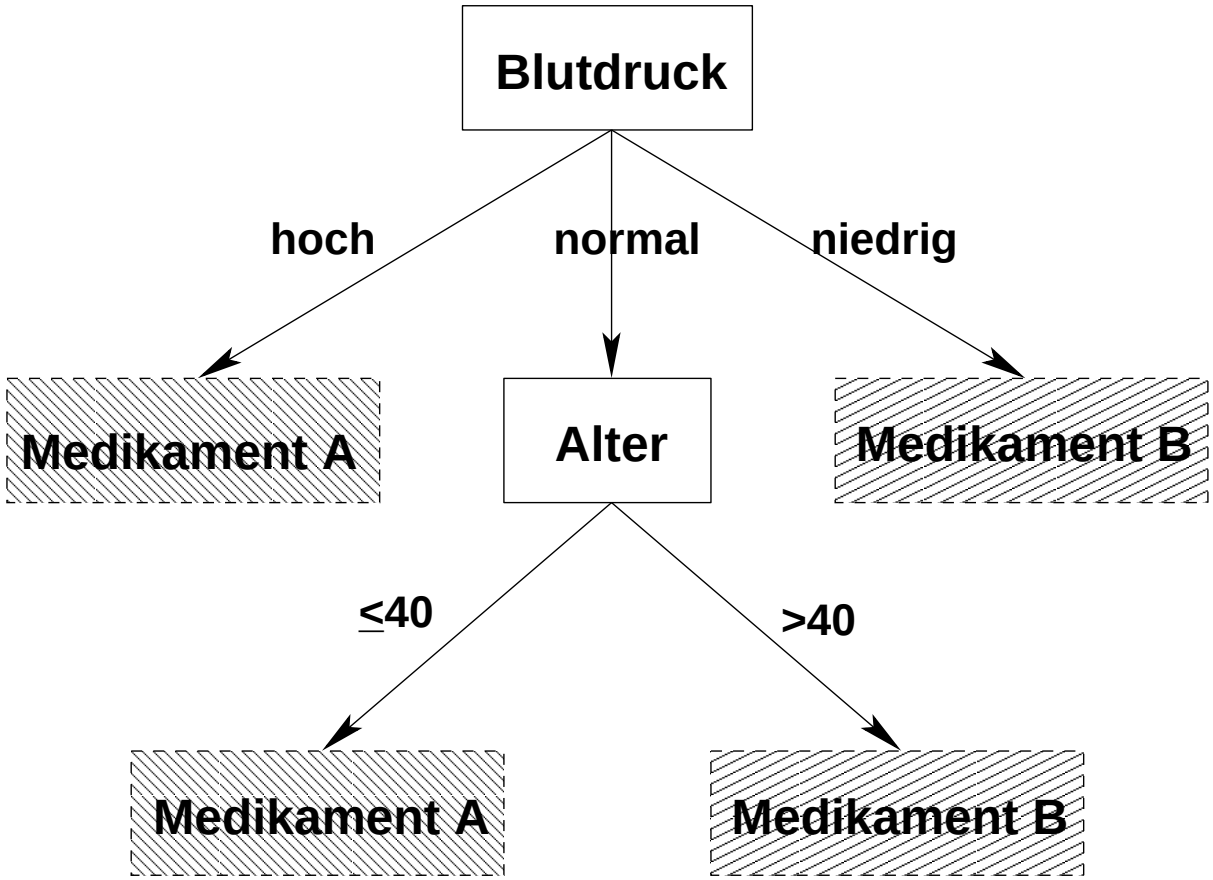
Patientendatenbank zusammen mit Medikament, das wirksam in Bezug auf eine Krankheit ist.

Nr.	Geschlecht	Alter	Blutdruck	Medikament
1	m	20	normal	A
2	w	73	normal	B
3	w	37	hoch	A
4	m	33	niedrig	B
5	w	48	hoch	A
6	m	29	normal	A
7	w	52	normal	B
8	m	42	niedrig	B
9	m	61	normal	B
10	w	30	normal	A
11	w	26	niedrig	B
12	m	54	hoch	A

Entscheidungsbaum - Resultat

Zuerst Aufteilung nach **Blutdruck**, dann nach **Alter** liefert

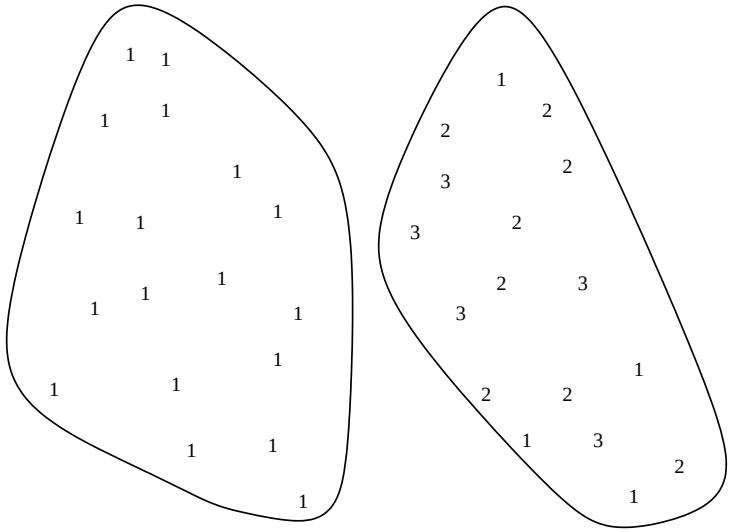
Nr.	Blutdruck	Alter	Medikament
3	hoch	37	A
5	hoch	48	A
12	hoch	54	A
1	normal	20	A
6	normal	29	A
10	normal	30	A
7	normal	52	B
9	normal	61	B
2	normal	73	B
11	niedrig	26	B
4	niedrig	33	B
8	niedrig	42	B



Wie kommt man zu solchen Entscheidungsbäumen?

Inhomogene Teilmengen

Beispiel: $\Omega = \{1, 2, 3\}$ Bewertung der Homogenität von Mengen; homogene Menge (links) nicht homogene Menge (rechts)



Q ein Homogenitätsmaß (auch *impurity measure*) einer Menge R ist von den relativen Häufigkeiten $p_j = |K_j|/|R|$ für $j \in \Omega$ mit $K_j := \{x^\mu \in X \ : \ y^\mu = j\}$ abhängig, also

$$Q(R) = Q(p_1(R), \ldots, p_L(R)) = Q(p_1, \ldots, p_L) = Q(p)$$

Anforderungen/Beispiele für Inhomogenitätsmaße

1. $Q(p)$ ist maximal, gdw. $p_j = 1/L$ für $j = 1, \ldots, L$.

2. $Q(p)$ ist minimal, gdw. $p = e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ ein Einheitsvektor.

3. $Q(p)$ ist symmetrisch, d.h. für eine Permutation $\tau : \{1, \ldots, L\} \to \{1, \ldots, L\}$ gilt $Q(p_1, \ldots, p_L) = Q(p_{\tau(1)}, \ldots, p_{\tau(L)})$

Beispiele:

1. $Q_m(p) := 1 - \max_j p_j$ (Misclassification index)

2. $Q_g(p) := 2 \sum_{i=1}^{L} \sum_{j=1+1}^{L} p_i p_j = 1 - \sum_{i=1}^{L} p_i^2$ (Gini index)

3. $Q_e(p) := -\sum_{j=1}^{L} p_j \log_2 p_j$ (Entropy index)

Satz: Q_m, Q_g und Q_e erfüllen die drei genannten (Breimann'schen) Anforderungen für Homogenitätsmaße Q.

- Bei Entscheidungsbäumen werden Mengenaufteilungen gesucht, so dass ein maximaler Gewinn an Homogenität erzielt wird.

- In den meisten Verfahren werden die Aufteilungen der numerischen Merkmale achsenparallel durchgeführt, d.h. **Aufteilung nach einem Merkmal**, d.h. achsenparallele Aufteilung!

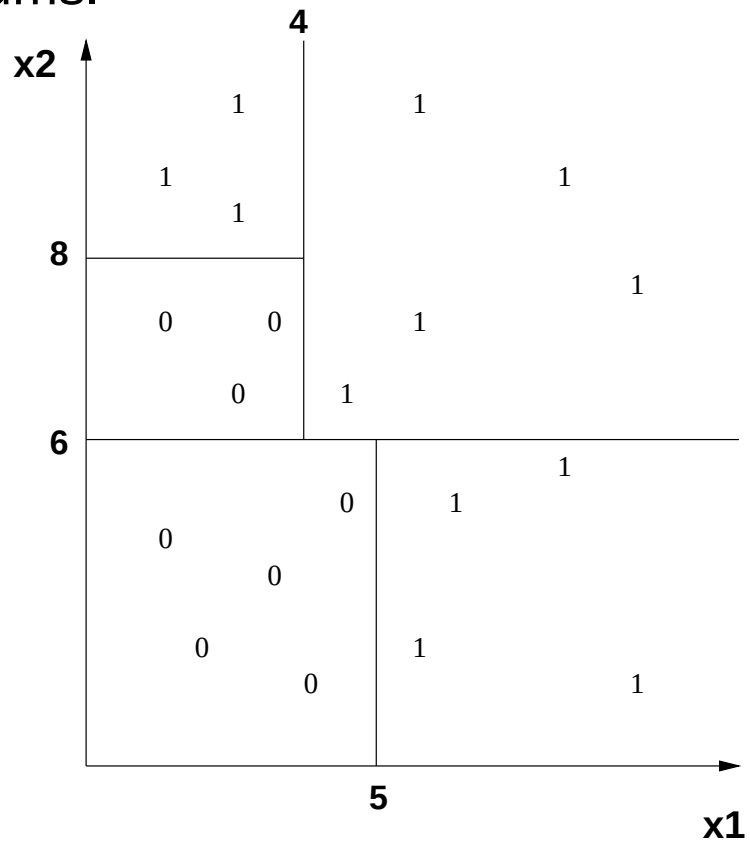
- Aufteilung kann in zwei oder mehr Untermengen erfolgen.

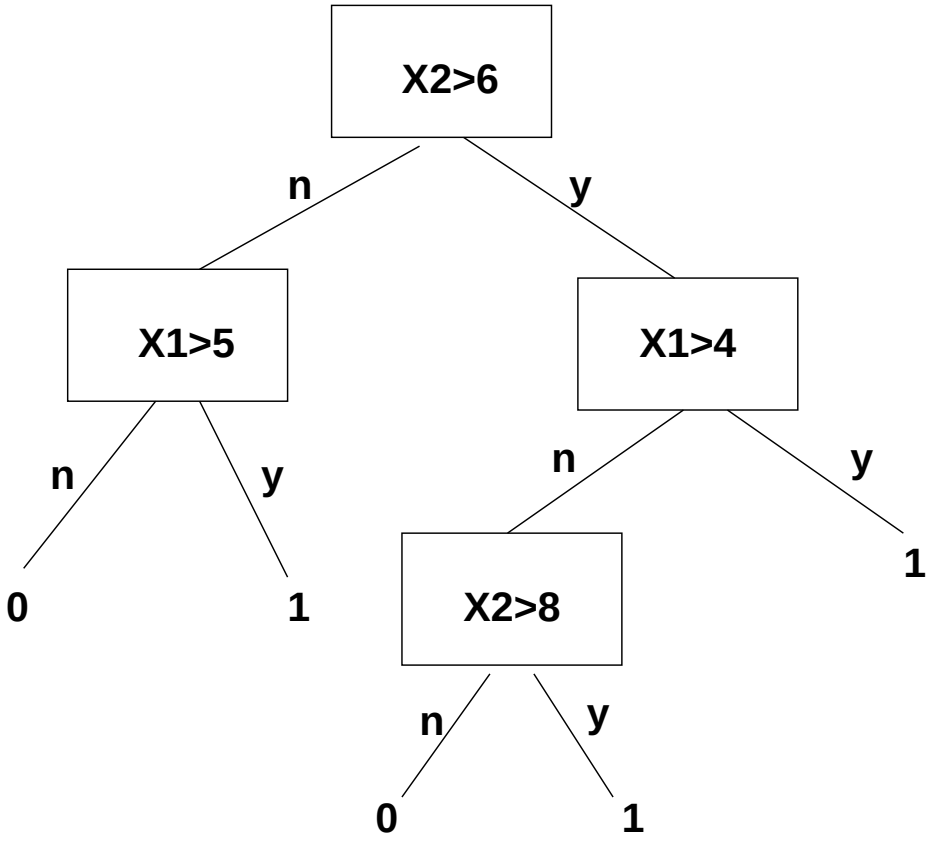
- Bei nicht numerisch skalierten Merkmalen, also nominal oder ordinal skalierten Merkmalen, ist die Anzahl der Unterteilungen meist gleich der Zahl der Merkmalsausprägungen.

- Binäre Aufteilung (bei metrischen Merkmalen) ist am weitesten verbreitet.

Entscheidungsbaum

Aufteilung in achsenparallele Rechtecke. Jedem Rechteck ist schließlich eine der L Klassen zugewiesen, dieses sind die Blattknoten des Entscheidungsbaums.





Maximierung der Homogenität

- Gegeben sei nun ein Knoten des Entscheidungsbaums (kann auch der Wurzelknoten sein), der eine Region im $\mathbb{R}^p$ definiert, und eine Menge $R \subset X$ der Trainingsdaten repräsentiert
  Frage: Wie kann R in (zwei) möglichst homogene $R_r \subset R$ und $R_l \subset R$ mit $R_r \cup \mathbb{R}_l = R$ aufgeteilt werden?

- Für ein (numerisches) Merkmal j und einen Schwellwert θ_j (für dieses Merkmal j) wird die Menge $R \subset X$ in zwei disjunkte Mengen $R_r = \{x^\mu \in R \ : \ x_j^\mu \geq \theta_j\}$ und $R_l = \{x^\mu \in R \ : \ x_j^\mu < \theta_j\} = R \setminus R_r$ zerlegt wird.

  Die Homogenitätsgewinn durch eine solche Aufteilung ist dann:

$$\Delta Q(R, R_l, R_r) := Q(R) - p_{R_l} Q(R_l) - p_{R_r} Q(R_r)$$

  Hierbei ist $p_{R_l} = |R_l|/|R|$ und $p_{R_r} = |R_r|/|R|$

- Beim einer Aufteilung in B Unterregionen, etwa bei B-nären Merkmalsausprägungen, sei $R_1, \ldots R_B$ eine Partition von R mit $p_{R_i} = |R_i|/|R|$.

  Dann ist der Homogenitätsgewinn so definiert:

$$\Delta Q(R, R_1, \ldots, R_B) := Q(R) - \sum_{i=1}^{B} p_{R_i} Q(R_i)$$

- Eine maximal homogene Aufteilung ist erreicht, falls die Datenmengen in den Blättern des Entscheidungsbaumes nur jeweils Datenpunkte einer einzigen Klasse enthalten.

  Im *worst case* ist dann in einem Blatt nur noch genau ein Datenpunkte repräsentiert.

- Bei stark verrauschten Datensätzen führt dies zu einer Überanpassung des Entscheidungsbaumes an die Trainingsdaten (*overfitting*). Dies führt zu einem Entscheidungsbaum mit vielen Knoten, der ggf. einen hohen Klassifikationsfehler auf unbekannten Daten zeigt.

Pruning in Entscheidungsbäumen

- Eine Möglichkeit die Anzahl der Knoten im Entscheidungsbaum zu reduzieren ist das *pruning*.

- Beim sogenannten *post pruning* wird zuerst der Entscheidungsbaum aufgebaut (bis eine gewählte Schranke erreicht wurde).

- Anschließend werden von den Blättern, die Kinder eines Knotens wieder zusammengefasst.

- Dabei verwendet man allerdings ein modifiziertes Güte-Maß, z.B.

$$E_{mod} = E_{emp} + \lambda |T|$$

  hierbei ist E_{emp} der Fehler auf dem Trainingsdatensatz, und $|T|$ ein Maß für die Größe des Entscheidungsbaumes, etwa die Zahl der (Blatt-)Knoten.

6.3 Prototypbasierte Klassifikatoren

Gegeben sei eine Trainings-Menge (x^μ, y^μ), $\mu = 1, \ldots, n$
$x^\mu \in X$ Merkmalsvektoren, $y^\mu \in \Omega$ Klassenlabel.
d sei eine Distanzfunktion auf X.

- Der wichtigste protoypbasierte Klassifikator ist der *k-nearest-neighbour* Klassifikator. $k \geq 1$, typischerweise ist k ungerade

- Es soll der Vektor x klassifiziert werden, dann werden alle Distanzen $d_\mu = d(x, x^\mu)$ bestimmt. und die nächsten k Nachbarn von x unter den x^μ bestimmt.

- Unter diesen k Nachbarn wird nun das Klassenlabel ω ermittelt, das am häufigsten vorkommt.

- ω dient nun als die Klassifikation für x.

- $k-NN$ ist ein sehr einfaches Klassifikationsprinzip. Keine Trainingsphase! Die Suche der k nächsten Nachbarn ist aufwändig.

- Ausweg: LVQ-Training (siehe Kap.3) mit dem Datensatz X. D.h. Reduktion der n Daten auf $m << n$ gelabelter Prototypen $c_1, \ldots, c_m$ und dann $1-NN$ Suche unter den m Protoypen.

6.4 Lineare Klassifikation

1. Lernproblem linearer Klassifikatoren
2. Das Perzeptron
3. Support-Vektor-Lernen
4. SVM-Lernen im lineare nicht separierbaren Fall

Lernproblem

Merkmalsvektor: $x \in \mathbb{R}^p$

Ausgabe des Klassifikators: $z = f(\langle x, w \rangle + w_0)$. Hierbei sei f eine beliebige $\{0, 1\}$-wertige Funktion (z.B. Signum- oder Heaviside-Funktion).

Material zum Training des Klassifikators: $\mathcal{M} = \{(x^\mu, y^\mu) \ : \ \mu = 1, \ldots, n\}$

Gesucht ist $w^* \in \mathbb{R}^{p+1}$, als erweiterter Gewichtsvektor $(w_0, w_1, \ldots, w_p)$ mit $x_0^\mu = 1$ für alle $\mu = 1, \ldots, n$, so dass

$$E(w^*) \rightarrow \text{minimal}$$

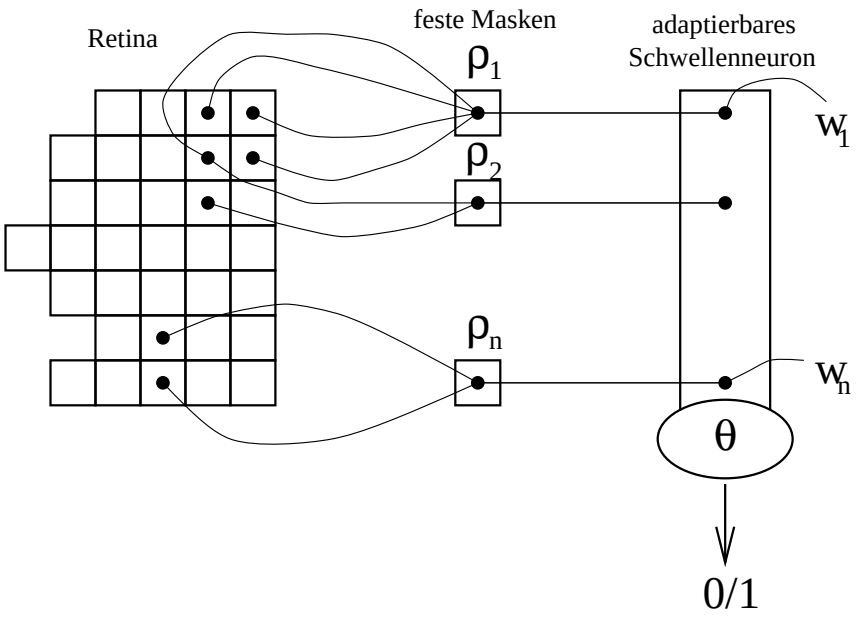
für eine a priori definierte Fehlerfunktionen $E : \mathbb{R}^{p+1} \rightarrow \mathbb{R}$.

Wir fassen im Folgenden w und x als Vektoren des $\mathbb{R}^{p+1}$ auf. Wobei wir eben die Eingabevektoren erweitern um $x_0^\mu = 1$.

Damit hat die Ausgabe einfach die Form $z = f(\langle w, x \rangle)$.

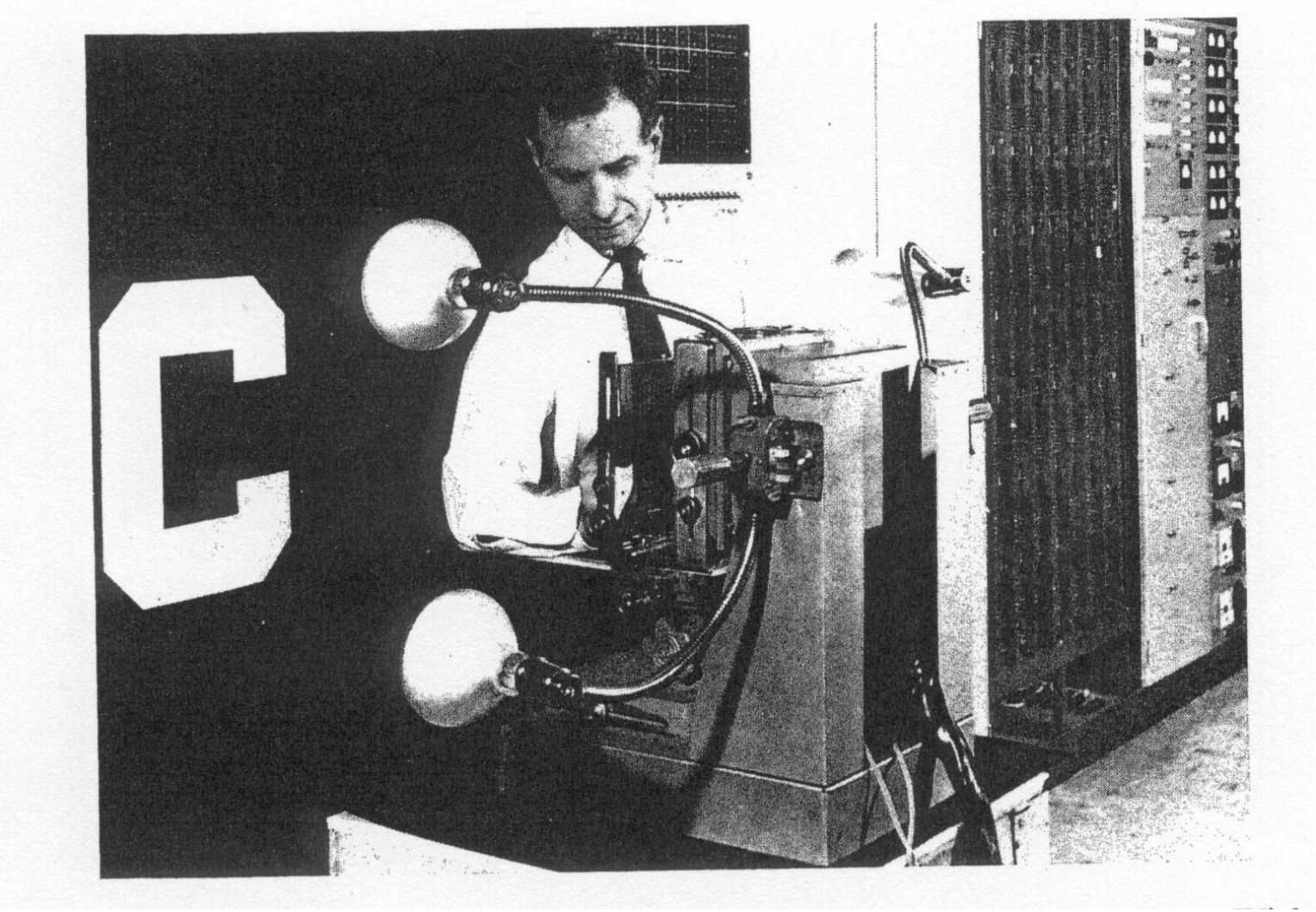
Das Perzeptron

- Eingabe- oder Sensorschicht (häufig auch *Retina* genannt)
- Masken mit festen Kopplungen zur Sensorschicht
- Schwellenneuron mit adaptierbaren Gewichten w und Schwellwert θ
- $z = 1$ falls $\langle w, x \rangle \geq \theta$ gilt und sonst $z = 0$

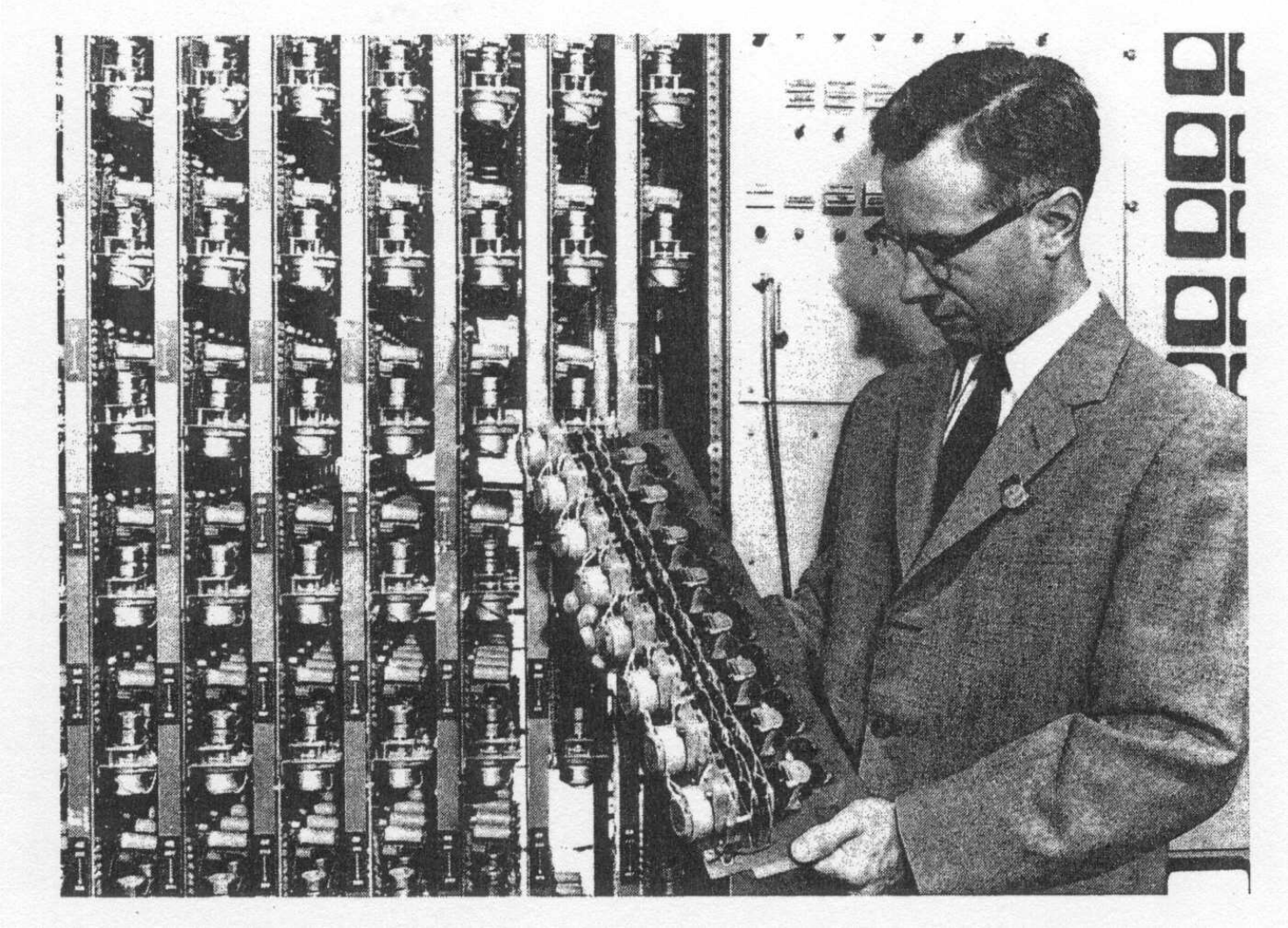


Frank Rosenblatt—Der Erfinder

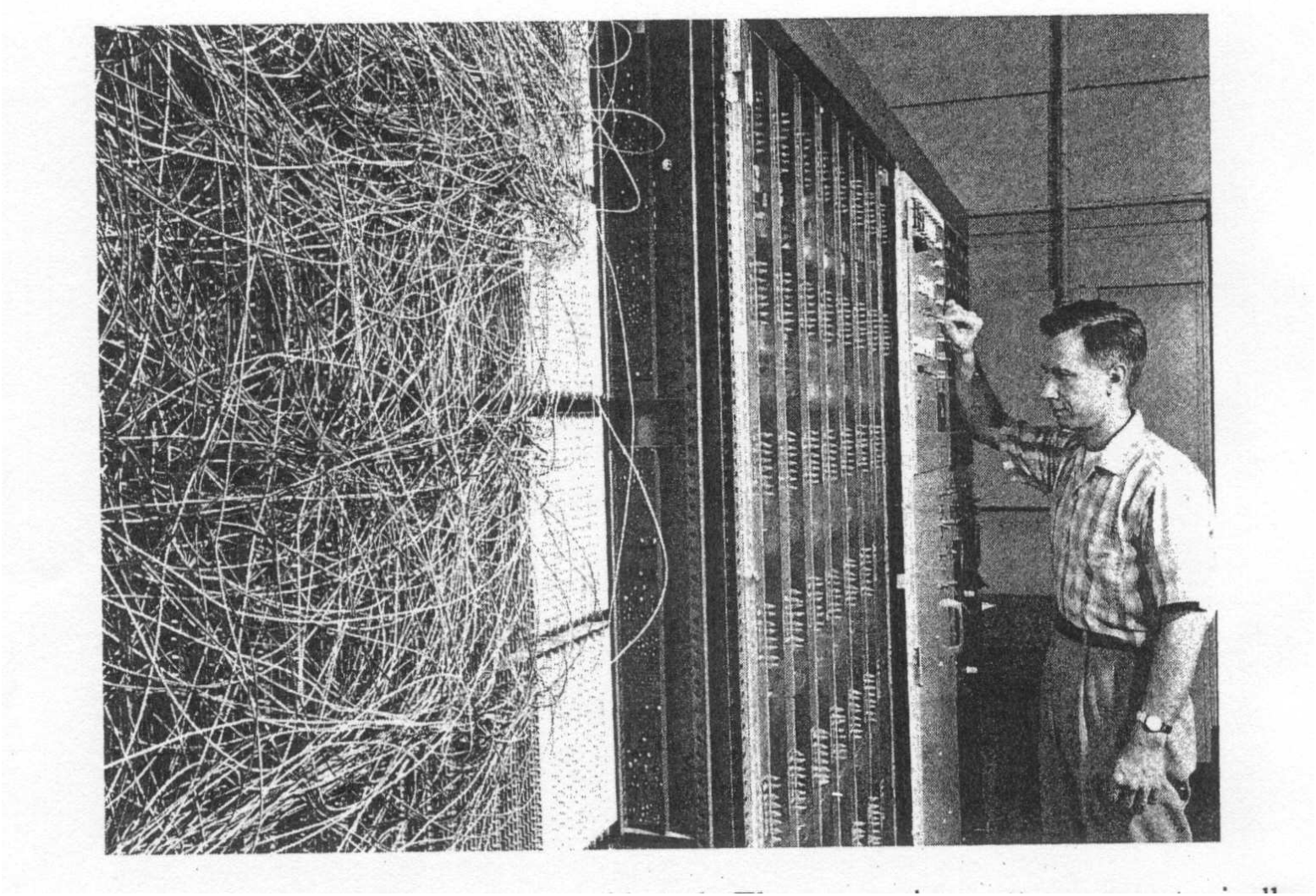
Perceptron - Retina



Perceptron - Adaptable Weights



Perceptron - Random Connections



Perzeptron Lernalgorithmus

Input: (x^μ, y^μ), $\mu = 1, \ldots, n$. $x^\mu \in \mathbb{R}^{p+1}$ (erweiterter Vektor), $y \in \{-1, 1\}$

$w = 0 \in \mathbb{R}^{p+1}$

Repeat

$L = 0$

For $\mu = 1$ To n

$\delta = (y^\mu - \text{sign}\langle x^\mu, w\rangle)$

If $\delta \neq 0$ **Then**

$L := L + 1;$

$w := w + \delta x^\mu;$

Until $L = 0$

Output: $w \in \mathbb{R}^{p+1}$

Perzeptron Lerntheorem

Lernregel:

$$\Delta w = l\,(y - z) \cdot x \text{ mit Lehrersignal } y \in \{1, -1\} \tag{3}$$

andere Schreibweise der Lernregel:

$$\Delta w = -l\,\text{sign}(x \cdot w) \cdot x = l\,y \cdot x \quad \text{falls } z \neq y \text{ (Änderungsschritt)} \tag{4}$$

Zu bestimmen: $S =$ Anzahl der Änderungsschritte

Problem lösbar, falls $\exists\, w$ mit $\text{sign}(x^\mu \cdot w) = y^\mu\ \forall \mu$,
d.h. $y^\mu\,(x^\mu \cdot w) > 0\ \forall \mu$, d.h. $D(w) := \min_{\mu=1}^{n} y^\mu\,(x^\mu \cdot w) > 0$.

$D(w)$ nimmt auf der Einheitskugel $K = \{w\ :\ w \cdot w = 1\}$ das Maximum d an.
Also gibt es w^* mit $w^* \cdot w^* = 1$ und $D(w^*) = d$.
Problem lösbar, falls $d > 0$. Sei nun $c := \max_{\mu=1}^{n}(x^\mu \cdot x^\mu)$.

Betrachte das Gewicht w_S nach S Änderungsschritten: $w_S = \sum_{i=1}^{S}(\Delta w)_i$.

Dann gilt:

$$(\Delta w) \cdot w^* \stackrel{(2)}{=} l\, y^\mu\, (x^\mu \cdot w^*) \geq l\, D(w^*) = l\, d \tag{5}$$

$$\begin{aligned} (w + \Delta w) \cdot (w + \Delta w) - w \cdot w &= 2\,((\Delta w) \cdot w) + (\Delta w) \cdot (\Delta w) \\ &\stackrel{(2)}{=} -2\, l\, \mathrm{sign}(x^\mu \cdot w)\ (x^\mu \cdot w) + l^2\, (x^\mu \cdot u^\mu) \\ &\leq l^2\, (x^\mu \cdot x^\mu) \leq l^2\, c \end{aligned} \tag{6}$$

Also gilt: $w_S \cdot w_S \stackrel{(4)}{\leq} S l^2 c$ und $w_S \cdot w^* \stackrel{(3)}{\geq} S l d$. Daraus folgt:

$$S\, l\, d \stackrel{(3)}{\leq} w_S \cdot w^* \leq \sqrt{(w_S \cdot w_S)(w^* \cdot w^*)} = \sqrt{w_S \cdot w_S} \leq \sqrt{S\, l^2\, c} \quad \Longrightarrow S \leq c/d^2$$

Support Vektor Lernen

Ist zunächst einmal eine spezielle Form des Perzeptron-Lernverfahrens.

Lernverfahren entsteht durch eine Kombination von 2 Zielen, diese legen im Fall linear separierbarer Mengen eine eindeutige Trennhyperebene fest.

Wieder gegeben Trainingsdaten

$$\mathcal{M} = \{(x^\mu, y^\mu) \ : \ \mu = 1, \ldots, n\} \subset \mathbb{R}^d \times \{-1, 1\}$$

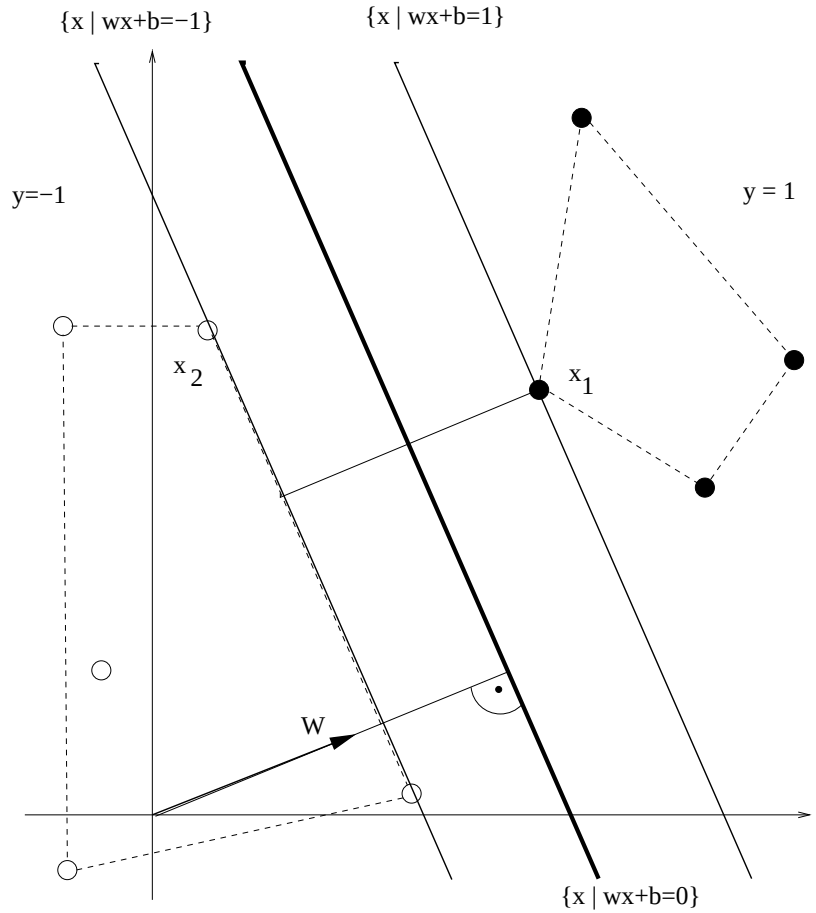
Wir nehmen zunächst an, die Mengen

$$P = \{x^\mu \mid y^\mu = 1\} \quad \text{und} \quad N = \{x^\mu \mid y^\mu = -1\}$$

seien linear separierbar.

Perzeptron-Lerntheorem sichert die Konverganz gegen eine Lösung w .

Support Vektor Lernen - Das Bild



$$\langle x_1, w \rangle + w_0 = 1$$

$$\langle x_2, w \rangle + w_0 = -1$$

$$\Longrightarrow$$

$$\langle x1 - x2, w \rangle = 2$$

$$\Longrightarrow$$

$$\langle x1 - x2, \frac{w}{\|w\|} \rangle = \frac{2}{\|w\|}$$

Support Vektor Lernen - Die Formeln

Wir suchen nun nach einer Lösung $w \in \mathbb{R}^p$ und $w_0 \in \mathbb{R}$,

- Die Separationsbedingungen

$$y^\mu(\langle w, x^\mu \rangle + w_0) > 0 \quad \text{für alle } \mu = 1, \ldots, n$$

  erfüllt, und

- möglichst weit von den Mengen N und P entfernt ist (*maximal margin*)

Es sein

$$\min_\mu \quad y^\mu(\langle w, x^\mu \rangle + w_0) = \delta > 0$$

Nun reskalieren wir und erhalten mit $w := \frac{1}{\delta} w$ und $w_0 := \frac{1}{\delta} w_0$

$$y^\mu(\langle w, x^\mu \rangle + w_0) \geq 1 \quad \text{für alle } \mu = 1, \ldots, n$$

Offenbar gibt es mindestens einen Punkt $x^\nu \in P$ und $x^\mu \in N$ mit

$$\langle w, x^\nu \rangle + w_0 = 1$$

und mit

$$\langle w, x^\mu \rangle + w_0 = -1$$

Daraus folgt $\langle w, x^\nu - x^\mu \rangle = 2$ und damit ist $D(w)$ die Breite des Randes der separierenden Hyperebene gegeben durch

$$D(w) = \langle \frac{w}{\|w\|_2}, (x^\nu - x^\mu) \rangle = \frac{2}{\|w\|_2}$$

Also Maximierung des Randes bedeutet Minimierung der Norm, etwa

$$\varphi(w) = \frac{\|w\|_2^2}{2} \to \min$$

unter den n Nebenbedingungen (eine für jeden Datenpunkt)

$$y^\mu(\langle w, x^\mu \rangle + w_0) \geq 1 \quad \text{für alle}$$

Dies ist ein quadratisches Optimierungsproblem unter Nebenbedingungen.

Mit der Einführung von n sogenannten Lagrange-Multiplikatoren $\alpha_\mu \geq 0$ (eine für jede Nebenbedingung) wird es in folgendes Optimierungsproblem überführt:

$$L(w, w_0, \alpha) = \frac{\|w\|_2^2}{2} - \sum_{\mu=1}^{n} \alpha_\mu \left(y^\mu(\langle w, x^\mu \rangle + w_0) - 1\right)$$

Setzt man nun für die partiellen Ableitungen $\frac{\partial L}{\partial w} = 0$ und $\frac{\partial L}{\partial w_0} = 0$, so erhält man die Bedingungen

$$\sum_{\mu=1}^{n} \alpha_\mu y^\mu = 0 \quad \text{und} \quad w = \sum_{\mu=1}^{n} \alpha_\mu y^\mu x^\mu$$

Außerdem folgt aus der Optimierungstheorie (*Kuhn-Tucker-Bedingungen*):

$$\alpha_\mu \left[y^\mu \left(\langle w, x^\mu \rangle + w_0 \right) - 1 \right] = 0 \quad \text{für alle } \mu = 1, \ldots, n$$

Falls nun $\alpha_\mu \neq 0$ so folgt: $y^\mu \left(\langle w, x^\mu \rangle + w_0 \right) = 1$, d.h. x^μ liegt genau auf dem Rand.

Diese Vektoren heißen auch **Support Vektoren**, daher der Name des Lernverfahrens.

Offensichtlich ist w eine Linearkombination der Support Vektoren (SV):

$$w = \sum_{x^\mu \in SV} \alpha_\mu y^\mu x^\mu$$

Zwischenresultate dann in L einsetzen, so erhält man

$$W(\alpha) = \sum_{\mu=1}^{n} \alpha_\mu - \frac{1}{2} \sum_{\nu=1}^{n} \sum_{\mu=1}^{n} \alpha_\nu \alpha_\mu y^\nu y^\mu \langle x^\nu, x^\mu \rangle$$

das mit $\alpha_\mu \geq 0$ für alle $\mu = 1, \ldots, n$ zu maximieren ist.

Dieses Optimierungsproblem kann mit Standardmethoden gelöst werden und liefert$\alpha^* \in \mathbb{R}^n$.

Mit der Lösung α^* steht nun auch die Trennhyperebene fest:

$$w = \sum_{\mu=1}^{n} \alpha_\mu^* y^\mu x^\mu$$

Die Schwelle $w_0 \overset{\in}{\mathbb{R}}$ läßt sich mit Hilfe eines Support-Vektors x^{μ_0} bestimmen. Denn es gilt $\alpha_{\mu_0} = 0$ und damit

$$y^{\mu_0} \left(\langle w, x^{\mu_0} \rangle + w_0 \right) = 1$$

Hieraus folgt sofort

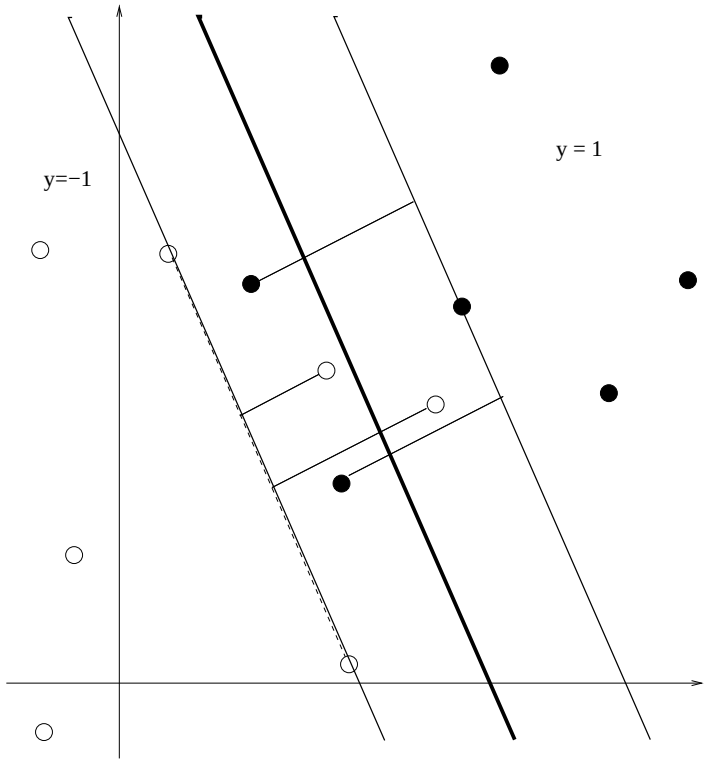
$$w_0^* = \frac{1}{y^{\mu_0}} - \langle w, x^{\mu_0} \rangle$$

damit liegt die die Entscheidungsfunktion fest:

$$f(x) = \text{sig}\left(\sum_{x^\mu \in SV} \alpha_\mu^* y^\mu \langle x^\mu, x\rangle + w_0^*\right).$$

Nicht separierbares Problem

$P = \{x^\mu \mid y^\mu = 1\}$ und $N = \{x^\mu \mid y^\mu = -1\}$ seien nun linear nicht separierbare Mengen:



Soft-Separationsbedingungen durch Schlupfvariable $\delta_\mu \geq 0$ (*slack variables*)

$$y^\mu \left(\langle w, x^\mu \rangle + w_0 \right) \geq 1 - \delta_\mu \quad \text{für alle } \mu = 1, \ldots, n$$

Nun minimieren wir mit $C > 0$

$$\varphi(w, \delta) = \frac{1}{2} \|w\|_2^2 + \frac{C}{n} \sum_{\mu=1}^{n} \delta_\mu$$

Dies führt wiederum auf die quadratische Funktion

$$W(\alpha) = \sum_{\mu=1}^{n} \alpha_\mu - \frac{1}{2} \sum_{\nu=1}^{n} \sum_{\mu=1}^{n} \alpha_\nu \alpha_\mu y^\nu y^\mu \langle x^\nu, x^\mu \rangle$$

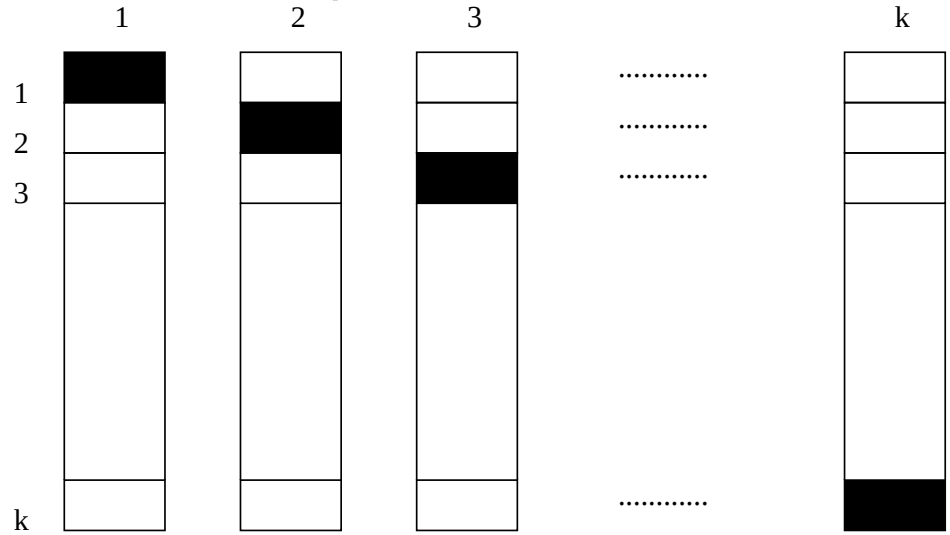
die mit $0 \leq \alpha_\mu \leq C/n$ für alle $\mu = 1, \ldots, n$ zu maximieren ist.

Bewertung von Klassifikatoren

- Nachdem ein Klassifikator berechnet (trainiert) wurde, sind wir daran interessiert, seine Generalisierungsfähigkeit zu überprüfen.
- Die Bewertung der Generalisierungsleistung sollte nicht mit dem Trainingsmaterial durchgeführt werden, da der Generalisierungsfehler auf dem Trainingsdatensatz unterschätzt wird.
- Beispiel: k-NN Klassifikator auf dem Trainingsdaten (etwa für $k = 1$).
- **Hold-Out-Methode** (HO)
  Die zur Verfügung stehende Datenmenge $\mathcal{X}$ wird in 2 disjunkte Mengen aufgeteilt: Traingsmenge $\mathcal{T}$ und Testmenge $\mathcal{V}$. Der Trainingsprozess wird mit $\mathcal{T}$ durchgeführt und der Generalisierungsfehler wird durch Testen des Klassifikators mit Daten aus $\mathcal{V}$ geschätzt.

  Problem: Für kleinere Datensätze ist die Hold-Out-Methode nicht durchführbar.

- **Cross-Validation-Methode** (CV)
  Genauer k-fach CV mit $2 \leq k \leq N$ wobei $N = |\mathcal{X}|$. Hierbei wird nur $\mathcal{X}$ in k disjunkte Mengen $\mathcal{X}_1, \ldots, \mathcal{X}_k$ partitioniert.



Es werden dann k Klassifikatoren erzeugt, wobei beim i-ten Klassifikatortraining die Datenmenge $\mathcal{X}_i$ nicht verwendet wird , sondern zum Test des Klassifikators eingesetzt wird.

Die Anzahl der Fehler wird für die Datenmengen $\mathcal{X}_i$ ermittelt und zum Gesamtfehler kummuliert. Hieraus ergibt sich nun eine Schätzung für den Generalisierungsfehler der Klassifikatorarchitektur.

7. Prognose (Regression)

1. Zielsetzung
2. Lineare Regression
3. Nichtlineare Regression
4. Bewertung

7.1 Zielsetzung

- Die gesuchte (aber unbekannte) Klassifikationsabbildung ist von der Form $c : X \to Y$ ist nur auf einer endlichen Menge von Datenpunkten bekannt.
- Bei der Prognose ist die Ausgabemenge kontinuierlich, also $Y = \mathbb{R}^m$.
- Die Eingabemenge ist ebenfalls, also $X = \mathbb{R}^n$.
- Beispiel: Prognose der Auslastung in einem Netzwerk für die nächste Zeiteinheit.
- **Überwachtes Lernen der Regressionsabbildung**
  Gegeben eine (endliche) Stichprobe von Eingabe-Ausgabe-Paaren (x^μ, y^μ) (Trainingsmenge) mit dem Ziel eine Funktion f zu lernen, die für jede (unbekannte) Eingabe x einen Funktionswert y bestimmt (möglichst $= c(x)$).

  Hierbei stammt f aus einer vorgegebenen Menge $\mathcal{F}$, z.B. dem Raum der Polynome vom Grad ≤ 2 auf einem Intervall $[a, b]$.

7.2 Lineare Regression

Zunächst betrachten wir das Problem für Funktionen $c : \mathbb{R} \to \mathbb{R}$.

Material zum Training: $\mathcal{M} = \{(x^\mu, y^\mu) \quad : \quad \mu = 1, \ldots, n\}$, d.h. die gesuchte Funktion c ist nur für $x^1, \ldots x^n$ bekannt, also $c(x^\mu) = y^\mu$.

Gesucht ist nun eine lineare Funktion $f(x) := ax + b$, mit $a, b \in \mathbb{R}$ so dass

$$E(a, b) = \sum_{\mu=1}^{n} (ax^\mu + b - y^\mu)^2 \to \min$$

für die festgelegte quadratische Fehlerfunktionen (*Methode der kleinsten Quadrate*).

Berechnung der Lösung

Ableitungen berechnen:

$$\frac{\partial}{\partial a}E = \sum_{\mu=1}^{n}(ax^{\mu} + b - y^{\mu})x^{\mu} = 0 \tag{7}$$

$$\frac{\partial}{\partial b}E = \sum_{\mu=1}^{n}(ax^{\mu} + b - y^{\mu}) = 0 \tag{8}$$

Aus der letzten Gleichung folgt, $nb = \sum_{\mu=1}^{n}(y^{\mu} - ax^{\mu})$ und damit:

$$b = \bar{y} - a\bar{x} \tag{9}$$

hierbei sind $\bar{y}, \bar{x}$ die Mittelwerte von y^{μ} bzw. x^{μ}.

Nun (9) in (7) einsetzen :

$$\sum_{\mu=1}^{n}(ax^{\mu}+(\bar{y}-a\bar{x})-y^{\mu})x^{\mu}=0$$

Und damit folgt:

$$a\sum_{\mu=1}^{n}(x^{\mu}-\bar{x})x^{\mu}=\sum_{\mu=1}^{n}(y^{\mu}-\bar{y})x^{\mu}$$

Ausmultiplizieren liefert dann das Ergebnis:

$$a=\frac{\sum_{\mu=1}^{n}x^{\mu}y^{\mu}-n\bar{x}\bar{y}}{\sum_{\mu=1}^{n}(x^{\mu})^{2}-n\bar{x}^{2}}=\frac{s_{xy}}{s_{x}^{2}}$$

und damit ist

$$b=\bar{y}-\frac{s_{xy}}{s_{x}^{2}}\bar{x}$$

Damit sind a und b bestimmt, da die 2. Ableitungen positiv sind, handelt es sich hierbei um ein Minimum und nicht um ein Maximum.

Offenbar muss gelten $s_x^2 > 0$ damit a definiert ist, d.h. die x^μ dürfen nicht alle gleich sein, die Forderung stellt also keine besondere Einschränkung dar.

Sei nun $f(x) = ax + b$ mit a, b wie hergeleitet, dann gilt

$$f(\bar{x}) = \frac{s_{xy}}{s_x^2}\bar{x} + \bar{y} - \frac{s_{xy}}{s_x^2}\bar{x} = \bar{y}$$

D.h. f geht durch den Schwerpunkt $(\bar{x}, \bar{y})$ der Daten (x^μ, y^μ)

Pseudo-Inversen-Lösung

Wir betrachten nun Funktionen mit p Eingaben und mit m Ausgaben, also $f : \mathbb{R}^p \to \mathbb{R}^m$, dann ist $f = (f_1, \ldots, f_m)$ mit $f_j : \mathbb{R}^p \to \mathbb{R}$.

Die Koordinatenfunktionen f_j sollen wieder lineare Funktionen sein, also

$$z_j = \sum_{i=1}^{p} x_i w_{ij}$$

(die Konstante lassen wir weg (und fassen die Vektoren als erweitert auf)).

Als Fehlerfunktion setzen wir an:

$$E(w) = \sum_{\mu} \|y^\mu - z^\mu\|_2^2 = \sum_{\mu} \sum_{j=1}^{m} (y_j^\mu - z_j^\mu)^2$$

dabei ist z^μ die Ausgabe für Input x^μ.

Wir setzen $m = 1$. Damit hat die Fehlerfunktion die Form

$$E(w) = \sum_{\mu} (y^{\mu} - z^{\mu})^2 = \sum_{\mu} (y^{\mu} - \sum_{i=1}^{d} x_i^{\mu} w_i)^2 \to \min$$

Wir definieren die $n \times 1$ Matrix $Y = (y^{\mu})_{1 \leq \mu \leq n}$ und die $n \times p$ Matrix $X = (X_i^{\mu})_{\substack{1 \leq \mu \leq n \\ 1 \leq i \leq p}}$, dann können wir schreiben

$$E(w) = \|Y - Xw\|_2^2 \to \min$$

Falls nun X invertierbar ist, so folgt einfach

$$w = X^{-1} Y$$

als Lösung für w (sogar mit Fehler $= 0$). Diese Lösung ist nur für $n = p$ überhaupt möglich (eine nicht sonderlich interessante Lernaufgabe)

Falls wir nun $E(w)$ minimieren wollen, so können wir uns an folgende Bedingung erinnern:

Eine notwendige Bedingung für ein Optimum von E ist

$$\frac{\partial}{\partial w_k} E = 0 \quad \text{für alle } k = 1, \ldots, p$$

Es folgt sofort:

$$\frac{\partial}{\partial w_k} E = -2 \sum_{\mu} (y^{\mu} - \sum_{i=1}^{p} x_i^{\mu} w_i) x_k^{\mu} \quad \text{für alle } k = 1, \ldots, p$$

Somit folgt

$$\sum_{\mu} (y^{\mu} - \sum_{i=1}^{p} x_i^{\mu} w_i) x_k^{\mu} = 0 \quad \text{für alle } k = 1, \ldots, p$$

Hieraus folgt weiter

$$\sum_{\mu} x_k^{\mu} \sum_{i=1}^{p} x_i^{\mu} w_i = \sum_{\mu} x_k^{\mu} y^{\mu} \quad \text{für alle } k = 1, \ldots, p$$

Mit den oben definierten Matrizen folgt die Gleichung

$$X^t X w = X^t Y$$

Falls nun die symmetrische $p \times p$ Matrix $X^t X$ invertierbar ist, so ist die Lösung des quadratischen Fehlers

$$w = (X^t X)^{-1} X^t Y$$

Die Matrix $(X^t X)^{-1} X^t$ heißt die *Pseudoinverse* von X. Also gilt

$$w = X^{+} Y$$

ist Lösung der Minimierungsaufgabe

$$E(w) = \|Y - Xw\|_2^2 \to \min$$

Die Invertierbarkeit der Matrix $X^t X$ ist sicher, falls es p linear unabhängige Vektoren in der Menge der Eingabevektoren x^μ, $\mu = 1, \ldots, n$ gibt.

Falls nun $X^t X$ nicht invertierbar sein sollte, dann gibt es mehrere Lösung der Minimierungsaufgabe $E(w) = \|Y - Xw\|_2^2 \to \min$.

Die Eindeutigkeit durch eine veränderte Fehlerfunktion

$$E(w) = \|Y - Xw\|_2^2 + \alpha^2 \|w\|_2^2 \to \min$$

mit $\alpha > 0$.

Dann folgt offenbar

$$\frac{\partial E}{\partial w_k} = -2\sum_{\mu}(y^{\mu} - \sum_{i=1}^{p} x_i^{\mu} w_i) x_k^{\mu} + 2\alpha^2 w_k \quad \text{für alle } k = 1, \ldots, d$$

und

$$\sum_{\mu} x_k^{\mu} \sum_{i=1}^{d} x_i^{\mu} w_i + \alpha^2 w_k = \sum_{\mu} x_k^{\mu} y^{\mu} \quad \text{für alle } k = 1, \ldots, p$$

also in Matrixform Mit den oben definierten Matrizen folgt die Gleichung

$$(X^t X + \alpha^2 I) w = X^t Y$$

Für $\alpha \neq 0$ ist $X^t X + \alpha^2 I$ invertierbar (sogar positiv definit) und es gilt

$$w = (X^t X + \alpha^2 I)^{-1} X^t Y$$

Pseudoinversen-Lösung

1. Für eine beliebige Matrix X existiert die Pseudoinverse X^+.

$$X^+ = \lim_{\alpha \to 0} (X^t X + \alpha^2 I)^{-1} X^t$$

2. Falls $X^t X$ invertierbar ist, so gilt

$$X^+ = (X^t X)^{-1} X^t$$

3. Falls sogar X invertierbar ist, so gilt

$$X^+ = X^{-1}$$

4. In jedem Fall ist $w = X^+ Y$ Lösung der Minimierungsaufgabe

$$E(w) = \|Y - Xw\|_2^2 \to \min$$

7.3 Lineare Regression mit festgewählten Basisfunktionen

- Datenmenge $(x^\mu, y^\mu)_{\mu=1}^n$ mit $x^\mu \in \mathbb{R}^p$ und $y^\mu \in \mathbb{R}$.
- Gesucht ist eine Funktion $f : \mathbb{R}^p \to \mathbb{R}$, die die Datenpunkte möglichst gut approximiert (quadratischer Fehler).
- Haben linearen Ansatz $f(x) = \langle x, w \rangle$ durchgeführt und $w \in \mathbb{R}^p$ bestimmt.
- Ansatz übertragbar wenn N festgelegte Funktionen $h_i : \mathbb{R}^p \to \mathbb{R}$ existieren.
- Definieren $n \times N$ Matrix H durch $H_{\mu i} := h_i(x^\mu)$ für $\mu = 1, \ldots, n$ und $i = 1, \ldots, N$
- Nun verfolgen wir den Ansatz

$$f(x) = \sum_{i=1}^{N} w_i h_i(x) = \langle h(x), w \rangle \quad \text{mit } h(x) = (h_1(x), \ldots, h_N(x)).$$

- Die Lösung $w \in \mathbb{R}^N$ ist dann (wie in 7.2): $w = H^+ Y$

Einige Beispiele

1. Für beliebiges $p \geq 1$ setze $N = p+1$ und $h_i(x) := x_i$ für $i = 0, 1, \ldots, N$ mit der Konvention $h_0(x) := 1$. So ergibt sich der bisher diskutierte Fall als ein Spezialfall der Regression mit Basisfunktionen.
2. Sei $p = 1$ und $N \geq p$ beliebig. Dann definiert man $h_i(x) := x^i$ für $i = 0, 1, \ldots, N$, die Polynome vom Grad $\leq N$.
3. Radiale Basisfunktionen mit festen Stützstellen $c_1, \ldots, c_N \in \mathbb{R}^p$

$$h_i(x) = exp(-\frac{\|x - c_i\|^2}{2\sigma^2}) \quad i = 1, \ldots, N \quad h_0(x) = 1$$

4. Multilayer Perzeptrone mit festen Rampen $c_1, \ldots, c_N \in \mathbb{R}^p$

$$h_i(x) = \frac{1}{1 + exp(-\langle c_i, x \rangle)} \quad i = 1, \ldots, N \quad h_0(x) = 1$$

7.4 Nichtlineare Regression

- Situation wie eben: Endliche Datenmenge $(x^\mu, y^\mu)_{\mu=1}^n$ mit $x^\mu \in \mathbb{R}^p$ und $y^\mu \in \mathbb{R}$ ist gegeben. Gesucht ist eine Funktion $f : \mathbb{R}^p \to \mathbb{R}$, die die Daten im Sinne der quadratischen Fehlernorm möglichst gut approximiert.
- Nun setzen wir voraus, dass es N *parametrisierte* Funktionen $h_{c_i} : \mathbb{R}^p \to \mathbb{R}$ gibt. Nun verfolgen wir den Ansatz

$$f(x) = \sum_{i=1}^{N} w_i h_{c_i}(x)$$

zur Minimierung von

$$E(w_i, c_i) = \sum_{\mu} (y^\mu - \sum_{i=1}^{N} w_i h_{c_i}(x^\mu))^2$$

- Im Gegensatz zu den festgewählten Basisfunktionen h_i sind die Basisfunktionen h_{c_i} frei parametrisiert. D.h. es können sowohl die Koeffizienten w_i, als auch die Parameter c_i durch ein Optimierungsverfahren angepasst werden.

- Hier gibt es keine analytische Lösung mehr! Optimierung z.B. durch Gradientenabstieg:

$$\Delta w = -l\frac{\partial}{\partial w}E$$

und für die Parametervektoren

$$\Delta c_i = -l\frac{\partial}{\partial c_i}E$$

7.5 Regressionsbasierte Klassifikation

- Eingabemenge der $\mathbb{R}^p$; Ausgabemenge ist endlich $\Omega = \{1, \ldots, L\}$
- Klassenlabel i für $1 \leq i \leq L$ werden kodiert durch die Einheitsvektoren e_i.
- D.h. ist das Klassenlabel $y^\mu = i \in \Omega$ so setzen wir $y^\mu = e_i \in \{0,1\}^L$ als Sollausgabe an.
- Hierfür ist nun eine Abbildung $c : \mathbb{R}^p \to \{0,1\}^L$ bzw. $[0,1]^L$ bzw. $\mathbb{R}^L$ zu realsieren.
- Jetzt verfolgen wir die Ansätze aus 7.2-7.4 um die Funktion c durch eine Funktion f anzunäheren.
- Die Ausgabewerte $f(x) = (f_1(x), \ldots, f_L(x)) \in \mathbb{R}^L$ bzw. $[0,1]^L$ lassen sich dann als Zugehörigkeit der Eingabe x zu den Klassen interpretieren. interpretieren.

7.6 Bewertung

- Nachdem eine Regression/Prognosefunktion berechnet (trainiert) wurde, sind wir daran interessiert, ihre Generalisierungsfähigkeit zu überprüfen.
- Die Bewertung der Generalisierungsleistung sollte nicht mit dem Trainingsmaterial durchgeführt werden, da der Generalisierungsfehler auf dem Trainingsdatensatz unterschätzt wird.
- Wie bei der Klassifikation soll der Fehler auf dem gesamten Eingaberaum bestimmt werden. Das Problem hierbei ist, dass die zu approximierende Funktion (bzw. die Klassifikatorabbildung) nur auf einer endlichen Beispielmenge bekannt ist.
- Verfahren zur Schätzung sind **Hold-Out-Methode** oder **Cross-Validation-Methode** (CV) (siehe hierzu auch das Kapitel zur Klassifikation).