Clustering

Katja Markert, mit einigen Folien von Yannick Versley und Frank Keller

Institut für Computerlinguistik
Uni Heidelberg
markert@cl.uni-heidelberg.de

January 28, 2020

Bisher und heute

1. Bisher: Wortähnlichkeit mit Lexikonmethoden
2. Bisher: Wortähnlichkeit mit Vektorraummethoden
3. Dies ergibt die Ähnlichkeit zwischen einem Wortpaar
4. Heute: Clustering (Gruppiere Wörter oder andere Vektoren...)

Übersicht

Übersicht

Clustering von Worten: flach

Apfel	Apfel
Banane	Banane
Mann	Grapefruit
Grapefruit	Wassermelone
Frau	Traube
Baby	Mann
Wassermelone	Frau
Kind	Baby
Traube	Kind

Tue dies automatisch!

Paarweise Ähnlichkeitsberechnungen nur ein Teil des Problems

Clustering von Worten: Hierarchisch

Aus Jurafsky und Martin, Kapitel 6, 3rd edition

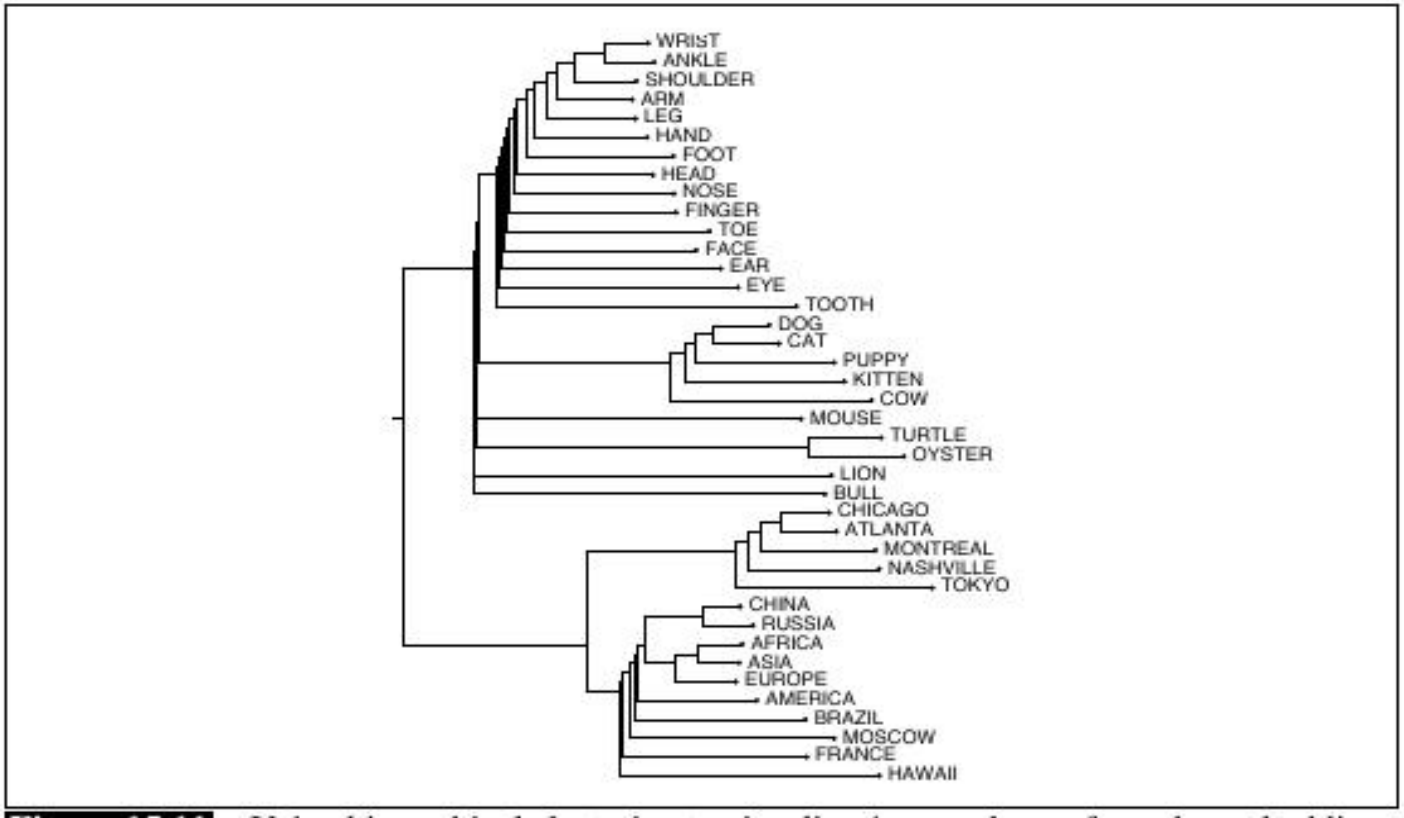
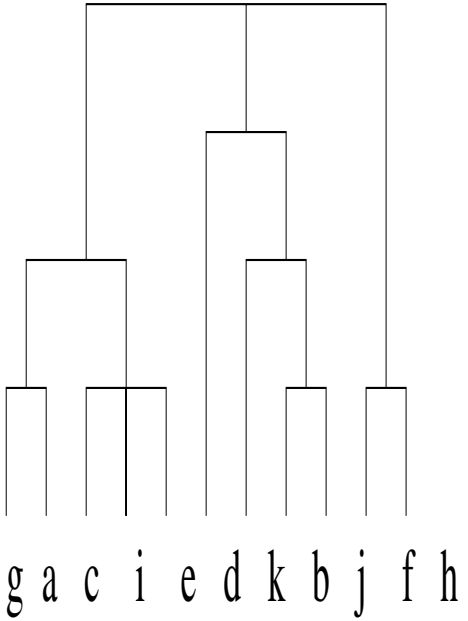


Figure 15.11 Using hierarchical clustering to visualize 4 noun classes from the embeddings produced by Rohde et al. (2006). These embeddings use a window size of ±4, and 14,000 dimensions, with 157 closed-class words removed. Rather than PPMI, these embeddings compute each cell via the positive correlation (the correlation between word pairs, with negative values replaced by zero), followed by a square root. This visualization uses hierarchical clustering, with correlation as the similarity function. From Rohde et al. (2006).

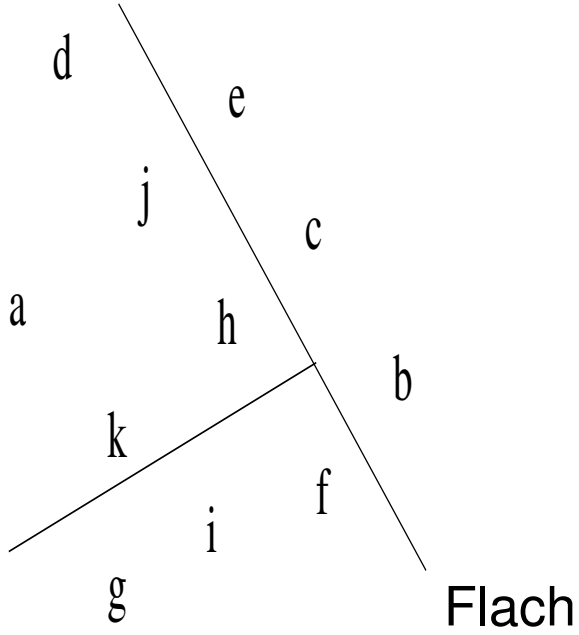
Clustering

- **Clustering:** lerne eine Klassifikation aus Daten
- **Unüberwachtes Lernen:** Trainingsdaten spezifizieren nicht, was gelernt werden soll, d.h. es gibt keine vorher definierten Klassen.
- Clustering Algorithmen teilen Daten in **natürliche Gruppen**
- Maximiere (durchschnittliche) Ähnlichkeit innerhalb eines Cluster
- Minimiere (durchschnittliche) Ähnlichkeit zwischen Clustern

Hierarchisches und Flaches Clustering



Hierarchisch (Dendrogramm)



Clustering: Schritte

1. Merkmalsselektion: repräsentiere Daten als m-dimensionale Vektoren.
2. Distanzmaß d zwischen Vektoren (oder Ähnlichkeitsmaß s): Distanzmaß aus Ähnlichkeitsmaß mit $d = 1 - s$ (wenn s zwischen 0 und 1)
3. Clustering Algorithmus
4. Evaluation

Übersicht

Agglomeratives hierarchisches Clustering

1. Anfangs: Jeder Vektor (Wort) in eigenem Cluster
2. Füge jeweils die beiden ähnlichsten Cluster zusammen
3. Berechne Ähnlichkeit zwischen zwei Clustern: Verschiedene Möglichkeiten
   - Single Link:

$$sim(C_1, C_2) := \max_{w_1 \in C_1, w_2 \in C_2} sim(w_1, w_2)$$

   - Complete Link:

$$sim(C_1, C_2) := \min_{w_1 \in C_1, w_2 \in C_2} sim(w_1, w_2)$$

   - Average Link:

$$sim(C_1, C_2) := \frac{1}{|C_1||C_2|} \sum_{w_1 \in C_1, w_2 \in C_2} sim(w_1, w_2)$$

4. Gehe zurück nach 2. und wiederhole bis vollständig geclustered.

Merken der Reihenfolge ergibt Hierarchie!

Beispiel

Man hat 5 Punkte mit folgenden vorberechneten euklidischen Distanzen zueinander, hier dargestellt in Matrix I:

s	s_1	s_2	s_3	s_4	s_5
s_1	–	17	21	31	23
s_2	–	–	30	34	21
s_3	–	–	–	28	39
s_4	–	–	–	–	43

Single Link Clustering I

Matrix I:

s	s_1	s_2	s_3	s_4	s_5
s_1	–	17	21	31	23
s_2	–	–	30	34	21
s_3	–	–	–	28	39
s_4	–	–	–	–	43

Bestimme den minimalen Clusterabstand aus Matrix I (17) $\longrightarrow$ Merge I: Neuer Cluster $\{s_1, s_2\}$.

Single Link Clustering II

Matrix II mit neuen Clustern: Clusterabstand = Minimum aller Distanzen zwischen einzelnen Punkten der Cluster

c	$\{s_1, s_2\}$	s_3	s_4	s_5
$\{s_1, s_2\}$	–	21	31	21
s_3	–	–	28	39
s_4	–	–	–	43

Merge II: Wir haben nun zwei Möglichkeiten zum Merge (zweimal 21 als Minimum). Wir könnten frei wählen. Ich wähle immer den ersten Matrixeintrag, wenn man sie von oben nach unten, links nach rechts durchgeht $\longrightarrow$ Merge $\{s_1, s_2\}$ mit s_3.

Single Link Clustering III

Matrix III:

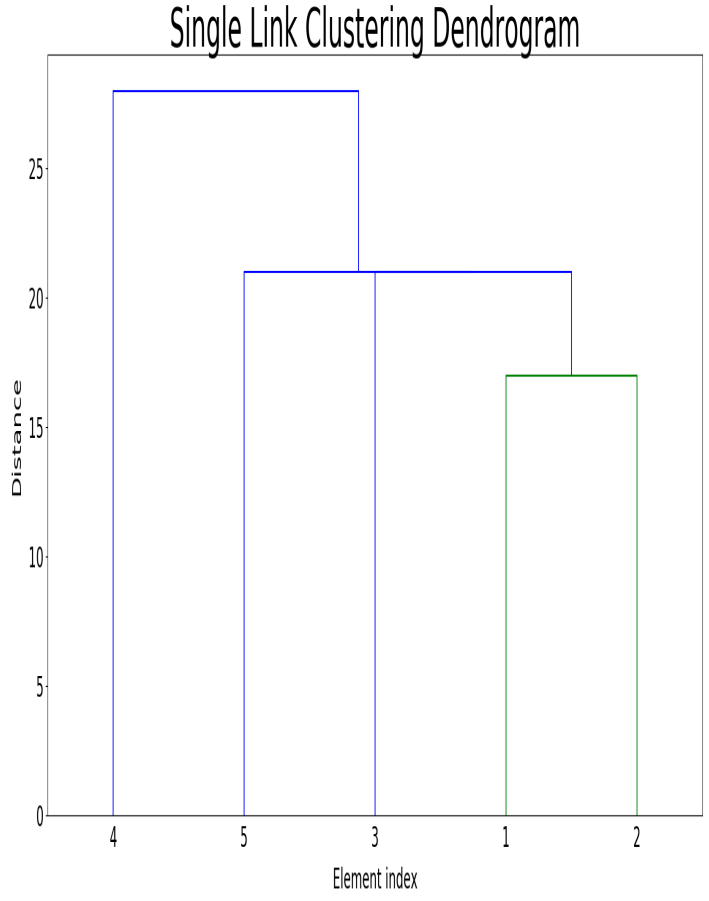
c	$\{s_1, s_2, s_3\}$	s_4	s_5
$\{s_1, s_2, s_3\}$	–	28	21
s_4	–	–	43

Merge III: $\{s_1, s_2, s_3\}$ mit s_5 ergibt den Cluster $\{s_1, s_2, s_3, s_5\}$.

Merge IV: Am Schluss wird dann als letztes s_4 (der unähnlichste Punkt) dazu gemergt (mit Abstand 28).

Single Link Clustering iV

Dendrogramm des Besipiels:



Single Link Clustering Eigenschaften

- Betrachtet nur die zwei ähnlichsten Punkte zwischen zwei Clustern
- Ignoriert andere Punkte
- Monoton: Die ersten merge-similarities sind nie größer als die in späteren Schritten. Im Beispiel: 17, dann 21, dann 21, dann 28
- Bei gleichen Abständen könnte man anstatt paarweise auch gleich alle zusammen mergen (siehe die beiden Schritte mit jeweils 21 Abstand). Dies ist bei complete und average link nicht möglich!

Single Link Clustering und Chaining

Chaining: eine negative Eigenschaft von single link clustering, da nur jeweils ein Punktepaar berücksichtig wird

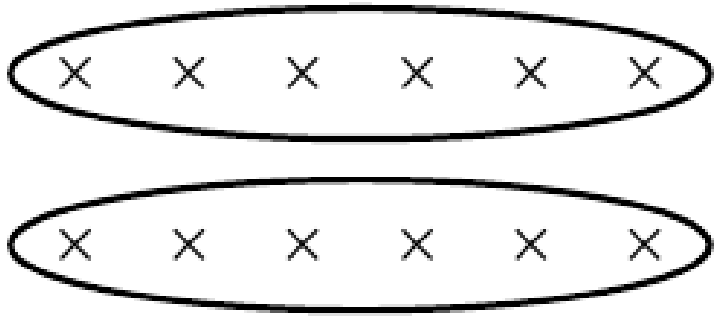


Bild aus Manning et al: Introduction to Information retrieval, Fig 17.6

Complete Link Clustering I

Matrix I:

s	s_1	s_2	s_3	s_4	s_5
s_1	–	17	21	31	23
s_2	–	–	30	34	21
s_3	–	–	–	28	39
s_4	–	–	–	–	43

Bestimme den minimalen Clusterabstand aus Matrix I (17) $\longrightarrow$ Merge I: Neuer Cluster $\{s_1, s_2\}$.

Im ersten Schritt ändert sich nichts im Vergleich zu single-link, da die verschiedenen Distanzen für alle Clusterpaare, die jeweils aus einem Punkt bestehen, gleich sind.

Complete Link Clustering II

Matrix II mit neuen Clustern: Clusterabstand = Maximum (!) aller Distanzen zwischen einzelnen Punkten der Cluster

c	$\{s_1, s_2\}$	s_3	s_4	s_5
$\{s_1, s_2\}$	–	30	34	23
s_3	–	–	28	39
s_4	–	–	–	43

Merge II: Merge $\{s_1, s_2\}$ mit s_5

Complete Link Clustering III

Matrix III mit neuen Clustern:

c	$\{s_1, s_2, s_5\}$	s_3	s_4
$\{s_1, s_2, s_5\}$	–	39	43
s_3	–	–	28

Merge III: Merge s_3 mit s_4

Complete Link Clustering IV

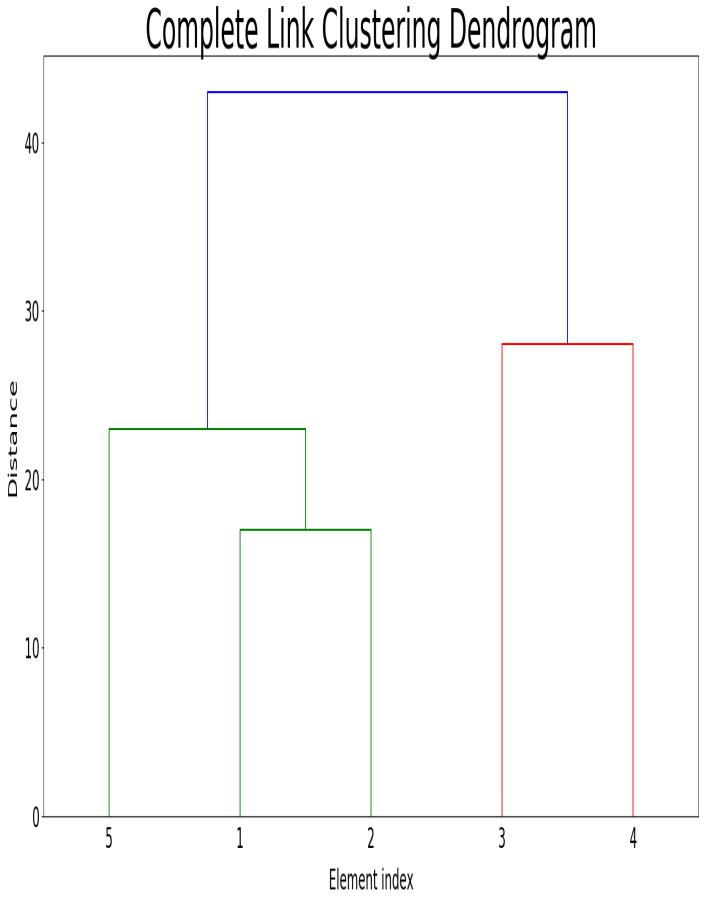
Matrix IV:

c	$\{s_1, s_2, s_5\}$	$\{s_3, s_4\}$
$\{s_1, s_2, s_5\}$	–	43

Merge IV: Merge $\{s_1, s_2, s_5\}$ mit $\{s_3, s_4\}$

Complete Link Clustering 5

Dendrogramm des Besipiels:



Complete Link Clustering Eigenschaften

Vermeidet Chaining (links single link clustering, rechts complete link clustering):

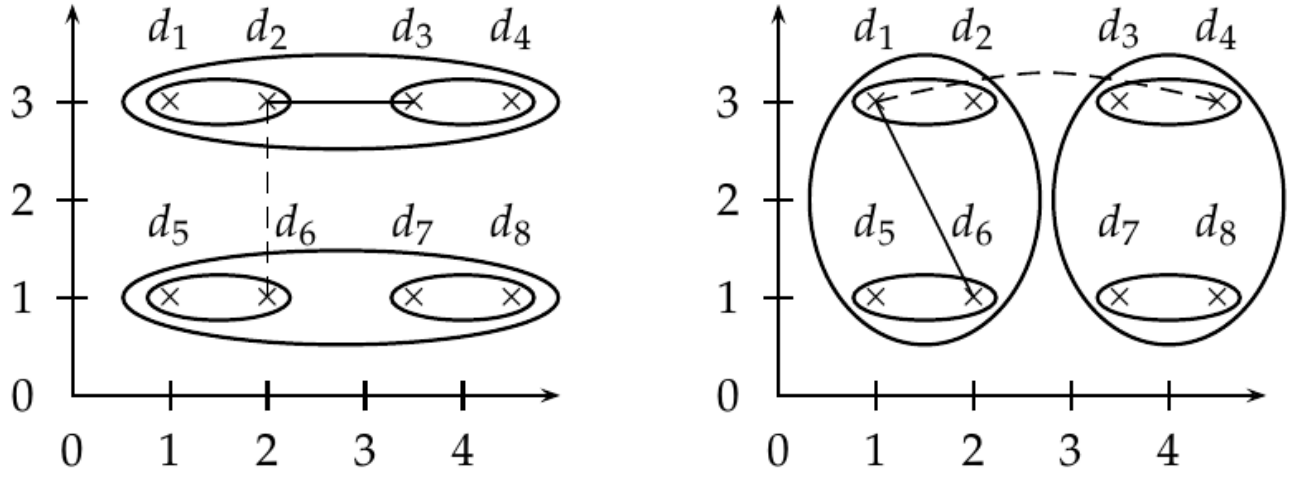
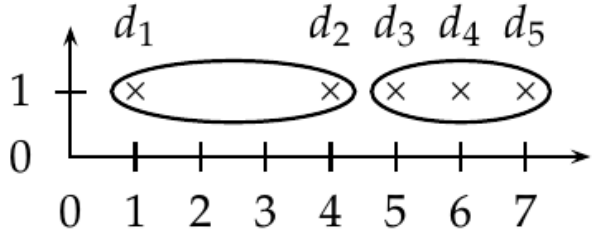


Bild aus Manning et al: Introduction to Information retrieval, Fig 17.4

Auch monoton!

Complete Link Clustering Eigenschaften

Schlecht für outliers:



► **Figure 17.7** Outliers in complete-link clustering. The five documents have the x-coordinates $1 + 2\epsilon, 4, 5 + 2\epsilon, 6$ and $7 - \epsilon$. Complete-link clustering creates the two clusters shown as ellipses. The most intuitive two-cluster clustering is $\{\{d_1\}, \{d_2, d_3, d_4, d_5\}\}$, but in complete-link clustering, the outlier d_1 splits $\{d_2, d_3, d_4, d_5\}$ as shown.

Bild aus Manning et al: Introduction to Information retrieval, Fig 17.7

Average Link Clustering I

Matrix I:

s	s_1	s_2	s_3	s_4	s_5
s_1	–	17	21	31	23
s_2	–	–	30	34	21
s_3	–	–	–	28	39
s_4	–	–	–	–	43

Bestimme den minimalen Clusterabstand aus Matrix I (17) $\longrightarrow$ Merge I: Neuer Cluster $\{s_1, s_2\}$.

Im ersten Schritt ändert sich nichts, da bei Clusterpaaren, die jeweils aus einem Punkt bestehen, average, complete oder single link Distanz gleich sind

Average Link Clustering Beispiel II

Beispielberechnung: Abstand zwischen $\{s_1, s_2\}$ und s_3 als $\frac{1}{2} \cdot (d(s_1, s_3) + d(s_2, s_3)) = 25.5$ mit Abständen aus Vorgängermatrix I.

Matrix II

c	$\{s_1, s_2\}$	s_3	s_4	s_5
$\{s_1, s_2\}$	–	25.5	32.5	22
s_3	–	–	28	39
s_4	–	–	–	43

Merge II: Merge $\{s_1, s_2\}$ mit s_5

Average Link Clustering Beispiel III

Beispielberechnung: Abstand zwischen $\{s_1, s_2, s_5\}$ und s_3 als $\frac{1}{3} \cdot (d(s_1, s_3) + d(s_2, s_3) + d(s_5, s_3)) = 30$ (mit Abständen aus Vorgängermatrix I).

Matrix III:

c	$\{s_1, s_2, s_5\}$	s_3	s_4
$\{s_1, s_2, s_5\}$	–	30	36
s_3	–	–	28

Merge III: Merge $\{s_3, s_4\}$

Average Link Clustering IV

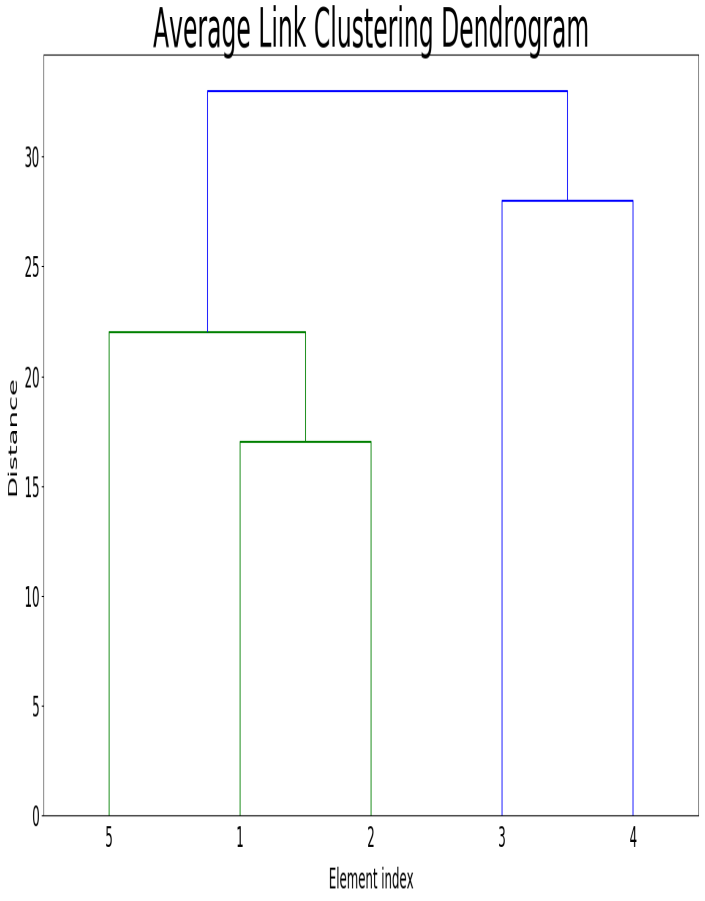
Matrix IV:

c	$\{s_1, s_2, s_5\}$	$\{s_3, s_4\}$
$\{s_1, s_2, s_5\}$	–	33

Merge IV: Merge $\{s_1, s_2, s_5\}$ mit $\{s_3, s_4\}$

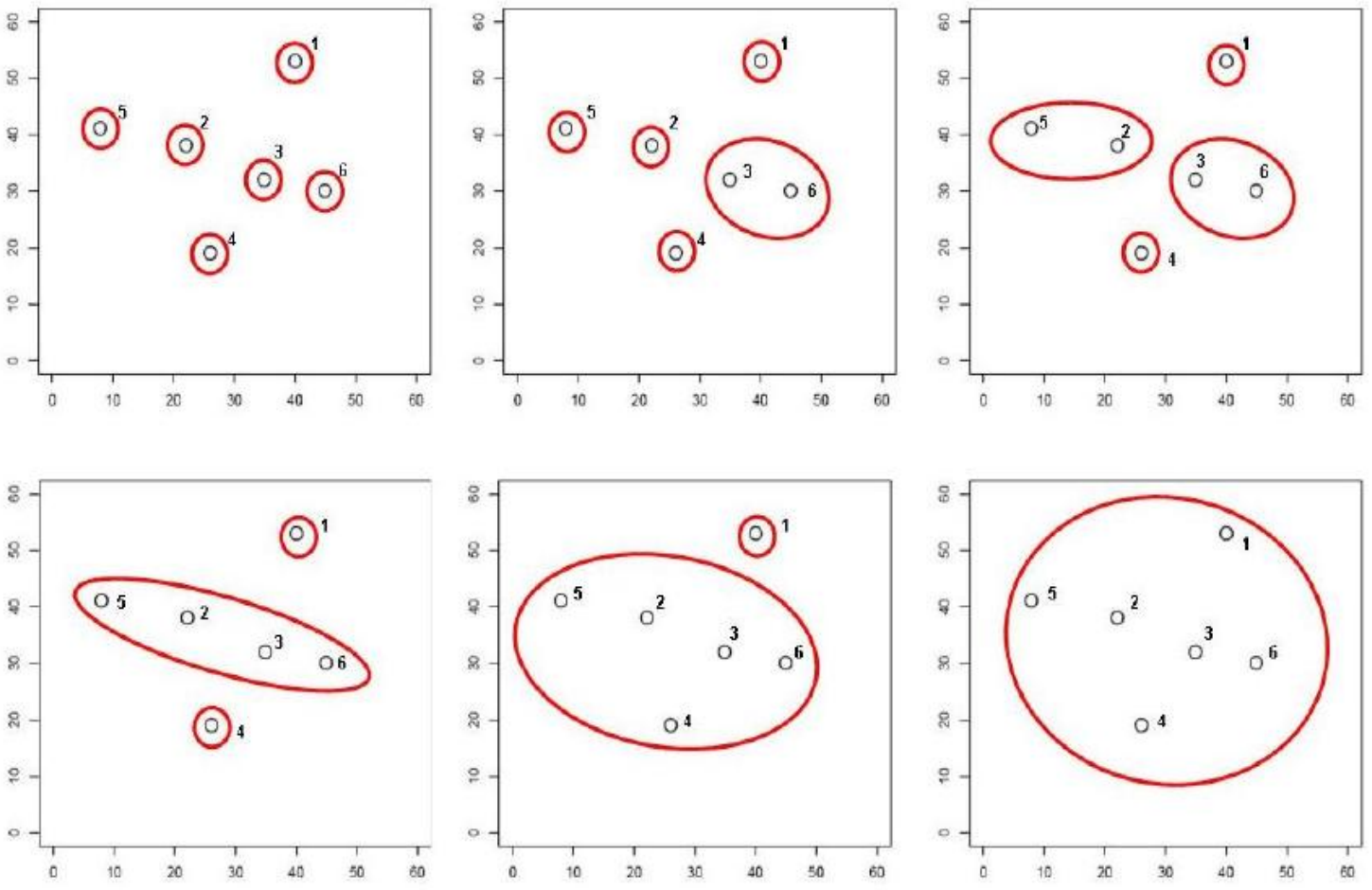
Average Link Clustering 5

Dendrogramm des Besipiels:



Noch ein visuelles Beispiel: Single Link Clustering

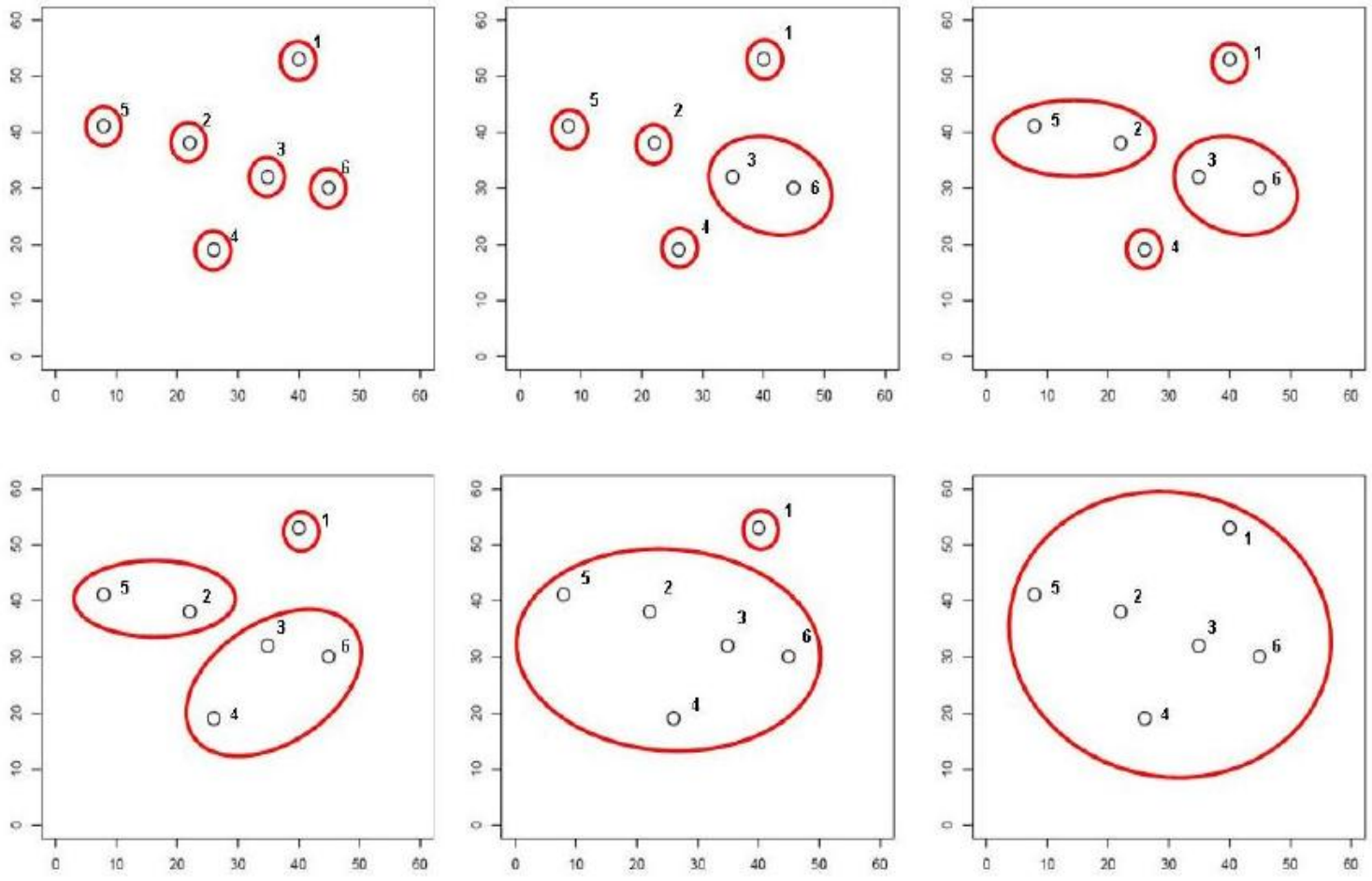
Annahme: Euklidische Distanz



Graphik von Clustergruppe TU München

Noch ein visuelles Beispiel: Average Link Clustering

Annahme. Euklidische Distanz



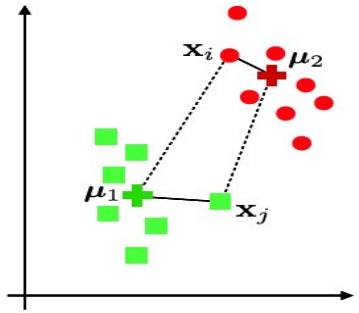
Graphik von Clustergruppe TU München

Übersicht

Flaches Clustering: k-means

D hat n Beobachtungen $\vec{x}_i$ (z.B. Wörter), repräsentiert als Vektoren der Länge m. Ziel: bilde k Cluster

- Teile D in Cluster, so dass jeder Punkt zu dem Cluster gehört, dessen Zentrum $\vec{\mu}_j$ er am nähesten ist



- Ziel: Finde Clusterzuweisung sowie Zentrumsvektoren $\{\vec{\mu}_j\}$ mit $j \in \{1, \ldots k\}$ so dass

$$J = \sum_{i=1}^{n} \|\vec{x}_i - \vec{\mu}_i\|^2$$

minimal ist

k-means: Idee

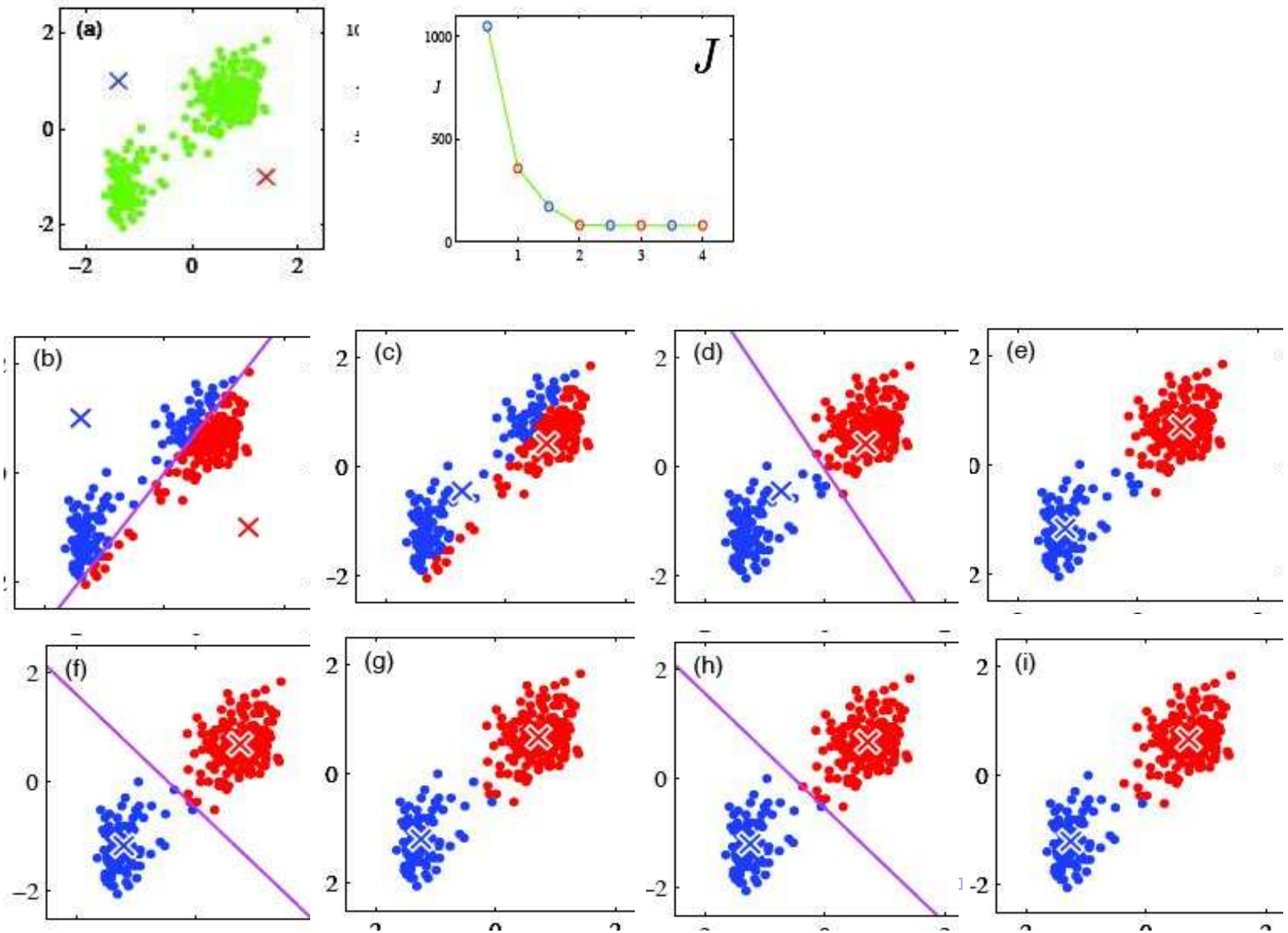
Iterative refinement procedure

1. Initialisierung: Spezifiziere k, die Anzahl der Cluster
2. Initialisierung: Wähle k Punkte $\vec{\mu}_i (i \in \{1...k\})$ als anfängliche Clusterzentren
3. **Assignment step:** Weise jedes $\vec{x}_i$ einem $\vec{\mu}_j$ zu, so dass $\|\vec{x}_i - \vec{\mu}_j\|^2$ minimal. Meist wird hier euklidische Distanz benutzt, d.h.

$$\|\vec{x} - \vec{\mu}\| = \sqrt{\sum_{l=1...m} (x^l - \mu^l)^2}$$

4. **Update step:** Berechne Zentren μ_j der Cluster neu. wobei das Zentrum μ einer Menge von Vektoren c_l definiert ist als $\mu = \frac{1}{|c_l|} \sum_{\vec{x} \in c_l} \vec{x}$
5. Gehe zurück nach 3 und wiederhole bis Konvergenz erreicht ist

K-means illustration

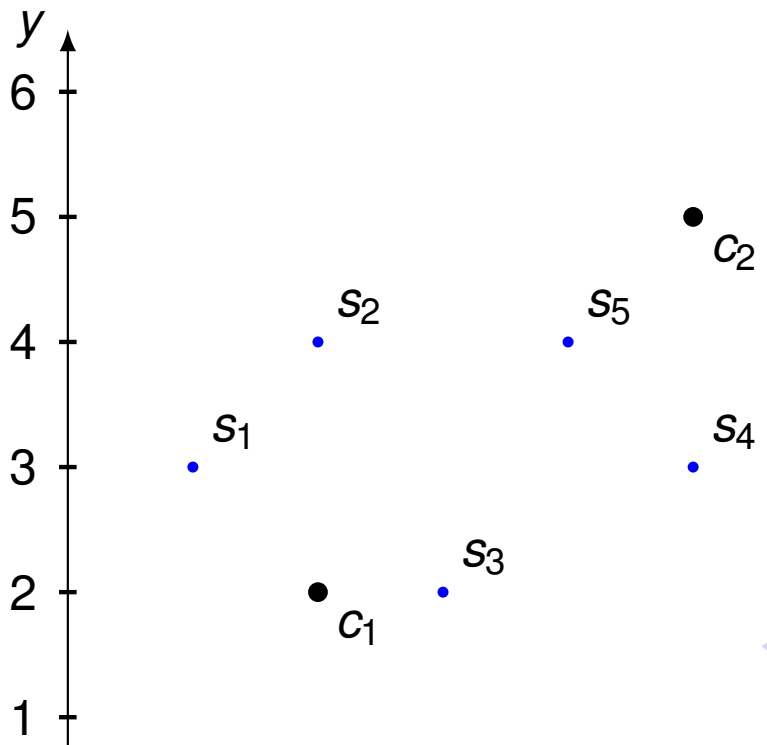


Kleine Beispielrechnung I

#	x	y
s_1	1	3
s_2	2	4
s_3	3	2
s_4	5	3
s_5	4	4

Magie: Initialisierung mit zwei zufällig gewählten Zentroiden (also $k = 2$)

Centroid	x	y
c_1	2	2
c_2	5	5



Kleines Beispiel II

Assignment Step I:

Sample	dist c_1	dist c_2
s_1	**1.4142**	4.4721
s_2	**2.0**	3.1622
s_3	**1.0**	3.6055
s_4	3.1622	**2.0**
s_5	2.8284	**1.4142**

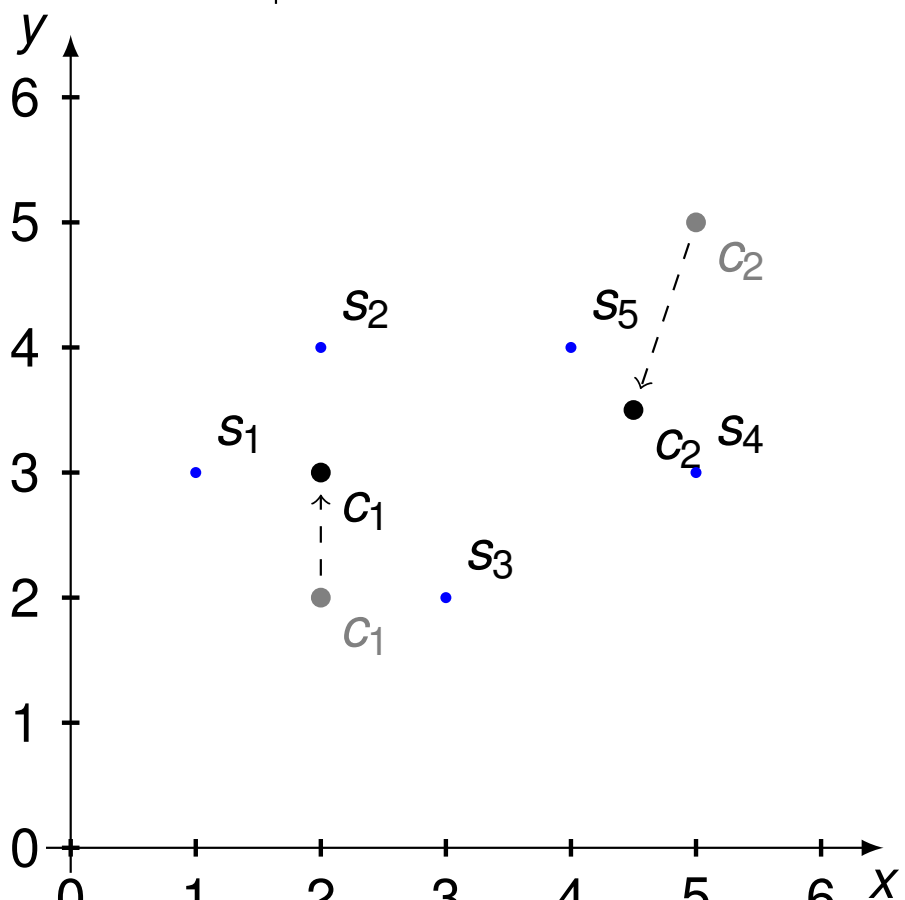
Die Fehlerfunktion J zu diesem Zeitpunkt errechnet sich aus den Distanzen der Punkte zu den jeweiligen Zentroiden.

$J = 1.4142^2 + 2^2 + 1^2 + 2^2 + 1.4142^2 = 13$

Kleines Beispiel III

Update Step I: Berechnung der neuen Zentroiden (Mittelpunkt der bisher gebildeten Cluster (s1, s2, s3) bzw (s4, s5):

Centroid	Coordinate
c_1^1	$(2, 3)$
c_2^1	$(4.5, 3.5)$



Kleines Beispiel IV

Assignment Step II:

Sample	dist c_1^1	dist c_2^1
s_1	**1**	3.53
s_2	**1**	2.54
s_3	**1.41**	2.12
s_4	3	**0.70**
s_5	2.23	**0.70**

Nichts ändert sich: Konvergenz und Stop!

$J = 1^2 + 1^2 + 1.41^2 + 0.7^2 + 0.7^2 = 5$

k-means Details: Stopping Kriterien

- Fixe Anzahl an Iterationen I (Vorteile, Probleme?)
- Keine Änderung der Zuweisung der x_i zu Clustern
- Oder: keine oder sehr kleine Änderung in Zentren
- Fehlerfunktion J wird “sehr klein” (Problem?)
- Fehlerfunktion J ändert sich nicht mehr viel zwischen Iterationen

k-means Details: Konvergenz

- k-means konvergiert immer, da J monoton kleiner wird oder gleich bleibt (und es nur endliche viele Clusterings gibt)
- Voraussetzung: Wenn ein Punkt äquidistant zu zwei Clustern, konsistente Regel
- Konvergiert nicht unbedingt zum globalen Optimum!

k-means Details: Wieviele Cluster?

Initialisierung von k:

- Sehr wichtig
- Meist: **lasse den Algorithmus mit verschiedenen k laufen** und wähle das beste k (Bayesian Information Criterion, Minimum Description length). Am beliebtesten: wenige Cluster, aber auch wenig Varianz innerhalb eines Clusters.

k-means Details: Anfangszentroiden?

Initialisierung der k Clusterzentren

- Sehr wichtig für Performanz
- Einfachste Strategie: zufällig
- Zufällig, aber aus D
- Zufällig, aber aus D und wähle keine outliers
- Besser $k - means++$: zufällig aus D aber sukzessive, wobei Wahrscheinlichkeit der Wahl proportional zu der (quadratischen) Distanz von schon gewählten Zentroiden ist.

k means: Eigenschaften

- Einfach und schnell: $O(lknm)$ wobei l die Anzahl der Iterationen ist und m die Dimensionalität des Vektorraumes
- Hierarchisches Clustering geht durch mehrmaliges Clustern
- Euklidische Distanz nur eine Möglichkeit
- Findet nur **lokales Maximum**, nicht globales.
- Hängt von Initialisierung ab
- Empfindlich gegenüber outliers
- Nicht gut mit **nicht-konvexen Clustern**: braucht ein Zentrum, Kugeln
- Demo: `http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html` oder `stanford.edu/class/ee103/visualizations/kmeans/kmeans.html`

Übersicht

Evaluation: Vergleich mit Expertenclustern

Problem: Wie wissen wir, ob Cluster gut sind (z.B. von gefundenen Wortgruppen).

Eine Möglichkeit: Vergleich mit vorher erstellten Goldclusters! Danach kann man verschiedene Maße verwenden.

Beispiel: Systemcluster in “Bubbles”, Goldcluster indiziert mit x,o und Raute

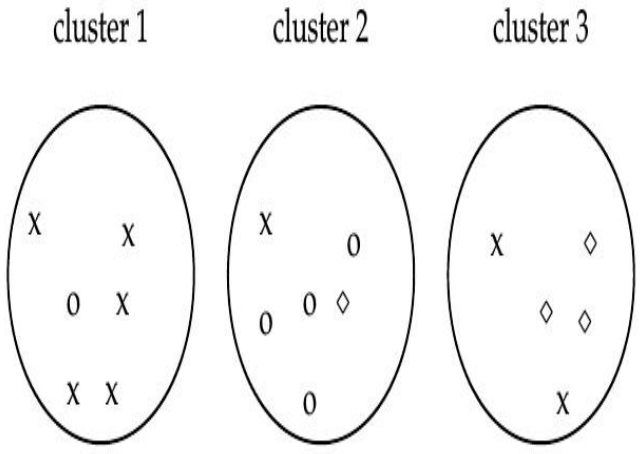


Bild aus Manning et al: Introduction to Information Retrieval Fig 16.4

Maß I: Purity

Jeder Systemcluster wird dem Goldcluster, dessen Einträge er am häufigsten enthält, zugewiesen.

$$purity(S, G) = \frac{1}{n} \sum_{K} \max_{l} |s_k \cap g_l|$$

wobei

- n Anzahl der Datenpunkte
- $S = \{s_1, s_2 \ldots s_K\}$ die Systemcluster
- $G = \{g_1, g_2, \ldots g_L\}$ die Goldcluster

Purity Beispielrechnung

Systemcluster in “Bubbles”, Goldcluster indiziert mit x,o und Raute

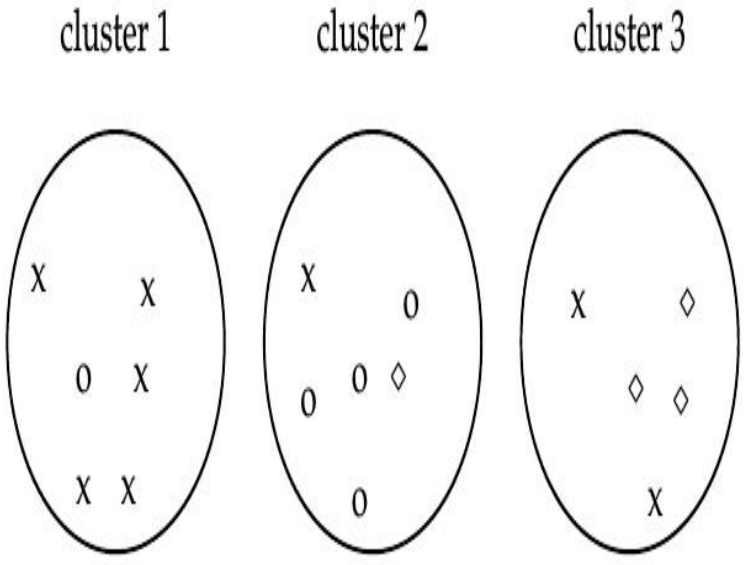


Bild aus Manning et al: Introduction to Information Retrieval Fig 16.4

$$\mathit{Purity} = \frac{1}{17} \cdot (5+4+3) = 0.71$$

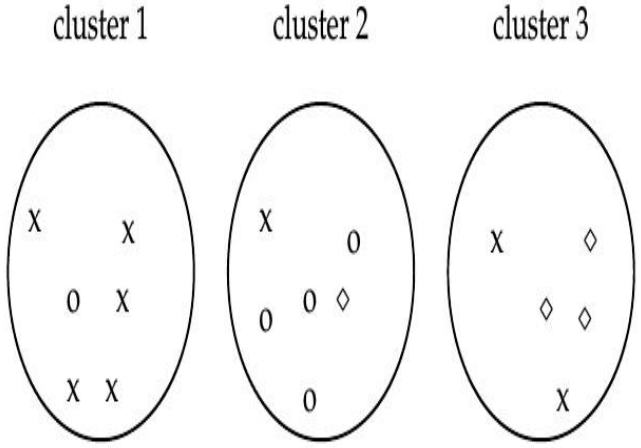
Problem: Eine Purity von 1 ist leicht zu erreichen. Wie?

Maß II: Clustering als Linkklassifikation

Man behandle die Clusteringentscheidung als eine Serie von Klassifikationsentscheidungen einzelner Links:

- Positiv (System): zwei Punkte sind im gleichen Systemcluster
- Negativ (System): zwei Punkte sind in unterschiedlichem Systemcluster
- Positiv (Gold): zwei Punkte sind im gleichen Goldcluster
- Negativ (Gold): zwei Punkte sind in unterschiedlichem Goldcluster

Im Beispiel



Alle Paare: $\binom{17}{2} = 136$

	pos im System	neg im System	
pos im Gold	TP= 20	FN = 24	44
neg im Gold	FP= 20	TN= 72	92
	40	96	136

Accuracy = Randindex = 92/136 = 0.68

Nachteil?

Maß III: Informationstheoretische Evaluation

Trade-off zwischen Purity und Anzahl der Cluster via Normalized Mutual Information

$$NMI(S,G) = \frac{MI(S,G)}{(H(S)+H(G))/2}$$

wobei

$$MI(S,G) = \sum_k \sum_l \frac{|s_k \cap g_l|}{n} \log \frac{n \cdot |s_k \cap g_l|}{|s_k||g_l|}$$

und

$$H(S) = -\sum_k \frac{|s_k|}{n} \log \frac{|s_k|}{n}$$

Siehe auch Manning et al 329 ff

$0 \cdot \log 0 := 0$

Im Beispiel I

$$
\begin{aligned}
MI(S,G) &= P(k=1,l=x)\log\frac{P(k=1,l=x)}{P(k=1)P(l=x)} \\
&+ P(k=1,l=kreis)\log\frac{P(k=1,l=kreis)}{P(k=1)P(l=kreis)} \\
&+ P(k=1,l=raute)\log\frac{P(k=1,l=raute)}{P(k=1)P(l=raute)} \\
&+ P(k=2,l=x)\log\frac{P(k=2,l=x)}{P(k=2)P(l=x)} \\
&+ P(k=2,l=kreis)\log\frac{P(k=2,l=kreis)}{P(k=2)P(l=kreis)} \\
&+ P(k=2,l=raute)\log\frac{P(k=2,l=raute)}{P(k=2)P(l=raute)} \\
&+ P(k=3,l=x)\log\frac{P(k=3,l=x)}{P(k=3)P(l=x)} \\
&+ P(k=3,l=kreis)\log\frac{P(k=3,l=kreis)}{P(k=3)P(l=kreis)} \\
&+ P(k=3,l=raute)\log\frac{P(k=3,l=raute)}{P(k=3)P(l=raute)} \\
&= 5/17\log\frac{5/17}{6/17\cdot 8/17}+\ldots = 00.5625
\end{aligned}
$$

Im Beispiel II

Entropie der Goldcluster:

$$H(G) = -(\frac{8}{17}\log\frac{8}{17} + \frac{5}{17}\log\frac{5}{17} + \frac{4}{17}\log\frac{4}{17}) = 1.51$$

Entropie der Systemcluster:

$$H(S) = -(\frac{6}{17}\log\frac{6}{17} + \frac{6}{17}\log\frac{6}{17} + \frac{5}{17}\log\frac{5}{17}) = 1.57$$

Und damit

$$NMI(G,S) = \frac{0.5635}{(1.51+1.57)/2} = 0.36$$

Zusammenfassung

- Clustering vs. Klassifikation: Clustering ist unüberwacht, Klassen sind noch nicht bekannt
- Hierarchisches vs. flaches Clustering
- Hierarchisches Clustering: single-link vs. complete link vs. average-link
- ***k*-means algorithm**: flaches Clustering
- Evaluation: entweder mit expertengenerierten Clustern oder innerhalb einer Anwendung

Leseempfehlung: Manning et al, Introduction to IR Kapitel 16 (bis einschl. 16.3) und 17