Kapitel ML: IX

IX. Clusteranalyse
- Einordnung Data Mining
- Einführung in die Clusteranalyse
- Hierarchische Verfahren
- Iterative Verfahren
- Dichtebasierte Verfahren
- Cluster Evaluation
- Constrained Cluster Analysis

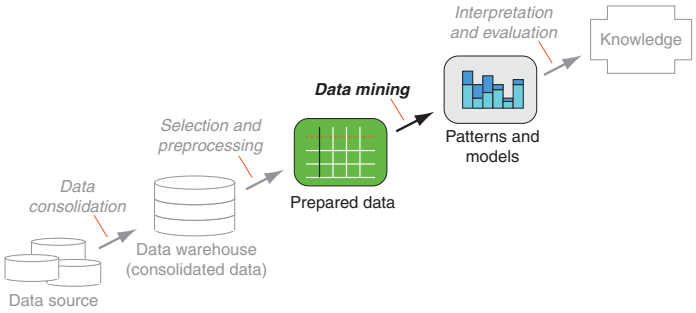
Einordnung Data Mining

Definition 1 (Data Mining)
Unter Data Mining versteht man das systematische, in der Regel automatisierte oder halbautomatische Entdecken und Extrahieren bislang unbekannter Zusammenhänge aus großen Mengen von Daten.

Data Mining umfasst folgende Schritte:

1. Aufgabendefinition
2. Datenselektion
3. Datenvorbereitung und -transformation
4. Mustererkennung
5. Kommunikation, Präsentation

Einordnung Data Mining



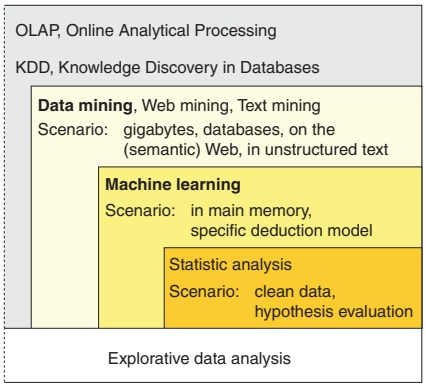
Definition 2 (Knowledge Discovery in Databases, KDD)
Wissensentdeckung in Datenbanken *(Knowledge Discovery in Databases)* ist der nichttriviale Prozess der Identifikation gültiger, neuer, potentiell nützlicher und schlussendlich verständlicher Muster in großen Datenbeständen.

[vgl. Fayyad 1996, Wrobel 1998]

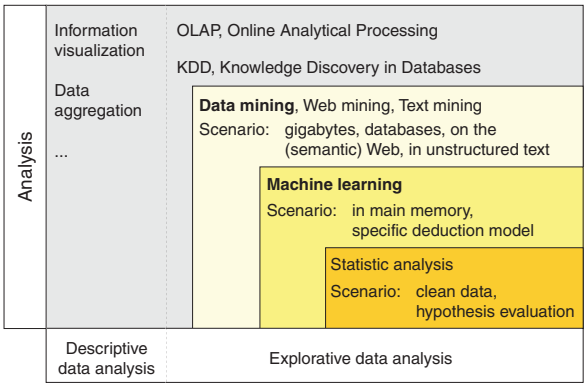
Bemerkungen:

- Data-Mining-Techniken werden der *explorativen Datenanalyse* zugeordnet. Ziel der explorativen Datenanalyse ist – über die Darstellung der Daten hinaus – die Suche nach Strukturen und Besonderheiten. Die explorative Datenanalyse wird eingesetzt, wenn die Fragestellung oder die Wahl eines geeigneten statistischen Modells unklar ist.
- In der Data-Mining-Definition wird auf den Begriff der Information verzichtet: Data Mining wird der sigmatischen Ebene der Semiotik zugeordnet.
  Die *Interpretation* der entdeckten Muster, also die im Rahmen der explorativen Datenanalyse stattfindende Auseinandersetzung mit Informationen im Sinne eines subjektiven Wissenszuwachses, die auf der pragmatischen Ebene abläuft, gehört in das Gebiet des Knowledge Discovery in Databases, KDD.
- Vor allem im kommerziellen Bereich wird der Begriff des Data Mining synonym zu Wissensentdeckung in Datenbanken (KDD) verwendet. Data Mining ist aber nur ein Teilschritt innerhalb des KDD-Prozesses, nämlich der Analyseschritt zur Mustererkennung.
- Unter Web Mining versteht man die Übertragung von Techniken des Data Mining zur (teil)automatischen Extraktion von Informationen aus dem Internet, speziell dem World Wide Web.
  Mit Text Mining wird die Entdeckung neuer und für den Benutzer relevanter Informationen aus Textdaten bezeichnet.

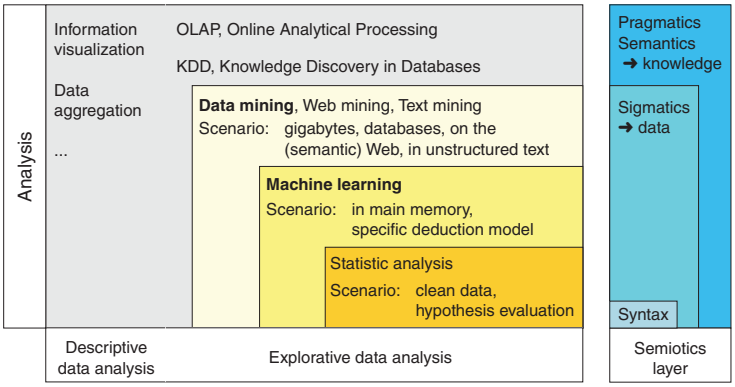
Einordnung Data Mining



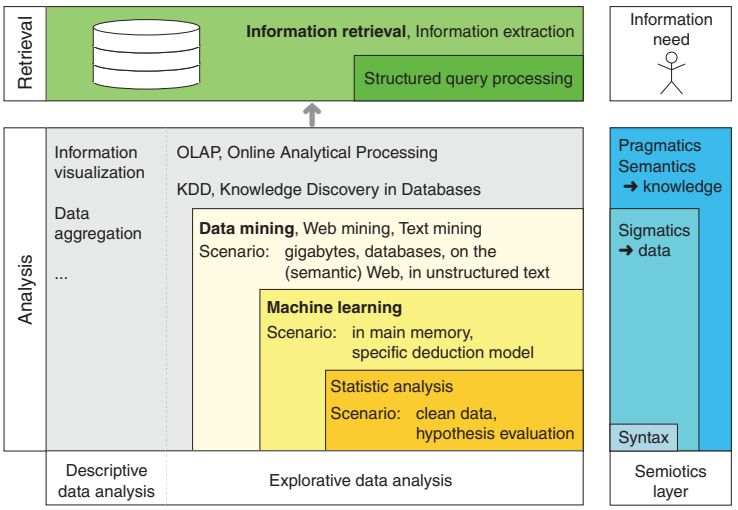
Einordnung Data Mining



Einordnung Data Mining



Einordnung Data Mining



Bemerkungen:

- Eine klare Abgrenzung von Machine Learning und Data Mining ist nicht immer möglich; ein wichtiger Unterschied resultiert aus der Größe der behandelten Datenmengen: Anwendungen des Machine Learning laufen typischerweise im Hauptspeicher ab; die Disziplin des Data Mining entstand aus der Notwendigkeit, maschinelle Analyseverfahren auf große Datenbanken anzuwenden.
- Ein Schwerpunkt des Machine Learning ist der eigentliche Lern- bzw. Deduktionsprozess wie die Theorie des analogen Schließens, das Lernen aus Beispielen, oder der Einfluss der Umwelt auf das Lernen. Hingegen ist die treibende Kraft hinter Data Mining die Industrie- und Geschäftswelt mit ihren großen Datenbanken.
- Zu den bekannten Aufgabenstellung des Data Mining gehören: ungerichtete Abhängigkeitsanalyse zur Identifikation signifikanter Dependenzen zwischen den Attributen eines Informationsobjektes (Beispiel: Warenkorbanalyse), Gruppenbildung und Klassifikationsprobleme, Filtern von Prozessdaten, Prognoseaufgaben.

Einordnung Data Mining

Methoden und Techniken

- Clusteranalyse
- propositionale (oder relative) Regellernverfahren
- assoziative Regellernverfahren
- Hauptkomponenten- und Faktoranalyse

Kapitel ML: IX (Fortsetzung)

Einführung in die Clusteranalyse

Clusteranalyse ist die unüberwachte Klassifikation einer Menge von Objekten in Gruppen; dabei wird folgendes Ziel verfolgt:

1. Ähnlichkeit innerhalb der Gruppen maximieren
2. Ähnlichkeit zwischen den Gruppen minimieren

Einführung in die Clusteranalyse

Clusteranalyse ist die unüberwachte Klassifikation einer Menge von Objekten in Gruppen; dabei wird folgendes Ziel verfolgt:

1. Ähnlichkeit innerhalb der Gruppen maximieren
2. Ähnlichkeit zwischen den Gruppen minimieren

Anwendungen

- Identifikation gleichartiger Käufergruppen
- „höhere“ Bildverarbeitung, im Sinne von Objekterkennung
- Suche nach ähnlichen Genprofilen
- Spezifikation von Syndromen
- Analyse von Verkehrsdaten in Computernetzen
- Visualisierung komplexer Graphen
- Textkategorisierung im Information-Retrieval

Bemerkungen:

- Die Problemstellung der Clusteranalyse ist umgekehrt zur Problemstellung der Varianzanalyse. Ziel der Varianzanalyse ist die Überprüfung, ob eine gegebene nominalskalierte Variable Gruppen definiert, deren Mitglieder sich in abhängigen (intervallskalierten) Variablen unterscheiden. Ziel der Clusteranalyse ist die Erzeugung einer solchen nominalskalierten Variable durch die Entdeckung der Ausprägungen dieser Variable. Jedes Cluster korrespondiert zu einer Variablenausprägung.
- Die Clusteranalyse ist ein Verfahren zur Struktur*erzeugung*: Man weiß quasi nichts über die zu entdeckende Variable, insbesondere nichts über die Anzahl ihrer Werteausprägungen. Die Varianzanalyse ist ein Verfahren zur Struktur*prüfung*.

Einführung in die Clusteranalyse

$\mathbf{x}_1, \ldots \mathbf{x}_n$ sind die zu n Objekten gehörenden p-dimensionalen Merkmalsvektoren:

	Merkmal 1	Merkmal 2	...	Merkmal p
$\mathbf{x}_1$	x_{1_1}	x_{1_2}	...	x_{1_p}
$\mathbf{x}_2$	x_{2_1}	x_{2_2}	...	x_{2_p}
$\vdots$				
$\mathbf{x}_n$	x_{n_1}	x_{n_2}	...	x_{n_p}

Einführung in die Clusteranalyse

$\mathbf{x}_1, \ldots \mathbf{x}_n$ sind die zu n Objekten gehörenden p-dimensionalen Merkmalsvektoren:

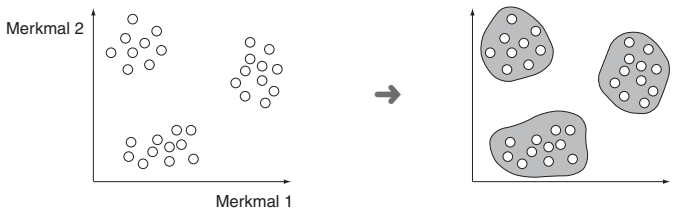
	Merkmal 1	Merkmal 2	...	Merkmal p	kein Zielkonzept
$\mathbf{x}_1$	x_{1_1}	x_{1_2}	...	x_{1_p}	c_1
$\mathbf{x}_2$	x_{2_1}	x_{2_2}	...	x_{2_p}	c_2
$\vdots$					$\vdots$
$\mathbf{x}_n$	x_{n_1}	x_{n_2}	...	x_{n_p}	c_n

Einführung in die Clusteranalyse

$\mathbf{x}_1, \ldots \mathbf{x}_n$ sind die zu n Objekten gehörenden p-dimensionalen Merkmalsvektoren:

	Merkmal 1	Merkmal 2	...	Merkmal p	kein Zielkonzept
$\mathbf{x}_1$	x_{1_1}	x_{1_2}	...	x_{1_p}	c_1
$\mathbf{x}_2$	x_{2_1}	x_{2_2}	...	x_{2_p}	c_2
$\vdots$					$\vdots$
$\mathbf{x}_n$	x_{n_1}	x_{n_2}	...	x_{n_p}	c_n

30 zweidimensionale Merkmalsvektoren ($n = 30$, $p = 2$) :



Einführung in die Clusteranalyse

Definition 2 (exklusives Clustering [vgl. Splitting]**)**
Sei X eine Menge von Merkmalsvektoren. Ein exklusives Clustering $\mathcal{C}$ von X, $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$, $C_i \subseteq X$, ist eine Aufteilung von X in nichtleere und paarweise disjunkte Teilmengen C_i mit $\bigcup_{C_i \in \mathcal{C}} C_i = X$.

Einführung in die Clusteranalyse

Definition 2 (exklusives Clustering [vgl. Splitting]**)**
Sei X eine Menge von Merkmalsvektoren. Ein exklusives Clustering $\mathcal{C}$ von X, $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$, $C_i \subseteq X$, ist eine Aufteilung von X in nichtleere und paarweise disjunkte Teilmengen C_i mit $\bigcup_{C_i \in \mathcal{C}} C_i = X$.

Algorithmen zur Clusteranalyse sind unüberwachte Lernverfahren:

- der Lernprozess ist selbstorganisiert
- es gibt keinen externen Lehrer
- es gibt ein aufgaben*unabhängiges* Optimierungskriterium

Einführung in die Clusteranalyse

Definition 2 (exklusives Clustering [vgl. Splitting]**)**
Sei X eine Menge von Merkmalsvektoren. Ein exklusives Clustering $\mathcal{C}$ von X, $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$, $C_i \subseteq X$, ist eine Aufteilung von X in nichtleere und paarweise disjunkte Teilmengen C_i mit $\bigcup_{C_i \in \mathcal{C}} C_i = X$.

Algorithmen zur Clusteranalyse sind unüberwachte Lernverfahren:

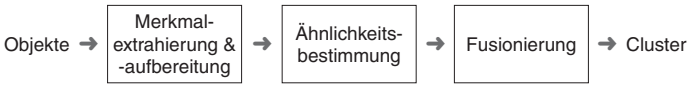
- der Lernprozess ist selbstorganisiert
- es gibt keinen externen Lehrer
- es gibt ein aufgaben*unabhängiges* Optimierungskriterium

Überwachtes Lernen:

- es gibt Lernziele: das Zielkonzept, gewünschte Reaktionen, etc.
- es gibt ein aufgaben*abhängiges* Optimierungskriterium
- es gibt Information darüber, *wie* eine Verbesserung im Optimierungskriterium zu erzielen ist. Stichwort: Instructive Feedback

Einführung in die Clusteranalyse

Grundschritte einer Clusteranalyse



Einführung in die Clusteranalyse

Merkmalextrahierung und -aufbereitung

Gesucht sind evtl. neue Merkmale mit hoher Varianz. Techniken:

- Analyse von Streuungsparametern
- Dimensionsreduktion: Faktoranalyse, multidimensionale Skalierung
- Visuelle Analyse: Scatter-Plots, Box-plots

Einführung in die Clusteranalyse

Merkmalextrahierung und -aufbereitung

Gesucht sind evtl. neue Merkmale mit hoher Varianz. Techniken:

- Analyse von Streuungsparametern
- Dimensionsreduktion: Faktoranalyse, multidimensionale Skalierung
- Visuelle Analyse: Scatter-Plots, Box-plots

Standardisierung von Variablen (Merkmalen) ist problematisch:

Einführung in die Clusteranalyse

Berechnung von Distanzen oder Ähnlichkeiten

	Merkmal 1	Merkmal 2	...	Merkmal p
$\mathbf{x}_1$	x_{1_1}	x_{1_2}	...	x_{1_p}
$\mathbf{x}_2$	x_{2_1}	x_{2_2}	...	x_{2_p}
$\vdots$				
$\mathbf{x}_n$	x_{n_1}	x_{n_2}	...	x_{n_p}

$\rightarrow$

	$\mathbf{x}_1$	$\mathbf{x}_2$	...	$\mathbf{x}_n$
$\mathbf{x}_1$	0	$d(\mathbf{x}_1, \mathbf{x}_2)$	...	$d(\mathbf{x}_1, \mathbf{x}_n)$
$\mathbf{x}_2$	-	0	...	$d(\mathbf{x}_2, \mathbf{x}_n)$
$\vdots$				
$\mathbf{x}_n$	-	-	...	0

Bemerkungen:

- Die Distanzmatrix ist oft implizit durch eine Metrik auf dem Merkmalsraum definiert.
- Die Distanzmatrix kann als Adjazenzmatrix eines gewichteten, ungerichteten Graphen G, $G = \langle V, E, w \rangle$, interpretiert werden: Die Menge X der Merkmalsvektoren wird bijektiv auf eine Knotenmenge V abgebildet; eine Distanz $d(\mathbf{x}_i, \mathbf{x}_j)$ entspricht dem Gewicht $w(\{u, v\})$ der Kante $\{u, v\} \in E$ zwischen den mit $\mathbf{x}_i$ und $\mathbf{x}_j$ assoziierten Knoten u und v.

Einführung in die Clusteranalyse

Berechnung von Distanzen oder Ähnlichkeiten (Fortsetzung)

Anforderungen an eine Distanzfunktion:

1. $d(\mathbf{x}_1, \mathbf{x}_2) \geq 0$
2. $d(\mathbf{x}_1, \mathbf{x}_1) = 0$
3. $d(\mathbf{x}_1, \mathbf{x}_2) = d(\mathbf{x}_2, \mathbf{x}_1)$
4. $d(\mathbf{x}_1, \mathbf{x}_3) \leq d(\mathbf{x}_1, \mathbf{x}_2) + d(\mathbf{x}_2, \mathbf{x}_3)$

Einführung in die Clusteranalyse

Berechnung von Distanzen oder Ähnlichkeiten (Fortsetzung)

Anforderungen an eine Distanzfunktion:

1. $d(\mathbf{x}_1, \mathbf{x}_2) \geq 0$
2. $d(\mathbf{x}_1, \mathbf{x}_1) = 0$
3. $d(\mathbf{x}_1, \mathbf{x}_2) = d(\mathbf{x}_2, \mathbf{x}_1)$
4. $d(\mathbf{x}_1, \mathbf{x}_3) \leq d(\mathbf{x}_1, \mathbf{x}_2) + d(\mathbf{x}_2, \mathbf{x}_3)$

Bei intervallskalierten Variablen Verwendung der Minkowsky-Metrik:

$$d(\mathbf{x}_1, \mathbf{x}_2) = \Big(\sum_{i=1}^{p} |x_{1_i} - x_{2_i}|^r \Big)^{1/r}$$

mit

- $r = 1$. Manhatten- oder Hamming-Distanz, L_1-Norm
- $r = 2$. Euklidische Distanz, L_2-Norm
- $r = \infty$. Maximum-Distanz, L_∞-Norm bzw. $L_{\max}$-Norm

Einführung in die Clusteranalyse

Berechnung von Distanzen oder Ähnlichkeiten (Fortsetzung)

Eine Clusteranalyse verlangt kein spezielles Skalenniveau der Merkmale.

➜ Verallgemeinerung der Distanzfunktion zur (Un)Ähnlichkeitsfunktion durch Verzicht auf die Dreiecksungleichung. (Un)Ähnlichkeiten lassen sich zwischen binären, nominalen und ordinalen Variablen quantifizieren.

Einführung in die Clusteranalyse

Berechnung von Distanzen oder Ähnlichkeiten (Fortsetzung)

Eine Clusteranalyse verlangt kein spezielles Skalenniveau der Merkmale.

➜ Verallgemeinerung der Distanzfunktion zur (Un)Ähnlichkeitsfunktion durch Verzicht auf die Dreiecksungleichung. (Un)Ähnlichkeiten lassen sich zwischen binären, nominalen und ordinalen Variablen quantifizieren.

Ähnlichkeitskoeffizienten für zwei Vektoren, $\mathbf{x}_1$, $\mathbf{x}_2$, mit binären Merkmalen:

$$\text{Simple Matching Coefficient (SMC)} = \frac{f_{11} + f_{00}}{f_{11} + f_{00} + f_{01} + f_{10}}$$

$$\text{Jaccard-Koefficient (J)} = \frac{f_{11}}{f_{11} + f_{01} + f_{10}}$$

mit

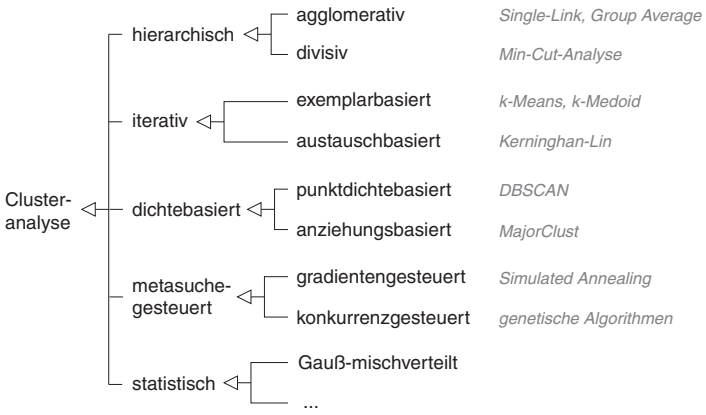
- f_{11} = Anzahl der Merkmale mit Ausprägung 1 sowohl in $\mathbf{x}_1$ als auch in $\mathbf{x}_2$
- f_{00} = Anzahl der Merkmale mit Ausprägung 0 sowohl in $\mathbf{x}_1$ als auch in $\mathbf{x}_2$
- f_{01} = Anzahl der Merkmale mit Ausprägung 0 in $\mathbf{x}_1$ und Ausprägung 1 in $\mathbf{x}_2$
- f_{10} = Anzahl der Merkmale mit Ausprägung 1 in $\mathbf{x}_1$ und Ausprägung 0 in $\mathbf{x}_2$

Bemerkungen:

- Die Definition der Ähnlichkeitskoeffizienten lässt sich auf nominale Variablen erweitern.
- Heterogene Metriken wie HEOM und HVDM ermöglichen die kombinierte Verrechnung verschiedener Skalenniveaus.
- Die Berechnung von Korrelationskoeffizienten zwischen zwei Vektoren über alle Merkmale (also nicht zwischen zwei Merkmalen über alle Vektoren) ermöglicht den Vergleich von Profilen. Beispiel: Q-Korrelationskoeffizient
- Die Konstruktion eines geeigneten Ähnlichkeitsmaßes stellt oft die größte Herausforderung bei der Cluster-Analyse dar. Typische Problemfelder:
  - Normalisierung
  - Empfindlichkeit bei Ausreißern
  - Korrelationen zwischen Merkmalen
  - unterschiedliche Wichtigkeit der Merkmale
- Ähnlichkeitsmaße lassen sich kanonisch in Unähnlichkeitsmaße umrechnen – und umgekehrt.

Einführung in die Clusteranalyse

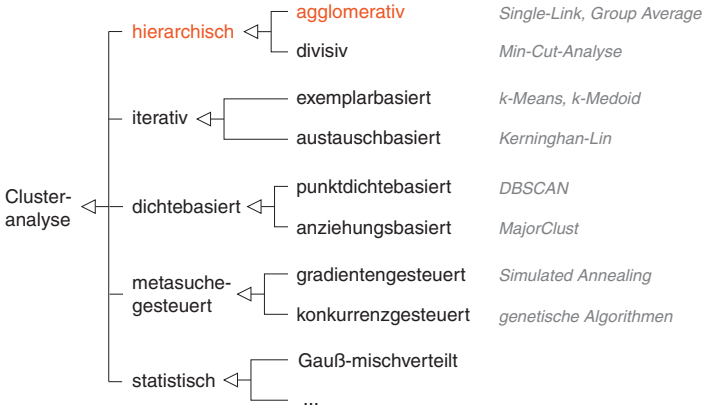
Prinzipien der Fusionierung



Kapitel ML: IX (Fortsetzung)

Hierarchische Verfahren

Prinzipien der Fusionierung



Hierarchische Verfahren

Algorithmus zur hierarchisch-agglomerativen Clusteranalyse

Input: $G = \langle V, E, w \rangle$. Weighted graph.
$d_{\mathcal{C}}$. Distance measure between two clusters.
Output: $T = \langle V_T, E_T \rangle$. Cluster hierarchy or dendrogram.

1. $\mathcal{C} = \{\{v\} \mid v \in V\}$ `// initial clustering`
2.
3. **WHILE** $|\mathcal{C}| > 1$ **DO**
4. *update_distance_matrix*$(\mathcal{C}, G, d_{\mathcal{C}})$
5. $\{C, C'\} = \underset{\{C_i, C_j\} \in \mathcal{C}: C_i \neq C_j}{\text{argmin}} d_{\mathcal{C}}(C_i, C_j)$
6. $\mathcal{C} = (\mathcal{C} \setminus \{C, C'\}) \cup \{C \cup C'\}$ `// merging`
7.
8. **ENDDO**
9. **RETURN**(T)

Vergleiche hierzu den Algorithmus zur hierarchisch-divisiven Clusteranalyse.

Hierarchische Verfahren

Algorithmus zur hierarchisch-agglomerativen Clusteranalyse

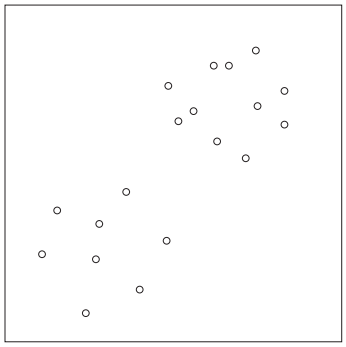
Input: $G = \langle V, E, w \rangle$. Weighted graph.
$d_{\mathcal{C}}$. Distance measure between two clusters.
Output: $T = \langle V_T, E_T \rangle$. Cluster hierarchy or dendrogram.

1. $\mathcal{C} = \{\{v\} \mid v \in V\}$ `// initial clustering`
2. $V_T = \{v_C \mid C \in \mathcal{C}\}, \; E_T = \emptyset$ `// initial dendrogram`
3. **WHILE** $|\mathcal{C}| > 1$ **DO**
4. *update_distance_matrix*$(\mathcal{C}, G, d_{\mathcal{C}})$
5. $\{C, C'\} = \underset{\{C_i, C_j\} \in \mathcal{C}: C_i \neq C_j}{\text{argmin}} d_{\mathcal{C}}(C_i, C_j)$
6. $\mathcal{C} = (\mathcal{C} \setminus \{C, C'\}) \cup \{C \cup C'\}$ `// merging`
7. $V_T = V_T \cup \{v_{C,C'}\}, \; E_T = E_T \cup \{\{v_{C,C'}, v_C\}, \{v_{C,C'}, v_{C'}\}\}$ `// dendrogram`
8. **ENDDO**
9. **RETURN**(T)

Vergleiche hierzu den Algorithmus zur hierarchisch-divisiven Clusteranalyse.

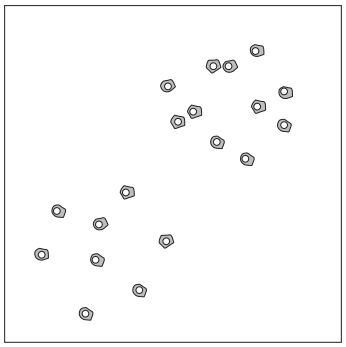
Hierarchische Verfahren

Single-Link: Cluster-Distanzmaß $d_{\mathcal{C}}$ = Nearest-Neighbor



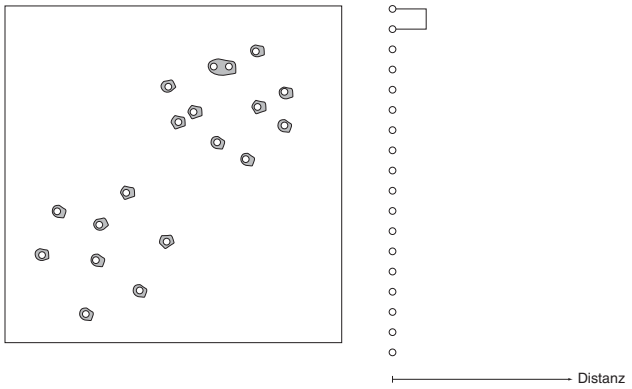
Hierarchische Verfahren

Single-Link: Cluster-Distanzmaß $d_{\mathcal{C}}$ = Nearest-Neighbor



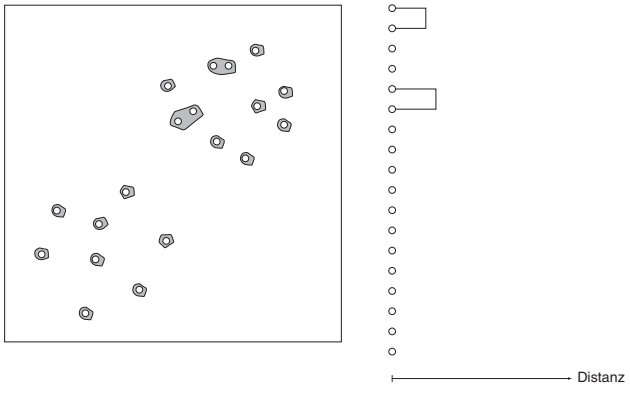
Hierarchische Verfahren

Single-Link: Cluster-Distanzmaß $d_{\mathcal{C}}$ = Nearest-Neighbor



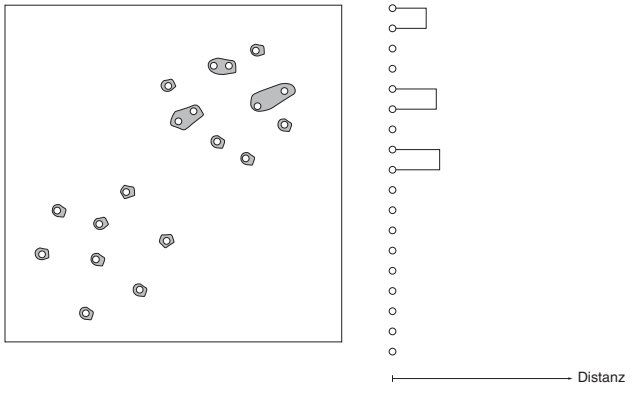
Hierarchische Verfahren

Single-Link: Cluster-Distanzmaß $d_{\mathcal{C}}$ = Nearest-Neighbor



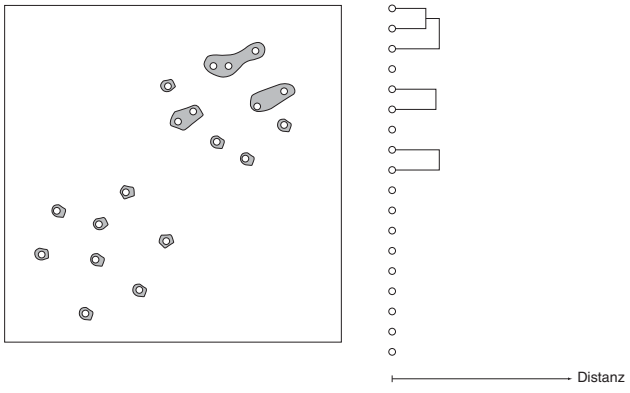
Hierarchische Verfahren

Single-Link: Cluster-Distanzmaß $d_{\mathcal{C}}$ = Nearest-Neighbor



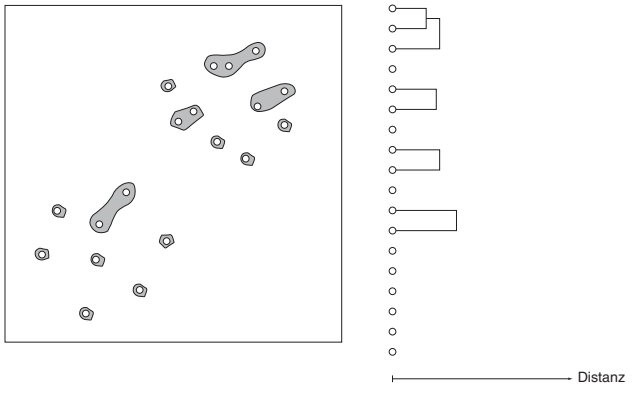
Hierarchische Verfahren

Single-Link: Cluster-Distanzmaß $d_{\mathcal{C}}$ = Nearest-Neighbor



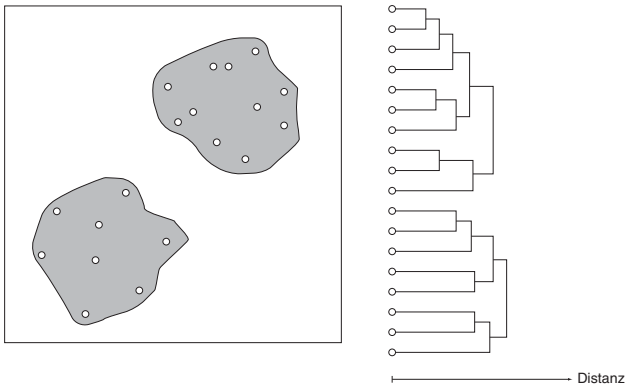
Hierarchische Verfahren

Single-Link: Cluster-Distanzmaß $d_{\mathcal{C}}$ = Nearest-Neighbor



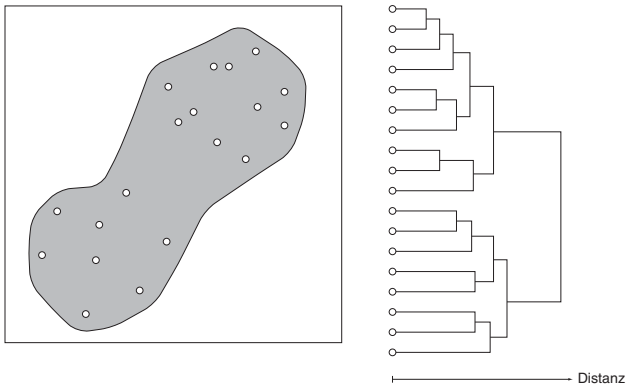
Hierarchische Verfahren

Single-Link: Cluster-Distanzmaß $d_{\mathcal{C}}$ = Nearest-Neighbor



Hierarchische Verfahren

Single-Link: Cluster-Distanzmaß $d_{\mathcal{C}}$ = Nearest-Neighbor



Hierarchische Verfahren

Distanzmaße hierarchisch-agglomerativer Verfahren [Eigenschaften]

$d_{\mathcal{C}}(C, C') = \min_{\substack{u \in C \\ v \in C'}} d(u, v)$	Single-Link (Nearest-Neighbor)
$d_{\mathcal{C}}(C, C') = \max_{\substack{u \in C \\ v \in C'}} d(u, v)$	Complete-Link (Furthest-Neighbor)
$d_{\mathcal{C}}(C, C') = \frac{1}{\lvert C\rvert \cdot \lvert C'\rvert} \sum_{\substack{u \in C \\ v \in C'}} d(u, v)$	(Group-)Average-Link
$d_{\mathcal{C}}(C, C') = \sqrt{\frac{2 \cdot \lvert C\rvert \cdot \lvert C'\rvert}{\lvert C\rvert + \lvert C'\rvert}} \cdot \lvert\lvert \bar{u} - \bar{v}\rvert\rvert$	Ward (Varianz)

Siehe Verwendung im Algorithmus:

- hierarchisch-agglomerative Clusteranalyse
- hierarchisch-divisive Clusteranalyse

Hierarchische Verfahren

Ward-Kriterium

Ward ist ein Varianzkriterium; es entspricht der doppelten Zunahme der Wurzel aus der Fehlerquadratsumme, *SSE*, in dem neuen Cluster, der durch die Vereinigung der beiden Cluster C und C' entsteht. Herleitung:

$$\mathit{SSE}(C) = \sum_{u \in C} ||\bar{u} - u||^2$$

Hierarchische Verfahren

Ward-Kriterium

Ward ist ein Varianzkriterium; es entspricht der doppelten Zunahme der Wurzel aus der Fehlerquadratsumme, *SSE*, in dem neuen Cluster, der durch die Vereinigung der beiden Cluster C und C' entsteht. Herleitung:

$$\begin{aligned}\mathit{SSE}(C) &= \sum_{u \in C} ||\bar{u} - u||^2 = \textstyle\sum_{u \in C}(||\bar{u}||^2 - 2 \cdot \langle u, \bar{u}\rangle + ||u||^2) \\ &= |C| \cdot ||\bar{u}||^2 - 2|C| \cdot ||\bar{u}||^2 + \sum_{u \in C} ||u||^2 = \sum_{u \in C} ||u||^2 \; - |C| \cdot ||\bar{u}||^2\end{aligned}$$

Hierarchische Verfahren

Ward-Kriterium

Ward ist ein Varianzkriterium; es entspricht der doppelten Zunahme der Wurzel aus der Fehlerquadratsumme, *SSE*, in dem neuen Cluster, der durch die Vereinigung der beiden Cluster C und C' entsteht. Herleitung:

$$\begin{aligned}\mathit{SSE}(C) &= \sum_{u \in C} ||\bar{u} - u||^2 = \textstyle\sum_{u \in C}(||\bar{u}||^2 - 2 \cdot \langle u, \bar{u}\rangle + ||u||^2) \\ &= |C| \cdot ||\bar{u}||^2 - 2|C| \cdot ||\bar{u}||^2 + \sum_{u \in C} ||u||^2 = \sum_{u \in C} ||u||^2 \; - |C| \cdot ||\bar{u}||^2\end{aligned}$$

$$\mathit{SSE}(C') = \sum_{v \in C'} ||v||^2 \; - |C'| \cdot ||\bar{v}||^2$$

$$\mathit{SSE}(C \cup C') = \sum_{w \in (C \cup C')} ||w||^2 \; - |C \cup C'| \cdot ||\bar{w}||^2, \quad \text{mit } \bar{w} = \frac{|C| \cdot \bar{u} + |C'| \cdot \bar{v}}{|C| + |C'|}$$

$$\mathit{SSE}(C \cup C') - \mathit{SSE}(C) - \mathit{SSE}(C') \; = \ldots = \frac{|C| \cdot |C'|}{|C| + |C'|} \cdot ||\bar{u} - \bar{v}||^2$$

$\bar{u}$ bzw. $\bar{v}$ bezeichnen den Mittelwert der Punkte $u \in C$ bzw. $v \in C'$.

Hierarchische Verfahren

Update-Formel für Cluster-Distanzen

Nach Vereinigung der Cluster C und C' zu einem neuen Cluster sind dessen Distanzen, in Zeichen: $d_{\mathcal{C}}(C \cup C', C_i)$, zu den anderen Clustern C_i zu berechnen.

Die folgende Update-Formel (Lance-Williams-Formel) ermöglicht eine effiziente Berechnung dieser Distanzen:

$$\begin{aligned} d_{\mathcal{C}}(C \cup C', C_i) = \; & \alpha \cdot d_{\mathcal{C}}(C, C_i) + \\ & \beta \cdot d_{\mathcal{C}}(C', C_i) + \\ & \gamma \cdot d_{\mathcal{C}}(C, C') + \\ & \delta \cdot |d_{\mathcal{C}}(C, C_i) - d_{\mathcal{C}}(C', C_i)| \end{aligned}$$

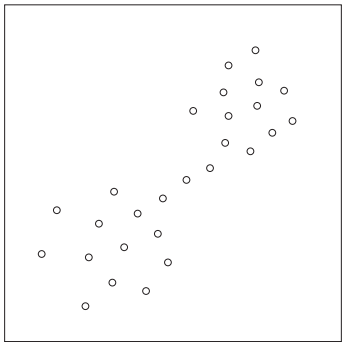
Die Konstanten $\alpha, \beta, \gamma, \delta$ sind spezifisch für Single-Link, Complete-Link, Average-Link und Ward, und lassen sich auf Basis der entsprechenden Berechnungsvorschriften für $d_{\mathcal{C}}$ herleiten.

Bemerkungen:

- Link-basierte Verfahren arbeiten sowohl mit beliebigen Distanz- als auch Ähnlichkeitsmaßen.
- Single-Link kann unmittelbar mit einem Minimum-Spanning-Tree-Algorithmus realisiert werden.
- Varianzbasierte Verfahren sind nur sinnvoll, falls alle Merkmale Intervallskalennivau besitzen.
- Idee der Lance-Williams-Formel: Anstatt immer wieder alle Elemente von zwei Clustern zu betrachten, nimmt die Update-Formel Bezug auf die im vorherigen Schritt berechneten Distanzen.

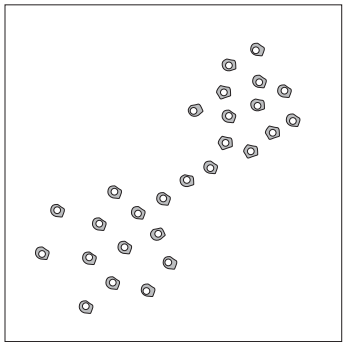
Hierarchische Verfahren

Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = Nearest-Neighbor)



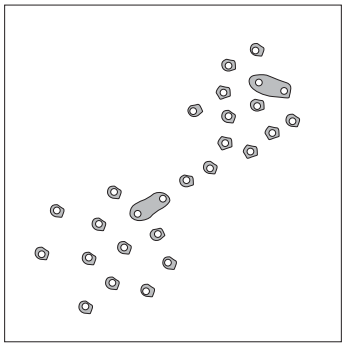
Hierarchische Verfahren

Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = Nearest-Neighbor)



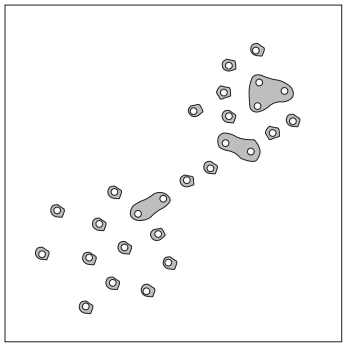
Hierarchische Verfahren

Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = Nearest-Neighbor)



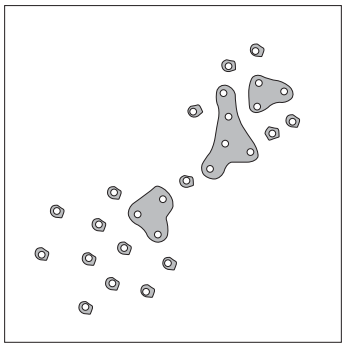
Hierarchische Verfahren

Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = Nearest-Neighbor)



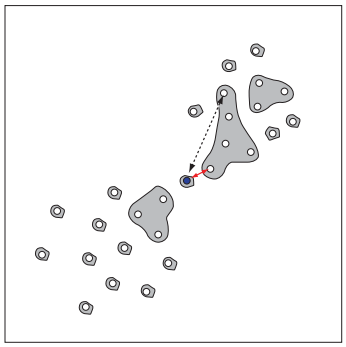
Hierarchische Verfahren

Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = Nearest-Neighbor)



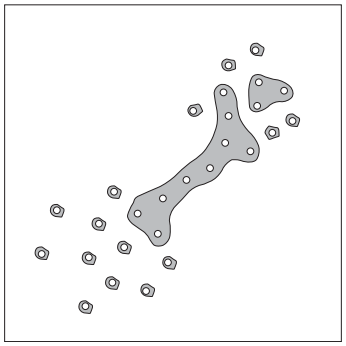
Hierarchische Verfahren

Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = Nearest-Neighbor)



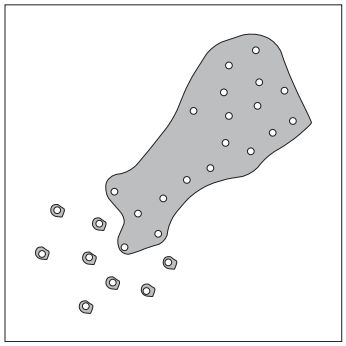
Hierarchische Verfahren

Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = Nearest-Neighbor)



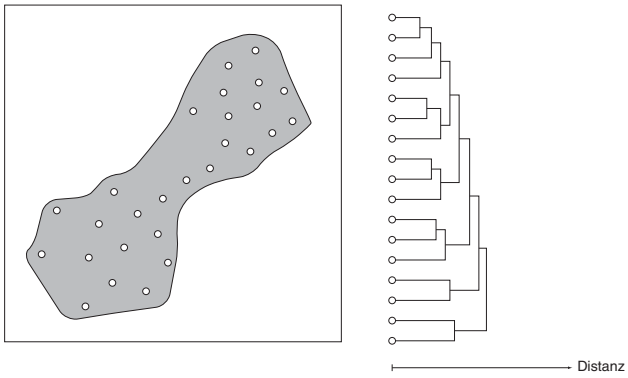
Hierarchische Verfahren

Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = Nearest-Neighbor)



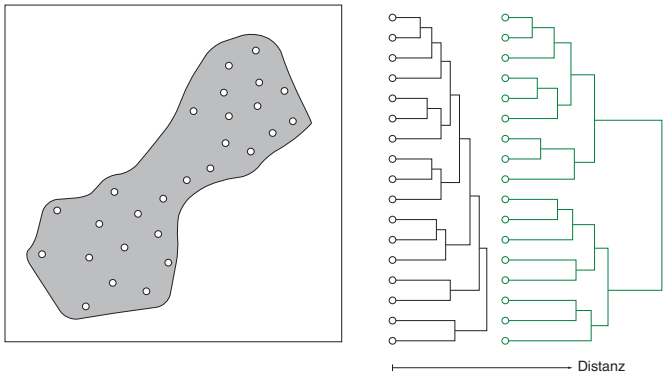
Hierarchische Verfahren

Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = Nearest-Neighbor)



Hierarchische Verfahren

Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = Nearest-Neighbor)

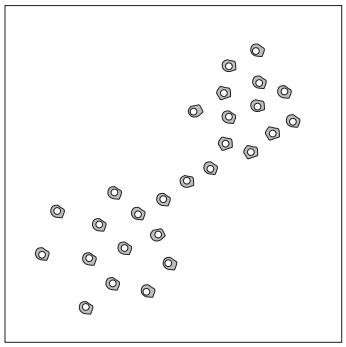


Bemerkungen:

- Mit einer k-Nearest-Neighbor-Variante könnte man das Problem entschärfen.
- Bei k-Nearest-Neighbor werden größere Cluster bei der Agglomeration bevorzugt, da sie mehr und damit – statistisch gesehen – auch mehr nähere Nachbarn besitzen.

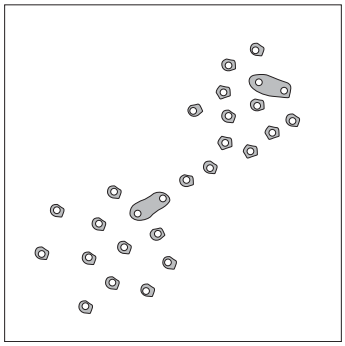
Hierarchische Verfahren

Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



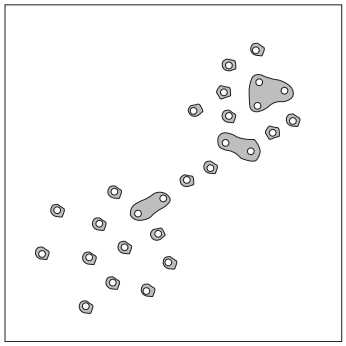
Hierarchische Verfahren

Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



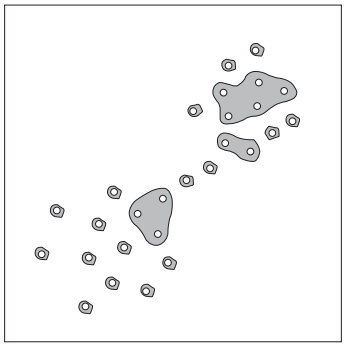
Hierarchische Verfahren

Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



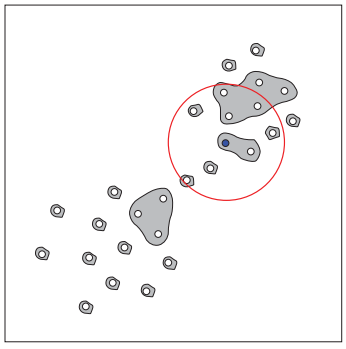
Hierarchische Verfahren

Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



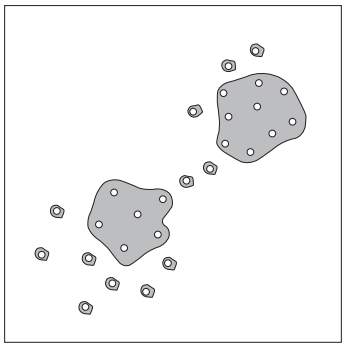
Hierarchische Verfahren

Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



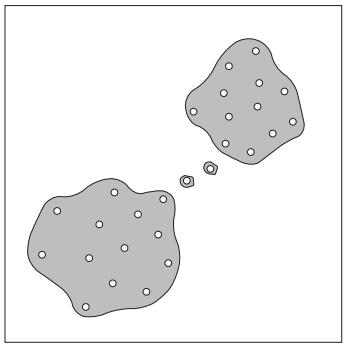
Hierarchische Verfahren

Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



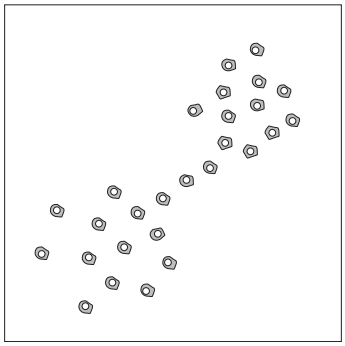
Hierarchische Verfahren

Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



Hierarchische Verfahren

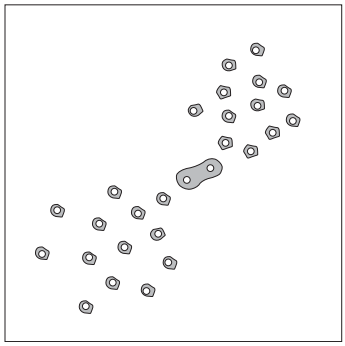
Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



In speziellen Situationen kann auch k-Nearest-Neighbor versagen.

Hierarchische Verfahren

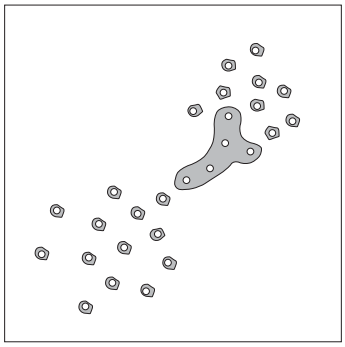
Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



In speziellen Situationen kann auch k-Nearest-Neighbor versagen.

Hierarchische Verfahren

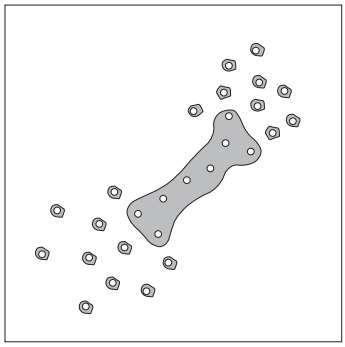
Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



In speziellen Situationen kann auch k-Nearest-Neighbor versagen.

Hierarchische Verfahren

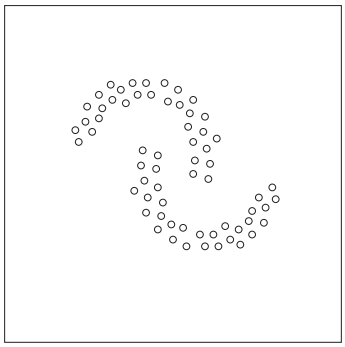
Chaining-Problematik bei Single-Link ($d_{\mathcal{C}}$ = k-Nearest-Neighbor)



In speziellen Situationen kann auch k-Nearest-Neighbor versagen.

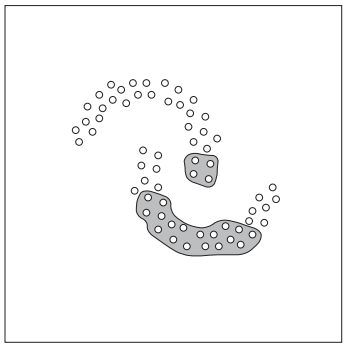
Hierarchische Verfahren

Überlappungsproblematik bei Complete-Link ($d_{\mathcal{C}}$ = Furthest-Neighbor)



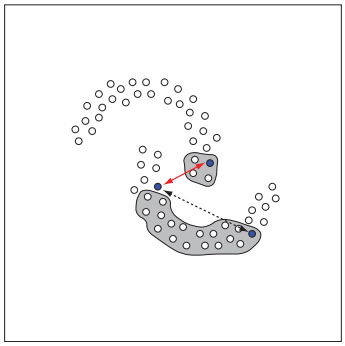
Hierarchische Verfahren

Überlappungsproblematik bei Complete-Link ($d_{\mathcal{C}}$ = Furthest-Neighbor)



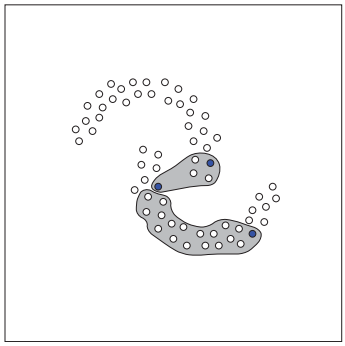
Hierarchische Verfahren

Überlappungsproblematik bei Complete-Link ($d_{\mathcal{C}}$ = Furthest-Neighbor)



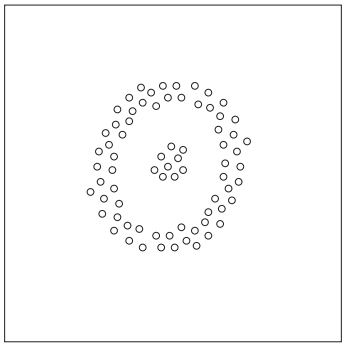
Hierarchische Verfahren

Überlappungsproblematik bei Complete-Link ($d_{\mathcal{C}}$ = Furthest-Neighbor)



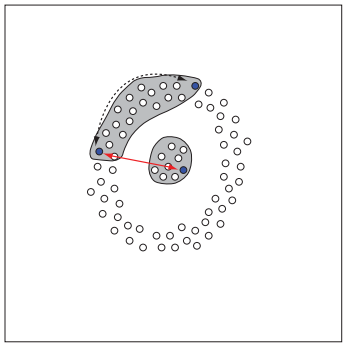
Hierarchische Verfahren

Überlappungsproblematik bei Complete-Link ($d_{\mathcal{C}}$ = Furthest-Neighbor)



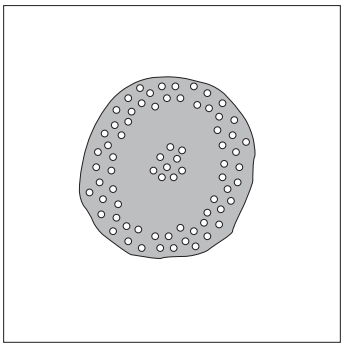
Hierarchische Verfahren

Überlappungsproblematik bei Complete-Link ($d_{\mathcal{C}}$ = Furthest-Neighbor)



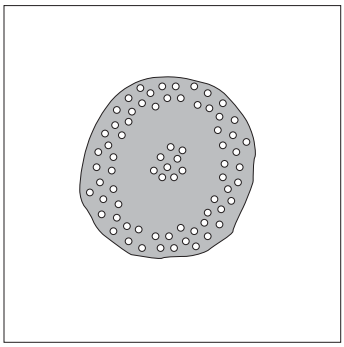
Hierarchische Verfahren

Überlappungsproblematik bei Complete-Link (d_C = Furthest-Neighbor)

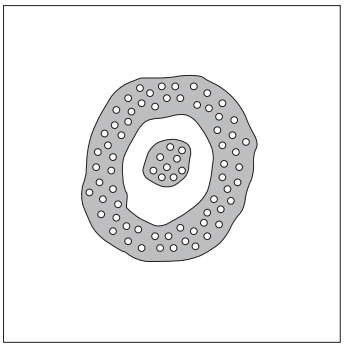


Hierarchische Verfahren

Überlappungsproblematik bei Complete-Link (d_C = Furthest-Neighbor)



Wirklichkeit



Wunsch

Hierarchische Verfahren

Eigenschaften hierarchisch-agglomerativer Verfahren [Distanzmaße]

Geometrische Eigenschaften:

	Single-Link	Complete-Link	Average-Link	Ward
Charakteristik	kontrahierend:	dilatierend:	konservativ:	konservativ:
Cluster-Zahl	niedrig	hoch	mittel	mittel
Cluster-Form	ausgedehnt	klein	kompakt	sphärisch
Verkettungstendenz	stark	niedrig	niedrig	niedrig
Ausreißerentdeckung	sehr gut	schlecht	mittel	mittel

Hierarchische Verfahren

Eigenschaften hierarchisch-agglomerativer Verfahren [Distanzmaße]

Geometrische Eigenschaften:

	Single-Link	Complete-Link	Average-Link	Ward
Charakteristik	kontrahierend:	dilatierend:	konservativ:	konservativ:
Cluster-Zahl	niedrig	hoch	mittel	mittel
Cluster-Form	ausgedehnt	klein	kompakt	sphärisch
Verkettungstendenz	stark	niedrig	niedrig	niedrig
Ausreißerentdeckung	sehr gut	schlecht	mittel	mittel

Datenbezogene Eigenschaften:

verrauschte Daten	empfindlich	empfindlich	unbeeinflusst	unbeeinflusst
Merkmaltransformation	invariant	invariant	–	–

Hierarchische Verfahren

Eigenschaften hierarchisch-agglomerativer Verfahren [Distanzmaße]

Geometrische Eigenschaften:

	Single-Link	Complete-Link	Average-Link	Ward
Charakteristik	kontrahierend:	dilatierend:	konservativ:	konservativ:
Cluster-Zahl	niedrig	hoch	mittel	mittel
Cluster-Form	ausgedehnt	klein	kompakt	sphärisch
Verkettungstendenz	stark	niedrig	niedrig	niedrig
Ausreißerentdeckung	sehr gut	schlecht	mittel	mittel

Datenbezogene Eigenschaften:

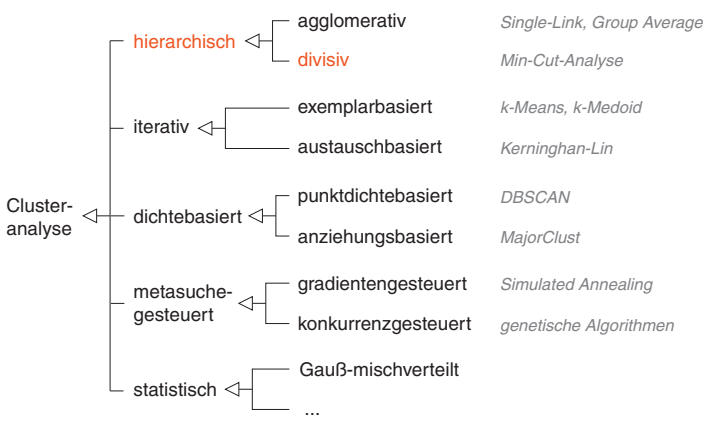
verrauschte Daten	empfindlich	empfindlich	unbeeinflusst	unbeeinflusst
Merkmaltransformation	invariant	invariant	–	–

Eigenschaften des Cluster-Distanzmaßes:

d_C monoton	✓	✓	✓	✓
d_C reihenfolgeunabh.	✓	✓	✓	✓
d_C konsistent	$\longrightarrow 0$	$\longrightarrow \infty$	✓	$\longrightarrow \infty$

Hierarchische Verfahren

Prinzipien der Fusionierung



Hierarchische Verfahren

Algorithmus zur hierarchisch-divisiven Clusteranalyse

Input: $G = \langle V, E, w \rangle$. Weighted graph.
$d_{\mathcal{C}}$. Distance measure between two clusters.
Output: $T = \langle V_T, E_T \rangle$. Cluster hierarchy or dendrogram.

1. $\mathcal{C} = \{V\}$ // initial clustering
2.
3. **WHILE** $\exists C_x : (C_x \in \mathcal{C} \ \wedge \ |C_x| > 1)$ **DO**
4. $\{C, C'\} = \underset{\substack{\{C_i, C_j\}: \\ C_i \cup C_j = C_x \ \wedge \ C_i \cap C_j = \emptyset}}{\text{argmax}} d_{\mathcal{C}}(C_i, C_j)$
5. $\mathcal{C} = (\mathcal{C} \setminus \{C_x\}) \ \cup \ \{C, C'\}$ // splitting
6.
7. **ENDDO**
8. **RETURN**(T)

Vergleiche hierzu den Algorithmus zur hierarchisch-agglomerativen Clusteranalyse.

Hierarchische Verfahren

Algorithmus zur hierarchisch-divisiven Clusteranalyse

Input: $G = \langle V, E, w \rangle$. Weighted graph.
$d_{\mathcal{C}}$. Distance measure between two clusters.
Output: $T = \langle V_T, E_T \rangle$. Cluster hierarchy or dendrogram.

1. $\mathcal{C} = \{V\}$ // initial clustering
2. $V_T = \{v_C \mid C \in \mathcal{C}\}, \ E_T = \emptyset$ // initial dendrogram
3. **WHILE** $\exists C_x : (C_x \in \mathcal{C} \ \wedge \ |C_x| > 1)$ **DO**
4. $\{C, C'\} = \underset{\substack{\{C_i, C_j\}: \\ C_i \cup C_j = C_x \ \wedge \ C_i \cap C_j = \emptyset}}{\text{argmax}} d_{\mathcal{C}}(C_i, C_j)$
5. $\mathcal{C} = (\mathcal{C} \setminus \{C_x\}) \ \cup \ \{C, C'\}$ // splitting
6. $V_T = V_T \cup \{v_C, v_{C'}\}, \ E_T = E_T \cup \{\{v_{C_x}, v_C\}, \{v_{C_x}, v_{C'}\}\}$ // dendrogram
7. **ENDDO**
8. **RETURN**(T)

Vergleiche hierzu den Algorithmus zur hierarchisch-agglomerativen Clusteranalyse.

Bemerkungen:

- Im Prinzip kann $d_{\mathcal{C}}$ wie bei den hierarchisch-agglomerativen Verfahren gewählt werden. Die Worst-Case-Komplexität ist exponentiell statt quadratisch.
- Hierarchisch-divisive Verfahren werden oft als *monothetische* Variante konstruiert: in jedem Entscheidungsschritt wird nur *eine* Variable betrachtet.
- Im Gegensatz zu hierarchisch-agglomerativen Verfahren darf ein hierarchisch-divisives Verfahren keinen „Fehler“ bei den ersten Schritten machen.
- Ein mächtiges hierarchisch-divisives Clusteranalyse-Verfahren entsteht mit

$$sim_{\mathcal{C}}(C, C') = \sum_{e \in cut(\{C, C'\})} w(e) \quad \text{bzw.} \quad d_{\mathcal{C}}(C, C') = \frac{1}{sim_{\mathcal{C}}(C, C')}$$

Hierarchische Verfahren

MinCut-Clusteranalyse

Definition 3 (Cut, minimaler Cut)
Sei $G = \langle V, E, w \rangle$ ein Graph mit nicht-negativer Gewichtsfunktion w; weiterhin sei $U \subset V$ eine nichtleere Teilmenge der Knotenmenge V und $\bar{U} = V \setminus U$. Der Cut zwischen U und $\bar{U}$ ist wie folgt definiert:

$$cut(\{U, \bar{U}\}) = \{\{u, v\} \mid \{u, v\} \in E, u \in U, v \in \bar{U}\}$$

Hierarchische Verfahren

MinCut-Clusteranalyse

Definition 3 (Cut, minimaler Cut)
Sei $G = \langle V, E, w \rangle$ ein Graph mit nicht-negativer Gewichtsfunktion w; weiterhin sei $U \subset V$ eine nichtleere Teilmenge der Knotenmenge V und $\bar{U} = V \setminus U$. Der Cut zwischen U und $\bar{U}$ ist wie folgt definiert:

$$cut(\{U, \bar{U}\}) = \{\{u, v\} \mid \{u, v\} \in E, u \in U, v \in \bar{U}\}$$

Weiterhin bezeichne $w(\{U, \bar{U}\})$ das Gewicht oder die Kapazität von $cut(\{U, \bar{U}\})$:

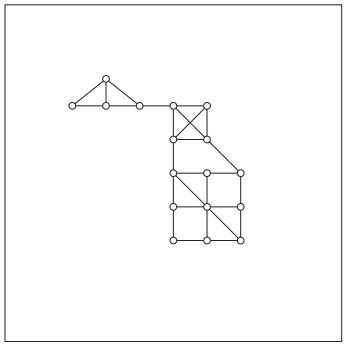
$$w(\{U, \bar{U}\}) = \sum_{e \in cut(\{U, \bar{U}\})} w(e)$$

$cut(\{U, \bar{U}\})$ heißt minimaler Cut von G *(Minium Capacity Cut)*, wenn für alle Zerlegungen $\{W, \bar{W}\}$, $W, \bar{W} \neq \emptyset$ gilt:

$$w(\{U, \bar{U}\}) \ \leq \ w(\{W, \bar{W}\})$$

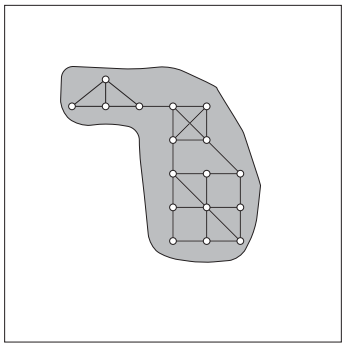
Hierarchische Verfahren

MinCut-Clusteranalyse



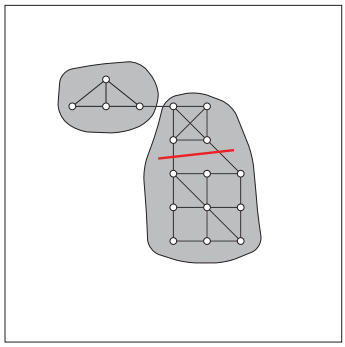
Hierarchische Verfahren

MinCut-Clusteranalyse



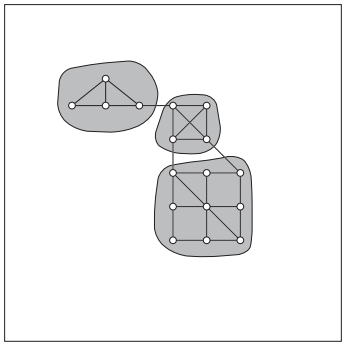
Hierarchische Verfahren

MinCut-Clusteranalyse



Hierarchische Verfahren

MinCut-Clusteranalyse

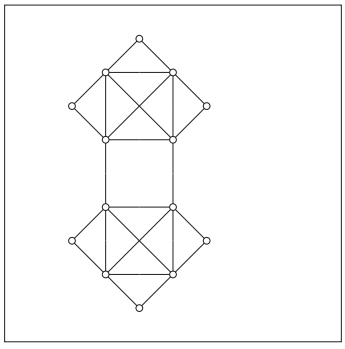


Bemerkungen:

- Jede Aufteilung erfordert die Berechnung eines Cuts minimaler Kapazität *(Minimum-Capacity-Cut)*. Beachte, dass kein Knoten als Quelle oder Senke vorgegeben ist.
- Die Laufzeitkomplexität des besten, bekannten Algorithmus zur Berechnung eines minimalen Cuts ist in $O(|V| \cdot |E| + |V|^2 \cdot \log |V|)$. [Nagamochi/Ono/Ibaraki 1994]
- Es sind $|V| - 1$ Berechnungen eines Minimum-Capacity-Cuts notwendig, um eine vollständige Zerlegung (= ein Knoten pro Cluster) herzustellen.
- Der Aufwand zur Berechnung eines Minimum-s-t-Cuts, also eines Cuts bei vorgegebener Quelle s und Senke t, ist in $O(|V|^2 \log(|E|))$.
- Der Aufwand zur Berechnung eines Balanced Min-Cut (k-way, $k \geq 2$) ist NP-vollständig.

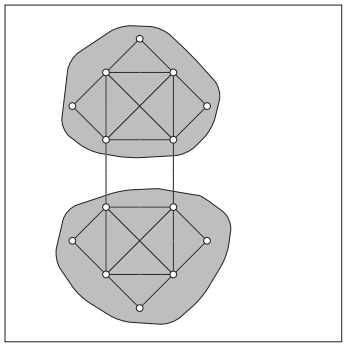
Hierarchische Verfahren

Splitting-Problematik bei MinCut-Clusteranalyse



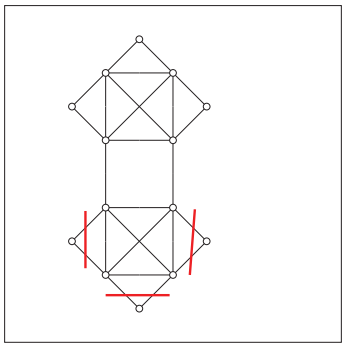
Hierarchische Verfahren

Splitting-Problematik bei MinCut-Clusteranalyse



Hierarchische Verfahren

Splitting-Problematik bei MinCut-Clusteranalyse



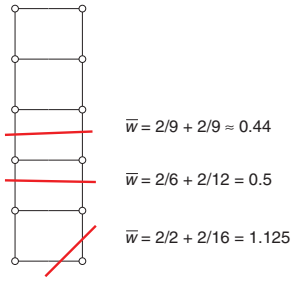
Ausweg: Normalisierung der Cut-Kapazität bzgl. der Größe der Knotenmengen.

Hierarchische Verfahren

Splitting-Problematik bei MinCut-Clusteranalyse

Normalisierte Cut-Kapazität: $\overline{w}(\{U, \bar{U}\}) = \frac{w(\{U, \bar{U}\})}{w(\{U, V\})} + \frac{w(\{U, \bar{U}\})}{w(\{\bar{U}, V\})}$

Illustration von $\overline{w}$:



Bemerkungen:

- Die Bestimmung eines Cuts minimaler, normalisierter Kapazität ist NP-vollständig.
- Es gibt effiziente Näherungslösungen zur Berechnung von $\overline{w}(\{U, \bar{U}\})$. Das Verfahren wird angewandt im Bereich der Bildsegmentierung und der Gene-Expression-Cluster-Analyse. [Shi/Malik 2000]

Hierarchische Verfahren

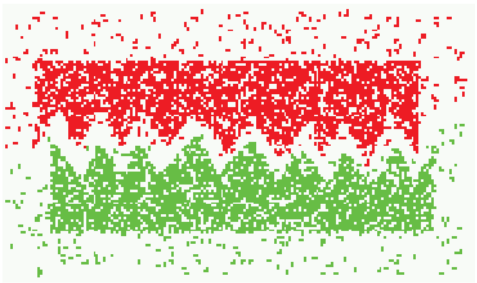
Kombination hierarchischer Verfahren

Das System Chameleon kombiniert die Schritte Graphausdünnung, Graphpartionierung und hierarchische Cluster-Analyse [Karypis/Han/Kumar 2000] :

Hierarchische Verfahren

Kombination hierarchischer Verfahren

Das System Chameleon kombiniert die Schritte Graphausdünnung, Graphpartionierung und hierarchische Cluster-Analyse [Karypis/Han/Kumar 2000] :

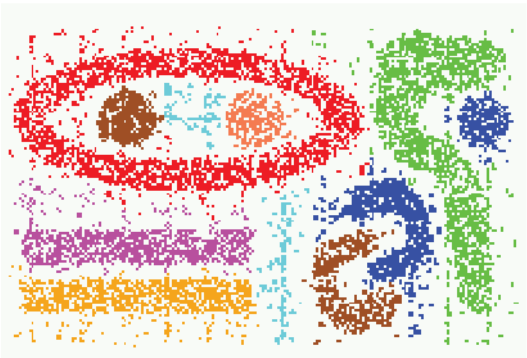


Die Clusterdistanz $d_{\mathcal{C}}(C, C')$ ist definiert als $d_{\mathcal{C}} = \frac{1}{R_I(C, C') \cdot (R_C(C, C'))^{\alpha}}$

Hierarchische Verfahren

Kombination hierarchischer Verfahren

Chameleon [Karypis/Han/Kumar 2000] :

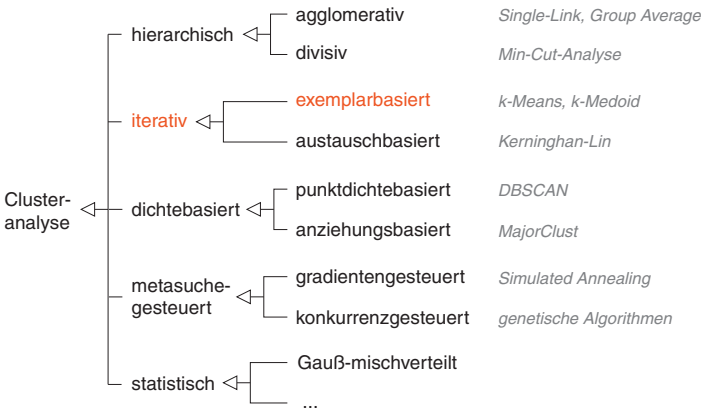


Der Parameter α in $d_{\mathcal{C}}$ ist vom Anwender problemabhängig zu bestimmen.

Kapitel ML: IX (Fortsetzung)

Iterative Verfahren

Prinzipien der Fusionierung



Iterative Verfahren

Algorithmus zur exemplarbasierten Clusteranalyse

Input: $G = \langle V, E, w \rangle$. Weighted graph.
d. Distance function for nodes in V.
e. Minimization criterion for cluster representatives, based on d.
k. Number of desired clusters.

Output: $r_1, \ldots, r_k$. Cluster representatives.

```
1.
2. FOR i = 1 to k DO r_i(t) = choose(V)   // init representatives
3.
4.
5.

6.   FOREACH v ∈ V DO   // find nearest representative (cluster)
7.     i = argmin_{j: j∈{1,...,k}} d(r_j(t), v), C_i = C_i ∪ {v}
8.   ENDDO
9.   FOR i = 1 to k DO r_i(t) = minimize(e(C_i))   // update
10.
11.
```


Iterative Verfahren

Algorithmus zur exemplarbasierten Clusteranalyse

Input: $G = \langle V, E, w \rangle$. Weighted graph.
d. Distance function for nodes in V.
e. Minimization criterion for cluster representatives, based on d.
k. Number of desired clusters.

Output: $r_1, \ldots, r_k$. Cluster representatives.

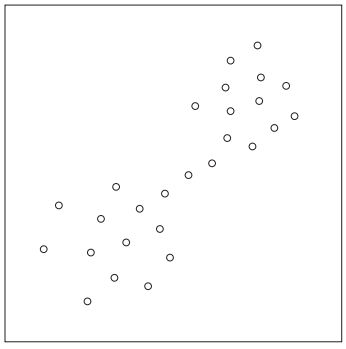
```
1. t = 0
2. FOR i = 1 to k DO r_i(t) = choose(V)   // init representatives
3. REPEAT
4.   t = t + 1
5.   FOR i = 1 to k DO C_i = ∅

6.   FOREACH v ∈ V DO   // find nearest representative (cluster)
7.     i = argmin_{j: j∈{1,...,k}} d(r_j(t), v), C_i = C_i ∪ {v}
8.   ENDDO
9.   FOR i = 1 to k DO r_i(t) = minimize(e(C_i))   // update
10. UNTIL (convergence(r_1(t), ..., r_k(t)) OR t > t_max)
11. RETURN({r_1(t), ..., r_k(t)})
```


Bemerkungen:

- Die Cluster-Repräsentanten werden Centroide bzw. allgemein Medoide genannt.
- Die Funktion *choose*(V) realisiert eine zufällige Auswahl ohne Zurücklegen.
- Als Distanzfunktion d wird bei metrischen Daten meistens der euklidische Abstand zwischen zwei Punkten gewählt. Ein alternativer und allgemeiner Ansatz ist die Verwendung des kürzesten Weges in G.
- Als Minimierungskriterium e wird bei metrischen Daten meistens die Summe der quadrierten Abweichungen zum Cluster-Repräsentanten gewählt (Varianzkriterium). Sind die Punkte $v \in V$ Vektoren aus dem $\mathbf{R}^p$, lässt sich der optimale Cluster-Repräsentant durch komponentenweise Berechnung des arithmetischen Mittels ermitteln.

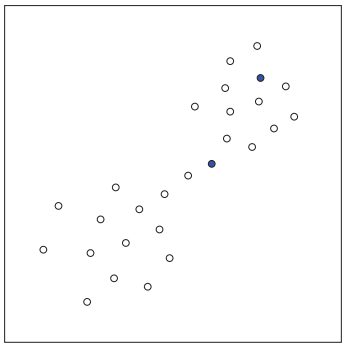
Iterative Verfahren

k-Means mit Minimierungskriterium e = Varianz



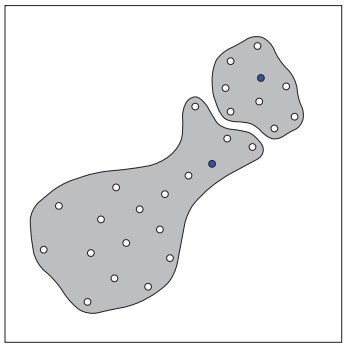
Iterative Verfahren

k-Means mit Minimierungskriterium e = Varianz



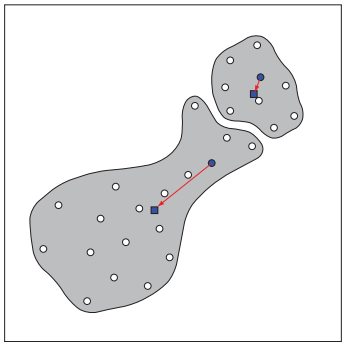
Iterative Verfahren

k-Means mit Minimierungskriterium e = Varianz



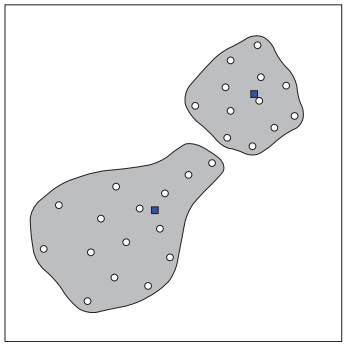
Iterative Verfahren

k-Means mit Minimierungskriterium e = Varianz



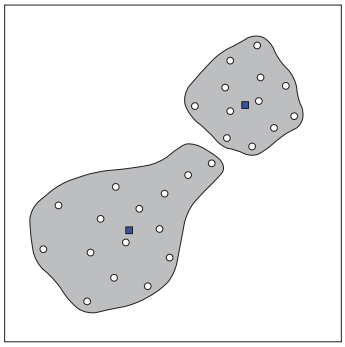
Iterative Verfahren

k-Means mit Minimierungskriterium e = Varianz



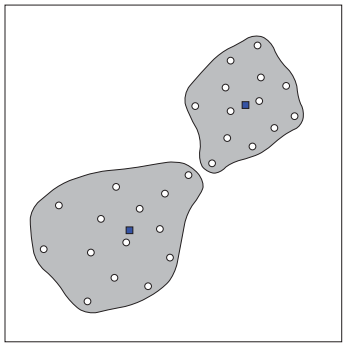
Iterative Verfahren

k-Means mit Minimierungskriterium e = Varianz



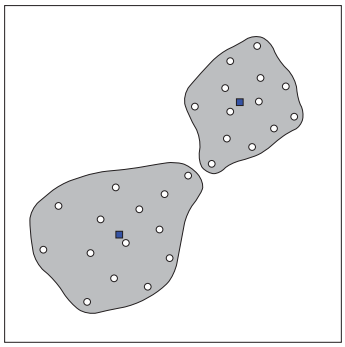
Iterative Verfahren

k-Means mit Minimierungskriterium e = Varianz



Iterative Verfahren

k-Means mit Minimierungskriterium e = Varianz



Iterative Verfahren

Minimierungskriterien exemplarbasierter Verfahren

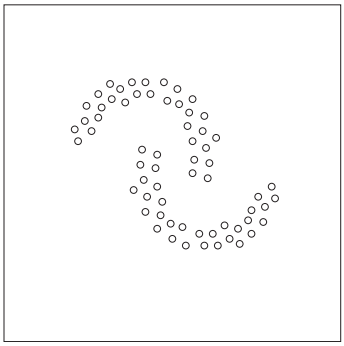
$e(C_i) = \sum_{v \in C_i} (v - r_i)^2$	$r_i = \bar{v}_{C_i}$	Centroid-Berechnung mittels Varianzminimierung (k-Means)
$e(C_i) = \sum_{v \in C_i} \lvert v - r_i \rvert$	$r_i \in C_i$	Medoid-Berechnung (k-Medoid)
$e(C_i) = \max_{v \in C_i} \lvert v - r_i \rvert$	$r_i \in C_i$	k-Center
$e(C_i) = \sum_{v \in V} (\mu_i(v))^2 \cdot (v - r_i)^2$	$r_i = \frac{\sum_{v \in V} (\mu_i(v))^2 \cdot v}{\sum_{v \in V} (\mu_i(v))^2}$	Fuzzy-k-Means

Bemerkungen:

- $\bar{v}_{C_i}$ bezeichne den Mittelwert der Punkte $v \in C_i$.
- Der Cluster-Repräsentant ist abkürzend mit r_i anstatt mit $r_i(t)$ bezeichnet.
- Die Centroid-Berechnung bei k-Means als Mittelwert der Cluster-Elemente entspricht einer lokalen, also Cluster-spezifischen Varianzminimierung.
- Der Medoid (Zentralelement) eines Cluster ist dasjenige Element, für das die Summe aller Distanzen zu diesem Element minimal ist. Ein Vorteil bei Verwendung von Medoiden ist das robustere Verhalten gegenüber Ausreißern und damit eventuell eine in der Anzahl der Iterationen schnellere Konvergenz.
- Bei Fuzzy-k-Means bezeichnet $\mu_i(v)$ den Zugehörigkeitswert von $v \in V$ zu Cluster C_i.
- k-Medoid und k-Center arbeiten sowohl mit beliebigen Distanz- als auch mit Ähnlichkeitsmaßen.
- k-Means und Fuzzy-k-Means setzen intervallskalierte Merkmale voraus.
- k-Means kann unmittelbar als Kohonen Self-Organizing-Map, SOM, einem speziellen neuronalen Netz, umgesetzt werden:
  - Die Netztopologie besteht aus einer Eingangsschicht mit Knoten in der Anzahl der Merkmale und einer Verarbeitungsschicht *(Competitive Layer)* mit k Knoten.
  - Der Lernalgorithmus bestimmt für einen Merkmalsvektor auf Basis der Kantengewichte das *Winning Neuron*, dessen Gewichte gemäß des Lernparameters η erhöht werden.

Iterative Verfahren

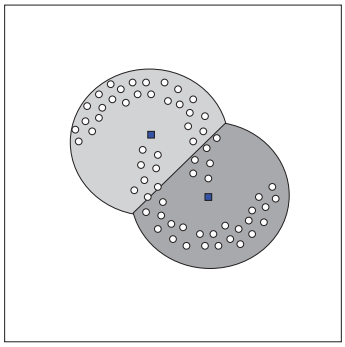
k-Means versus Single-Link



Exemplarbasierte Verfahren versagen bei verschränkt liegenden Clustern.

Iterative Verfahren

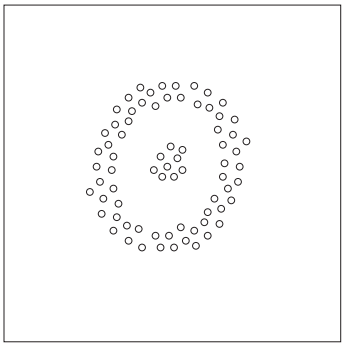
k-Means versus Single-Link



Exemplarbasierte Verfahren versagen bei verschränkt liegenden Clustern.

Iterative Verfahren

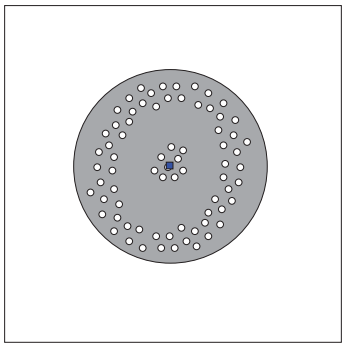
k-Means versus Single-Link



Exemplarbasierte Verfahren versagen bei verschränkt liegenden Clustern.

Iterative Verfahren

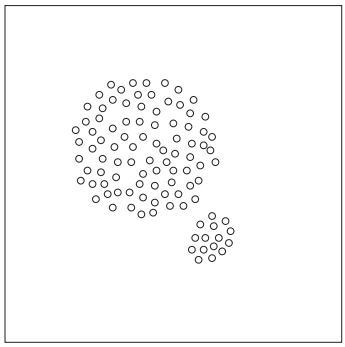
k-Means versus Single-Link



Exemplarbasierte Verfahren versagen bei verschränkt liegenden Clustern.

Iterative Verfahren

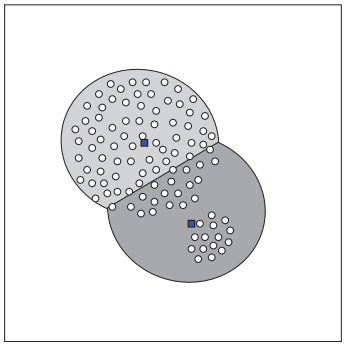
k-Means versus Single-Link



Exemplarbasierte Verfahren versagen bei extremen Größendifferenzen.

Iterative Verfahren

k-Means versus Single-Link



Exemplarbasierte Verfahren versagen bei extremen Größendifferenzen.

Iterative Verfahren

Exklusive versus nicht-exklusive Clusteranalyse

Sei $\mathcal{C} = \{C_1, \ldots, C_k\}$ eine Partionierung einer Menge V mit $\bigcup_{i=1\ldots k} C_i = V$.

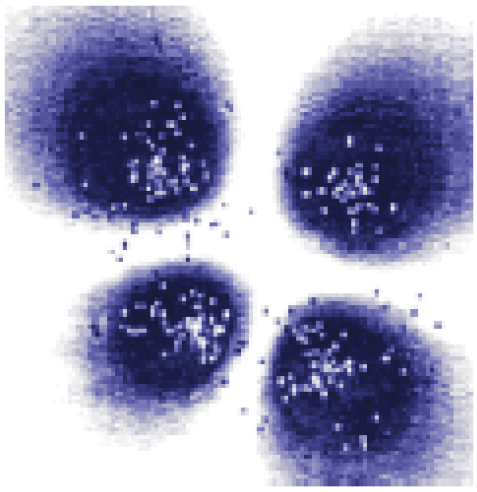
- exklusive Clusteranalyse: $\forall i, j \in \{1, \ldots, k\} : i \neq j$ impliziert $C_i \cap C_j = \emptyset$
- nicht-exklusive Clusteranalyse erlaubt mehrfache Cluster-Zugehörigkeit

Iterative Verfahren

Exklusive versus nicht-exklusive Clusteranalyse

Sei $\mathcal{C} = \{C_1, \ldots, C_k\}$ eine Partionierung einer Menge V mit $\bigcup_{i=1\ldots k} C_i = V$.

- exklusive Clusteranalyse: $\forall i, j \in \{1, \ldots, k\} : i \neq j$ impliziert $C_i \cap C_j = \emptyset$
- nicht-exklusive Clusteranalyse erlaubt mehrfache Cluster-Zugehörigkeit
- Fuzzy-Clusteranalyse quantifiziert die Cluster-Zugehörigkeit der $v \in V$ mit Zugehörigkeitsfunktionen $\mu_i(v),\ i \in \{1, \ldots, k\}$.



[Höppner/Klawonn/Kruse 1997]

Iterative Verfahren

Exklusive versus nicht-exklusive Clusteranalyse

Anwendung der Fuzzy-Clusteranalyse zur Darstellung von Gehirngewebe:

[Pham/Prince/Dagher/Xn 1996]

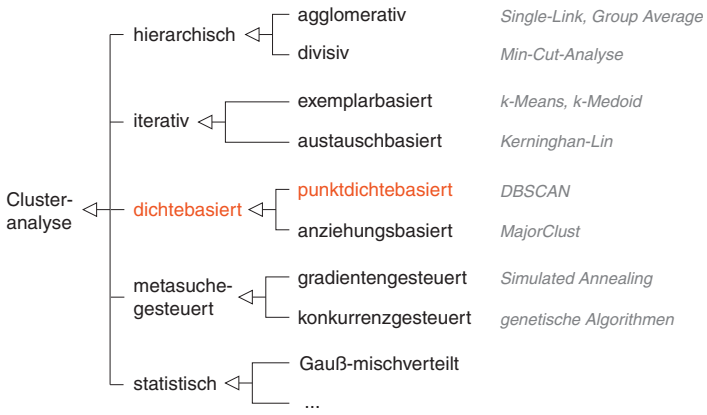
Bemerkungen:

- Die linguistische Variable bei der Fuzzy-Modellierung hat k Ausprägungen, entsprechend den Clustern $C_1, \ldots, C_k$.
- Üblich ist eine Normalisierungsrestriktion für Zugehörigkeitsfunktionen: $\sum_{i=1\ldots k} \mu_i(v) = 1$
- Varianten von Fuzzy-k-Means ohne die Normalisierungsrestriktion haben den Nachteil, dass Punkte mit kleinen Zugehörigkeitswerten zu einem Cluster wie Ausreißer behandelt werden, anstatt den Cluster in ihre Richtung zu bewegen. Deshalb ist es sinnvoll, das Iterationsverfahren in einer Initialisierungsphase zunächst mit der Restriktion anzuwenden.
- Eine Klassifikation durch eine unscharfe Cluster-Analyse ist vorteilhaft, wenn keine klar ausgebildete Klassenstruktur vorliegt bzw. wenn sich nicht alle Vektoren eindeutig einer Klasse zuordnen lassen.
- Ein Nachteil der Fuzzy-Clusteranalyse ist, dass keine Repräsentanten für Cluster ermittelt werden.

Kapitel ML:IX (Fortsetzung)

Dichtebasierte Verfahren

Prinzipien der Fusionierung



Dichtebasierte Verfahren

Dichtebasierte Verfahren versuchen, den Graphen $G = \langle V, E, w \rangle$ bzw. die Punktmenge V in Bereiche gleicher Dichte aufzuteilen.

Ansätze zur Dichteschätzung:

- parameterbasiert: die unterliegende Verteilung ist bekannt
- parameterlos: Histogramme (Konstruktion von Bar-Charts), Kerndichteschätzer (Überlagerung kontinuierlicher Funktionen)

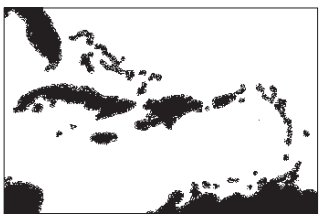
Dichtebasierte Verfahren

Dichtebasierte Verfahren versuchen, den Graphen $G = \langle V, E, w \rangle$ bzw. die Punktmenge V in Bereiche gleicher Dichte aufzuteilen.

Ansätze zur Dichteschätzung:

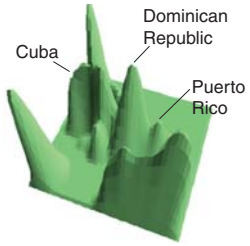
- parameterbasiert: die unterliegende Verteilung ist bekannt
- parameterlos: Histogramme (Konstruktion von Bar-Charts), Kerndichteschätzer (Überlagerung kontinuierlicher Funktionen)

Beispiel (karibische Inseln):



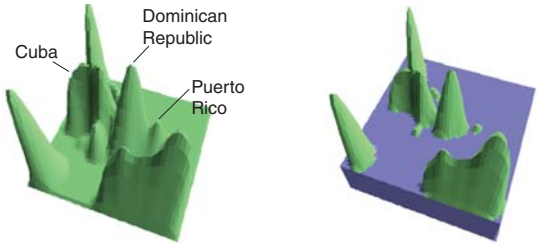
Dichtebasierte Verfahren

Dichteschätzung mit Gauß'schem Kern für das Beispiel



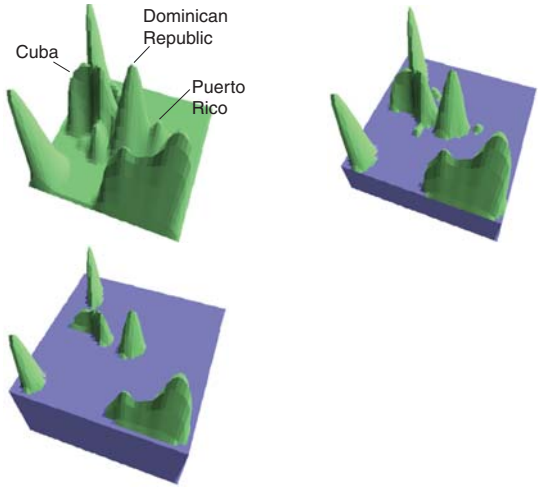
Dichtebasierte Verfahren

Dichteschätzung mit Gauß'schem Kern für das Beispiel



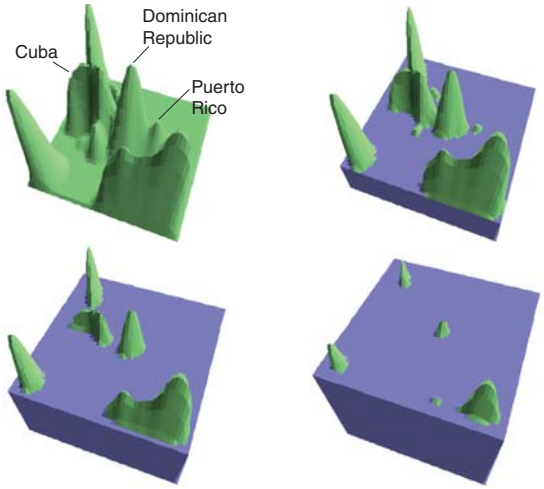
Dichtebasierte Verfahren

Dichteschätzung mit Gauß'schem Kern für das Beispiel



Dichtebasierte Verfahren

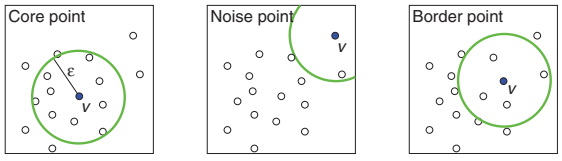
Dichteschätzung mit Gauß'schem Kern für das Beispiel



Dichtebasierte Verfahren

DBSCAN: Prinzip der Dichteschätzung [Ester et al. 1996]

Sei $N_\varepsilon(v)$ die ε-Nachbarschaft eine Punktes v. Unterscheidung von drei Punkttypen:



1. v ist Kernpunkt *(core point)* $\Leftrightarrow$ $|N_\varepsilon(v)| \geq$ *MinPts*
2. v ist Rauschen *(noise point)* $\Leftrightarrow$
   v ist von keinem Kernpunkt aus dichteerreichbar *(density-reachable)*
3. v ist Randpunkt *(border point)* in allen anderen Fällen

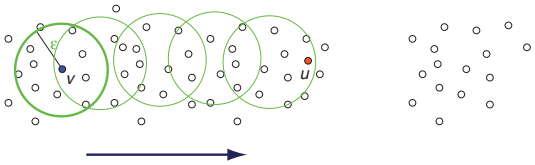
Dichtebasierte Verfahren

DBSCAN: Prinzip der Dichteschätzung

Ein Punkt u ist dichteerreichbar von einem Punkt v, falls gilt:

(a) $u \in N_\varepsilon(v)$, wobei v ein Kernpunkt ist, oder

(b) es gibt eine Menge von Punkten $\{v_1, \ldots, v_l\}$:
$v_{i+1} \in N_\varepsilon(v_i)$ und v_i ist Kernpunkt, $i = 1, \ldots, l\text{-}1$, mit $v_1 = v$, $v_l = u$.



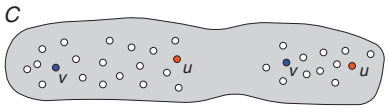
Bedingung (b) lässt sich als transitive Anwendung von Bedingung (a) auffassen.

Dichtebasierte Verfahren

DBSCAN: Cluster-Interpretation

Ein Cluster $C \subseteq V$ erfüllt folgende Bedingungen:

1. $\forall u, v$: Falls $v \in C$ und u dichteerreichbar von v, dann ist $u \in C$.



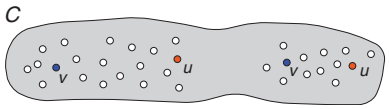
Maximalitäts-bedingung

Dichtebasierte Verfahren

DBSCAN: Cluster-Interpretation

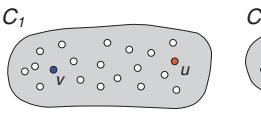
Ein Cluster $C \subseteq V$ erfüllt folgende Bedingungen:

1. $\forall u, v$: Falls $v \in C$ und u dichteerreichbar von v, dann ist $u \in C$.



Maximalitäts-bedingung

2. $\forall u, v$: u ist dichteverbunden mit v, d. h.,
   es existiert ein Punkt t von dem u und v dichteerreichbar sind.



Zusammenhangs-bedingung

Dichtebasierte Verfahren

DBSCAN: Algorithmus

Input: $G = \langle V, E, w \rangle$. Weighted graph.
d. Distance function for nodes in V.
ε. Neighborhood radius.
MinPts. Lower bound for point number in ε-neighborhood.

Output: $\gamma : V \to \mathbf{Z}$. Cluster assignment function.

```
1.
2.
3.

4.   N_ε(v) = neighborhood(G, d, v, ε)
5.   IF |N_ε(v)| ≥ MinPts THEN   // v is core point
6.     i = i + 1
7.     C_i = density_reachable_hull(G, d, N_ε(v))   // form a new cluster
8.     FOREACH v ∈ C_i DO γ(v) = i
9.   ELSE γ(v) = −1   // v is _preliminarily_ classified as noise

10.
11.
```


Dichtebasierte Verfahren

DBSCAN: Algorithmus

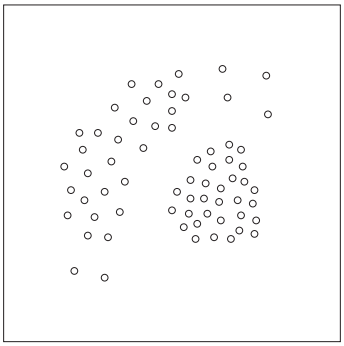
Input: $G = \langle V, E, w \rangle$. Weighted graph.
d. Distance function for nodes in V.
ε. Neighborhood radius.
MinPts. Lower bound for point number in ε-neighborhood.

Output: $\gamma : V \to \mathbf{Z}$. Cluster assignment function.

```
1.  i = 0
2.  WHILE ∃v : (v ∈ V AND γ(v) =⊥) DO   // ⊥ = unclassified
3.    v = choose_unclassified_point(V)

4.    N_ε(v) = neighborhood(G, d, v, ε)
5.    IF |N_ε(v)| ≥ MinPts THEN   // v is core point
6.      i = i + 1
7.      C_i = density_reachable_hull(G, d, N_ε(v))   // form a new cluster
8.      FOREACH v ∈ C_i DO γ(v) = i
9.    ELSE γ(v) = −1   // v is _preliminarily_ classified as noise

10. ENDDO
11. RETURN(γ)
```

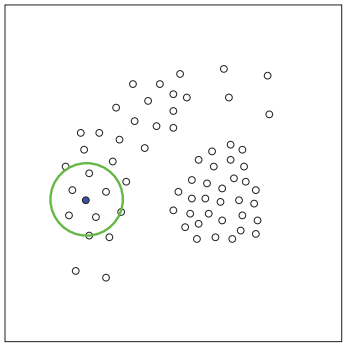

Dichtebasierte Verfahren

DBSCAN



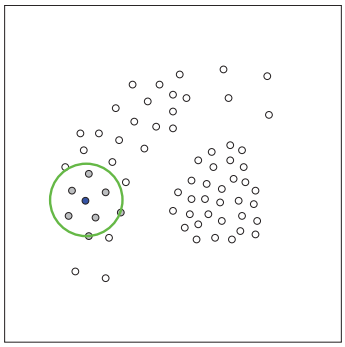
Dichtebasierte Verfahren

DBSCAN



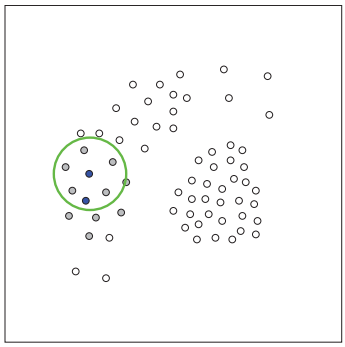
Dichtebasierte Verfahren

DBSCAN



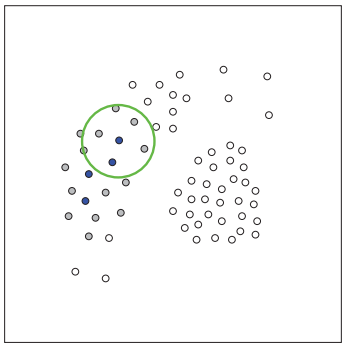
Dichtebasierte Verfahren

DBSCAN



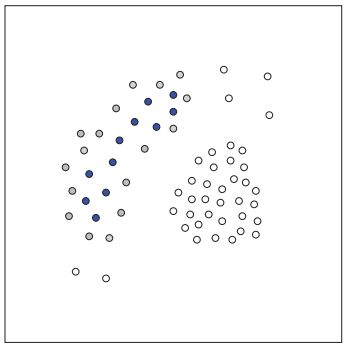
Dichtebasierte Verfahren

DBSCAN



Dichtebasierte Verfahren

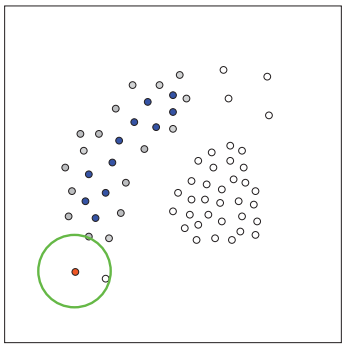
DBSCAN



- Core point
- Border point

Dichtebasierte Verfahren

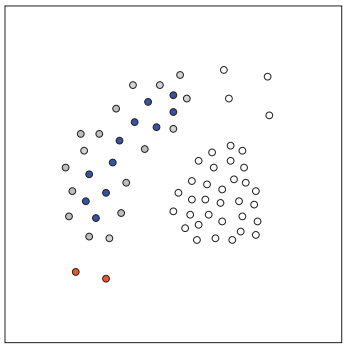
DBSCAN



- Core point
- Border point

Dichtebasierte Verfahren

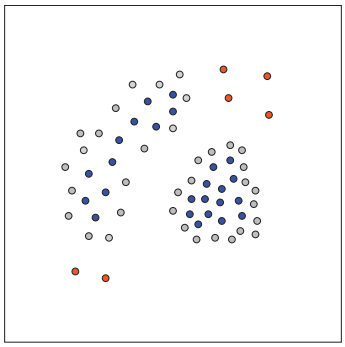
DBSCAN



- Core point
- Border point
- Noise point

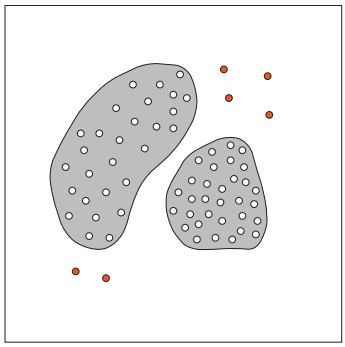
Dichtebasierte Verfahren

DBSCAN



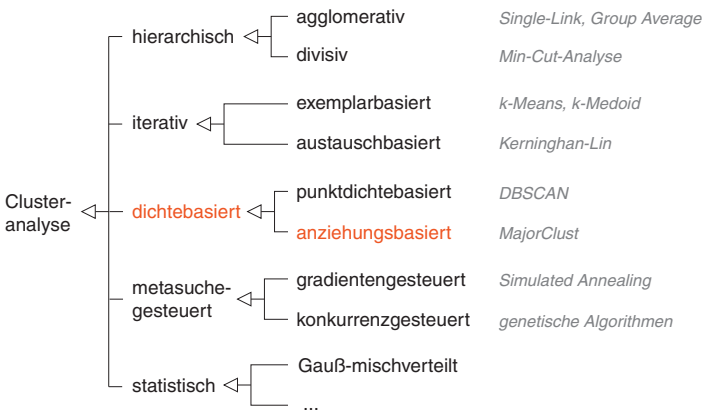
Dichtebasierte Verfahren

DBSCAN



Dichtebasierte Verfahren

Prinzipien der Fusionierung



Dichtebasierte Verfahren

MajorClust: Prinzip der Dichteschätzung [Stein/Niggemann 1999]

Die gewichteten Kanten im Graph $G = \langle V, E, w \rangle$ werden als Kräfte interpretiert; Knoten desselben Clusters bündeln ihrer Kräfte. Illustration:

eindeutige Zugehörigkeitsentscheidung (mit Agglomeration):

Dichtebasierte Verfahren

MajorClust: Prinzip der Dichteschätzung [Stein/Niggemann 1999]

Die gewichteten Kanten im Graph $G = \langle V, E, w \rangle$ werden als Kräfte interpretiert; Knoten desselben Clusters bündeln ihrer Kräfte. Illustration:

eindeutige Zugehörigkeitsentscheidung (mit Agglomeration):

eindeutige Zugehörigkeitsentscheidung (mit Cluster-Wechsel):

Dichtebasierte Verfahren

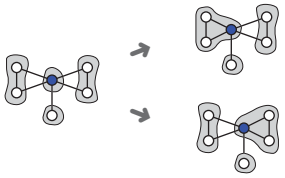
MajorClust: Prinzip der Dichteschätzung [Stein/Niggemann 1999]

Die gewichteten Kanten im Graph $G = \langle V, E, w \rangle$ werden als Kräfte interpretiert; Knoten desselben Clusters bündeln ihrer Kräfte. Illustration:

eindeutige Zugehörigkeitsentscheidung (mit Agglomeration):

eindeutige Zugehörigkeitsentscheidung (mit Cluster-Wechsel):

mehrdeutige Zugehörigkeit:



Dichtebasierte Verfahren

MajorClust: Algorithmus

Input: $G = \langle V, E, w \rangle$. Weighted graph.
d. Distance function for nodes in V.
Output: $\gamma : V \to \mathbf{N}$. Cluster assignment function.

1.
2.
3.
4.
5. **FOREACH** $v \in V$ **DO**
6. $\gamma^* = \underset{i:\, i \in \{1,\ldots,|V|\}}{\text{argmax}} \sum_{\{u,v\}:\, \{u,v\} \in E \wedge \gamma(u) = i} w(u,v)$
7. **IF** $\gamma(v) \neq \gamma^*$ **THEN** $\gamma(v) = \gamma^*$, $t =$ *False* **ENDIF** // relabeling
8. **ENDDO**
9.
10.

Dichtebasierte Verfahren

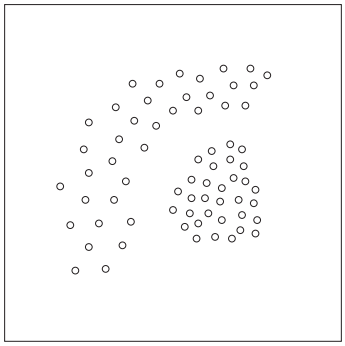
MajorClust: Algorithmus

Input: $G = \langle V, E, w \rangle$. Weighted graph.
d. Distance function for nodes in V.
Output: $\gamma : V \to \mathbf{N}$. Cluster assignment function.

1. $i = 0$, $t =$ *False*
2. **FOREACH** $v \in V$ **DO** $i = i + 1$, $\gamma(v) = i$ **ENDDO**
3. **UNLESS** t **DO**
4. $t =$ *True*
5. **FOREACH** $v \in V$ **DO**
6. $\gamma^* = \underset{i:\, i \in \{1,\ldots,|V|\}}{\text{argmax}} \sum_{\{u,v\}:\, \{u,v\} \in E \wedge \gamma(u) = i} w(u,v)$
7. **IF** $\gamma(v) \neq \gamma^*$ **THEN** $\gamma(v) = \gamma^*$, $t =$ *False* **ENDIF** // relabeling
8. **ENDDO**
9. **ENDDO**
10. **RETURN**(γ)

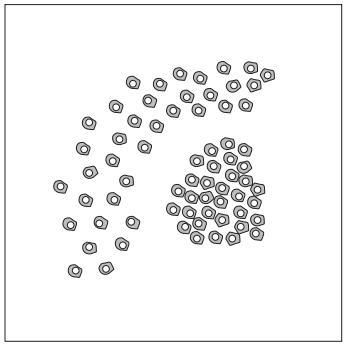
Dichtebasierte Verfahren

MajorClust



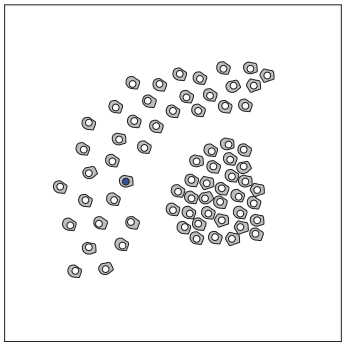
Dichtebasierte Verfahren

MajorClust



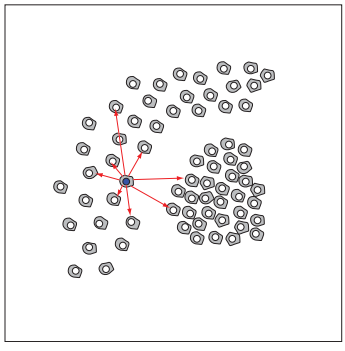
Dichtebasierte Verfahren

MajorClust



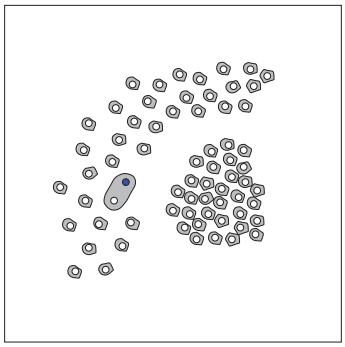
Dichtebasierte Verfahren

MajorClust



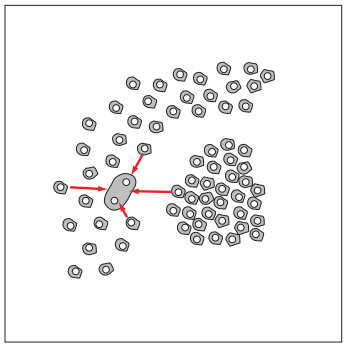
Dichtebasierte Verfahren

MajorClust



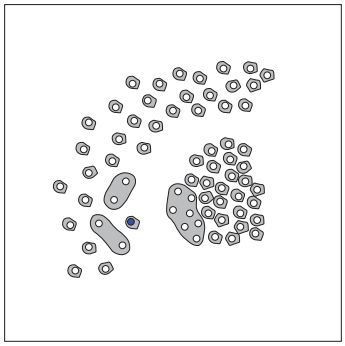
Dichtebasierte Verfahren

MajorClust



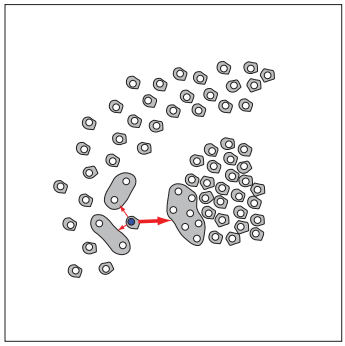
Dichtebasierte Verfahren

MajorClust



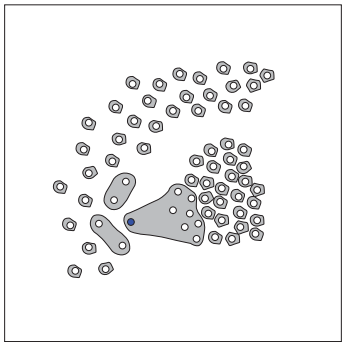
Dichtebasierte Verfahren

MajorClust



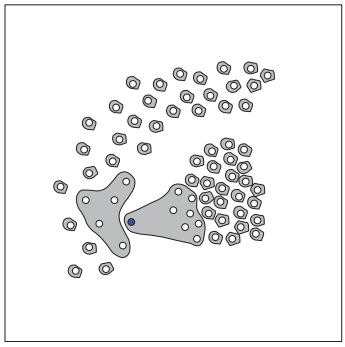
Dichtebasierte Verfahren

MajorClust



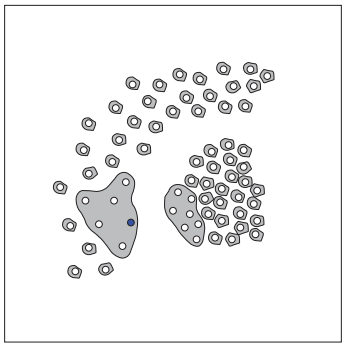
Dichtebasierte Verfahren

MajorClust



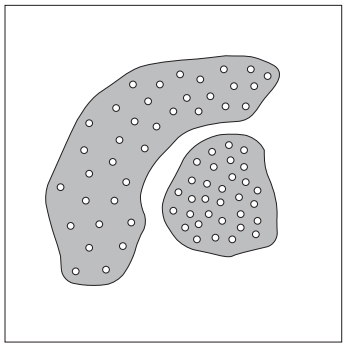
Dichtebasierte Verfahren

MajorClust



Dichtebasierte Verfahren

MajorClust



Dichtebasierte Verfahren

MajorClust: Prinzip der Dichteschätzung (Fortsetzung)

Jedes $\mathcal{C} = \{C_1, \ldots, C_k\}$ induziert k Teilgraphen. MajorClust ist eine Heuristik zur Maximierung des *gewichteten partiellen Kantenzusammenhangs*, $\Lambda(\mathcal{C})$.

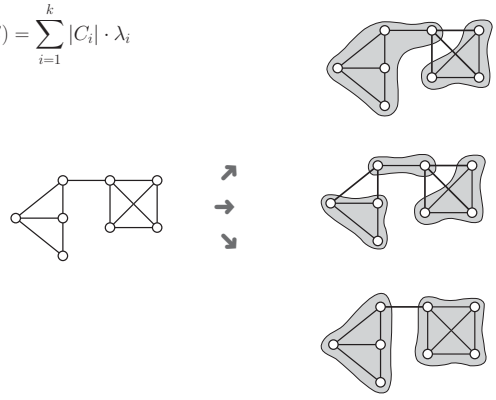
$$\Lambda(\mathcal{C}) = \sum_{i=1}^{k} |C_i| \cdot \lambda_i$$

Dichtebasierte Verfahren

MajorClust: Prinzip der Dichteschätzung (Fortsetzung)

Jedes $\mathcal{C} = \{C_1, \ldots, C_k\}$ induziert k Teilgraphen. MajorClust ist eine Heuristik zur Maximierung des *gewichteten partiellen Kantenzusammenhangs*, $\Lambda(\mathcal{C})$.

$$\Lambda(\mathcal{C}) = \sum_{i=1}^{k} |C_i| \cdot \lambda_i$$

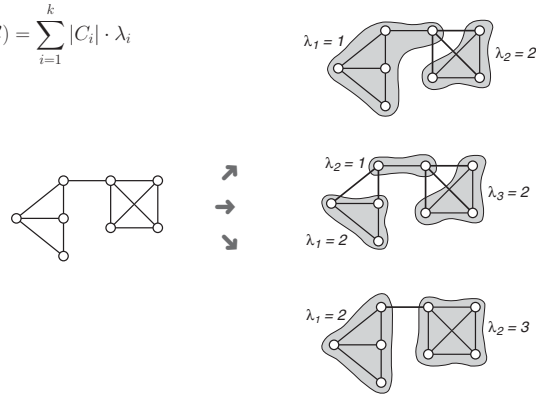


Dichtebasierte Verfahren

MajorClust: Prinzip der Dichteschätzung (Fortsetzung)

Jedes $\mathcal{C} = \{C_1, \ldots, C_k\}$ induziert k Teilgraphen. MajorClust ist eine Heuristik zur Maximierung des *gewichteten partiellen Kantenzusammenhangs*, $\Lambda(\mathcal{C})$.

$$\Lambda(\mathcal{C}) = \sum_{i=1}^{k} |C_i| \cdot \lambda_i$$

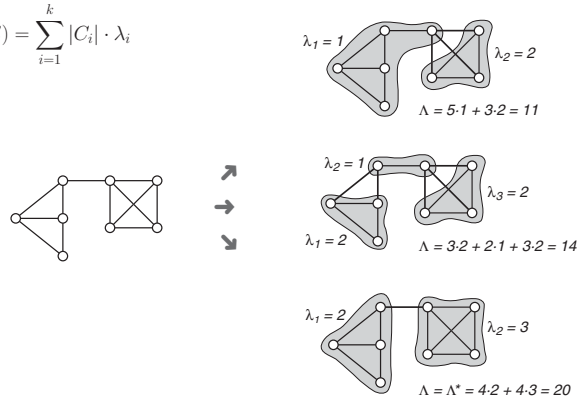


Dichtebasierte Verfahren

MajorClust: Prinzip der Dichteschätzung (Fortsetzung)

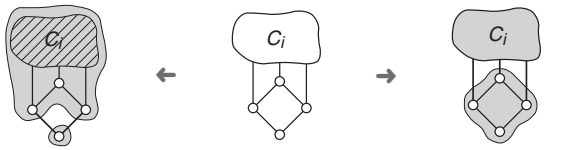
Jedes $\mathcal{C} = \{C_1, \ldots, C_k\}$ induziert k Teilgraphen. MajorClust ist eine Heuristik zur Maximierung des *gewichteten partiellen Kantenzusammenhangs*, $\Lambda(\mathcal{C})$.

$$\Lambda(\mathcal{C}) = \sum_{i=1}^{k} |C_i| \cdot \lambda_i$$



Dichtebasierte Verfahren

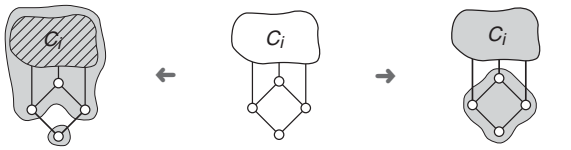
MajorClust: Prinzip der Dichteschätzung (Fortsetzung)



Min-Cut-Clustering Λ-Maximierung

Dichtebasierte Verfahren

MajorClust: Prinzip der Dichteschätzung (Fortsetzung)



Min-Cut-Clustering Λ-Maximierung

Satz 1 (Strong Splitting Condition [Stein/Niggemann 1999]**)**
Sei $\mathcal{C} = \{C_1, \ldots, C_k\}$ eine Partitionierung eines Graphen $G = \langle V, E, w \rangle$; weiterhin bezeichne $\lambda(G)$ den Kantenzusammenhang von G und $\lambda_1, \ldots, \lambda_k$ die Kantenzusammenhänge der von den $C_1, \ldots, C_k$ induzierten Subgraphen.

Gilt $\lambda(G) < \min\{\lambda_1, \ldots, \lambda_k\}$ (Strong Splitting Condition), so liefert Λ-Maximierung eine Aufteilung am minimalen Cut.

Dichtebasierte Verfahren

DBSCAN versus MajorClust: niedrigdimensionale Daten

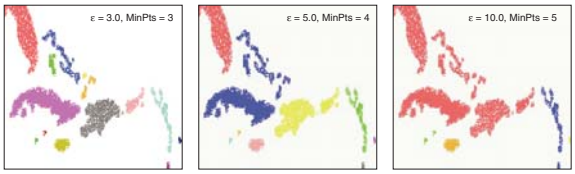
Karte der karibischen Inseln, etwa 20.000 Punkte:

Dichtebasierte Verfahren

DBSCAN versus MajorClust: niedrigdimensionale Daten

Karte der karibischen Inseln, etwa 20.000 Punkte:

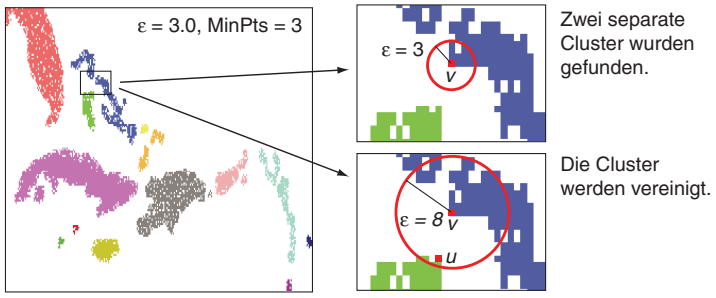
DBSCAN:



Dichtebasierte Verfahren

DBSCAN versus MajorClust: niedrigdimensionale Daten

Problematik geeigneter ε-Werte bei DBSCAN:



Zwei separate Cluster wurden gefunden.

Die Cluster werden vereinigt.

Dichtebasierte Verfahren

DBSCAN versus MajorClust: niedrigdimensionale Daten

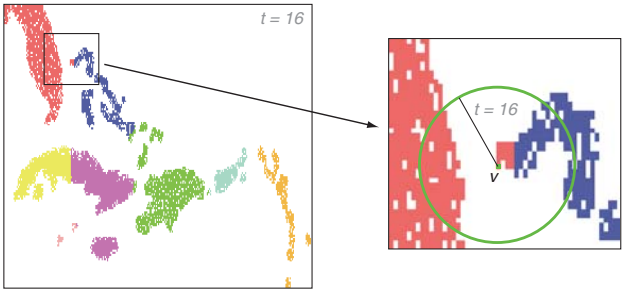
Karte der karibischen Inseln, etwa 20.000 Punkte:

MajorClust:

Dichtebasierte Verfahren

DBSCAN versus MajorClust: niedrigdimensionale Daten

Problematik der globalen Analyse (keine Beschränkung auf eine ε-Nachbarschaft) bei MajorClust:



Dichtebasierte Verfahren

DBSCAN versus MajorClust: hochdimensionale Daten

Dokumentenkategorisierung mit dem Reuters-Korpus:

- 1000 Dokumente
- 10 Kategorien: Politik, Kultur, Wirtschaft, etc.
- die Dokumente sind gleichverteilt, gehören genau zu einer Kategorie
- Dimension des Merkmalsraums: > 10.000

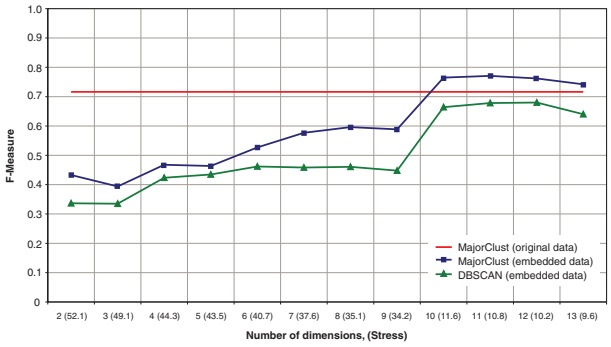
DBSCAN:

- degeneriert mit steigender Zahl der Dimensionen
- Ursache ist die Bestimmung der ε-Nachbarschaft
- Ausweg ist eine Dimensionreduktion, z. B. eine Einbettung der Daten mittels multidimensionaler Skalierung (MDS)

Dichtebasierte Verfahren

DBSCAN versus MajorClust: hochdimensionale Daten

Klassifikationsergebnisse (F-Measure), aufgetragen über Dimensionalität:



[Stein/Busch 2005]

Bemerkungen:

- Das Problem der Nachbarschaftssuche in hochdimensionalen Räumen ist meistens nicht effizient lösbar: Ab Dimensionen größer als 10-20 ist das lineare Durchsuchen aller Merkmalsvektoren effizienter als die Verwendung von hochentwickelten, raumpartitionierenden Datenstrukturen wie R-Tree, X-Tree, Quadtree, KD-Tree, etc.
  Einen Ausweg bieten die Ansätze wie Locality-Sentitve-Hashing oder Fuzzy-Fingerprinting.
  Siehe auch: [Weber 99] [Gionis/Indyk/Motwani 99-04] [Stein 05] [Stein/SMZE 05]
- DBSCAN verwendet zur Bestimmung der ε-Nachbarschaft die R-Tree-Datenstruktur. Diese Datenstruktur leistet einen wesentlichen (wenn nicht den größten) Teil der Cluster-Analyse innerhalb von DBSCAN.
- Möchte man DBSCAN für hochdimensionale Daten verwenden, ist eine Einbettung der Daten in einen niedrigdimensionalen Raum unvermeidbar. Dabei ist zu bedenken, dass eine Dimensionsreduktion durch Einbettung rechenintensiv ist und das gute Laufzeitverhalten von DBSCAN zunichte macht.

Chapter ML: IX (continued)

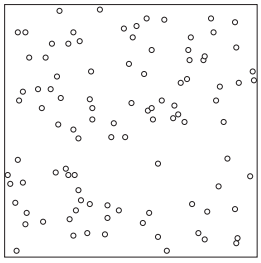
Cluster Evaluation

Overview

"The validation of clustering structures is the most difficult and frustrating part of cluster analysis. Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

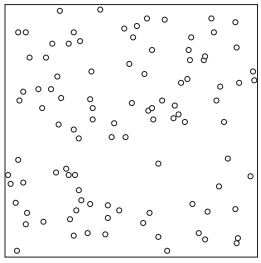
[Jain/Dubes 1990]

Cluster Evaluation [from Tan/Steinbach/Kumar 05]

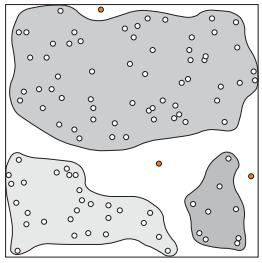


Random points

Cluster Evaluation [from Tan/Steinbach/Kumar 05]

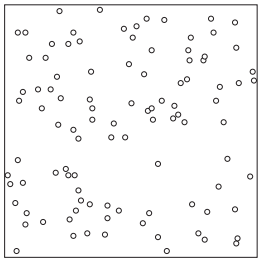


Random points

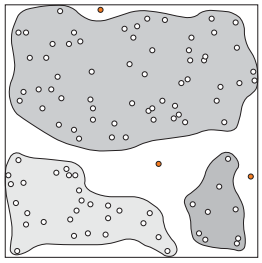


DBSCAN

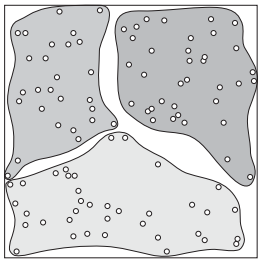
Cluster Evaluation [from Tan/Steinbach/Kumar 05]



Random points

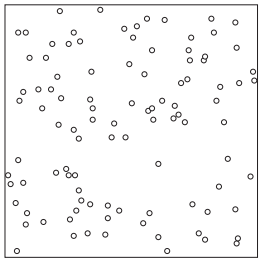


DBSCAN

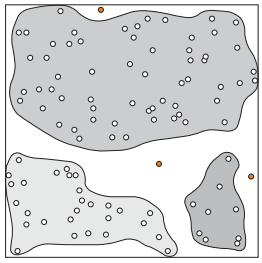


k-means

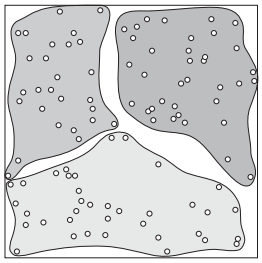
Cluster Evaluation [from Tan/Steinbach/Kumar 05]



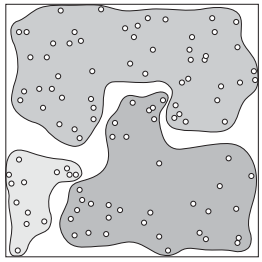
Random points



DBSCAN



k-means



Complete link

Cluster Evaluation

Overview

Cluster evaluation can address different issues:

- Provide evidence whether data contains non-random structures.
- Relate found structures in the data to externally provided class information.
- Rank alternative clusterings with regard to their quality.
- Determine the ideal number of clusters.
- Provide information to choose a suited clustering approach.

Cluster Evaluation

Overview

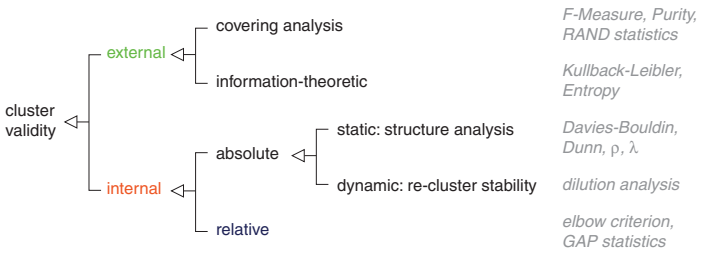
Cluster evaluation can address different issues:

- Provide evidence whether data contains non-random structures.
- Relate found structures in the data to externally provided class information.
- Rank alternative clusterings with regard to their quality.
- Determine the ideal number of clusters.
- Provide information to choose a suited clustering approach.

1. External validity measures:
   Analyze how close is a clustering to a reference.
2. Internal validity measures:
   Analyze intrinsic characteristics of a clustering.
3. Relative validity measures:
   Analyze the sensitivity (of internal measures) during clustering generation.

Cluster Evaluation

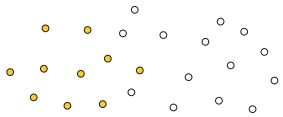
Overview



1. External validity measures:
   Analyze how close is a clustering to a reference.
2. Internal validity measures:
   Analyze intrinsic characteristics of a clustering.
3. Relative validity measures:
   Analyze the sensitivity (of internal measures) during clustering generation.

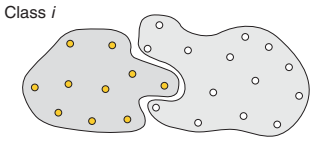
Cluster Evaluation

External Validity Measures: F-Measure



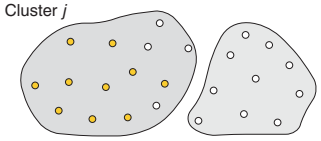
Cluster Evaluation

External Validity Measures: F-Measure



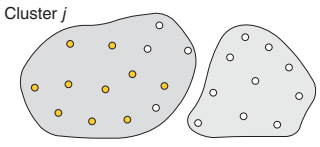
Cluster Evaluation

External Validity Measures: F-Measure



Cluster Evaluation

External Validity Measures: F-Measure

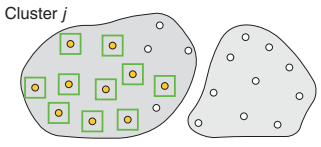


(node-based analysis)

		Truth	
		P	N
Hypothesis	P		
	N		

Cluster Evaluation

External Validity Measures: F-Measure

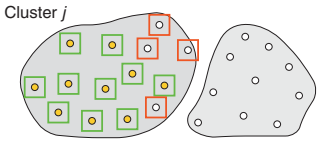


(node-based analysis)

		Truth	
		P	N
Hypothesis	P	TP (a)	
	N		

Cluster Evaluation

External Validity Measures: F-Measure

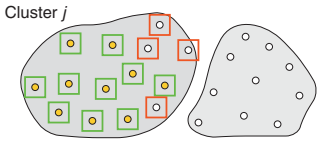


(node-based analysis)

		Truth	
		P	N
Hypothesis	P	TP (a)	FP (b)
	N		

Cluster Evaluation

External Validity Measures: F-Measure

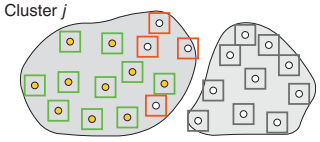


(node-based analysis)

		Truth	
		P	N
Hypothesis	P	TP (a)	FP (b)
	N	FN (c)	

Cluster Evaluation

External Validity Measures: F-Measure

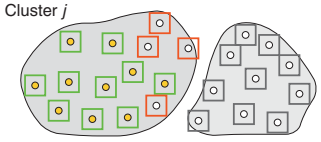


(node-based analysis)

		Truth	
		P	N
Hypothesis	P	TP (a)	FP (b)
	N	FN (c)	TN (d)

Cluster Evaluation

External Validity Measures: F-Measure



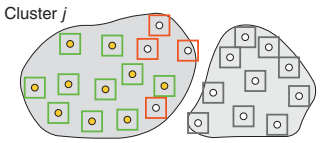
(node-based analysis)

		Truth	
		P	N
Hypothesis	P	TP (a)	FP (b)
	N	FN (c)	TN (d)

Precision: $\frac{a}{a+b}$ Recall: $\frac{a}{a+c}$

Cluster Evaluation

External Validity Measures: F-Measure



(node-based analysis)

		Truth	
		P	N
Hypothesis	P	TP (a)	FP (b)
	N	FN (c)	TN (d)

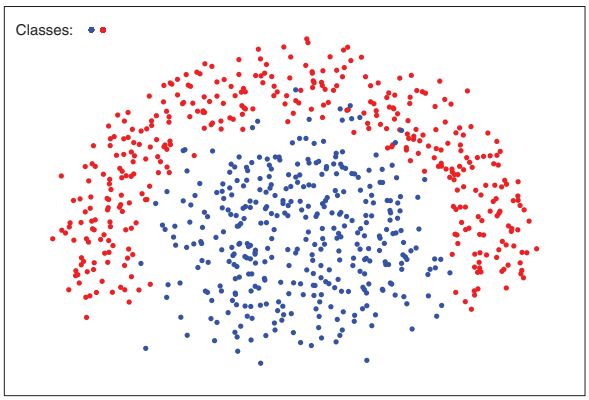
Precision: $\frac{a}{a+b}$ Recall: $\frac{a}{a+c}$ F-measure:

$$F_\alpha = \frac{1+\alpha}{\frac{1}{precision} + \frac{\alpha}{recall}}$$

$\alpha = 1$	harmonic mean
$\alpha \in (0;1)$	favor precision over recall
$\alpha > 1$	favor recall over precision

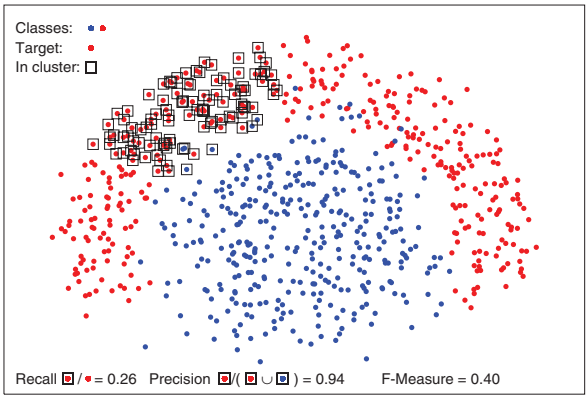
Cluster Evaluation

External Validity Measures: F-Measure



Cluster Evaluation

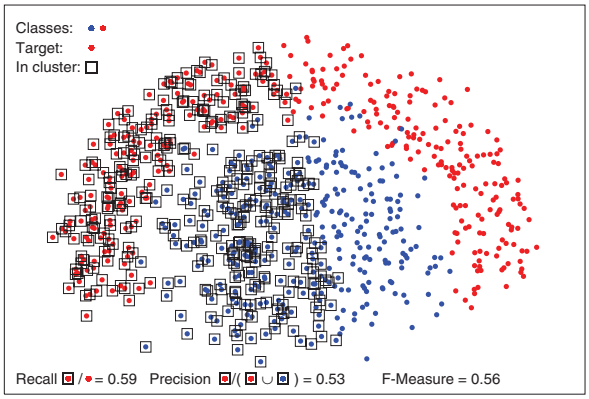
External Validity Measures: F-Measure



High precision, low recall $\Rightarrow$ low F-measure.

Cluster Evaluation

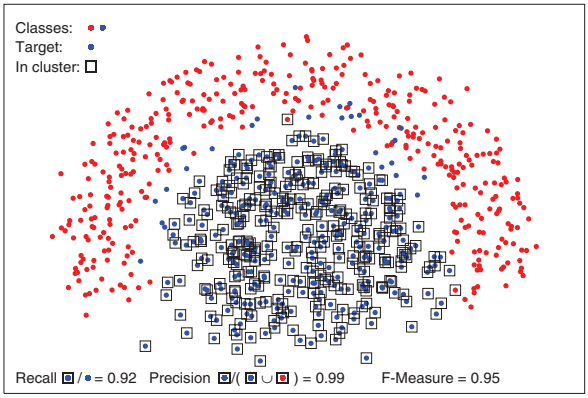
External Validity Measures: F-Measure



Low precision, low recall $\Rightarrow$ low F-measure.

Cluster Evaluation

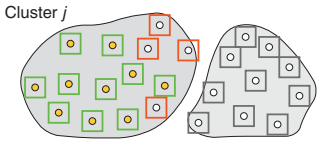
External Validity Measures: F-Measure



High precision, high recall $\Rightarrow$ high F-measure.

Cluster Evaluation

External Validity Measures: F-Measure



(node-based analysis)

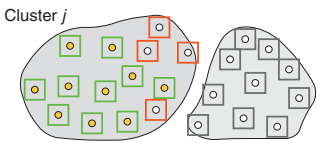
- Clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ and classification $\mathcal{C}^* = \{C_1^*, \ldots, C_l^*\}$ of D.
- $F_{i,j}$ is the F-measure of a cluster j computed *with respect to a class* i.

  Recall of cluster j with respect to class i is $|C_j \cap C_i^*|/|C_i^*|$

  Precision of cluster j with respect to class i is $|C_j \cap C_i^*|/|C_j|$

Cluster Evaluation

External Validity Measures: F-Measure



(node-based analysis)

- Clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ and classification $\mathcal{C}^* = \{C_1^*, \ldots, C_l^*\}$ of D.
- $F_{i,j}$ is the F-measure of a cluster j computed *with respect to a class* i.

  Recall of cluster j with respect to class i is $|C_j \cap C_i^*|/|C_i^*|$

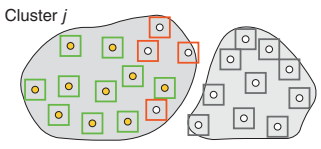
  Precision of cluster j with respect to class i is $|C_j \cap C_i^*|/|C_j|$

- Micro-averaged F-measure for $\langle D, \mathcal{C}, \mathcal{C}^* \rangle$:

$$F = \sum_{i=1}^{l} \frac{|C_i^*|}{|D|} \cdot \max_{j=1,\ldots,k} \{F_{i,j}\}$$

Cluster Evaluation

External Validity Measures: F-Measure



(node-based analysis)

- Clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ and classification $\mathcal{C}^* = \{C_1^*, \ldots, C_l^*\}$ of D.
- $F_{i,j}$ is the F-measure of a cluster j computed *with respect to a class* i.

  Recall of cluster j with respect to class i is $|C_j \cap C_i^*|/|C_i^*|$

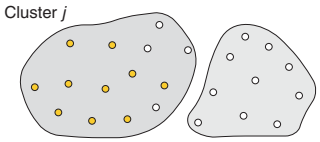
  Precision of cluster j with respect to class i is $|C_j \cap C_i^*|/|C_j|$

- Macro-averaged F-measure for $\langle D, \mathcal{C}, \mathcal{C}^* \rangle$:

$$F = \frac{1}{l} \sum_{i=1}^{l} \max_{j=1,\ldots,k} \{F_{i,j}\}$$

Cluster Evaluation

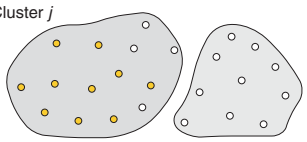
External Validity Measures: Entropy



(node-based analysis)

Cluster Evaluation

External Validity Measures: Entropy



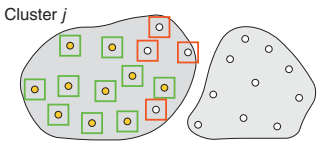
(node-based analysis)

- A cluster C acts as information source $\mathcal{L}$.

  $\mathcal{L}$ emits cluster labels $L_1, \ldots, L_l$ with probabilities $P(L_1), \ldots, P(L_l)$.

Cluster Evaluation

External Validity Measures: Entropy



(node-based analysis)

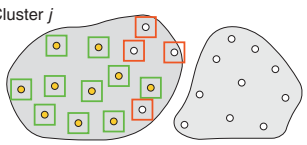
- A cluster C acts as information source $\mathcal{L}$.

  $\mathcal{L}$ emits cluster labels $L_1, \ldots, L_l$ with probabilities $P(L_1), \ldots, P(L_l)$.

  $\hat{P}(\square) = 10/14, \quad \hat{P}(\square) = 4/14$

Cluster Evaluation

External Validity Measures: Entropy



(node-based analysis)

- A cluster C acts as information source $\mathcal{L}$.

  $\mathcal{L}$ emits cluster labels $L_1, \ldots, L_l$ with probabilities $P(L_1), \ldots, P(L_l)$.

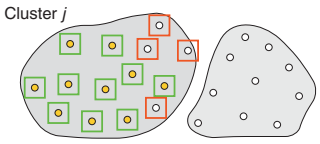
  $\hat{P}(\square) = 10/14, \quad \hat{P}(\square) = 4/14$

- Entropy of $\mathcal{L}$: $H(\mathcal{L}) = -\sum_{i=1}^{l} P(L_i) \cdot \log_2(P(L_i))$

  Entropy of C_j wrt. $\mathcal{C}^*$: $H(C_j) = -\sum_{C_j \cap C_i^* \neq \emptyset} |C_j \cap C_i^*|/|C_j| \cdot \log_2(|C_j \cap C_i^*|/|C_j|)$

Cluster Evaluation

External Validity Measures: Entropy



(node-based analysis)

- A cluster C acts as information source $\mathcal{L}$.

  $\mathcal{L}$ emits cluster labels $L_1, \ldots, L_l$ with probabilities $P(L_1), \ldots, P(L_l)$.

  $\hat{P}(\square) = 10/14, \quad \hat{P}(\square) = 4/14$

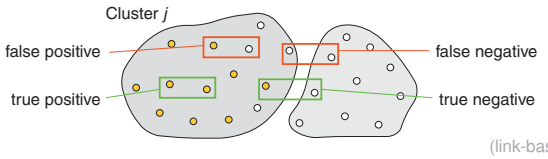
- Entropy of $\mathcal{L}$: $H(\mathcal{L}) = -\sum_{i=1}^{l} P(L_i) \cdot \log_2(P(L_i))$

  Entropy of C_j wrt. $\mathcal{C}^*$: $H(C_j) = -\sum_{C_j \cap C_i^* \neq \emptyset} |C_j \cap C_i^*|/|C_j| \cdot \log_2(|C_j \cap C_i^*|/|C_j|)$

- Entropy of $\mathcal{C}$ wrt. $\mathcal{C}^*$: $H(\mathcal{C}) = \sum_{C_j \in \mathcal{C}} |C_j|/|D| \cdot H(C_j)$

Cluster Evaluation

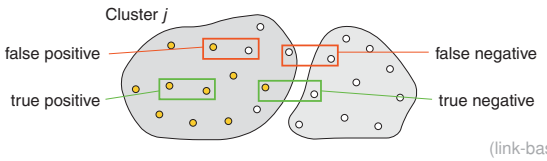
External Validity Measures: Rand, Jaccard



(link-based analysis)

Cluster Evaluation

External Validity Measures: Rand, Jaccard

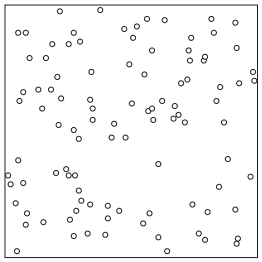


(link-based analysis)

- $R(\mathcal{C}) = \dfrac{|TP| + |TN|}{|TP| + |TN| + |FP| + |FN|} = \dfrac{|TP| + |TN|}{n(n-1)/2}$, with $n = |D|$
- $J(\mathcal{C}) = \dfrac{|TP|}{|TP| + |FP| + |FN|}$

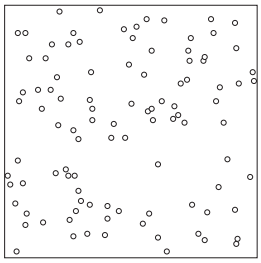
Cluster Evaluation

Internal Validity Measures: Link Correlation [from Tan/Steinbach/Kumar 05]



Cluster Evaluation

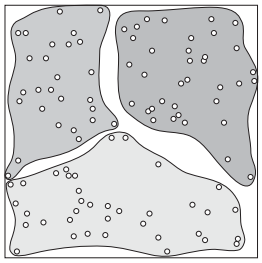
Internal Validity Measures: Link Correlation [from Tan/Steinbach/Kumar 05]



$$\begin{pmatrix} 1.0 & 0.2 & 0.1 & 0.3 & \dots & 0.1 & 0.0 \\ - & 1.0 & 0.1 & 0.0 & \dots & 0.0 & 0.2 \\ & & & & \vdots & & \\ - & - & - & - & - & 1.0 & 0.6 \\ - & - & - & - & - & - & 1.0 \end{pmatrix}$$

Cluster Evaluation

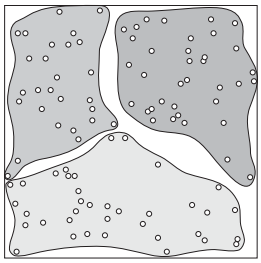
Internal Validity Measures: Link Correlation [from Tan/Steinbach/Kumar 05]



$$\begin{pmatrix} 1.0 & 0.2 & 0.1 & 0.3 & \dots & 0.1 & 0.0 \\ - & 1.0 & 0.1 & 0.0 & \dots & 0.0 & 0.2 \\ & & & & \vdots & & \\ - & - & - & - & - & 1.0 & 0.6 \\ - & - & - & - & - & - & 1.0 \end{pmatrix}$$

Cluster Evaluation

Internal Validity Measures: Link Correlation [from Tan/Steinbach/Kumar 05]

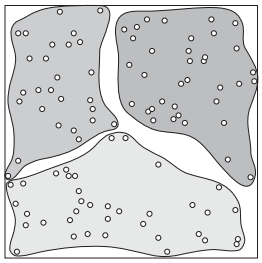


$$\begin{pmatrix} 1.0 & 0.2 & 0.1 & 0.3 & \dots & 0.1 & 0.0 \\ - & 1.0 & 0.1 & 0.0 & \dots & 0.0 & 0.2 \\ & & & & \vdots & & \\ - & - & - & - & - & 1.0 & 0.6 \\ - & - & - & - & - & - & 1.0 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 1 & \dots & 0 & 0 \\ - & 1 & 0 & 0 & \dots & 0 & 1 \\ & & & & \vdots & & \\ - & - & - & - & - & 1 & 1 \\ - & - & - & - & - & - & 1 \end{pmatrix}$$

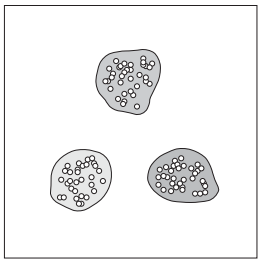
- Construct occurrence matrix based on cluster analysis.
- Compare similarity matrix to occurrence matrix: correlation τ

Cluster Evaluation

Internal Validity Measures: Link Correlation [from Tan/Steinbach/Kumar 05]



k-means
$\tau = 0.5810$



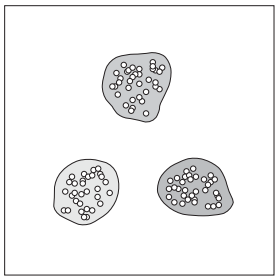
k-means
$\tau = 0.9235$

$$\begin{pmatrix} 1.0 & 0.2 & 0.1 & 0.3 & \dots & 0.1 & 0.0 \\ - & 1.0 & 0.1 & 0.0 & \dots & 0.0 & 0.2 \\ & & & & \vdots & & \\ - & - & - & - & - & 1.0 & 0.6 \\ - & - & - & - & - & - & 1.0 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 1 & \dots & 0 & 0 \\ - & 1 & 0 & 0 & \dots & 0 & 1 \\ & & & & \vdots & & \\ - & - & - & - & - & 1 & 1 \\ - & - & - & - & - & - & 1 \end{pmatrix}$$

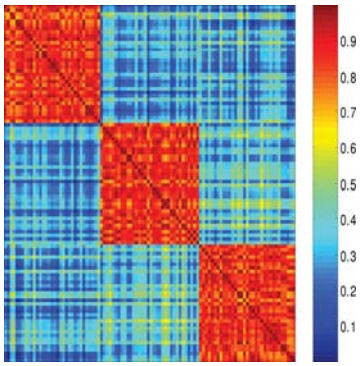
- Construct occurrence matrix based on cluster analysis.
- Compare similarity matrix to occurrence matrix: correlation τ

Cluster Evaluation

Internal Validity Measures: Link Correlation [from Tan/Steinbach/Kumar 05]



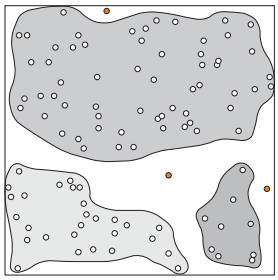
k-means at structured data.



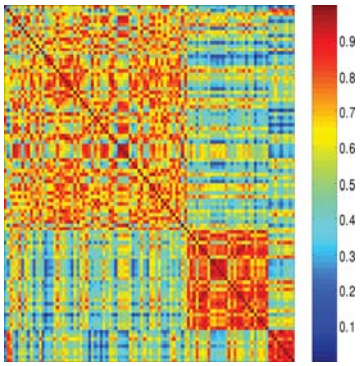
Similarity matrix sorted by cluster label.

Cluster Evaluation

Internal Validity Measures: Link Correlation [from Tan/Steinbach/Kumar 05]



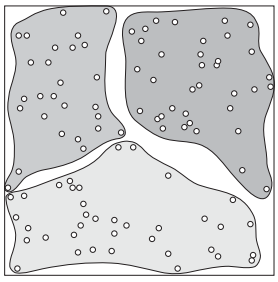
DBSCAN at random data.



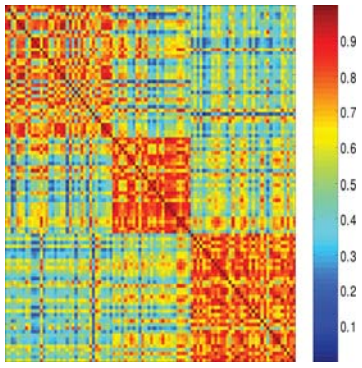
Similarity matrix sorted by cluster label.

Cluster Evaluation

Internal Validity Measures: Link Correlation [from Tan/Steinbach/Kumar 05]



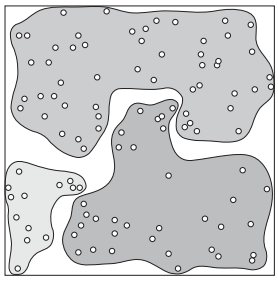
k-means at random data.



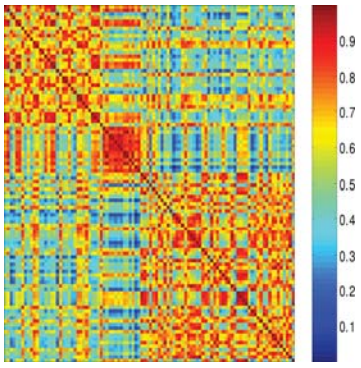
Similarity matrix sorted by cluster label.

Cluster Evaluation

Internal Validity Measures: Link Correlation [from Tan/Steinbach/Kumar 05]



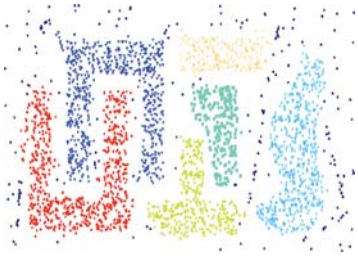
Complete link at random data.



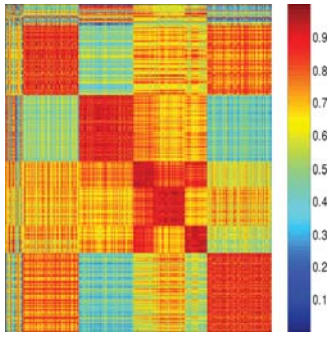
Similarity matrix sorted by cluster label.

Cluster Evaluation

Internal Validity Measures: Link Correlation [from Tan/Steinbach/Kumar 05]



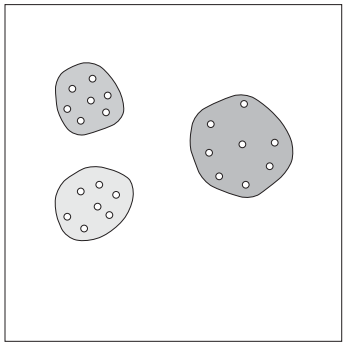
DBSCAN at structured data.



Similarity matrix sorted by cluster label.

Cluster Evaluation

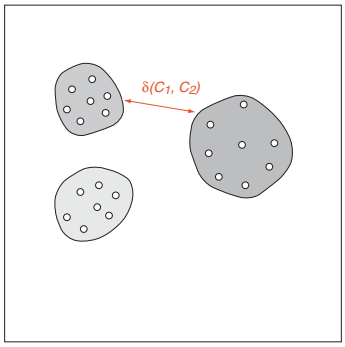
Internal Validity Measures: Structural Analysis



- Distance between two clusters, $\delta(C_1, C_2)$.
- Diameter of a cluster, $\Delta(C)$.
- Scatter within a cluster, $\sigma^2(C)$, SSE.

Cluster Evaluation

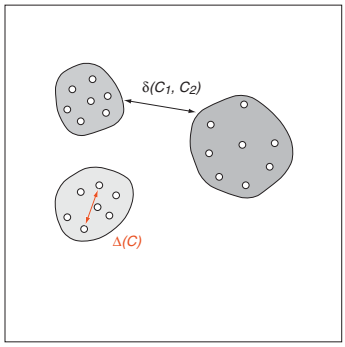
Internal Validity Measures: Structural Analysis



- Distance between two clusters, $\delta(C_1, C_2)$.
- Diameter of a cluster, $\Delta(C)$.
- Scatter within a cluster, $\sigma^2(C)$, SSE.

Cluster Evaluation

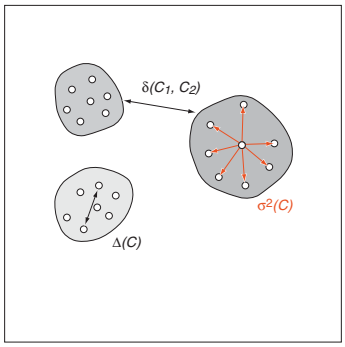
Internal Validity Measures: Structural Analysis



- Distance between two clusters, $\delta(C_1, C_2)$.
- Diameter of a cluster, $\Delta(C)$.
- Scatter within a cluster, $\sigma^2(C)$, SSE.

Cluster Evaluation

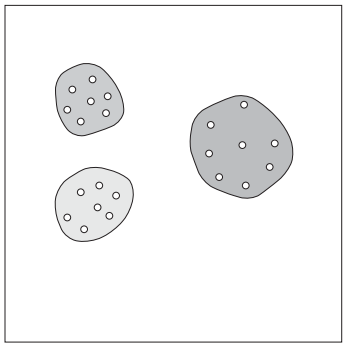
Internal Validity Measures: Structural Analysis



- Distance between two clusters, $\delta(C_1, C_2)$.
- Diameter of a cluster, $\Delta(C)$.
- Scatter within a cluster, $\sigma^2(C)$, SSE.

Cluster Evaluation

Internal Validity Measures: Dunn Index

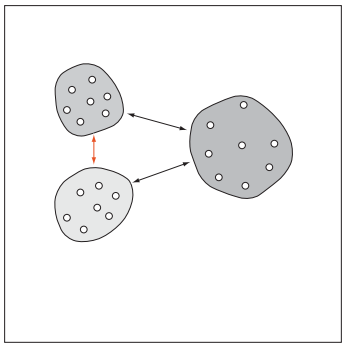


$$I(\mathcal{C}) = \frac{\min_{i \neq j}\{\delta(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}},$$

$I(\mathcal{C}) \rightarrow \max$

Cluster Evaluation

Internal Validity Measures: Dunn Index

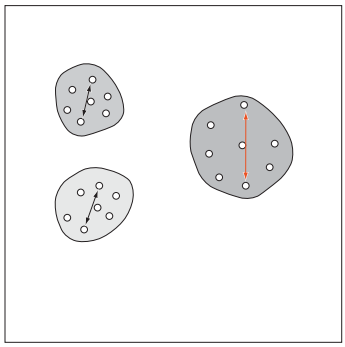


$$I(\mathcal{C}) = \frac{\min_{i \neq j}\{\delta(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}},$$

$I(\mathcal{C}) \rightarrow \max$

Cluster Evaluation

Internal Validity Measures: Dunn Index

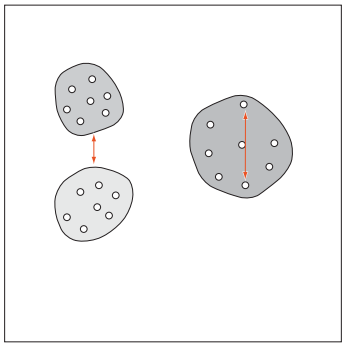


$$I(\mathcal{C}) = \frac{\min_{i \neq j}\{\delta(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}},$$

$I(\mathcal{C}) \rightarrow \max$

Cluster Evaluation

Internal Validity Measures: Dunn Index



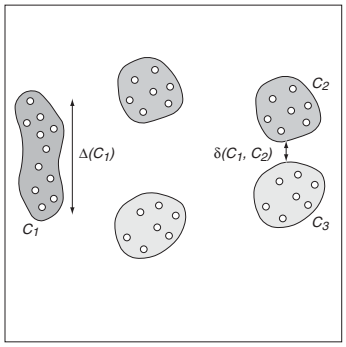
$$I(\mathcal{C}) = \frac{\min_{i \neq j}\{\delta(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}},$$

$I(\mathcal{C}) \rightarrow \max$

- Dunn is susceptible to noise.
- Dunn is biased towards the worst substructure in a clustering (cf. min)
- Dunn cannot put distances and diameters into relation.

Cluster Evaluation

Internal Validity Measures: Dunn Index



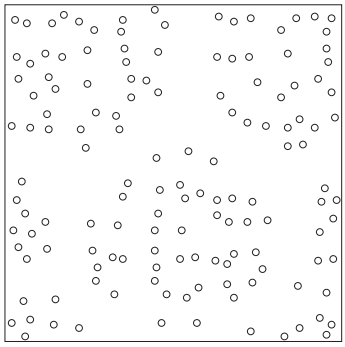
$$I(\mathcal{C}) = \frac{\min_{i \neq j}\{\delta(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}},$$

$I(\mathcal{C}) \rightarrow \max$

- Dunn is susceptible to noise.
- Dunn is biased towards the worst substructure in a clustering (cf. min)
- Dunn cannot put distances and diameters into relation.

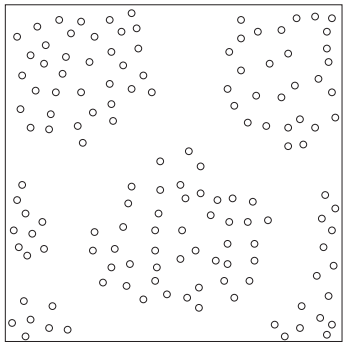
Cluster Evaluation

Internal Validity Measures: Expected Density ρ [Stein/Meyer zu Eissen 2007]



Cluster Evaluation

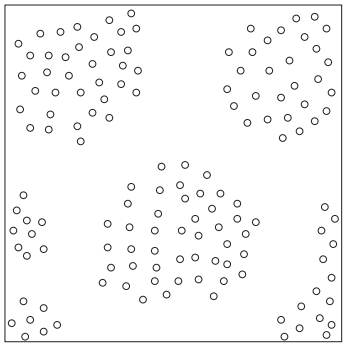
Internal Validity Measures: Expected Density ρ [Stein/Meyer zu Eissen 2007]



Different retrieval models yield different similarity graphs.

Cluster Evaluation

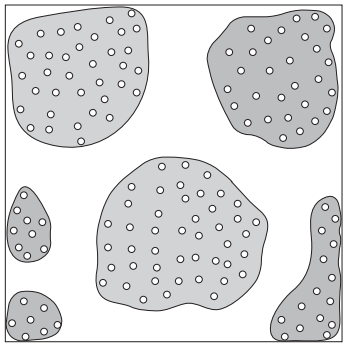
Internal Validity Measures: Expected Density ρ [Stein/Meyer zu Eissen 2007]



Different retrieval models yield different similarity graphs.

Cluster Evaluation

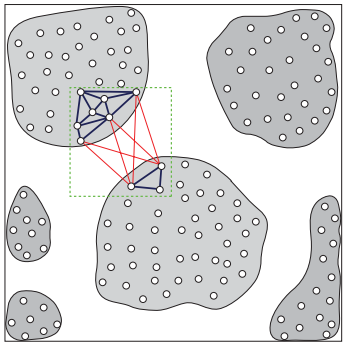
Internal Validity Measures: Expected Density ρ



Compare (for alternative clusterings) the similarity density within the clusters to the average similarity of the entire graph.

Cluster Evaluation

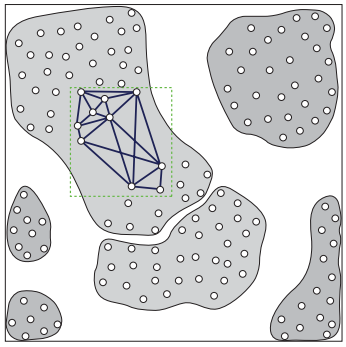
Internal Validity Measures: Expected Density ρ



Compare (for alternative clusterings) the similarity density within the clusters to the average similarity of the entire graph.

Cluster Evaluation

Internal Validity Measures: Expected Density ρ



Compare (for alternative clusterings) the similarity density within the clusters to the average similarity of the entire graph.

Cluster Evaluation

Internal Validity Measures: Expected Density ρ

Graph $G = \langle V, E \rangle$

- G is called sparse [dense] if $|E| = O(|V|)$ $[O(|V|^2)]$
- the density θ computes from the equation $|E| = |V|^\theta$

Cluster Evaluation

Internal Validity Measures: Expected Density ρ

Graph $G = \langle V, E \rangle$

- G is called sparse [dense] if $|E| = O(|V|)$ $[O(|V|^2)]$
- the density θ computes from the equation $|E| = |V|^\theta$

Similarity graph $G = \langle V, E, w \rangle$, $\quad |E| \sim w(G) := \sum_{e \in E} w(e)$

- the density θ computes from the equation $w(G) = |V|^\theta$

Cluster Evaluation

Internal Validity Measures: Expected Density ρ

Graph $G = \langle V, E \rangle$

- G is called sparse [dense] if $|E| = O(|V|)$ $[O(|V|^2)]$
- the density θ computes from the equation $|E| = |V|^\theta$

Similarity graph $G = \langle V, E, w \rangle$, $\quad |E| \sim w(G) := \sum_{e \in E} w(e)$

- the density θ computes from the equation $w(G) = |V|^\theta$

Induced subgraph G_i for class C_i

- the expected density ρ compares class C_i to the density average in D

$$\rho(G_i) = \frac{w(G_i)}{|V_i|^\theta}$$

Cluster Evaluation

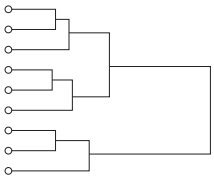
Relative Validity Measures: Elbow Criterion

1. Hyperparameters of a clustering algorithm: $p_1, \ldots, p_m$
   - number of centroids for k-means
   - stopping level for hierarchical algorithms
   - neighborhood size for DBSCAN
2. Clusterings $\mathcal{C} = \{\mathcal{C}_{p_1}, \ldots, \mathcal{C}_{p_m}\}$ associated with $p_1, \ldots, p_m$.
3. Points of an error curve $\{(p_i, e(\mathcal{C}_{p_i})) \mid i = 1, \ldots, m\}$.

Cluster Evaluation

Relative Validity Measures: Elbow Criterion

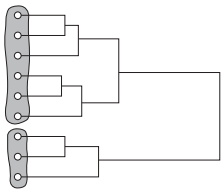
1. Hyperparameters of a clustering algorithm: $p_1, \ldots, p_m$
   - number of centroids for k-means
   - stopping level for hierarchical algorithms
   - neighborhood size for DBSCAN
2. Clusterings $\mathcal{C} = \{\mathcal{C}_{p_1}, \ldots, \mathcal{C}_{p_m}\}$ associated with $p_1, \ldots, p_m$.
3. Points of an error curve $\{(p_i, e(\mathcal{C}_{p_i})) \mid i = 1, \ldots, m\}$.

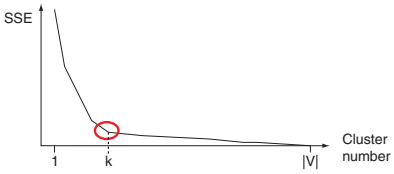


Cluster Evaluation

Relative Validity Measures: Elbow Criterion

1. Hyperparameters of a clustering algorithm: $p_1, \ldots, p_m$
   - number of centroids for k-means
   - stopping level for hierarchical algorithms
   - neighborhood size for DBSCAN
2. Clusterings $\mathcal{C} = \{\mathcal{C}_{p_1}, \ldots, \mathcal{C}_{p_m}\}$ associated with $p_1, \ldots, p_m$.
3. Points of an error curve $\{(p_i, e(\mathcal{C}_{p_i})) \mid i = 1, \ldots, m\}$.

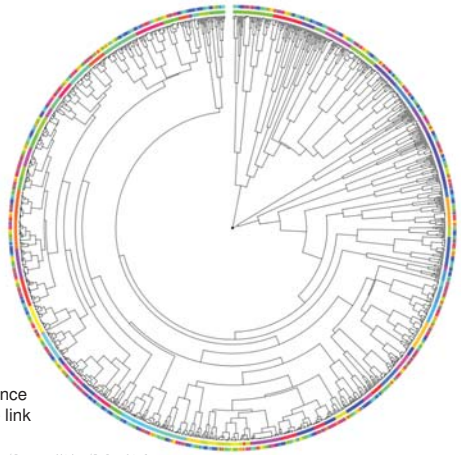




4. Find point that maximizes error drop with respect to its predecessor.

Cluster Evaluation

Relative Validity Measures: Elbow Criterion



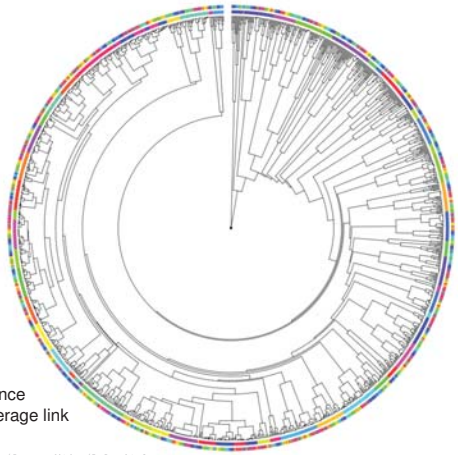
d_C: Hamming distance
Merging: complete link

http://cs.jhu.edu/~razvanm/fs-expedition/2.6.x.html

Relations between 1377 file systems for Linux Kernel 2.6.0. [Razvan Musaloiu-E. 2009]

Cluster Evaluation

Relative Validity Measures: Elbow Criterion



d_C: Hamming distance
Merging: group average link

http://cs.jhu.edu/~razvanm/fs-expedition/2.6.x.html

Relations between 1377 file systems for Linux Kernel 2.6.0. [Razvan Musaloiu-E. 2009]

Cluster Evaluation

Correlation between External and Internal Measures

In the wild, we are not given a reference classification.

- ➜ An external evaluation is not possible.
  (though many papers report on such experiments)
- ➜ Resort to an internal evaluation.
  (connectivity, squared error sums, distance-diameter heuristics, etc.)

Cluster Evaluation

Correlation between External and Internal Measures

In the wild, we are not given a reference classification.

- ➜ An external evaluation is not possible.
  (though many papers report on such experiments)
- ➜ Resort to an internal evaluation.
  (connectivity, squared error sums, distance-diameter heuristics, etc.)

"To which extent can an internal evaluation be used to predict for a clustering its distance from the best reference classification—say, to predict the F-measure?"

$$\operatorname*{argmax}_{\phi} \{\tau\langle X, Y\rangle \mid x = F(\mathcal{C}),\ y = \phi(\mathcal{C}),\ \mathcal{C} \in \boldsymbol{\mathcal{C}}\}$$

[Stein/Meyer zu Eissen 2007]

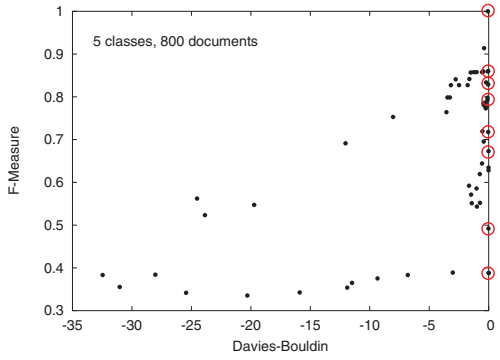
Cluster Evaluation

Correlation between External and Internal Measures

Perfect correlation (desired).

Cluster Evaluation

Correlation between External and Internal Measures

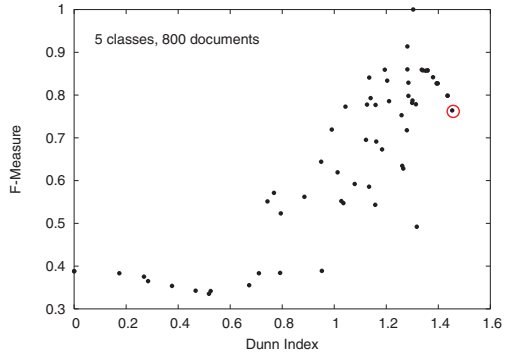


Davies-Bouldin: $\frac{1}{k} \cdot \sum_{i=1}^{k} \max_{j} \frac{s(C_i) + s(C_j)}{\delta(C_i, C_j)}$

Prefers spherical clusters.

Cluster Evaluation

Correlation between External and Internal Measures

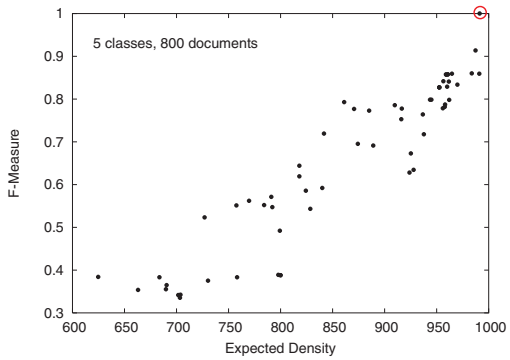


Dunn Index: $\frac{\min_{i \neq j}\{\delta(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}}$

Maximizes dilatation = inter/intra-cluster-diameter.

Cluster Evaluation

Correlation between External and Internal Measures



Expected Density: $\bar{\rho} = \sum_{i=1}^{k} \frac{|V_i|}{|V|} \cdot \frac{w(G_i)}{|V_i|^{\theta}}$

Independent of cluster forms and sizes.

Chapter ML: IX (continued)

Constrained Cluster Analysis

Person Resolution Task

Constrained Cluster Analysis

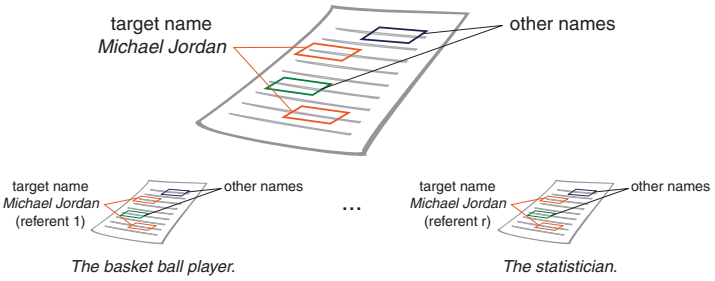
Person Resolution Task

Constrained Cluster Analysis

Person Resolution Task

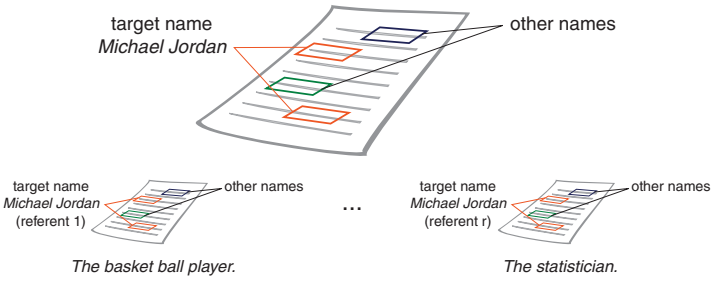
Constrained Cluster Analysis

Person Resolution Task



Constrained Cluster Analysis

Person Resolution Task

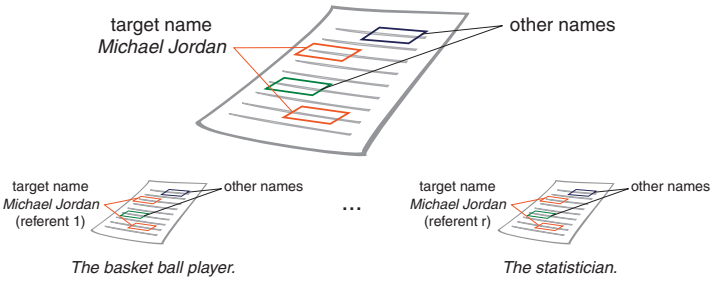


- **Multi-document resolution task:**

Names, Target names:	$N = \{n_1, \ldots, n_l\},$	$T \subset N$
Referents:	$R = \{r_1, \ldots, r_m\},$	$\tau : R \to T,\ \ \lvert R\rvert \gg \lvert T\rvert$
Documents:	$D = \{d_1, \ldots, d_n\},$	$\nu : D \to \mathcal{P}(N),\ \ \lvert\nu(d_i) \cap T\rvert = 1$
A solution:	$\gamma : D \to R,$	s.t. $\tau(\gamma(d_i)) \in \nu(d_i)$

Constrained Cluster Analysis

Person Resolution Task

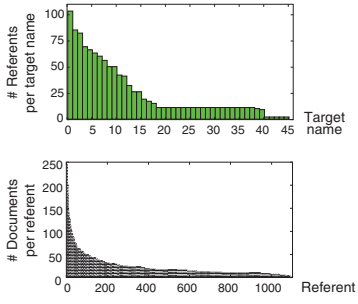


- **Facts about the Spock data mining challenge:**

Target names:	$\lvert T\rvert = 44$
Referents:	$\lvert R\rvert = 1\,101$
Documents:	$\lvert D_{train}\rvert = 27\,000$ (labeled $\approx$ 2.3GB)
	$\lvert D_{test}\rvert = 75\,000$ (unlabeled $\approx$ 7.8GB)

Constrained Cluster Analysis

Person Resolution Task



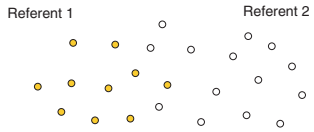
- up to 105 referents for a single target name
- about 25 referents on average per target name
- about 23 documents on average per referent

- **Facts about the Spock data mining challenge:**

Target names:	$\lvert T\rvert = 44$
Referents:	$\lvert R\rvert = 1\,101$
Documents:	$\lvert D_{train}\rvert = 27\,000$ (labeled $\approx$ 2.3GB)
	$\lvert D_{test}\rvert = 75\,000$ (unlabeled $\approx$ 7.8GB)

Constrained Cluster Analysis

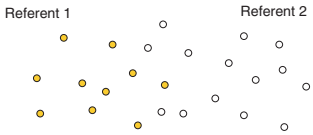
Applied to Multi-Document Resolution



1. Model similarities ➜ new and established retrieval models:
   - ❑ global and context-based vector space models
   - ❑ explicit semantic analysis
   - ❑ ontology alignment
2. Learn class memberships (supervised) ➜ logistic regression
3. Find equivalence classes (unsupervised) ➜ cluster analysis:
   - (a) adaptive graph thinning
   - (b) multiple, density-based cluster analysis
   - (c) clustering selection by expected density maximization

Constrained Cluster Analysis

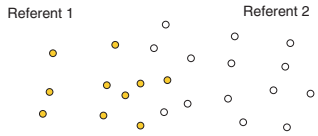
Applied to Multi-Document Resolution



1. Model similarities ➜ new and established retrieval models:
   - ❑ global and context-based vector space models
   - ❑ explicit semantic analysis
   - ❑ ontology alignment
2. Learn class memberships (supervised) ➜ logistic regression
3. Find equivalence classes (unsupervised) ➜ cluster analysis:
   - (a) adaptive graph thinning
   - (b) multiple, density-based cluster analysis
   - (c) clustering selection by expected density maximization

Constrained Cluster Analysis

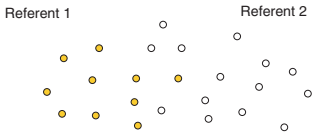
Applied to Multi-Document Resolution



1. Model similarities ➜ new and established retrieval models:
   - ❑ global and context-based vector space models
   - ❑ explicit semantic analysis
   - ❑ ontology alignment
2. Learn class memberships (supervised) ➜ logistic regression
3. Find equivalence classes (unsupervised) ➜ cluster analysis:
   - (a) adaptive graph thinning
   - (b) multiple, density-based cluster analysis
   - (c) clustering selection by expected density maximization

Constrained Cluster Analysis

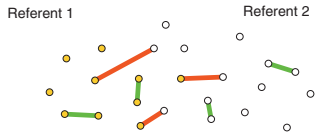
Applied to Multi-Document Resolution



1. Model similarities ➜ new and established retrieval models:
   - ❑ global and context-based vector space models
   - ❑ explicit semantic analysis
   - ❑ ontology alignment
2. Learn class memberships (supervised) ➜ logistic regression
3. Find equivalence classes (unsupervised) ➜ cluster analysis:
   - (a) adaptive graph thinning
   - (b) multiple, density-based cluster analysis
   - (c) clustering selection by expected density maximization

Constrained Cluster Analysis

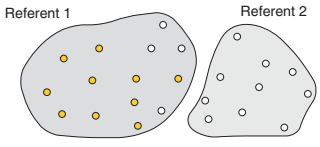
Applied to Multi-Document Resolution



1. Model similarities ➜ new and established retrieval models:
   - ❑ global and context-based vector space models
   - ❑ explicit semantic analysis
   - ❑ ontology alignment
2. Learn class memberships (supervised) ➜ logistic regression
3. Find equivalence classes (unsupervised) ➜ cluster analysis:
   - (a) adaptive graph thinning
   - (b) multiple, density-based cluster analysis
   - (c) clustering selection by expected density maximization

Constrained Cluster Analysis

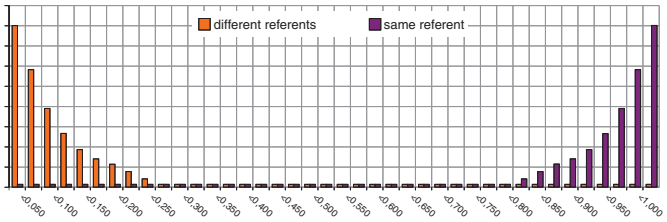
Applied to Multi-Document Resolution



1. Model similarities ➜ new and established retrieval models:
   - ❑ global and context-based vector space models
   - ❑ explicit semantic analysis
   - ❑ ontology alignment
2. Learn class memberships (supervised) ➜ logistic regression
3. Find equivalence classes (unsupervised) ➜ cluster analysis:
   - (a) adaptive graph thinning
   - (b) multiple, density-based cluster analysis
   - (c) clustering selection by expected density maximization

Constrained Cluster Analysis

Idealized Class Membership Distribution over Similarities



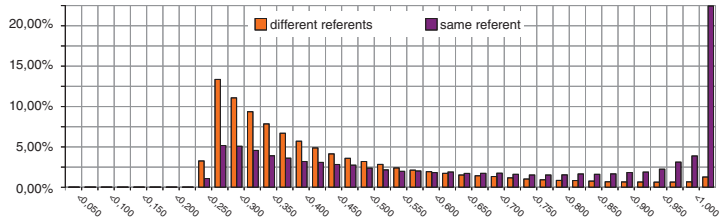
Similarity distributions for document pairs from different referents and same referent.

Logistic regression task:

- sample size: 400 000
- classes imbalance: non-target class : target class ≈ 25:1
- items are drawn uniformly distributed wrt. non-targets and targets
- items are uniformly distributed over the groups of target names

Constrained Cluster Analysis

Membership Distribution under *tf·idf* Vector Space Model

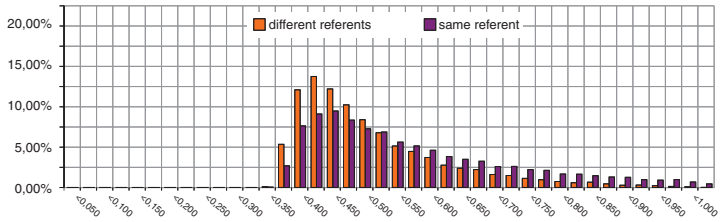


Model details:

- corpus size: 25 000 documents
- dictionary size: 1,2 Mio terms
- stopwords number: 850
- stopword volume: 36%

Constrained Cluster Analysis

Membership Distribution under Context-Based Vector Space Model

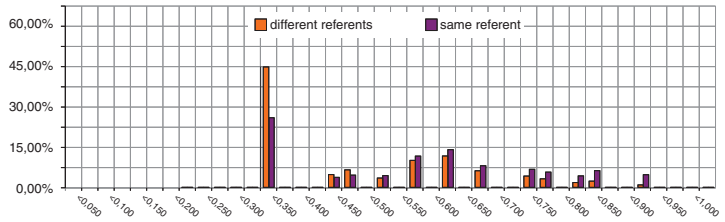


Model details:

- corpus size: 25 000 documents
- dictionary size: 1,2 Mio terms
- stopwords number: 850
- stopword volume: 36%

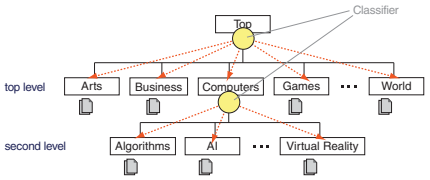
Constrained Cluster Analysis

Membership Distribution under Ontology Alignment Model



Model details:

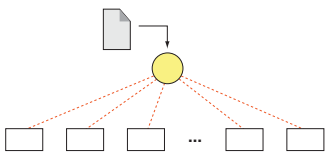
- DMOZ open directory project
- > 5 million documents
- 12 top-level categories
- 31 second level categories
- ML: hierarchical Bayes
- training set: 100 000 pages



Constrained Cluster Analysis

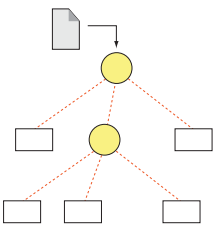
In-Depth: Multi-Class Hierarchical Classification

Flat (big-bang) classification



+ simple realization
– loss of discriminative power with increasing number of categories

Hierarchical (top-down) classification



+ specialized classifiers (divide and conquer)
– misclassification at higher levels can never become repaired

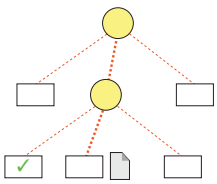
Constrained Cluster Analysis

In-Depth: Multi-Class Hierarchical Classification

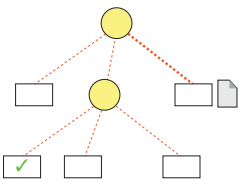
State of the art of effectiveness analyses:

1. independence assumption between categories
2. neglection of both hierarchical structure and degree of misclassification

(a)



(b)



Improvements:

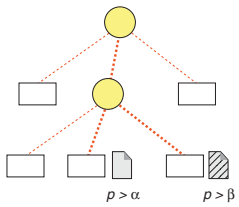
- Consider similarity $\varphi(C_i, C_j)$ between correct and wrong category.
- Consider graph distance $d(C_i, C_j)$ between correct and wrong category.

Constrained Cluster Analysis

In-Depth: Multi-Class Hierarchical Classification

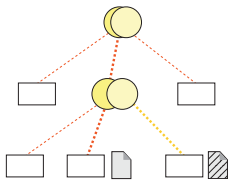
Improvements continued:

Multi-label (multi path) classification



- traverse more than one path and return all labels
- employ probabilistic classifiers with a threshold: split a path or not

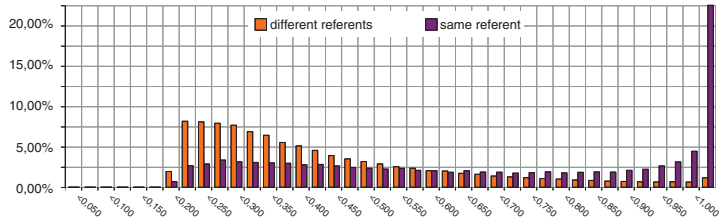
Multi-classifier (ensemble) classification



- classification result is a majority decision
- employ different classifier (different types or differently parameterized)

Constrained Cluster Analysis

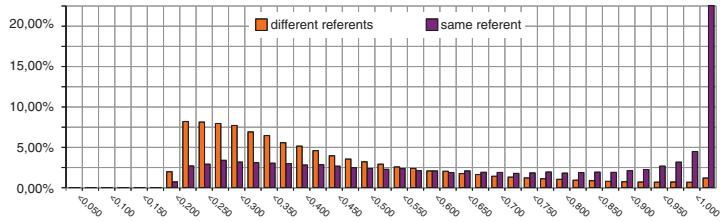
Membership Distribution under Optimized Retrieval Model Combination



Retrieval Model	$F_{1/3}$ -Measure
tf-idf vector space	0.39
context-based vector space	0.32
ESA Wikipedia persons	0.30
phrase structure grammar	0.17
ontology alignment	0.15
optimized combination	**0.42**

Constrained Cluster Analysis

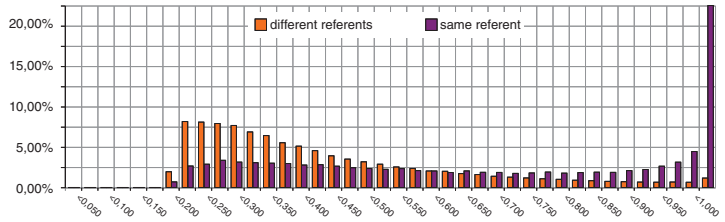
Membership Distribution under Optimized Retrieval Model Combination



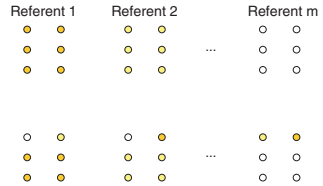
Retrieval Model	$F_{1/3}$ -Measure
tf-idf vector space	0.39
context-based vector space	0.32
ESA Wikipedia persons	0.30
phrase structure grammar	0.17
ontology alignment	0.15
optimized combination	**0.42**

Constrained Cluster Analysis

Membership Distribution under Optimized Retrieval Model Combination

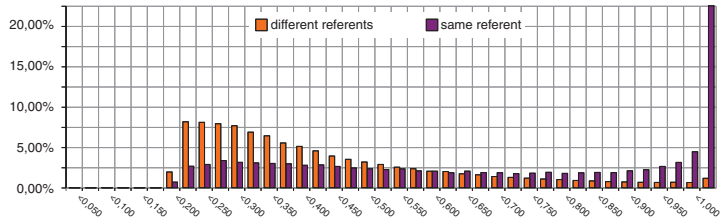


Retrieval Model	$F_{1/3}$ -Measure
tf-idf vector space	0.39
context-based vector space	0.32
ESA Wikipedia persons	0.30
phrase structure grammar	0.17
ontology alignment	0.15
optimized combination	**0.42**

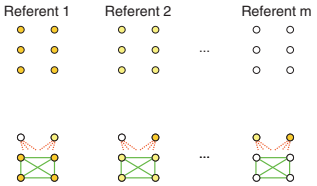


Constrained Cluster Analysis

Membership Distribution under Optimized Retrieval Model Combination

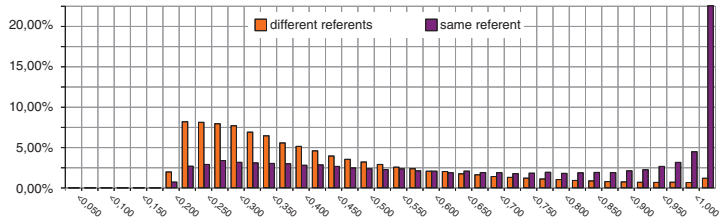


Retrieval Model	$F_{1/3}$ -Measure
tf-idf vector space	0.39
context-based vector space	0.32
ESA Wikipedia persons	0.30
phrase structure grammar	0.17
ontology alignment	0.15
optimized combination	**0.42**



Constrained Cluster Analysis

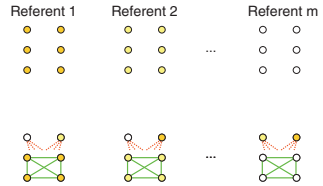
Membership Distribution under Optimized Retrieval Model Combination



In the example:

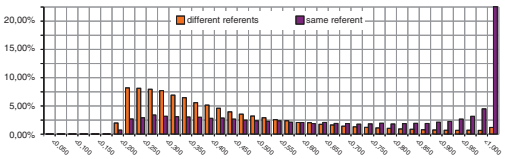
- precision = 0.4
- recall = 0.43
- $F_{1/3}$ = 0.41

(if false negatives are uniformly distributed)

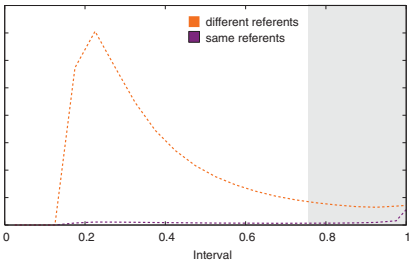


Constrained Cluster Analysis

In-Depth: Analysis of Classifier Effectiveness



Consideration of imbalance:



Constrained Cluster Analysis

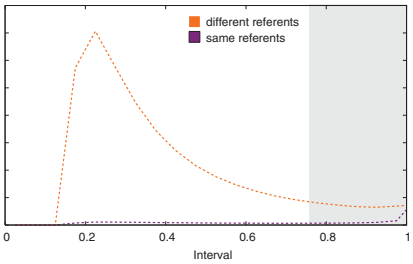
In-Depth: Analysis of Classifier Effectiveness

- class imbalance factor (*CIF*) of 25

⇒ precision in interval [0.725; 1] for edges between same referents: ≈ 0.17

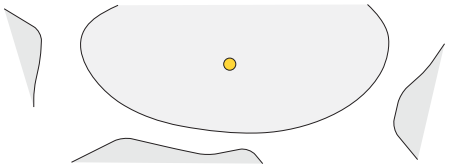
How can $F_{1/3} = 0.42$ be achieved via cluster analysis?

Consideration of imbalance:



Constrained Cluster Analysis

In-Depth: Analysis of Classifier Effectiveness



Assumption: uniform distribution of referents over documents (here: 25 clusters with $|C| = 23$)

⇒ $|TP|$ true 1-similarities per cluster (here: 130 @ threshold 0.725)

⇒ $\frac{|TP|}{|C|}$ degree of true positives per node (here: 11)

⇒ $|TP|(\frac{1}{precision} - 1)$ false 1-similarities per cluster (here: 760)

Density-based cluster analysis: effective false positives, FP^*, connect to same cluster

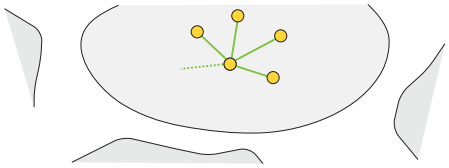
⇒ analyze $P(|FP^*| > k \mid D, R_{iid})$ (here: $E(|FP^*|) = 2.7$)

⇒ edge tie factor (*ETF*) specifies the excess of true positives until tie (here: $3 \ldots 5$)

$$ETF = \frac{|TP|}{|C| \cdot E(|FP^*|)}, \qquad \text{effective precision} = precision \cdot \frac{CIF}{ETF}$$

Constrained Cluster Analysis

In-Depth: Analysis of Classifier Effectiveness



Assumption: uniform distribution of referents over documents (here: 25 clusters with $|C| = 23$)

⇒ $|TP|$ true 1-similarities per cluster (here: 130 @ threshold 0.725)

⇒ $\frac{|TP|}{|C|}$ degree of true positives per node (here: 11)

⇒ $|TP|(\frac{1}{precision} - 1)$ false 1-similarities per cluster (here: 760)

Density-based cluster analysis: effective false positives, FP^*, connect to same cluster

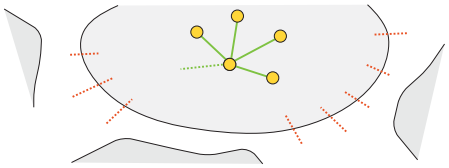
⇒ analyze $P(|FP^*| > k \mid D, R_{iid})$ (here: $E(|FP^*|) = 2.7$)

⇒ edge tie factor (*ETF*) specifies the excess of true positives until tie (here: $3 \ldots 5$)

$$ETF = \frac{|TP|}{|C| \cdot E(|FP^*|)}, \qquad \text{effective precision} = precision \cdot \frac{CIF}{ETF}$$

Constrained Cluster Analysis

In-Depth: Analysis of Classifier Effectiveness



Assumption: uniform distribution of referents over documents (here: 25 clusters with $|C| = 23$)

⇒ $|TP|$ true 1-similarities per cluster (here: 130 @ threshold 0.725)

⇒ $\frac{|TP|}{|C|}$ degree of true positives per node (here: 11)

⇒ $|TP|(\frac{1}{precision} - 1)$ false 1-similarities per cluster (here: 760)

Density-based cluster analysis: effective false positives, FP^*, connect to same cluster

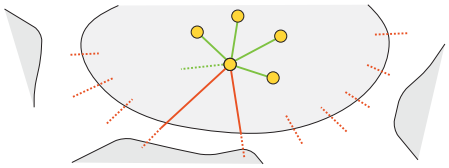
⇒ analyze $P(|FP^*| > k \mid D, R_{iid})$ (here: $E(|FP^*|) = 2.7$)

⇒ edge tie factor (*ETF*) specifies the excess of true positives until tie (here: $3 \ldots 5$)

$$ETF = \frac{|TP|}{|C| \cdot E(|FP^*|)}, \qquad \text{effective precision} = precision \cdot \frac{CIF}{ETF}$$

Constrained Cluster Analysis

In-Depth: Analysis of Classifier Effectiveness



Assumption: uniform distribution of referents over documents (here: 25 clusters with $|C| = 23$)

⇒ $|TP|$ true 1-similarities per cluster (here: 130 @ threshold 0.725)

⇒ $\frac{|TP|}{|C|}$ degree of true positives per node (here: 11)

⇒ $|TP|(\frac{1}{precision} - 1)$ false 1-similarities per cluster (here: 760)

Density-based cluster analysis: effective false positives, FP^*, connect to same cluster

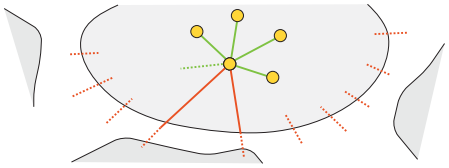
⇒ analyze $P(|FP^*| > k \mid D, R_{iid})$ (here: $E(|FP^*|) = 2.7$)

⇒ edge tie factor (*ETF*) specifies the excess of true positives until tie (here: $3 \ldots 5$)

$$ETF = \frac{|TP|}{|C| \cdot E(|FP^*|)}, \qquad \text{effective precision} = precision \cdot \frac{CIF}{ETF}$$

Constrained Cluster Analysis

In-Depth: Analysis of Classifier Effectiveness



Assumption: uniform distribution of referents over documents (here: 25 clusters with $|C| = 23$)

- $\Rightarrow$ $|TP|$ true 1-similarities per cluster (here: 130 @ threshold 0.725)
- $\Rightarrow$ $\frac{|TP|}{|C|}$ degree of true positives per node (here: 11)
- $\Rightarrow$ $|TP|(\frac{1}{precision} - 1)$ false 1-similarities per cluster (here: 760)

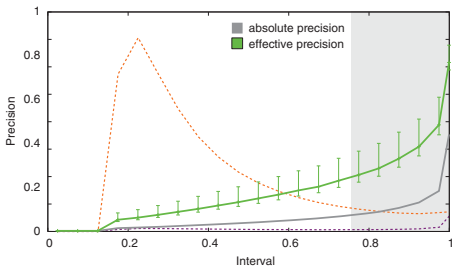
Density-based cluster analysis: effective false positives, FP^*, connect to same cluster

- $\Rightarrow$ analyze $P(|FP^*| > k \mid D, R_{iid})$ (here: $E(|FP^*|) = 2.7$)
- $\Rightarrow$ edge tie factor (ETF) specifies the excess of true positives until tie (here: $3 \ldots 5$)

$$ETF = \frac{|TP|}{|C| \cdot E(|FP^*|)}, \qquad \text{effective precision} = precision \cdot \frac{CIF}{ETF}$$

Constrained Cluster Analysis

In-Depth: Analysis of Classifier Effectiveness



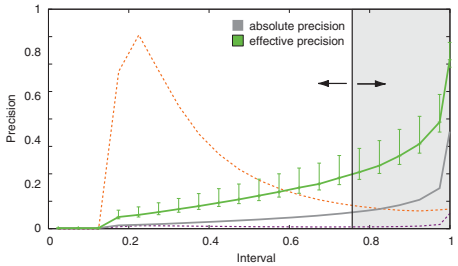
Density-based cluster analysis: effective false positives, FP^*, connect to same cluster

- $\Rightarrow$ analyze $P(|FP^*| > k \mid D, R_{iid})$ (here: $E(|FP^*|) = 2.7$)
- $\Rightarrow$ edge tie factor (ETF) specifies the excess of true positives until tie (here: $3 \ldots 5$)

$$ETF = \frac{|TP|}{|C| \cdot E(|FP^*|)}, \qquad \text{effective precision} = precision \cdot \frac{CIF}{ETF}$$

Constrained Cluster Analysis

In-Depth: Analysis of Classifier Effectiveness



Determine optimum similarity threshold for class-membership function:

$$\theta^* = \underset{\theta \in [0;1]}{\text{argmax}} \{ \frac{1+\alpha}{\frac{ETF}{precision_\theta \cdot CIF} + \frac{\alpha}{recall_\theta}} \}$$

θ^* considers co-variate shift, introduces model formation bias and sample selection bias.

Constrained Cluster Analysis

Model Selection: Our Risk Minimization Strategy

Retrieval Model	$F_{1/3}$-Measure
tfidf vector space	0.39
context-based vector space	0.32
ESA Wikipedia persons	0.30
phrase structure grammar	0.17
ontology alignment	0.15
optimized combination	**0.42**
Ensemble cluster analysis	**0.40**

Ensemble cluster analysis: higher bias, better generalization.

(1) Do we speculate on a better fit?

(2) Do we expect a significant covariate shift, more noise, etc.?

Constrained Cluster Analysis

Recap

1. Multi-document resolution can be tackled with constrained cluster analysis.
2. Constraints are derived from labeled examples.
3. Class membership function ties constraints to multiple retrieval models.
4. Advanced density-based clustering technology is key.

Constrained Cluster Analysis

References

- Disambiguating Web Appearances of People in a Social Network.
  [R. Bekkerman, A. McCallum. WWW 2005]
- A Bayesian Model for Supervised Clustering with the Dirichlet Process Prior.
  [H. Daumé III, D. Marcu. Journal MLR 2005]
- Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis.
  [E. Gabrilovich, S. Markovitch. IJCAI 2007]
- Unsupervised Discrimination of Person Names in Web Contexts.
  [T. Pedersen, A. Kulkarni. CICLing 2007]
- On Information Need and Categorizing Search.
  [S. Meyer zu Eissen. Dissertation, Paderborn University, 2007]
- Weighted Experts: A Solution for the Spock Data Mining Challenge.
  [B. Stein, S. Meyer zu Eissen. I-KNOW 2008]
- GRAPE: A System for Disambiguating and Tagging People Names in Web Search.
  [L. Jiang, W. Shen, J. Wang, N. An]