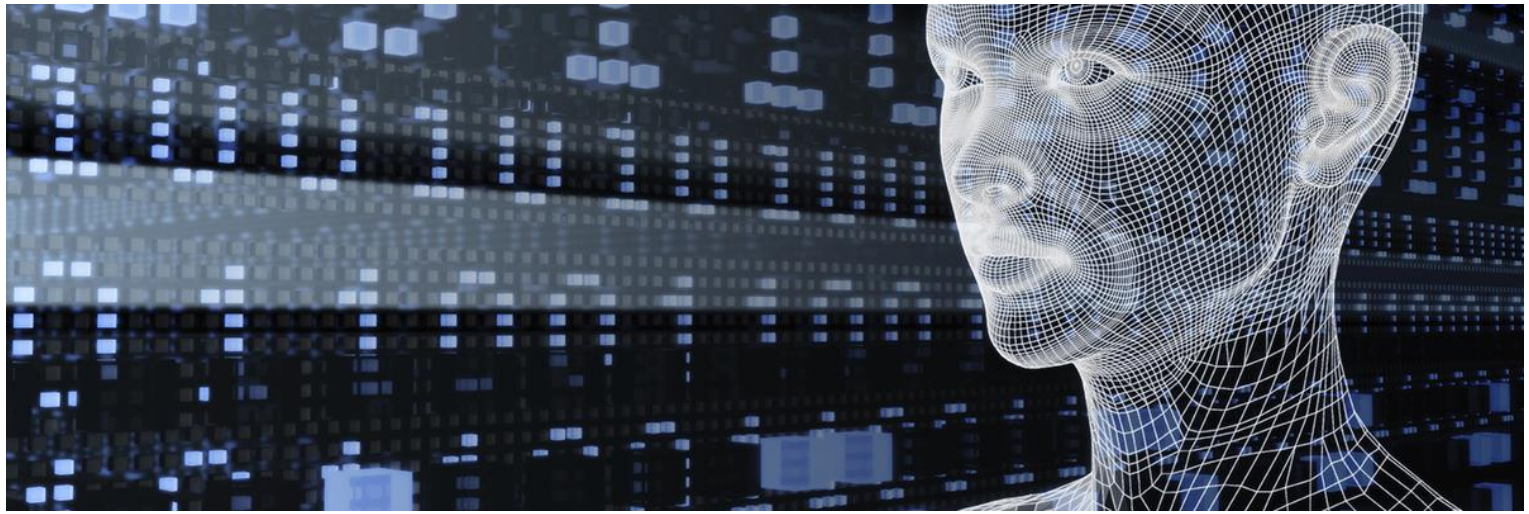
Künstliche Intelligenz

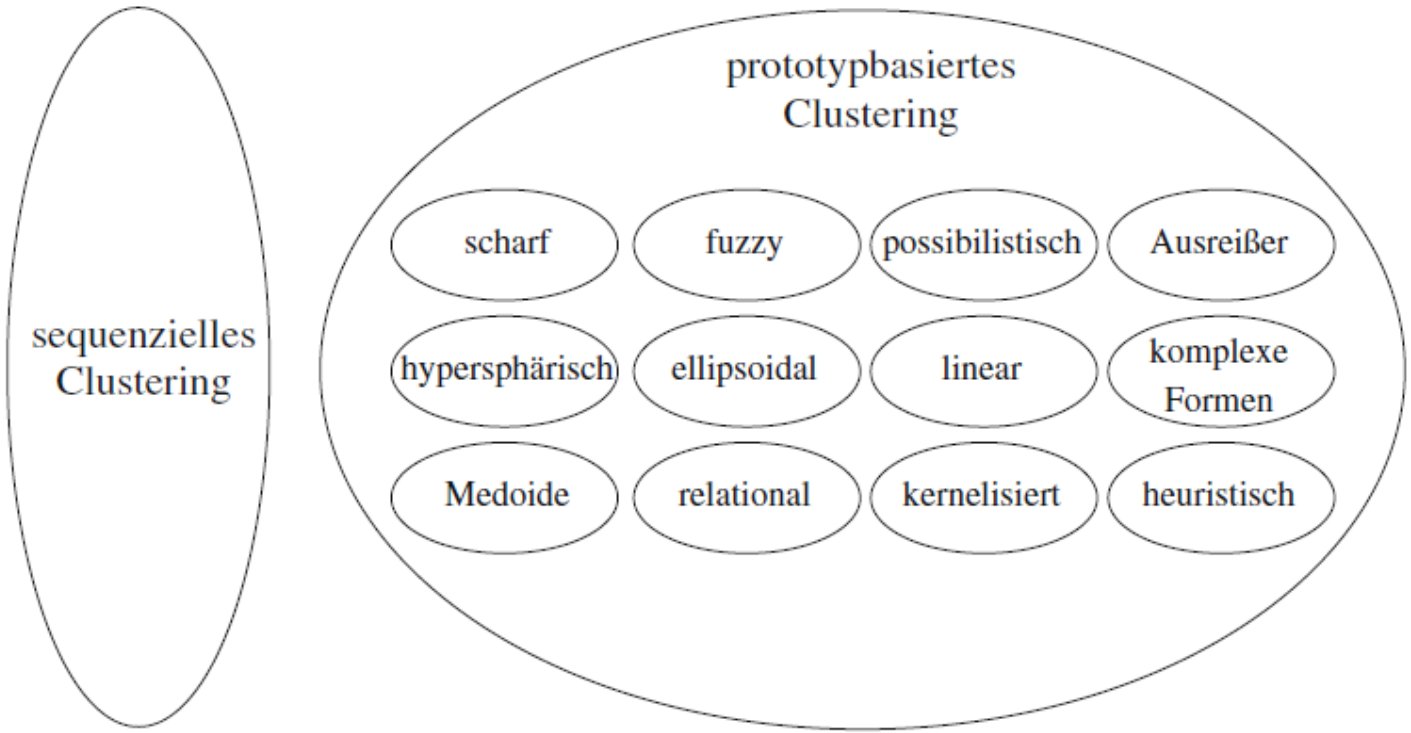
Vorlesung 9: Maschinelles Lernen und Data Mining



Wiederholung

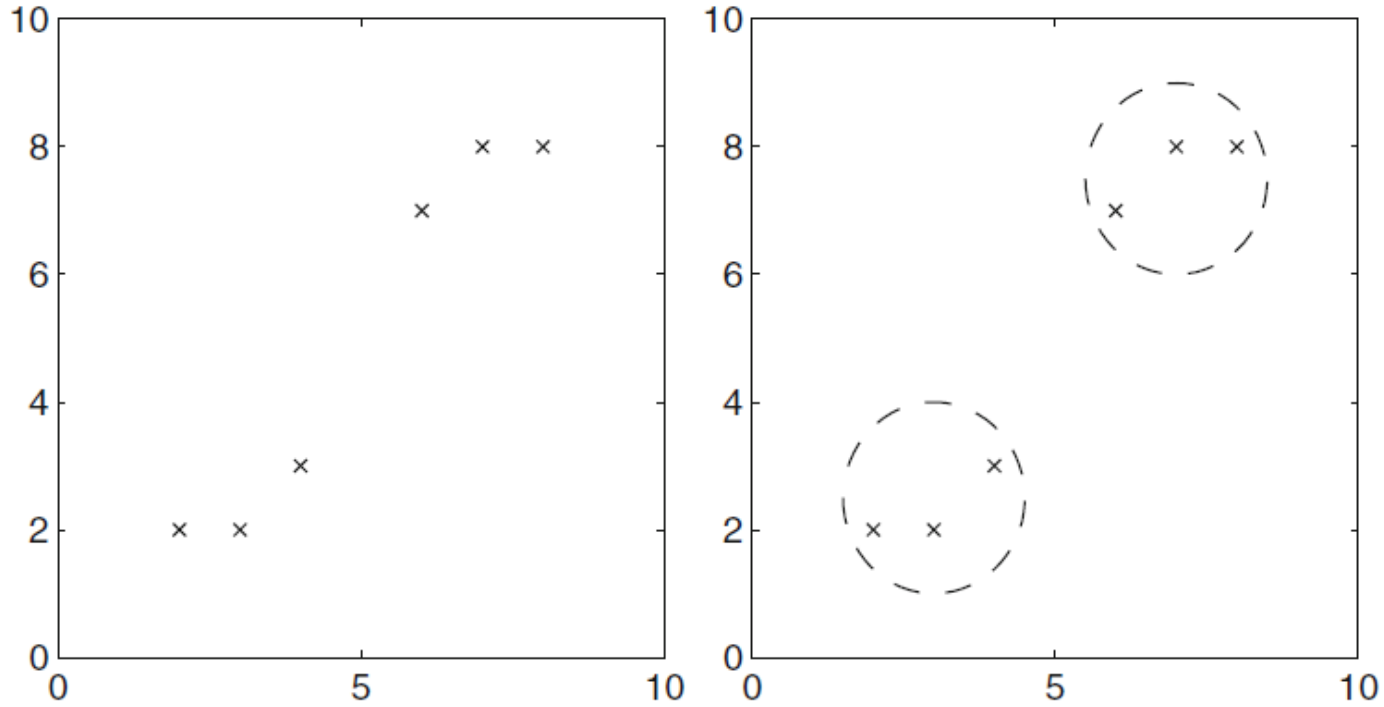
- Clustering ist ein unüberwachtes Lernverfahren, bei dem unmarkierte Daten Clustern zugeordnet werden.
- Falls die zu clusternden Daten auch Klassen zugeordnet sind, so können die erhaltenen Clusterzugehörigkeiten möglicherweise den Klassenzugehörigkeiten entsprechen.
- Cluster- und Klassenzugehörigkeiten können jedoch auch verschieden sein. Cluster können mathematisch mit Hilfe von Mengen, Partitionsmatrizen und/oder Cluster-Prototypen spezifiziert werden.
- Sequenzielles Clustering (z. B. Single-Linkage, Complete-Linkage, Average-Linkage, Ward-Methode) läßt sich einfach implementieren, hat aber einen hohen Rechenaufwand.
- Partitionsbasiertes Clustering kann mit scharfen, unscharfen, probabilistischen oder robusten Clustermodellen definiert werden. Clusterprototypen können verschiedene geometrische Formen annehmen (z. B. Hypersphären, Ellipsoide, Linien, Hyperebenen, Kreise oder kompliziertere Formen).
- Relationale Clustermodelle finden Cluster in relationalen Daten. Dabei kann auch der Kernel-Trick angewendet werden.
- Die Clustertendenz gibt an, ob die Daten überhaupt Cluster enthalten. Clustervaliditätsmaße quantifizieren die Güte des Clusterergebnisses und ermöglichen, die Anzahl der Cluster abzuschätzen.
- Auch heuristische Methoden wie die selbstorganisierende Karte können zum Clustering verwendet werden.

Clusterverfahren



Clusteringverfahren

- $X = \{(2, 2), (3, 2), (4, 3), (6, 7), (7, 8), (8, 8), (5, 5)\}$ (9.1)



Clusteringverfahren

- Die Clusterstruktur partitioniert diesen Datensatz X in die paarweise disjunkten Teilmengen $C_1 = \{x_1, x_2, x_3\}$ und $C_2 = \{x_4, x_5, x_6\}$.
- Im allgemeinen Fall ist die Zerlegung eines Datensatzes

$$X = \{x_1, \ldots, x_n\} \subset \mathrm{R}^p$$

in seine (scharfe) Clusterstruktur definiert als die Partition von X in

$c \in \{2, 3, \ldots, n-1\}$ paarweise disjunkte Teilmengen $C_1, \ldots, C_c$, so dass

$$X = C_1 \cup \ldots \cup C_c \quad (9.2)$$

$$C_i \text{ nichtleer, für alle } i = 1, \ldots, c \quad (9.3)$$

$$C_i \cap C_j = \{\} \text{ für alle } i, j = 1, \ldots, c, i \text{ und } j \text{ verschieden} \quad (9.4)$$

Wiederholung: K-means

Zahl der Cluster ist bekannt! **k-Means**-Verfahren:

- Zuerst werden die k Clustermittelpunkte $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ zufällig oder manuell initialisiert.
- Dann wird wiederholt:
  - Klassifikation aller Daten zum nächsten Clustermittelpunkt
  - Neuberechnung der Clustermittelpunkte

K-means Algorithmus

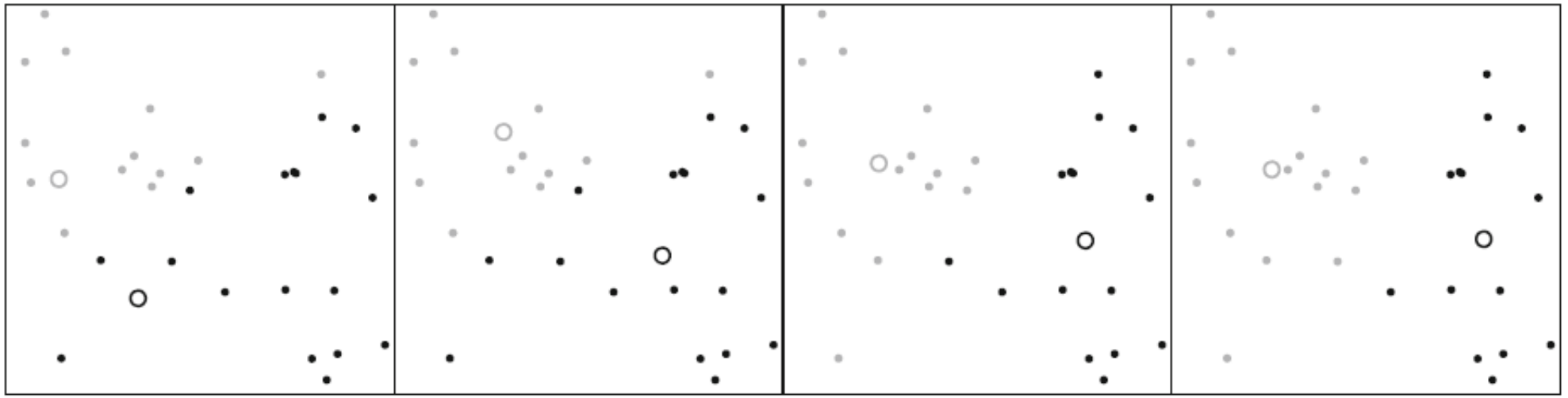
k-Means($\mathbf{x}_1, \ldots, \mathbf{x}_n, k$)

initialisiere $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ (z.B. zufällig)
Repeat
    Klassifiziere $\mathbf{x}_1, \ldots, \mathbf{x}_n$ zum jeweils nächsten $\boldsymbol{\mu}_i$
    Berechne $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ neu
Until keine Änderung in $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$
Return$(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k)$

Berechnung des Clustermittelpunktes:

$$\boldsymbol{\mu} = \frac{1}{l} \sum_{i=1}^{l} \mathbf{x}_i.$$

K-means



k-Means mit zwei Klassen (k = 2) angewendet auf 30 Datenpunkte. *Ganz links* die Datenmenge mit den initialen Zentren und *nach rechts* die Cluster nach jeder Iteration. Nach drei Iterationen ist die Konvergenz erreicht

K-means Algorithmus

- keine Konvergenzgarantie, aber meist recht schnelle Konvergenz.
- Zahl der Iterationsschritte meist viel kleiner als Zahl der Punkte.
- Komplexität: O(ndkt), mit
  - ✓n = Gesamtzahl der Punkte,
  - ✓d = Dimension des Merkmalsraumes
  - ✓t = Zahl der Iterationsschritte

EM-Algorithmus

- stetige Variante von k-Means
- liefert für jeden Punkt die Wahrscheinlichkeiten für die Zugehörigkeit zu den verschiedenen Klassen.
- Art der Wahrscheinlichkeitsverteilung der Daten ist bekannt. (Oft Normalverteilung)
- EM-Algorithmus bestimmt die Parameter (Mittelwert μ und Standardabweichungen σ_{ij} der k mehrdimensionalen Normalverteilungen) für jeden Cluster:

  Expectation: Für jeden Datenpunkt wird berechnet, mit welcher Wahrscheinlichkeit $P(C_j|\mathbf{x}_i)$ er zu jedem der Cluster gehört.

  Maximization: Unter Verwendung der neu berechneten Wahrscheinlichkeiten werden die Parameter der Verteilung neu berechnet.

EM-Algorithmus

- weicheres Clustering!
- Der EM-Algorithmus wird neben dem Clustering zum Beispiel für das Lernen von Bayes-Netzen eingesetzt

Hierarchisches Clustering

Anfang: n Cluster mit je einem Punkt. Wiederholt werden die beiden nächsten Nachbarcluster vereinigt:

Hierarchisches-Clustering($\mathbf{x}_1, \ldots, \mathbf{x}_n$)

initialisiere $C_1 = \{\mathbf{x}_1\}, \ldots, C_n = \{\mathbf{x}_n\}$

Repeat

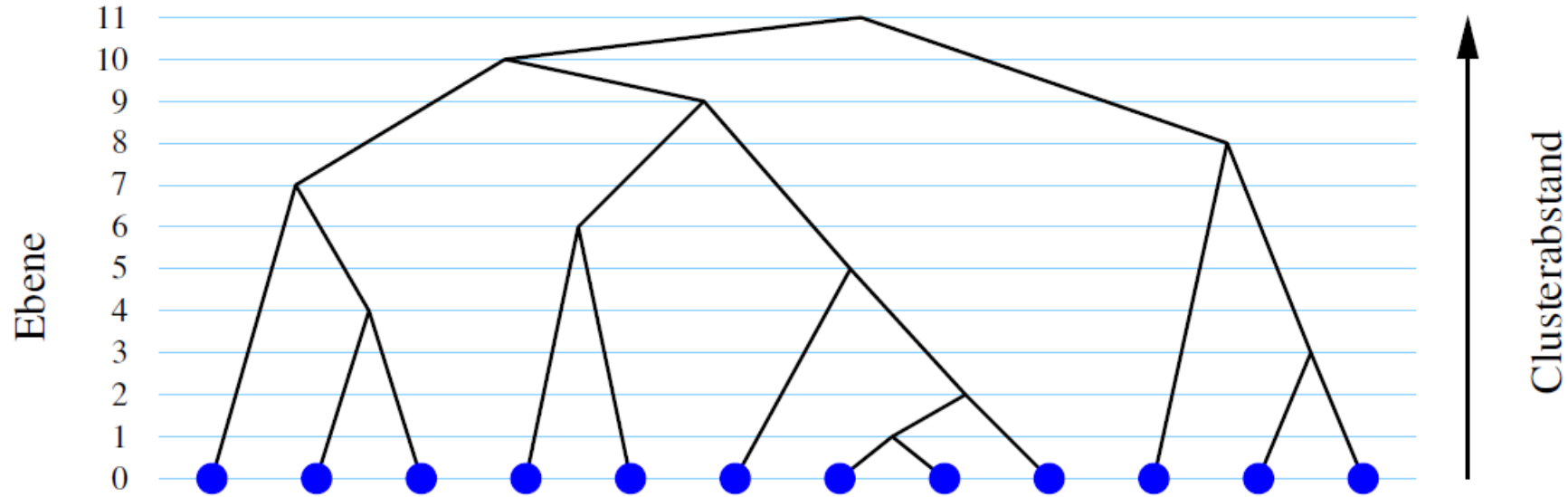
    Finde zwei Cluster C_i und C_j mit kleinstem Abstand

    Vereinige C_i und C_j

Until Abbruchbedingung erreicht

Return(Baum mit Clustern)

Hierarchisches Clustering



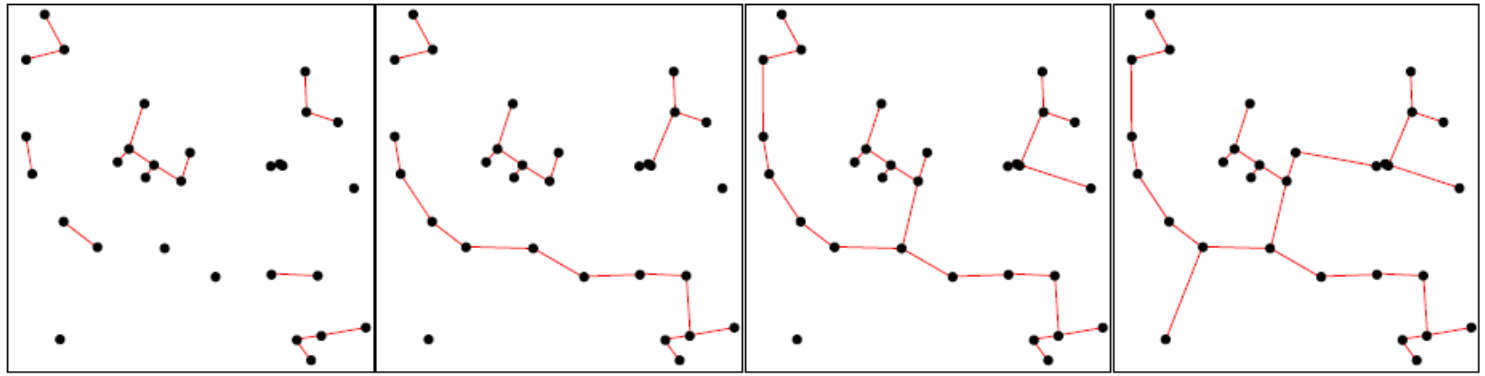
Abstände der Cluster

z.B. Abstand der zwei nächstliegenden Punkte aus den beiden Clustern C_i und C_j , also

$$d_{min}(C_i, C_j) = \min_{\mathbf{x}\in C_i,\ \mathbf{y}\in C_j} |\mathbf{x} - \mathbf{y}|.$$

Damit erhält man den Nearest Neighbour-Algorithmus.

Der Nearest Neighbour-Algorithmus, angewandt auf die Daten aus Slide 40 auf verschiedenen Ebenen mit 12, 6, 3, 1 Clustern



$d_{min} \geq 1.2$ $d_{min} \geq 1.6$ $d_{min} \geq 2$ minim. aufsp. Baum

- Algorithmus erzeugt minimalen aufspannenden Baum
- die beiden beschriebenen Algorithmen erzeugen ganz unterschiedliche Cluster
- arbeitet auf Adjazenzmatrix ($O(n^2)$ Zeit und Speicherplatz)
- Schleife wird n - 1 mal durchlaufen
- asymptotische Rechenzeit: $O(n^3)$

Farthest Neighbour-Algorithmus

- Abstands von zwei Clustern als Abstand der beiden entferntesten Punkte

$$d_{max}(C_i, C_j) = \max_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} |\mathbf{x} - \mathbf{y}|$$

- Alternativ: Abstand der Clustermittelpunkte
  $d_{\mu}(C_i, C_j) = |\boldsymbol{\mu}_\mathbf{i} - \boldsymbol{\mu}_\mathbf{j}|$

Wie bestimmt man die Anzahl der Cluster?

- Wie gut sind die Cluster getrennt?
- Finde k, das die Cluster optimal trennt.
- Silhouette Width Kriterium.
- Kreuzvalidierung zum Maximieren des Silhouette Width Kriteriums

Das Silhouette Width Kriterium

Gegeben:

- Daten: $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$.
- Zahl k der Cluster.
- Zuordnung $c : \mathbf{x}_i \mapsto c(\mathbf{x}_i)$ der Punkte zu Clustern.

Gesucht

- Funktion die die Qualität der Aufteilung der Punkte in Cluster misst!

Aufgabe:

- Finde unter den n^n Aufteilungen der Punkte in Cluster eine optimale.

Das Silhouette Width Kriterium

- Mißt die Qualität der Aufteilung der Punkte in Cluster!
- $\bar{d}(i, \ell)$ = mittlerer Abstand von $\mathbf{x}_i$ zu allen Punkten ($\neq \mathbf{x}_i$) in Cluster ℓ.
- $a(i) = \bar{d}(i, c(\mathbf{x}_i))$ = mittlerer Abstand von $\mathbf{x}_i$ zu allen anderen Punkten im gleichen Cluster.
- $b(i) = \min_{j \neq c(\mathbf{x}_i)}\{\bar{d}(i,j)\}$ = kleinster mittlerer Abstand von Punkt $\mathbf{x}_i$ zu einem Cluster, zu dem $\mathbf{x}_i$ nicht gehört.

$$s(i) = \begin{cases} 0 & \text{falls } a(i) = 0 \\ \dfrac{b(i) - a(i)}{\max\{a(i), b(i)\}} & \text{sonst} \end{cases}$$

Das Silhouette Width Kriterium

$$s(i) = \begin{cases} 0 & \text{falls } a(i) = 0 \\ \dfrac{b(i) - a(i)}{\max\{a(i), b(i)\}} & \text{sonst} \end{cases}$$

Es gilt: $-1 \leq s(i) \leq 1$.

$$s(i) = \begin{cases} 1 & \text{falls Punkt } \mathbf{x}_i \text{ in der Mitte seines eigenen Clusters ist.} \\ 0 & \text{falls Punkt } \mathbf{x}_i \text{ auf dem Rand zwischen zwei Clustern liegt.} \\ -1 & \text{falls Punkt } \mathbf{x}_i \text{ im „falschen" Cluster ist.} \end{cases}$$

Silhouette Width Kriterium

- Gesucht wird eine Partition, welche den Mittelwert von s(i) über alle Punkte maximiert:

$$S = \frac{1}{n} \sum_{i=1}^{n} s(i)$$

- Für den k-Means-Algorithmus wird dies durch den OMRk-Algorithmus erreicht.

Der OMRk Algorithmus

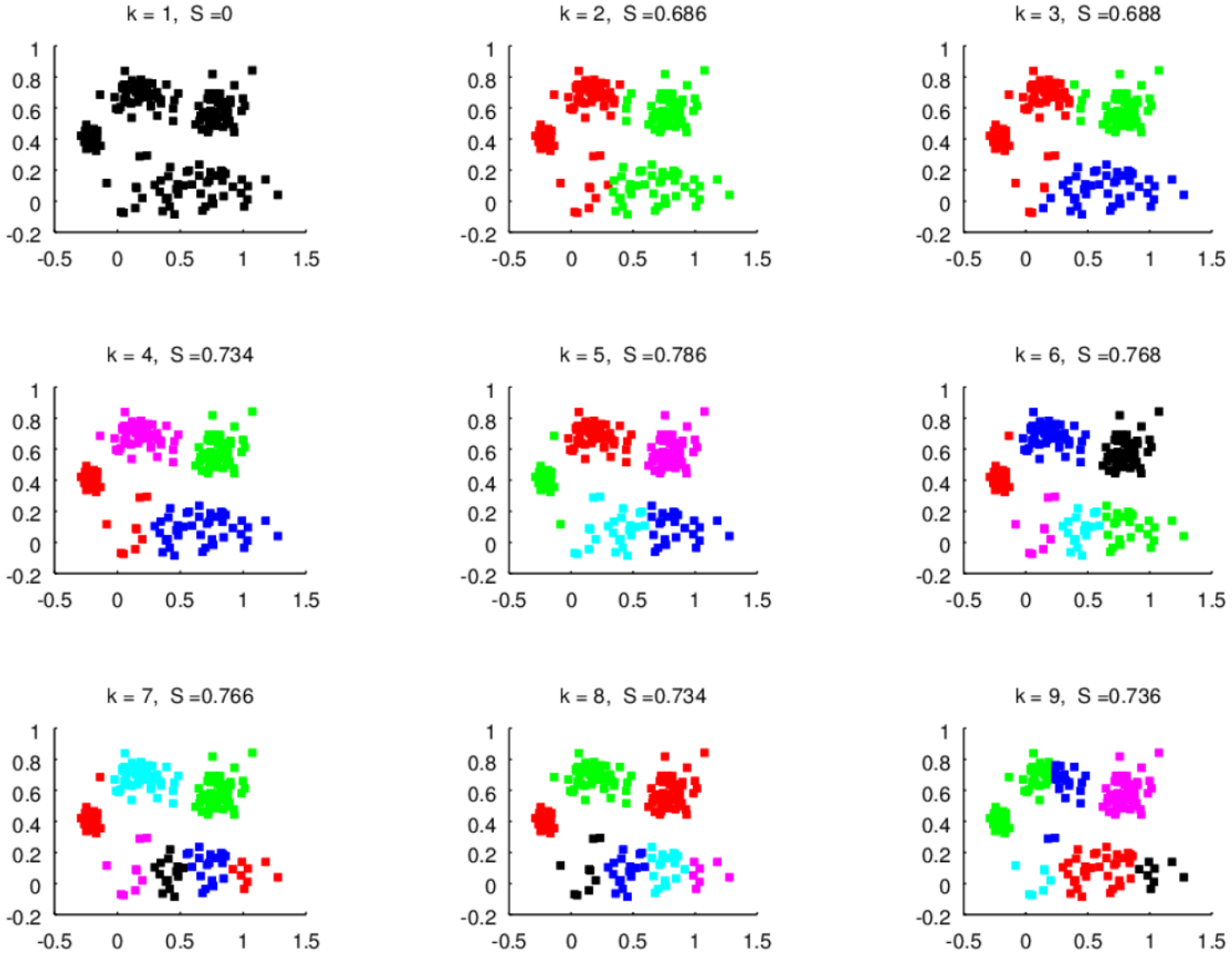
- Dieser Algorithmus wendet k-Means wiederholt für verschiedene k an.
- Da das Ergebnis von k-Means stark von der Initialisierung abhängt, werden in der inneren Schleife für jedes k noch p verschiedene zufällige Initialisierungen verwendet und dann das optimale k* ermittelt, welches bei der besten Initialisierung die beste Partition P* findet.

Der OMRk Algorithmus

Idee: Finde k, das $S = \frac{1}{n}\sum_{i=1}^{n} s(i)$ maximiert.

OMRk($\mathbf{x}_1, \ldots, \mathbf{x}_n, p, k_{max}$)

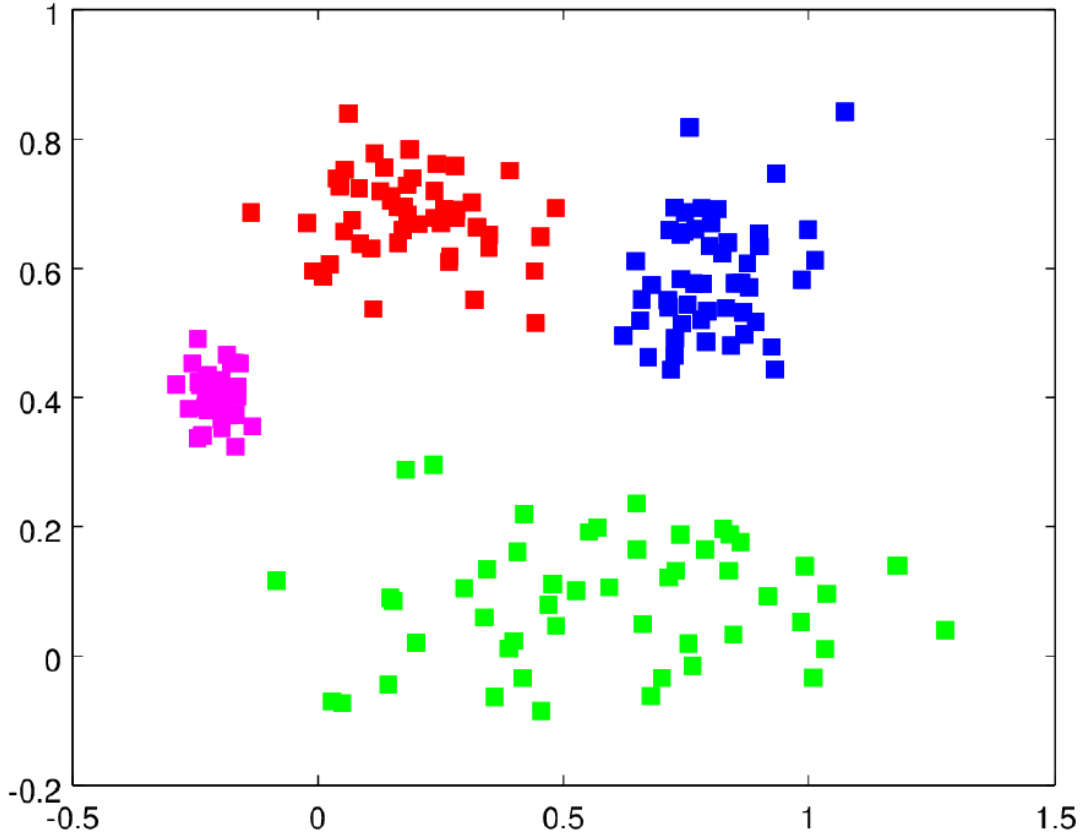
$S_* = -\infty$
For $k = 2$ **To** k_{max}
  For i=1 **To** p
    Erzeuge zufällige Partition mit k Clustern
    Ermittle mit k-Means eine Partition P
    Bestimme S für P
    If $S > S^*$ **Then**
      $S^* = S$; $k^* = k$; $P^* = P$
Return(k^*, P^*)



Ergebnisse des OMRk-Algorithmus für k = 2 bis 9 und p=30. Der beste Wert mit S = 0,786 wurde gefunden für k = 5

- Bei k=4, liegen mehrere Punkte, welche eigentlich zum blauen Cluster gehören, im roten Cluster.
- Dies liegt daran, dass bei k-Means der Abstand zum Clustermittelpunkt minimiert wird und die falsch zugeordneten Punkte näher am roten als am blauen Clustermittelpunkt liegen.
- Die höhere Dichte der Punkte im roten Cluster wird nicht berücksichtigt.

OMRk (mit EM-Algorithmus)



Wesentlich besser schneidet der EM-Algorithmus ab, welcher aufgrund der Verwendung von Normal-verteilungen die unterschiedliche Dichte der Punkte besser approximieren
kann.

Der EM-Algorithmus findet, für k = 4 mit hoher Wahrscheinlichkeit fast exakt die oben erwähnte natürliche Aufteilung.

Eine mit dem EM Algorithmus erzeugte Partition für k = 4

Sequenzielle aglomerative hierarchische nichtüberlappende (SAHN) Clusteranalyse

- Zu Beginn interpretiert SAHN jeden der n Punkte als einzelnen Cluster, so dass sich initial die Partition $n = \{\{x_1\}, \ldots, \{x_n\}\}$ ergibt.
- In jedem Schritt bestimmt SAHN aus der Menge der Cluster das Paar mit dem geringsten Abstand und fasst dieses Paar zu einem Cluster zusammen, so dass sich die Anzahl der Cluster in jedem Schritt um eins verringert.
- SAHN terminiert, wenn die gewünschte Anzahl von Clustern erreicht ist oder wenn alle Punkte zu einem einzigen Cluster zusammengefasst sind, was der Partition $1 = \{\{x_1, \ldots, x_n\}\}$ entspricht.
- Der Abstand zwischen einem Paar von Clustern kann auf verschiedene Weise aus den paarweisen Abständen der zugehörigen Punkte berechnet werden

Sequenzielle aglomerative hierarchische nichtüberlappende (SAHN) Clusteranalyse

1. Eingabe: $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^p$
2. Initialisiere $\Gamma_n = \{\{x_1\}, \ldots, \{x_n\}\}$
3. Für $c = n-1, n-2, \ldots, 1$
   - $(i, j) = \underset{(C_r, C_s) \in \Gamma_c}{\operatorname{argmin}} \, d(C_r, C_s)$
   - $\Gamma_c = (\Gamma_{c+1} \setminus C_i \setminus C_j) \cup (C_i \cup C_j)$
4. Ausgabe: Partitionen $\Gamma_1, \ldots, \Gamma_n$

Sequenzielle aglomerative hierarchische nichtüberlappende (SAHN) Clusteranalyse

- Minimalabstand (engl. *Single Linkage*)

$$d(C_r, C_s) = \min_{x \in C_r,\ y \in C_s} d(x, y) \tag{9.5}$$

- Maximalabstand (engl. *Complete Linkage*)

$$d(C_r, C_s) = \max_{x \in C_r,\ y \in C_s} d(x, y) \tag{9.6}$$

- mittlerer Abstand (engl. *Average Linkage*)

$$d(C_r, C_s) = \frac{1}{|C_r| \cdot |C_s|} \sum_{x \in C_r,\ y \in C_s} d(x, y) \tag{9.7}$$

Sequenzielle aglomerative hierarchische nichtüberlappende (SAHN) Clusteranalyse

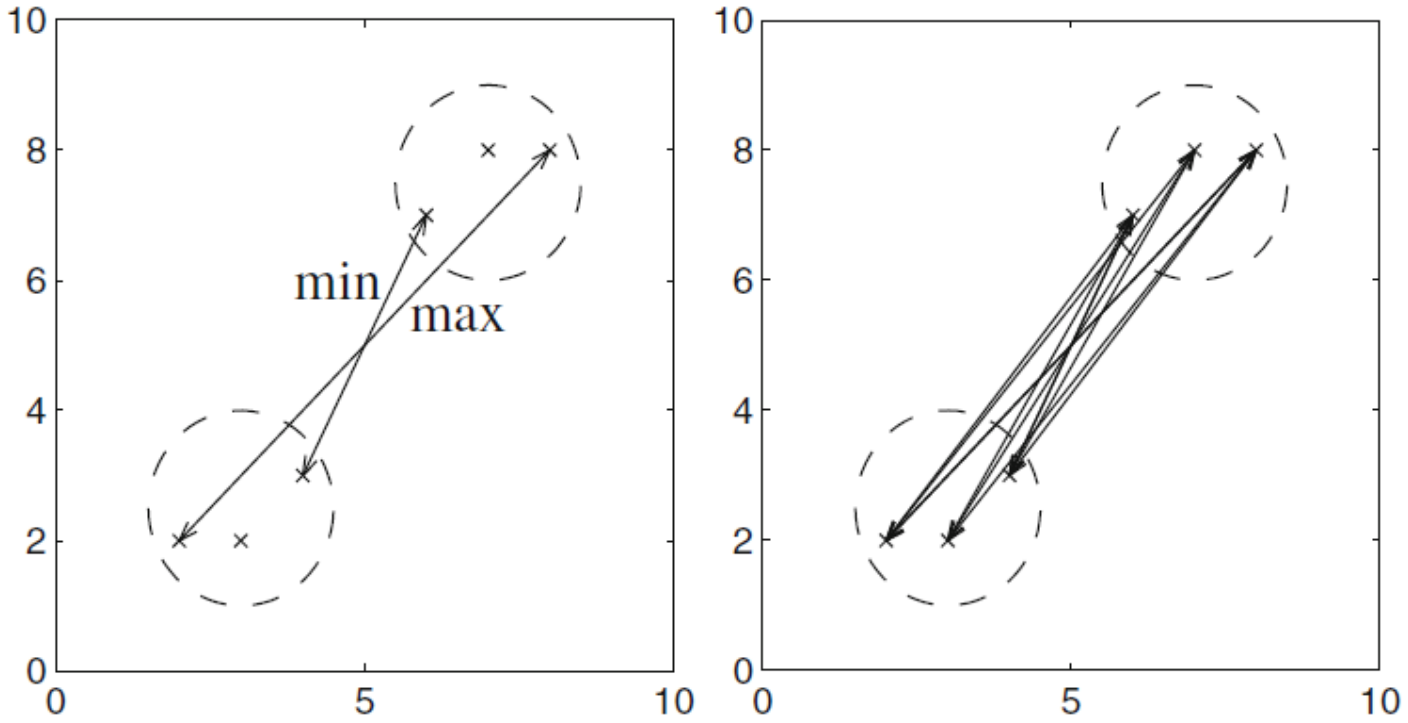
- Abstand der Zentren

$$d(C_r, C_s) = \left\| \frac{1}{|C_r|} \sum_{x \in C_r} x - \frac{1}{|C_s|} \sum_{x \in C_s} x \right\| \qquad (9.8)$$

- Ward-Methode

$$d(C_r, C_s) = \frac{|C_r| \cdot |C_s|}{|C_r| + |C_s|} \left\| \frac{1}{|C_r|} \sum_{x \in C_r} x - \frac{1}{|C_s|} \sum_{x \in C_s} x \right\| \qquad (9.9)$$

Abstände zwischen Clustern (*links*: Single und Complete Linkage, *rechts*: Average Linkage)

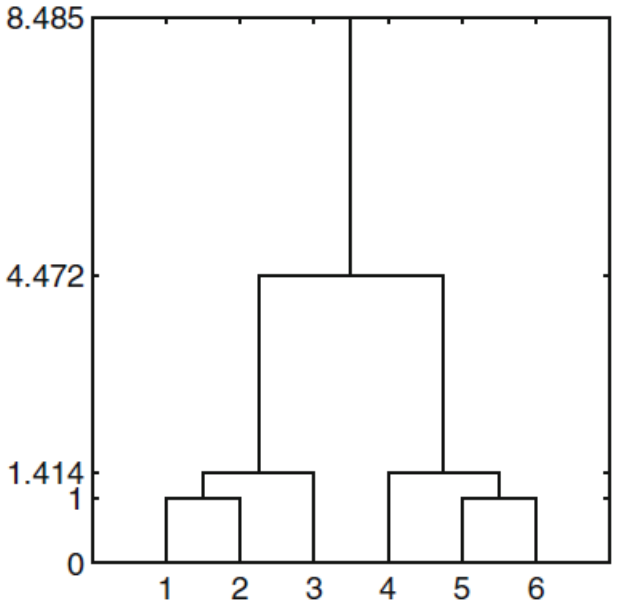


Sequenzielle aglomerative hierarchische nichtüberlappende (SAHN) Clusteranalyse

- SAHN-Algorithmen werden oft einfach nach dem verwendeten Abstandsmaß benannt, z. B. heißt SAHN mit (9.5) einfach *Single-Linkage-Clustering*.
- Die hierarchische Struktur der mit SAHN bestimmten Partitionen kann mit einem sogenannten *Dendrogramm* dargestellt werden.
- Es wird dargestellt wie die Punkte $x_1, \ldots, x_6$ (Indizes auf der horizontalen Achse) sukzessive zusammengefasst werden.
- Auf der vertikalen Achse sind die entsprechenden Single-Linkage-Abstände aufgetragen.

Sequenzielle aglomerative hierarchische nichtüberlappende (SAHN) Clusteranalyse

- $\Gamma_0 = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_6\}\}$ (9.10)
- $\Gamma_1 = \Gamma_2 = \{\{x_1, x_2\}, \{x_3\}, \{x_4\}, \{x_5, x_6\}\}$ (9.11)
- $\Gamma_3 = \Gamma_4 = \{\{x_1, x_2, x_3\}, \{x_4, x_5, x_6\}\}$ (9.12)
- $\Gamma_5 = \{\{x_1, x_2, x_3, x_4, x_5, x_6\}\} = \{X\}$ (9.13)



Prototypbasiertes Clustering

Im vorigen Abschnitt wurde eine Partition von X durch eine *Partitionsmenge* Γ von disjunkten Teilmengen von X dargestellt. Eine äquivalente Darstellung ist eine *Partitionsmatrix* U mit den Elementen

$$u_{ik} = \begin{cases} 1 & \text{falls } x_k \in C_i \\ 0 & \text{falls } x_k \notin C_i \end{cases} \tag{9.14}$$

$i = 1, \ldots, c$, $k = 1, \ldots, n$. Jeder *Zugehörigkeitswert* u_{ik} bestimmt, ob x_k zu C_i gehört. Für nichtleere Cluster fordern wir

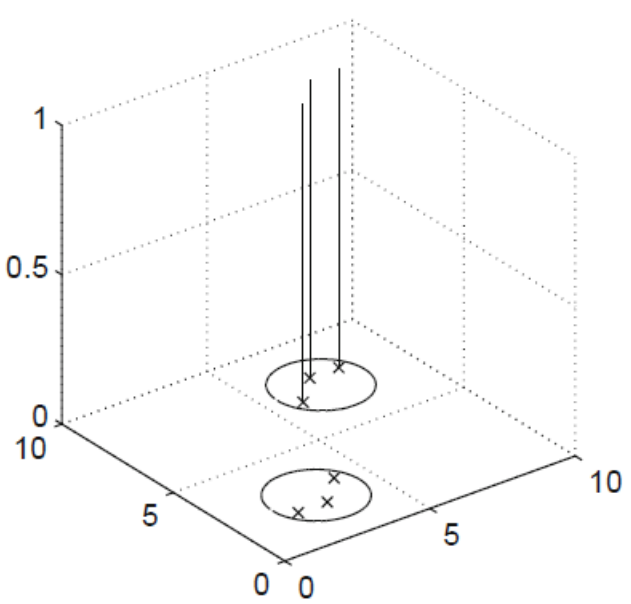
$$\sum_{k=1}^{n} u_{ik} > 0, \quad i = 1, \ldots, c \tag{9.15}$$

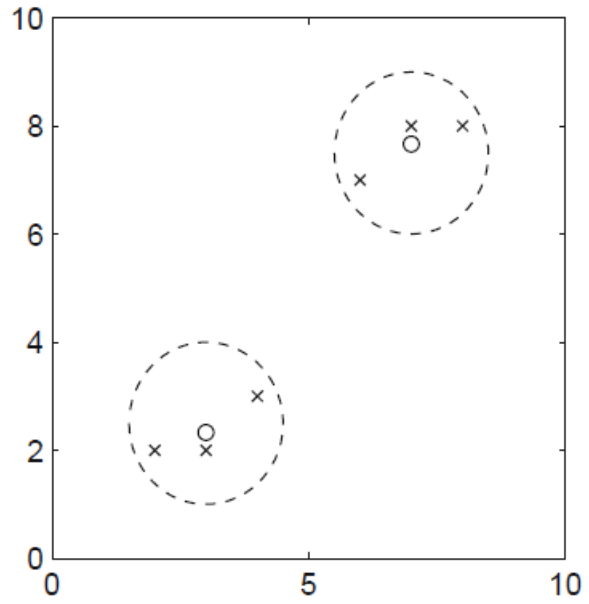
und für paarweise disjunkte Cluster

$$\sum_{i=1}^{c} u_{ik} = 1, \quad k = 1, \ldots, n \tag{9.16}$$

Clusterzugehörigkeiten und Clusterzentren

- Datensatz (9.1) mit vertikalen Balken, die die zweite Zeile der Partitionsmatrix darstellen.
- Die zweite Zeile der Partitionsmatrix ist eins für alle Elemente des zweiten Clusters und null für alle anderen Punkte.
- Die drei Balken an den Punkten x_4, . . . , x_6 haben daher die Höhe eins, und die übrigen Balken sind nicht sichtbar (Höhe null).





Prototypbasiertes Clustering

- Neben Partitionsmengen und Partitionsmatrizen können Cluster von Merkmalsdaten auch durch *Prototypen* repräsentiert werden.
- Z. B. kann jeder Cluster durch ein (einzelnes) Zentrum v_i , $i = 1, \ldots, c$, repräsentiert werden, so dass die Clusterstruktur durch die Menge der Clusterzentren $V = \{v_1, \ldots, v_c\} \subset \mathrm{R}^p$ (9.17) definiert wird.
- Für einen gegebenen Datensatz *X* können die Clusterzentren *V* und die Zuordnung der Datenpunkte *X* zu den *c* Clustern durch Optimierung des *k-Means-Clustermodells (CM)* gefunden werden.
- Die Kostenfunktion des k-Means-Clustermodells ist die Summe der quadratischen Abstände zwischen den Clusterzentren und den zugehörigen Datenpunkten.

$$J_{CM}(U, V; X) = \sum_{i=1}^{c} \sum_{x_k \in C_i} \|x_k - v_i\|^2 = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik} \|x_k - v_i\|^2 \qquad (9.18)$$

Prototypbasiertes Clustering

Zur Minimierung von J_{CM} bestimmen wir für jedes k das Minimum $\|x_k - v_i\|$, setzen die entsprechende Zugehörigkeit u_{ik} auf eins und alle übrigen Zugehörigkeiten u_{jk}, $j \neq i$, auf null.

$$u_{ik} = \begin{cases} 1 & \text{falls } \|x_k - v_i\| = \min_{j=1,\ldots,c} \|x_k - v_j\| \\ 0 & \text{sonst} \end{cases} \tag{9.19}$$

Im Falle mehrfacher Minima wird nur einer der Cluster ausgewählt, z. B. zufällig. Die notwendige Bedingung für Extrema von (9.18)

$$\frac{\partial J_{CM}(U, V; X)}{\partial v_i} = 0, \quad i = 1, \ldots, c \tag{9.20}$$

liefert die Clusterzentren

$$v_i = \frac{1}{|C_i|} \sum_{x_k \in C_i} x_k = \frac{\sum_{k=1}^{n} u_{ik}\, x_k}{\sum_{k=1}^{n} u_{ik}} \tag{9.21}$$

Prototypbasiertes Clustering

- Die Clusterzentren sind also die Mittelwerte der Punkte des entsprechenden Clusters, und die Datenpunkte werden dem Cluster mit dem nächsten Zentrum zugeordnet.
- Die optimale Partition *U* hängt also von den Clusterzentren *V* ab, und die optimalen Clusterzentren *V* hängen von der Partition *U* ab.
- *U* und *V* müssen daher mit einer alternierenden Optimierung (AO) bestimmt werden.
- Diese AO initialisiert *V*, berechnet abwechselnd *U* und *V* und terminiert in Abhängigkeit von *V* .
- Die umgekehrte Variante initialisiert *U*, berechnet abwechselnd *V* und *U* und terminiert in Abhängigkeit von *U*.

Alternierende Optimierung von Clustermodellen

1. Eingabe: Daten $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^p$,
   Clusteranzahl $c \in \{2, \ldots, n-1\}$,
   maximale Schrittzahl $t_{\max}$,
   Abstandsmaß $\|.\|$,
   Abstandsmaß zur Terminierung $\|.\|_\varepsilon$,
   Terminierungsgrenze ε
2. Initialisiere Prototypen $V^{(0)} \subset \mathbb{R}^p$
3. Für $t = 1, \ldots, t_{\max}$
   - Berechne $U^{(t)}(V^{(t-1)}, X)$
   - Berechne $V^{(t)}(U^{(t)}, X)$
   - Falls $\|V^{(t)} - V^{(t-1)}\|_\varepsilon \leq \varepsilon$, dann Ende
4. Ausgabe: Partitionsmatrix $U \in [0,1]^{c \times n}$,
   Prototypen $V = \{v_1, \ldots, v_c\} \in \mathbb{R}^p$

Clustertendenz

- Clusteralgorithmen liefern Partitionen, auch wenn die Daten eigentlich gar keine Cluster enthalten.
- Die Clustertendenz quantifiziert, zu welchem Grad die Daten geclustert sind.
- Hierzu eignet sich der sogenannte *Hopkins-Index*.
- Zur Berechnung des Hopkins-Index eines Datensatzes $X = \{x_1, \ldots, x_n\} \subset \mathrm{R}^p$ werden zunächst m Punkte $R = \{r_1, \ldots, r_m\}$ zufällig aus der konvexen Hülle oder (zur Vereinfachung) aus dem minimalen Hyperwürfel von X gewählt, $m << n$.
- Anschließend werden zufällig m Datenpunkte $S = \{s_1, \ldots, s_m\}$ aus X gezogen, so dass $S \subset X$. Für die Punkte beider Mengen R und S werden die Abstände zum jeweils nächsten Nachbarn aus X bestimmt.
- Auf Basis dieser Abstände $d_{r1}, \ldots, d_{rm}$ und $d_{s1}, \ldots, d_{sm}$ wird der Hopkins-Index $h \in [0, 1]$ berechnet als

$$h = \frac{\sum_{i=1}^{m} d_{r_i}^p}{\sum_{i=1}^{m} d_{r_i}^p + \sum_{i=1}^{m} d_{s_i}^p}$$

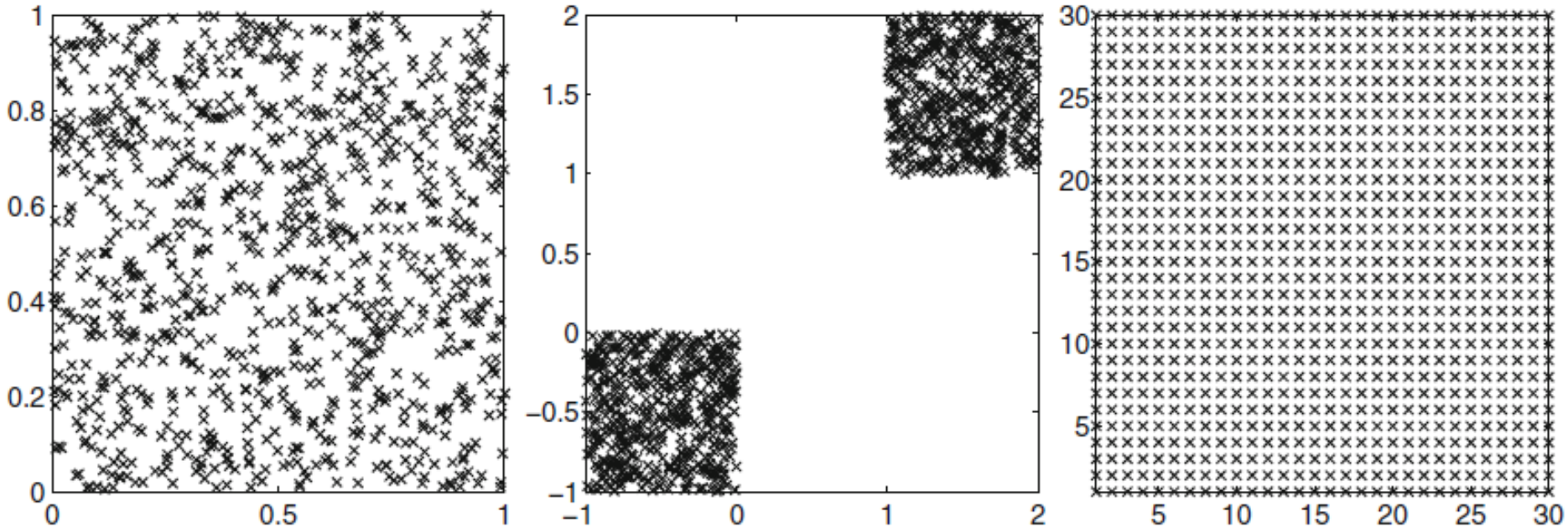
Clustertendenz

Es werden drei Fälle unterschieden

1. Für $h \approx 0.5$ sind die Abstände innerhalb X näherungsweise gleich den Abständen zwischen R und X, also sind R und X ähnlich verteilt. Da R gleichverteilt ist, ist auch X gleichverteilt und enthält daher keine Cluster.
2. Für $h \approx 1$ sind die Abstände innerhalb X viel kleiner als die Abstände zwischen R und X, also besitzt X eine Clusterstruktur.

3. Für $h \approx 0$ sind die Abstände zwischen R und X viel kleiner als die Abstände innerhalb X, also haben die Punkte in X näherungsweise gleiche Abstände zu ihren nächsten Nachbarn und liegen z. B. auf einem regelmäßigen Gitter.

Drei Datensätze und ihre Hopkins-Indizes



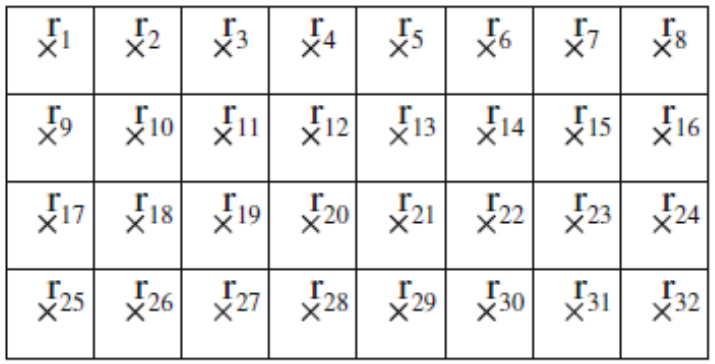
$h \approx 0.4229$ $h \approx 0.9988$ $h \approx 0.1664$

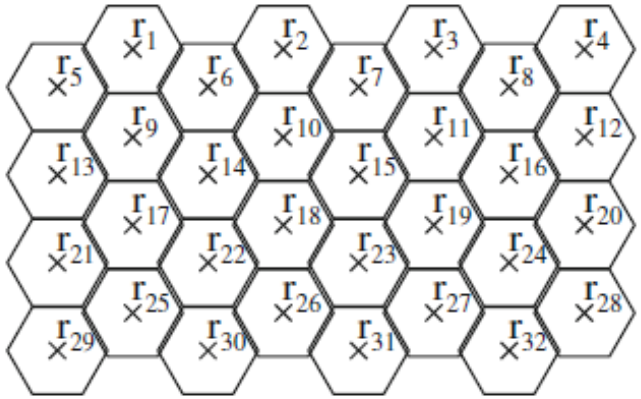
Drei Datensätze und ihre Hopkins-Indizes

- Drei unterschiedliche Datensätze und ihre Hopkins-Indizes (m = 10, n = 1000).
- Der linke Datensatz ist zufällig verteilt, daher ist $h \approx 0.5$.
- Der mittlere Datensatz enthält zwei deutlich getrennte Cluster, daher ist $h \approx 1$.
- Der rechte Datensatz ist auf einem regelmäßigen orthogonalen Gitter, daher ist h sehr klein.

Selbstorganisierende Karte

- Eine *selbstorganisierende Karte* ist ein heuristisches Projektions- und Clusterverfahren.
- Die Karte ist eine regelmäßige *q*-dimensionale Struktur mit *l* Referenzknoten.
- Für zweidimensionale Karten werden oft rechteckige oder hexagonale Strukturen verwendet





Selbstorganisierende Karte

- Jeder Knoten besitzt einen festen Ortsvektor $r_i \in R^q$, der die Position auf der Karte festlegt, und einen Referenzvektor $m_i \in R^p$, der sich auf die Daten X bezieht, $i = 1, \ldots, l$.
- In der Trainingsphase werden die Referenzvektoren $m_i \in R^p$ zunächst zufällig initialisiert. Dann wird für jeden Datenpunkt $x_k \in X$ der nächste Referenzvektor m_c gesucht, und es werden alle Referenzvektoren aktualisiert, die auf der Karte in der Nachbarschaft von r_c liegen.

Selbstorganisierende Karte

- Der Grad der Nachbarschaft in der Karte wird oft mit einer sogenannten *Blasenfunktion*

$$h_{ci} = \begin{cases} \alpha(t) & \text{falls } \|r_c - r_i\| < \rho(t) \\ 0 & \text{sonst} \end{cases}$$

- oder einer Gauß-Funktion

$$h_{ci} = \alpha(t) \cdot e^{-\frac{\|r_c - r_i\|^2}{2 \cdot \rho^2(t)}}$$

berechnet.

Selbstorganisierende Karte

- Der Nachbarschaftsradius $\rho(t)$ und die Lernrate $\alpha(t)$ werden während des Lernvorgangs kontinuierlich verringert, z. B. gemäß

$$\alpha(t) = \frac{A}{B + t}, \quad A, B > 0$$

Trainingsalgorithmus einer selbstorganisierenden Karte

1. Eingabe: Data $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^p$,
   Kartendimensionalität $q \in \{1, \ldots, p-1\}$,
   Knotenpositionen $R = \{r_1, \ldots, r_l\} \subset \mathbb{R}^q$
2. Initialisiere $M = \{m_1, \ldots, m_l\} \subset \mathbb{R}^p$, $t = 1$
3. Für jedes x_k, $k = 1, \ldots, n$,

   a. Bestimme Gewinnerknoten m_c mit

   $$\|x_k - m_c\| \leq \|x_k - m_i\| \quad \forall i = 1, \ldots, l$$

   b. Aktualisiere Gewinner und Nachbarn

   $$m_i = m_i + h_{ci} \cdot (x_k - m_c) \quad \forall i = 1, \ldots, l$$

4. $t = t + 1$
5. Wiederhole ab (3.), bis Terminierungsbedingung erfüllt
6. Ausgabe: Referenzvektoren $M = \{m_1, \ldots, m_l\} \subset \mathbb{R}^p$

Daten und Relationen

- Numerische Daten können als Mengen, Vektoren oder Matrizen repräsentiert werden.
- Viele Datenanalyseverfahren basieren auf Unähnlichkeitsmaßen (z. B. Matrixnormen, Lebesgue/Minkowski-Normen) oder
- Ähnlichkeitsmaßen (z. B. Cosinus, Überlapp, Dice, Jaccard, Tanimoto).
- Sequenzen können mit Sequenzrelationen analysiert werden (z. B. Hamming, Levenshtein/Edit-Abstand).

Maßskalen

- Numerische Informationen können unterschiedliche Bedeutung haben, auch wenn sie durch die gleichen numerischen Daten repräsentiert werden.
- Je nach semantischer Bedeutung können bestimmte mathematische Operationen zulässig oder unzulässig sein.

Skala	Operation		Beispiel	Statistisches Maß
Proportional	$\cdot$	$/$	273° K, 21 Jahre	Verallgemeinerter Mittelwert
Intervall	$+$	$-$	20° C, 2020 n. Chr.	Mittelwert
Ordinal	$>$	$<$	Sehr gut, gut, befriedigend	Median
Nominal	$=$	$\neq$	Müller, Meier, Schulz	Modus

Maßskalen

- Nominale Skalen: sind nur Teste auf Gleichheit und Ungleichheit zulässig. Beispiele für nominalskalierte Daten sind Namen von Personen oder Indizes von Objekten. Daten eines nominalskalierten Merkmals können durch den *Modus* (oder *Modalwert*), also dem am häufigsten vorkommenden Wert, repräsentiert werden.
- Für ordinalskalierte Daten (zweite Zeile von unten) sind die Ordnungsrelationen > bzw. < gültig.
- Daten eines ordinalskalierten Merkmals können durch den *Median* repräsentiert werden, also den Wert für den (ungefähr) so viele größere wie kleinereWerte vorliegen. Der *Mittelwert* ist für ordinalskalierte Daten nicht zulässig.

Maßskalen

- Für intervallskalierte Daten (dritte Zeile von unten) sind Addition und Subtraktion gültig.
- Intervallskalierte Merkmale haben kontextspezifisch definierte Nullpunkte.
- Beispiele sind Jahreszahlen nach Christus oder Temperaturen in Grad Celsius oder Grad Fahrenheit. Es also nicht sinnvoll zu sagen, dass 40◦ C doppelt so warm ist wie 20◦ C.
- Daten eines intervallskalierten Merkmals, z. B. die Daten $X = \{x_1, \ldots, x_n\}$, können durch den (arithmetischen) *Mittelwert*

$$\bar{x} = \frac{1}{n} \sum_{k=1}^{n} x_k$$

repräsentiert werden.

Maßskalen

- Für proportionalskalierte Daten sind Multiplikation und Division gültig.
- Beispiele für proportionalskalierte Merkmale sind Zeitdifferenzen (z. B. Alter) oder Temperaturen auf der Kelvin-Skala.
- Daten eines proportionalskalierten Merkmals können durch den *verallgemeinerten Mittelwert*

$$m_\alpha(X) = \sqrt[\alpha]{\frac{1}{n}\sum_{k=1}^{n} x_k^\alpha}$$

$\alpha \in R$, repräsentiert werden.

Der verallgemeinerte Mittelwert enthält die Sonderfälle Minimum ($\alpha \to -\infty$), harmonischer Mittelwert ($\alpha = -1$), geometrischer Mittelwert ($\alpha \to 0$), arithmetischer Mittelwert ($\alpha = 1$), quadratischer Mittelwert ($\alpha = 2$) und Maximum ($\alpha \to \infty$).

Mengen- und Matrixdarstellung

- Jeder numerischer Merkmalsdatensatz läßt sich als Menge

$$X = \{x_1, \ldots, x_n\} \subset \mathrm{R}^p$$

schreiben.

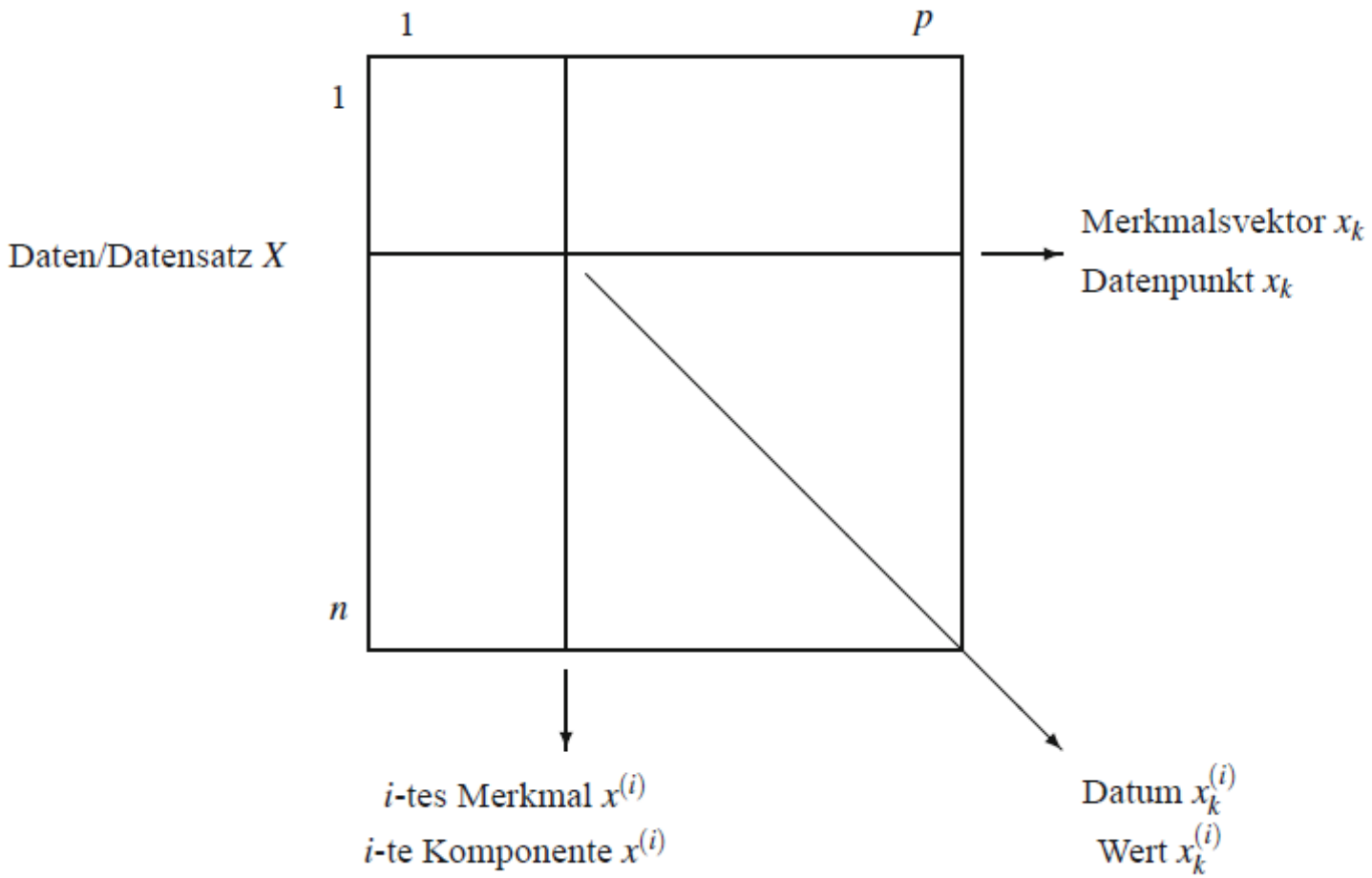
Ein solcher Datensatz enthält $n \in \{1, 2, \ldots\}$ Elemente. Jedes Element ist ein p dimensionaler reellwertiger Merkmalsvektor, $p \in \{1, 2, \ldots\}$.

- Für $p = 1$ heißt *X skalarer Datensatz*. Außer der Mengenschreibweise wird häufig auch die Matrixschreibweise

$$X = \begin{pmatrix} x_1^{(1)} & \cdots & x_1^{(p)} \\ \vdots & \ddots & \vdots \\ x_n^{(1)} & \cdots & x_n^{(p)} \end{pmatrix}$$

verwendet. Die Vektoren $x_1, \ldots, x_n$ sind also *Zeilenvektoren.*

Matrixschreibweise eines Datensatzes



Unähnlichkeitsmaße

Eine Funktion *d* heißt *Unähnlichkeitsmaß* oder *Distanzmaß* wenn für alle $x, y \in R^p$ gilt

$$d(x, y) = d(y, x)$$
$$d(x, y) = 0 \quad \Leftrightarrow \quad x = y$$
$$d(x, z) \leq d(x, y) + d(y, z)$$

Aus diesen Axiomen folgt $d(x, y) \geq 0$

Unähnlichkeitsmaße

- Eine Klasse von Unähnlichkeitsmaßen ist definiert durch eine *Norm* ||.|| von $x - y$, also

$$d(x, y) = \|x - y\|$$

Unähnlichkeitsmaße

Eine Funktion $\|.\|:\mathbb{R}^p \to \mathbb{R}^+$ heißt Norm genau dann, wenn

$$\|x\| = 0 \Leftrightarrow x = (0, \ldots, 0)$$
$$\|a \cdot x\| = |a| \cdot \|x\| \quad \forall a \in \mathbb{R}, x \in \mathbb{R}^p$$
$$\|x + y\| \leq \|x\| + \|y\| \quad \forall x, y \in \mathbb{R}^p$$

Häufig verwendete Normen

- Matrixnorm: $\|x\|_A = \sqrt{xAx^T}$
  - Euklidische Norm

$$A = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

  - *Frobenius-* oder *Hilbert-Schmidt-Norm*

$$A = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}$$

Häufig verwendete Normen

- *Diagonalnorm* mit merkmalsspezifischen Gewichten

$$A = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_p \end{pmatrix}$$

- Lebesgue- oder Minkowski-Norm

$$\|x\|_\alpha = \sqrt[\alpha]{\sum_{j=1}^{p} \left|x^{(j)}\right|^\alpha}$$

Häufig verwendete Normen

- Wichtige Sonderfälle der Lebesgue- oder Minkowski-Norm sind
  - die Infimum-Norm ($\alpha \to -\infty$)

$$\|x\|_{-\infty} = \min_{j=1,\ldots,p} x^{(j)}$$

  - *Manhattan*- oder *City-Block-Norm* ($\alpha = 1$)

$$\|x\|_1 = \sum_{j=1}^{p} \left|x^{(j)}\right|$$

  - *Euklidische Norm* ($\alpha = 2$) als einziger Schnittpunkt zwischen Matrixnormen und Lebesgue-/Minkowski-Normen

$$\|x\|_2 = \sqrt{\sum_{j=1}^{p} \left(x^{(j)}\right)^2}$$

  - Supremum-Norm ($\alpha \to \infty$)

$$\|x\|_\infty = \max_{j=1,\ldots,p} |x^{(j)}|$$

Häufig verwendete Normen

- Ein weiteres häufig verwendetes Unähnlichkeitsmaß ist der *Hamming-Abstand*

$$d_H(x, y) = \sum_{i=1}^{p} \rho(x^{(i)}, y^{(i)})$$

mit der diskreten Metrik

$$\rho(x, y) = \begin{cases} 0 & \text{falls } x = y \\ 1 & \text{sonst} \end{cases}$$

Der Hamming-Abstand ist also die Anzahl der unterschiedlichen Merkmalswerte.

Ähnlichkeitsmaße

Eine Funktion s heißt *Ähnlichkeitsmaß* wenn für alle $x, y \in \mathbb{R}^p$ gilt

$$s(x, y) = s(y, x)$$

$$s(x, y) \leq s(x, x)$$
$$s(x, y) \geq 0$$

Die Funktion s heißt *normalisiertes* Ähnlichkeitsmaß, wenn zusätzlich gilt

$$s(x, x) = 1$$

Ähnlichkeitsmaße

- Kosinus

$$s(x, y) = \frac{\sum_{i=1}^{p} x^{(i)} y^{(i)}}{\sqrt{\sum_{i=1}^{p} \left(x^{(i)}\right)^2 \sum_{i=1}^{p} \left(y^{(i)}\right)^2}}$$

- Überlapp

$$s(x, y) = \frac{\sum_{i=1}^{p} x^{(i)} y^{(i)}}{\min\left(\sum_{i=1}^{p} \left(x^{(i)}\right)^2, \sum_{i=1}^{p} \left(y^{(i)}\right)^2\right)}$$

- Dice

$$s(x, y) = \frac{2 \sum_{i=1}^{p} x^{(i)} y^{(i)}}{\sum_{i=1}^{p} \left(x^{(i)}\right)^2 + \sum_{i=1}^{p} \left(y^{(i)}\right)^2}$$

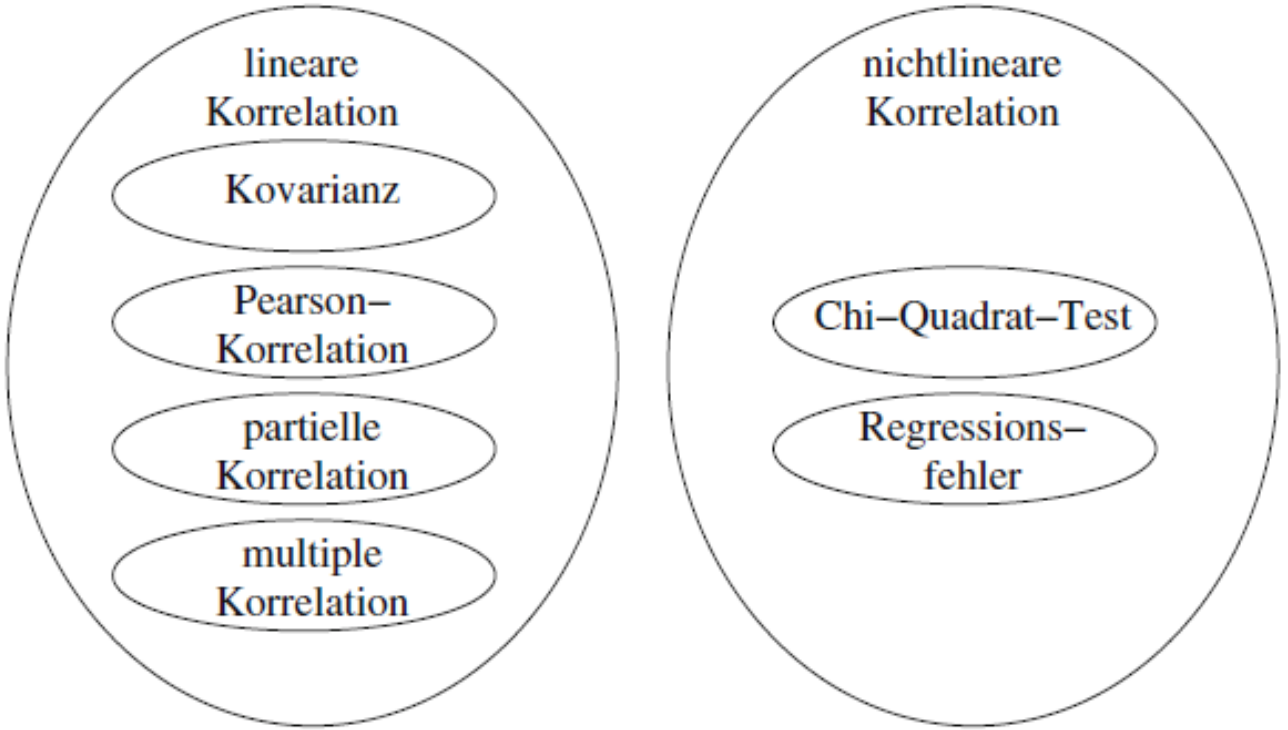
- Jaccard (auch Tanimoto)

$$s(x, y) = \frac{\sum_{i=1}^{p} x^{(i)} y^{(i)}}{\sum_{i=1}^{p} \left(x^{(i)}\right)^2 + \sum_{i=1}^{p} \left(y^{(i)}\right)^2 - \sum_{i=1}^{p} x^{(i)} y^{(i)}}$$

Korrelation

- Die Korrelationsanalyse quantifiziert den Zusammenhang zwischen Merkmalen.
- Die lineare Korrelationsanalyse ist robust und effizient, erfasst aber nur lineare Zusammenhänge.
- Die nichtlineare Korrelationsanalyse erfasst auch nichtlineare Zusammenhänge, muss aber sorgfältig parametriert werden.
  - Beispiel für nichtlineare Korrelationsverfahren: Chi-Quadrat-Test auf stochastische Unabhängigkeit vor.
- Die nichtlineare Korrelation kann auch durch den Validierungsfehler von Regressionsmodellen quantifiziert werden.
- Stark korrelierte Merkmale stehen nicht unbedingt in kausalem Zusammenhang, sondern können auch durch Scheinkorrelationen bedingt sein.
- Die partielle Korrelationsanalyse erlaubt es, Effekte von Scheinkorrelationen herauszurechnen.

Korrelationsverfahren



Lineare Korrelation

- Die Korrelation quantifiziert den Grad des Zusammenhangs zwischen Merkmalen.
- Ziel der Korrelationsanalyse ist es, zusammenhängende Merkmale zu identifizieren, um Ursachen für beobachtete Effekte zu erklären oder gezielt bestimmte Effekte herbeiführen zu können.
- In einer Produktionsanlage würde die Korrelationsanalyse beispielsweise diejenigen Merkmale identifizieren, die mit der Produktqualität zusammenhängen, so dass eine angestrebte Produktqualität durch gezielte Variation dieser Merkmale erzielt werden kann.

Lineare Korrelation

- Die Mahalanobis Norm $$A = \mathrm{cov}^{-1} X = \left(\frac{1}{n-1} \sum_{k=1}^{n} (x_k - \bar{x})^T (x_k - \bar{x}) \right)^{-1}$$

passt die Gewichtung der einzelnen Komponenten an die beobachtete Statistik an und berücksichtigt auch Korrelationen zwischen Merkmalen.

- Sei *X* ein Datensatz aus R^p. Die Covarianzmatrix von *X* ist gegeben durch

$$c_{ij} = \frac{1}{n-1} \sum_{k=1}^{n} (x_k^{(i)} - \bar{x}^{(i)})(x_k^{(j)} - \bar{x}^{(j)}), \quad i, j = 1, \ldots, p$$

Lineare Korrelation

- Große positive Werte von c_{ij} deuten auf eine starke positive Abhängigkeit zwischen $x^{(i)}$ und $x^{(j)}$ hin, d. h. wir beobachten größere Werte von $x^{(i)}$ zusammen mit größeren Werten von $x^{(j)}$ und kleinere Werte von $x^{(i)}$ zusammen mit kleineren Werten von $x^{(j)}$.
- Große negative Werte von c_{ij} deuten auf eine starke negative Abhängigkeit zwischen $x^{(i)}$ und $x^{(j)}$ hin, d. h. wir beobachten größere Werte von $x^{(i)}$ zusammen mit kleineren Werten von $x^{(j)}$ und kleinere Werte von $x^{(i)}$ zusammen mit größeren Werten von $x^{(j)}$.
- Kleine (positive oder negative) Werte von c_{ij} deuten auf eine schwache Abhängigkeit zwischen $x^{(i)}$ und $x^{(j)}$ hin.

Lineare Korrelation

- Wird ein Merkmal mit einem konstanten Faktor α multipliziert, beispielsweise weil das Merkmal in einer anderen Einheit gemessen wird (z. B. Meter statt Kilometer), so steigen die Kovarianzwerte zwischen diesem Merkmal und jedem anderen Merkmal ebenfalls um den Faktor α, obwohl der Zusammenhang zwischen den Merkmalen eigentlich der gleiche bleibt.
- Die Pearson-Korrelation kompensiert diesen Skalierungseffekt, indem die Kovarianz durch das Produkt der Standardabweichungen der beiden Merkmale geteilt wird.

Lineare Korrelation

$$s_{ij} = \frac{c_{ij}}{s^{(i)} s^{(j)}}$$

$$= \frac{\sum_{k=1}^{n} (x_k^{(i)} - \bar{x}^{(i)})(x_k^{(j)} - \bar{x}^{(j)})}{\sqrt{\left(\sum_{k=1}^{n} (x_k^{(i)} - \bar{x}^{(i)})^2\right)\left(\sum_{k=1}^{n} (x_k^{(j)} - \bar{x}^{(j)})^2\right)}}$$

$$= \frac{\sum_{k=1}^{n} x_k^{(i)} x_k^{(j)} - n\,\bar{x}^{(i)}\bar{x}^{(j)}}{\sqrt{\left(\sum_{k=1}^{n} \left(x_k^{(i)}\right)^2 - n\left(\bar{x}^{(i)}\right)^2\right)\left(\sum_{k=1}^{n} \left(x_k^{(j)}\right)^2 - n\left(\bar{x}^{(j)}\right)^2\right)}}$$

Korrelation und Kausalität

- Wir unterscheiden zwischen einer Korrelation und einer kausalen Beziehung zwischen zwei Merkmalen.
- Eine Korrelation zwischen *x* und *y* kann auf folgende kausale Beziehungen hindeuten:

1. Zufall
2. *x* wirkt auf *y*
3. *y* wirkt auf *x*
4. *z* wirkt auf *x* und *y*

Korrelation und Kausalität

- Sind die Merkmale $x(i)$ und $x(j)$ miteinander korreliert und zusätzlich mit $x(k)$ korreliert, dann interessiert oft die Korrelation zwischen $x^{(i)}$ und $x^{(j)}$ *ohne* den Einfluss von $x^{(k)}$.
- Diese sogenannte *partielle* oder *bedingte Korrelation* ist definiert als

$$s_{ij|k} = \frac{s_{ij} - s_{ik}s_{jk}}{\sqrt{(1 - s_{ik}^2)(1 - s_{jk}^2)}}$$

Korrelation und Kausalität

- Die Korrelation zwischen $x^{(i)}$ und $x^{(j)}$ ohne den Einfluss zweier Merkmale $x^{(k)}$ und $x^{(l)}$ heißt *bipartielle Korrelation* und ist definiert als

$$s_{i|k,\,j|l} = \frac{s_{ij} - s_{ik}s_{jk} - s_{il}s_{jl} + s_{ik}s_{kl}s_{jl}}{\sqrt{(1 - s_{ik}^2)(1 - s_{jl}^2)}}$$

Korrelation und Kausalität

- Die Korrelation von $x^{(i)}$ mit einer Menge von Merkmalen $x^{(j_1)}, \ldots, x^{(j_q)}$ heißt *multiple Korrelation* und ist definiert als

$$s_{i,(j_1,\ldots,j_q)} = \sqrt{(s_{ij_1} \ldots s_{ij_q}) \cdot \begin{pmatrix} 1 & s_{j_2 j_1} & \ldots & s_{j_1 j_q} \\ s_{j_1 j_2} & 1 & \ldots & s_{j_2 j_q} \\ \vdots & \vdots & \ddots & \vdots \\ s_{j_1 j_q} & s_{j_2 j_q} & \ldots & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} s_{ij_1} \\ s_{ij_2} \\ \vdots \\ s_{ij_q} \end{pmatrix}}$$

Chi-Quadrat-Test auf Unabhängigkeit

- Bisher: lineare Zusammenhänge.
- Welche Methoden für nichtlineare Zusammenhänge?
- *Chi-Quadrat-Test auf stochastische Unabhängigkeit*
  - Zur Bestimmung der nichtlinearen Korrelation zwischen zwei (kontinuierlichen) Merkmalen $x^{(1)}$und $x^{(2)}$ berechnen wir zunächst die Histogramme von $x^{(1)}$ mit r Intervallen und von $x^{(2)}$mit s Intervallen.

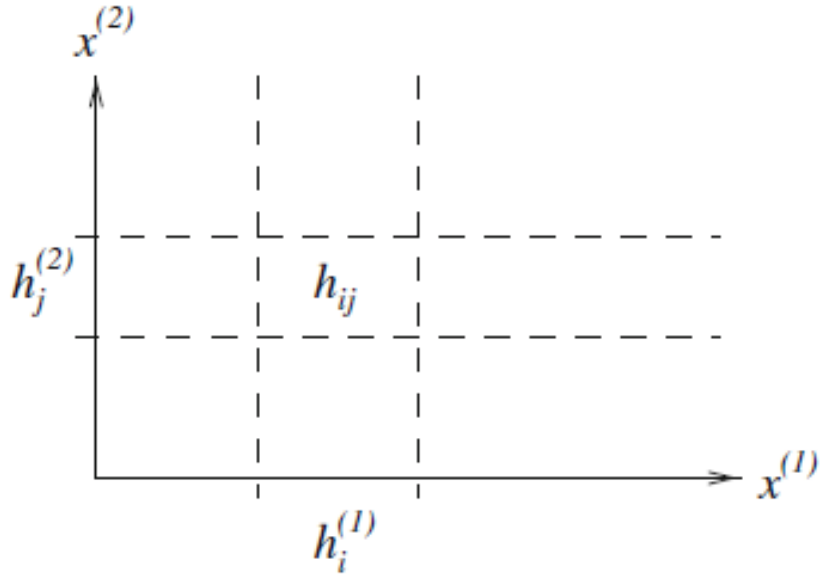
$$h^{(1)} = (h_1^{(1)}, \ldots, h_r^{(1)}), \quad h^{(2)} = (h_1^{(2)}, \ldots, h_s^{(2)})$$

Chi-Quadrat-Test auf Unabhängigkeit

Dann bestimmen wir $h_{ij}, i = 1, \ldots, r, j = 1, \ldots, s$, die Anzahl der Daten, die in das i-te Intervall von $x^{(1)}$ und das j-te Intervall von $x^{(2)}$ fallen, und schreiben diese Häufigkeiten als Matrix

$$H = \begin{pmatrix} h_{11} & h_{12} & \cdots & h_{1s} \\ h_{21} & h_{22} & \cdots & h_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ h_{r1} & h_{r2} & \cdots & h_{rs} \end{pmatrix}$$

Häufigkeiten für den Chi-Quadrat-Test auf stochastische Unabhängigkeit



Chi-Quadrat-Test auf Unabhängigkeit

- Die Histogramme von $x^{(1)}$ und $x^{(2)}$ entsprechen den Zeilen und Spaltensummen von H.

$$\sum_{j=1}^{s} h_{ij} = h_i^{(1)}, \quad i = 1, \ldots, r$$

$$\sum_{i=1}^{r} h_{ij} = h_j^{(2)}, \quad j = 1, \ldots, s$$

- Falls die Merkmale $x^{(1)}$ und $x^{(2)}$ stochastisch unabhängig sind, dann entspricht die Wahrscheinlichkeit, dass ein Datenpunkt in das Rechteck h_{ij} fällt, dem Produkt aus der Wahrscheinlichkeit, dass er in Intervall $h^{(1)}{}_i$ fällt, und der Wahrscheinlichkeit, dass er im Intervall $h^{(2)}{}_j$ fällt, also

Chi-Quadrat-Test auf Unabhängigkeit

$$\frac{h_{ij}}{n} = \frac{h_i^{(1)}}{n} \cdot \frac{h_j^{(2)}}{n} \quad \Rightarrow \quad h_{ij} = \frac{h_i^{(1)} \cdot h_j^{(2)}}{n}$$

wobei

$$n = \sum_{i=1}^{r} \sum_{j=1}^{s} h_{ij} = \sum_{i=1}^{r} h_i^{(1)} = \sum_{j=1}^{s} h_j^{(2)}$$

Chi-Quadrat-Test auf Unabhängigkeit

- Die Abweichung von h_{ij} von der stochastischen Unabhängigkeit können wir als
  - absoluten quadratischen Fehler $$E_1 = \left(h_{ij} - \frac{h_i^{(1)} \cdot h_j^{(2)}}{n} \right)^2$$
  - relativen quadratischen Fehler $$E_2 = \left(h_{ij} - \frac{h_i^{(1)} \cdot h_j^{(2)}}{n} \right)^2 / \left(\frac{h_i^{(1)} \cdot h_j^{(2)}}{n} \right)^2$$
  - oder als Kompromiss zwischen absolutem und relativem quadratischen Fehler $$E_3 = \left(h_{ij} - \frac{h_i^{(1)} \cdot h_j^{(2)}}{n} \right)^2 / \left(\frac{h_i^{(1)} \cdot h_j^{(2)}}{n} \right)$$

  definieren.

Chi-Quadrat-Test auf Unabhängigkeit

- Wir wählen E_3 und erhalten die Statistik des Chi-Quadrat-Tests als

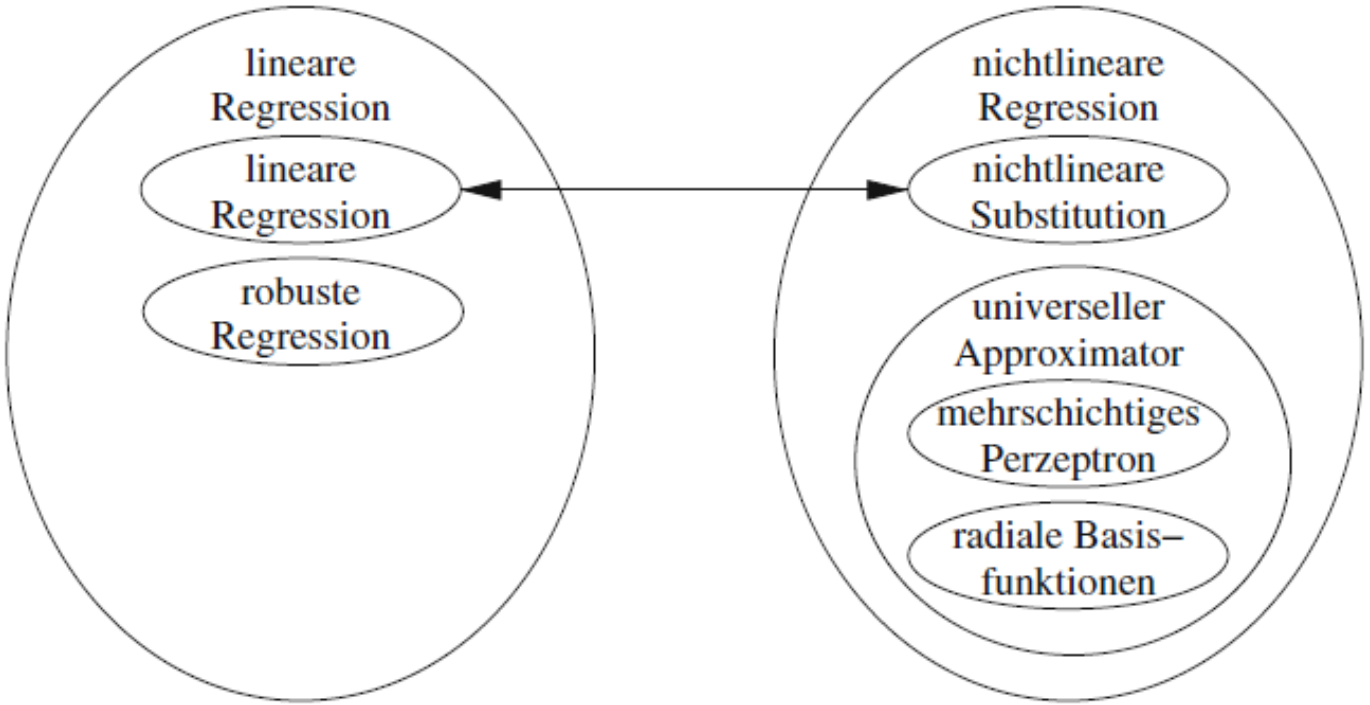
$$\chi^2 = \frac{1}{n} \sum_{i=1}^{r} \sum_{j=1}^{s} \frac{\left(n \cdot h_{ij} - h_i^{(1)} \cdot h_j^{(2)}\right)^2}{h_i^{(1)} \cdot h_j^{(2)}}$$

- Je kleiner der Wert von χ^2, desto größer ist die stochastische Unabhängigkeit zwischen den beiden Merkmalen.
- Um für p Merkmale eine sortierte Liste der Merkmalspaare mit den höchsten nichtlinearen Korrelationen zu finden, genügt es, die entsprechenden Werte von χ^2 zu berechnen und zu sortieren.

Regression

- Die Regressionsanalyse schätzt die funktionalen Abhängigkeiten zwischen Merkmalen, um Zusammenhänge zu verstehen und gezielt zu steuern.
- Lineare Regressionsmodelle können effizient aus den Kovarianzen berechnet werden, sind aber auf lineare Zusammenhänge beschränkt.
- Durch Substitution lassen sich auch bestimmte nichtlineare Regressionsmodelle durch lineare Regression finden.

Regressionsverfahren



Lineare Regression

- Korrelationsmethoden quantifizieren den Grad des Zusammenhangs zwischen Merkmalen.
- Die Regressionsanalyse: Abschätzung der Art des funktionalen Zusammenhangs zwischen Merkmalen.
- Wurden in der Korrelationsanalyse beispielsweise diejenigen Merkmale gefunden, die am stärksten mit der Produktqualität korrelieren, so besagen die Regressionsmodelle, auf welche Werte die betreffenden Merkmale gesetzt werden müssen, um eine bestimmte Zielqualität zu erreichen.

Lineare Regression

- Liefert lineare funktionale Zusammenhänge zwischen Merkmalen.
- Die Approximation eines Merkmals $x^{(i)}$ durch eine lineare Funktion f eines anderen Merkmals $x^{(j)}$, also $x^{(i)} \approx f(x^{(j)})$, $i, j \in \{1, \ldots, p\}$, $i, j \in \{1, \ldots, p\}$, kann geschrieben werden als

$$x_k^{(i)} \approx a \cdot x_k^{(j)} + b$$

- Die lineare Regression schätzt die Parameter $a, b \in \mathbb{R}$ dieser linearen Funktion aus X durch Minimierung einer geeigneten Fehlerfunktion.

Lineare Regression

- Für die lineare Regression wird gewöhnlich der quadratische Regressionsfehler minimiert:

$$E = \frac{1}{n}\sum_{k=1}^{n} e_k^2 = \frac{1}{n}\sum_{k=1}^{n}\left(x_k^{(i)} - a \cdot x_k^{(j)} - b\right)^2$$

Lineare Regression

Eine notwendige Bedingung für (lokale) Extrema von E lautet

$$\frac{\partial E}{\partial b} = -\frac{2}{n}\sum_{k=1}^{n}\left(x_k^{(i)} - a \cdot x_k^{(j)} - b\right) = 0 \quad \Rightarrow \quad b = \bar{x}^{(i)} - a \cdot \bar{x}^{(j)}$$

Damit können wir den Regressionsfehler schreiben als

$$E = \frac{1}{n}\sum_{k=1}^{n}\left(x_k^{(i)} - \bar{x}^{(i)} - a(x_k^{(j)} - \bar{x}^{(j)})\right)^2$$

Die andere notwendige Bedingung für (lokale) Extrema von E lautet

$$\frac{\partial E}{\partial a} = -\frac{2}{n}\sum_{k=1}^{n}(x_k^{(j)} - \bar{x}^{(j)})\left(x_k^{(i)} - \bar{x}^{(i)} - a(x_k^{(j)} - \bar{x}^{(j)})\right) = 0$$

Diese Gleichung liefert

$$a = \frac{\sum_{k=1}^{n}(x_k^{(i)} - \bar{x}^{(i)})(x_k^{(j)} - \bar{x}^{(j)})}{\sum_{k=1}^{n}(x_k^{(j)} - \bar{x}^{(j)})^2} = \frac{c_{ij}}{c_{jj}}$$

Lineare Regression

- Die Approximation eines Merkmals $x^{(i)}$ durch eine lineare Funktion f von $m \in \{2, 3, \ldots\}$ Merkmalen, also $x(i) \approx f\,(x^{(j_1)}, \ldots, x^{(j_m)})$, $i, j_1, \ldots, j_m \in \{1, \ldots, p\}$, kann geschrieben werden als

$$x_k^{(i)} \approx \sum_{l=1}^{m} a_l \cdot x_k^{(j_l)} + b$$

Die Parameter $a_1, \ldots, a_m, b \in \mathbb{R}$ werden bestimmt durch Minimierung von

$$E = \frac{1}{n}\sum_{k=1}^{n} e_k^2 = \frac{1}{n}\sum_{k=1}^{n}\left(x_k^{(i)} - \sum_{l=1}^{m} a_l \cdot x_k^{(j_l)} - b\right)^2$$

Eine notwendige Bedingung für (lokale) Extrema von E liefert

$$\frac{\partial E}{\partial b} = -\frac{2}{n}\sum_{k=1}^{n}\left(x_k^{(i)} - \sum_{l=1}^{m} a_l \cdot x_k^{(j_l)} - b\right) = 0 \quad \Rightarrow \quad b = \bar{x}^{(i)} - \sum_{l=1}^{m} a_l \cdot \bar{x}^{(j}$$

so dass

$$E = \frac{1}{n}\sum_{k=1}^{n}\left(x_k^{(i)} - \bar{x}^{(i)} - \sum_{l=1}^{m} a_l \cdot (x_k^{(j_l)} - \bar{x}^{(j_l)})\right)^2$$

Die andere notwendige Bedingung für (lokale) Extrema von E lautet

$$\frac{\partial E}{\partial a_r} = -\frac{2}{n}\sum_{k=1}^{n}(x_k^{(j_r)} - \bar{x}^{(j_r)})\left(x_k^{(i)} - \bar{x}^{(i)} - \sum_{l=1}^{m} a_l \cdot (x_k^{(j_l)} - \bar{x}^{(j_l)})\right) = 0$$

$r = 1, \ldots, m$, was als lineares Gleichungssystem geschrieben werden kann:

$$\sum_{l=1}^{m} a_l \sum_{k=1}^{n}(x_k^{(j_l)} - \bar{x}^{(j_l)})(x_k^{(j_r)} - \bar{x}^{(j_r)}) = \sum_{k=1}^{n}(x_k^{(i)} - \bar{x}^{(i)})(x_k^{(j_r)} - \bar{x}^{(j_r)})$$

$$\Leftrightarrow \quad \sum_{l=1}^{m} a_l c_{j_l j_r} = c_{i j_r}, \quad r = 1, \ldots, m$$

Lineare Regression

- Ein äquivalentes Ergebnis erhalten wir, wenn wir die lineare Regression in Matrixform schreiben.

$$Y = \begin{pmatrix} x_1^{(i)} - \bar{x}^{(i)} \\ \vdots \\ x_n^{(i)} - \bar{x}^{(i)} \end{pmatrix} \quad X = \begin{pmatrix} x_1^{(j_1)} - \bar{x}^{(j_1)} & \dots & x_1^{(j_m)} - \bar{x}^{(j_m)} \\ \vdots & \ddots & \vdots \\ x_n^{(j_1)} - \bar{x}^{(j_1)} & \dots & x_n^{(j_m)} - \bar{x}^{(j_m)} \end{pmatrix} \quad A = \begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix}$$

- Der lineare Regression Ausdruck wird zu

$$Y = X \cdot A$$
$$X^T \cdot Y = X^T \cdot X \cdot A$$
$$(X^T \cdot X)^{-1} \cdot X^T \cdot Y = A$$

Lineare Regression

Der Ausdruck $(X^T \cdot X)^{-1} \cdot X^T$ heißt *Pseudoinverse* von X. Die Matrix der Regressionsparameter A kann also als Produkt der Pseudoinversen von X und der Matrix Y berechnet werden.

Beispiel

$$X = \begin{pmatrix} 6 & 4 & -2 \\ 2 & 1 & -1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 2 & 4 & 2 \end{pmatrix}$$

Gesucht sei eine lineare Funktion $x^{(1)} = f(x^{(2)}, x^{(3)})$, also

$$x_k^{(1)} \approx \bar{x}^{(1)} + a_1(x_k^{(2)} - \bar{x}^{(2)}) + a_2(x_k^{(3)} - \bar{x}^{(3)})$$

Die Mittelwerte der Merkmale sind

$$\bar{x}^{(1)} = \frac{6+2+2}{5} = 2$$

$$\bar{x}^{(2)} = \frac{4+1+1+4}{5} = 2$$

$$\bar{x}^{(3)} = \frac{-2-1+1+2}{5} = 0$$

Beispiel

Die Kovarianzmatrix von X lautet

$$C = \begin{pmatrix} 6 & 3.5 & -2.5 \\ 3.5 & 3.5 & 0 \\ -2.5 & 0 & 2.5 \end{pmatrix}$$

Das lineare Gleichungssystem zur Bestimmung von a_1 und a_2 ist somit

$$\begin{aligned} c_{22}a_1 + c_{32}a_2 &= c_{12} \\ c_{23}a_1 + c_{33}a_2 &= c_{13} \end{aligned}$$

$$\Leftrightarrow \quad 3.5\,a_1 = 3.5 \quad \Leftrightarrow \quad a_1 = 1$$

$$\Leftrightarrow \quad 2.5\,a_2 = -2.5 \quad \Leftrightarrow \quad a_2 = -1$$

Beispiel

Wir erhalten also die multiple lineare Regressionsfunktion

$$x_k^{(1)} \approx 2 + (x_k^{(2)} - 2) - (x_k^{(3)} - 0) = x_k^{(2)} - x_k^{(3)}$$

Beispiel

Mit dem Ansatz der Pseudoinverse erhalten wir alternativ

$$Y = \begin{pmatrix} 6-2 \\ 2-2 \\ 0-2 \\ 0-2 \\ 2-2 \end{pmatrix} = \begin{pmatrix} 4 \\ 0 \\ -2 \\ -2 \\ 0 \end{pmatrix} \quad X = \begin{pmatrix} 4-2 & -2-0 \\ 1-2 & -1-0 \\ 0-2 & 0-0 \\ 1-2 & 1-0 \\ 4-2 & 2-0 \end{pmatrix} = \begin{pmatrix} 2 & -2 \\ -1 & -1 \\ -2 & 0 \\ -1 & 1 \\ 2 & 2 \end{pmatrix}$$

und damit

$$A = (X^T \cdot X)^{-1} \cdot X^T \cdot Y$$

$$= \left(\begin{pmatrix} 2 & -1 & -2 & -1 & 2 \\ -2 & -1 & 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} 2 & -2 \\ -1 & -1 \\ -2 & 0 \\ -1 & 1 \\ 2 & 2 \end{pmatrix} \right)^{-1} \begin{pmatrix} 2 & -1 & -2 & -1 & 2 \\ -2 & -1 & 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} 4 \\ 0 \\ -2 \\ -2 \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} 14 & 0 \\ 0 & 10 \end{pmatrix}^{-1} \begin{pmatrix} 14 \\ 10 \end{pmatrix} = \begin{pmatrix} \frac{1}{14} & 0 \\ 0 & \frac{1}{10} \end{pmatrix} \begin{pmatrix} 14 \\ 10 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

was der oben gefundenen Lösung $a_1 = 1, a_2 = -1$ entspricht.

Data Mining in der Praxis

- Alle vorgestellten Lernverfahren können zum Data Mining verwendet werden.
- teilweise mühsam
- Daten müssen immer passend formatiert werden

Data Mining-System:

- komfortable graphische Benutzeroberfläche
- Werkzeuge zur Visualisierung der Daten
- Vorverarbeitung, wie zum Beispiel die Manipulation von fehlenden Werten
- Analyse der Daten mit Lernverfahren

Data Mining Werkzeuge

- RapidMiner(www.rapidminer.com)
- Clementine (www.spss.com/clementine) (baut auf SPSS auf)
- KXEN Analytic Framework (www.kxen.com).
- KNIME (Konstanz Information Miner, www.knime.org)

Alternative: Open-Source Java-Programm Bibliothek WEKA

Zusammenfassung

Lernen mit Lehrer

faules Lernen (lazy learning)

- k-Nearest-Neighbour-Methode (Klass. + Approx.)
- Fallbasiertes Lernen (Klass. + Approx.)

eifriges Lernen (eager learning)

- Induktion von Entscheidungsbäumen (Klass.)
- Lernen von Bayes-Netzen (Klass. + Approx.)
- neuronale Netze (Klass. + Approx.)

Lernen ohne Lehrer (Clustering)

- Nearest Neighbour-Methode
- Farthest Neighbour-Methode
- k-Means
- neuronale Netze

Lernen durch Verstärkung

- Wert-Iteration
- Q-Lernen
- TD-Lernen
- Policy-Gradient-Methoden
- neuronale Netze

Offene Fragen / Forschung

automatische Merkmalsextraktion: (engl. feature selection)

- Kann eine Maschine neue Merkmale finden?
- z.B. mittels Berechnung des Informationsgewinns von Merkmalen
- Clustering zum automatischen kreativen „Entdecken” von Merkmalen