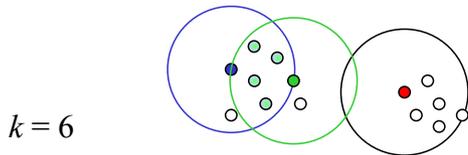


## 5.3 Dichtebasiertes Clustering

### *Shared Nearest Neighbor (SNN) Clustering*

- DBSCAN
  - Erkennt Cluster unterschiedlicher Form und Größe
  - Hat Probleme bei Clustern mit unterschiedlicher Dichte
- Verbesserung: anderer Ähnlichkeitsbegriff
  - Ähnlichkeit zwischen zwei Objekten, wenn sie beide sehr nahe zu einer Referenzmenge  $R$  sind
  - Ähnlichkeit wird durch die Referenzmenge  $R$  “bestätigt”
  - Ähnlichkeit z.B. durch die Anzahl gemeinsamer nächster Nachbarn definieren (d.h.  $R$  ist die Menge der nächsten Nachbarn)
  - Shared Nearest Neighbor (SNN) Ähnlichkeit:
    - $SNN_k\text{-similarity}(p,q) = |NN(p, k) \cap NN(q, k)|$
    - $NN(o, k) =$  Menge der  $k$ -nächsten Nachbarn von Objekt  $o$  (vgl. Kap. 2.2)



$$SNN_6\text{-similarity}(\bullet, \bullet) = 4$$

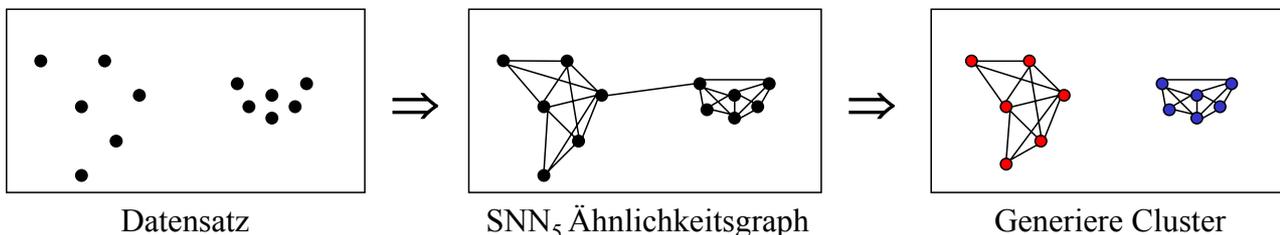
$$SNN_6\text{-similarity}(\bullet, \circ) = 0$$

222

## 5.3 Dichtebasiertes Clustering

### *Einfaches SNN-Clustering* [Jarvis, Patrick 73]:

1. Berechnung der Ähnlichkeitsmatrix und des Ähnlichkeitsgraphen
  - für alle Objekt-Paare  $p, q \in DB$ : berechne  $SNN_k\text{-similarity}(p, q)$
  - $SNN_k$ -Ähnlichkeitsgraph:
    - Knoten = Objekten
    - Kante zwischen jedem Objektpaar  $p, q$  mit Gewicht  $SNN_k\text{-similarity}(p, q)$
  - Keine Kanten mit Gewicht 0
2. Generiere Cluster
  - Lösche alle Kanten, deren Gewicht unterhalb eines Grenzwerts  $\tau$  liegen
  - Cluster = verbundenen Komponenten im resultierenden Graphen



223

## 5.3 Dichtebasiertes Clustering

### Problem:

- Threshold  $\tau$  schwer zu bestimmen
- kleine Variationen führen zu stark unterschiedlichen Ergebnissen

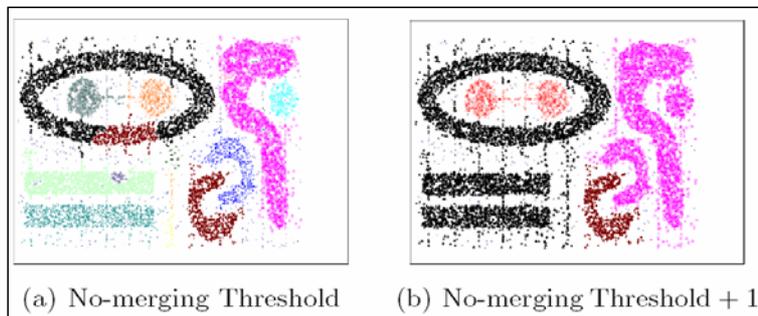


Bild aus:  
[Ertöz, Steinbach, Kumar 03]

### Lösung [Ertöz, Steinbach, Kumar 03]

- Kombiniere SNN-Ähnlichkeit mit dichtebasierten Konzepten
- SNN-Dichte:  
Anzahl der Punkte innerhalb eines spezifizierten Radius  $\varepsilon$  bzgl. SNN-Ähnlichkeit

$$\text{SNN}_k\text{-density}(p, \varepsilon) = |\{q \mid \text{SNN}_k\text{-similarity}(p, q) \geq \varepsilon\}|$$

224

## 5.3 Dichtebasiertes Clustering

### *SNN-Dichte*

- Beispiel:
  - 10 000 Daten (Bild (a))
  - $k = 50, \varepsilon = 20$
  - Bild (b): “Kernpunkte”  
alle Punkte mit SNN-Dichte  $\geq 34$
  - Bild (c): “Randpunkte”  
alle Punkte mit SNN-Dichte  $\geq 17$
  - Bild (d): “Rauschen”  
alle Punkte mit SNN-Dichte  $< 17$

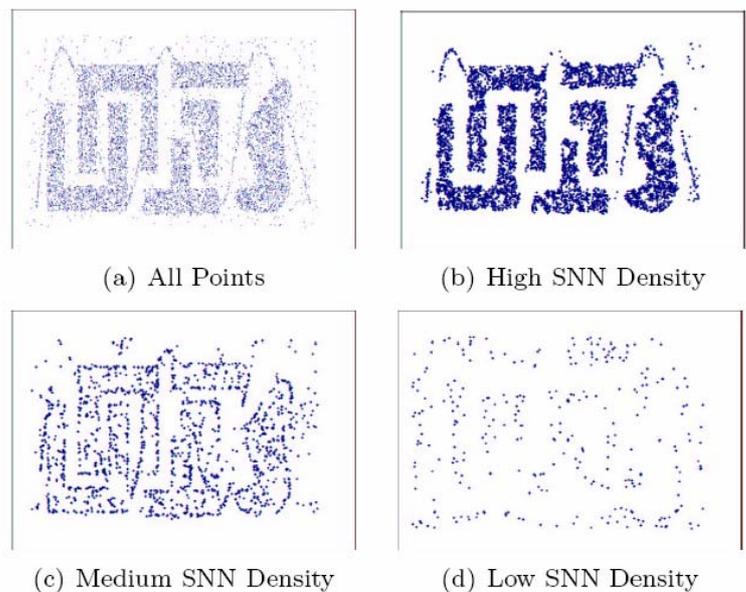


Bild aus: [Ertöz, Steinbach, Kumar 03]

- Analogie zu DBSCAN:  $\varepsilon = 20, \text{minPts} = 34$   
Kernpunkt  $p$ : mehr als  $\text{minPts}$  Punkte haben 20 oder mehr der 50 nächsten Nachbarn mit  $p$  gemeinsam

225

## 5.3 Dichtebasiertes Clustering

---

### *SNN-Clustering Algorithmus* [Ertöz, Steinbach, Kumar 03]

Eingabe:  $k$ ,  $\varepsilon$ ,  $minPts$

1. Berechne Ähnlichkeitsmatrix und -graph (siehe einfaches SNN-Clustering)
2. Berechne die  $SNN_k$ -Dichte für jeden Punkt bzgl.  $\varepsilon$
3. Bestimme Kernpunkte bzgl.  $minPts$  (alle Punkte mit einer SNN-Dichte  $\geq minPts$ )
4. Vereinige Kernpunkte  $p, q$ , wenn  $SNN_k\text{-similarity}(p, q) \geq \varepsilon$
5. Ordne Nicht-Kernpunkt  $p$  einem Cluster zu, wenn es ein Kernpunkt  $q$  gibt, mit  $SNN_k\text{-similarity}(q, p) \geq \varepsilon$
6. Alle anderen Nicht-Kernpunkte sind Rauschen

→ DBSCAN mit SNN-Ähnlichkeit

226

## 5.3 Dichtebasiertes Clustering

---

### *Diskussion*

- Unterschied zu DBSCAN
  - DBSCAN mit Euklidischer Distanz: nur Cluster, die dichter sind als der Grenzwert (spezifiziert durch  $minPts$  und  $\varepsilon$ )
  - SNN-Dichte eines Punktes  $p$ : Anzahl der Punkte, die mind.  $\varepsilon$  nächste Nachbarn mit  $p$  gemeinsam haben
    - ⇒ unabhängig von der eigentlichen Dichte
- Parametrisierung
  - Wahl von  $k$  ist kritisch:
    - Zu klein: auch relativ gleichverteilte Cluster werden wegen lokalen Variationen gesplittet ⇒ viele kleine Cluster
    - Zu groß: wenige große, gut-separierte Cluster
  - $minPts$ ,  $\varepsilon < k$

227

## 5.4 Hierarchische Verfahren

---

### Grundlagen

#### Ziel

Konstruktion einer Hierarchie von Clustern (*Dendrogramm*), so dass immer die Cluster mit minimaler Distanz verschmolzen werden

#### Dendrogramm

ein Baum, dessen Knoten jeweils ein Cluster repräsentieren, mit folgenden Eigenschaften:

- die Wurzel repräsentiert die ganze DB
- die Blätter repräsentieren einzelne Objekte
- ein innerer Knoten repräsentiert einen Cluster bestehend aus allen Objekten des darunter liegenden Teilbaums

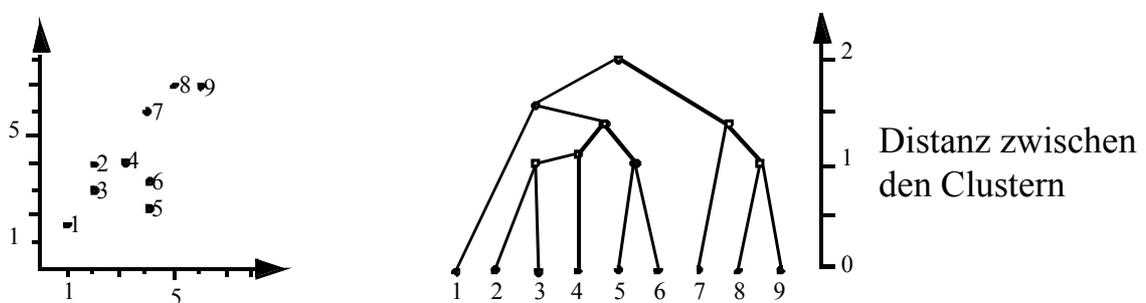
228

## 5.4 Hierarchische Verfahren

---

### Grundlagen

#### Beispiel eines Dendrogramms



#### Typen von hierarchischen Verfahren

- Bottom-Up Konstruktion des Dendrogramms (*agglomerative*)
- Top-Down Konstruktion des Dendrogramms (*divisive*)

229

### *Algorithmus Single-Link* [Jain & Dubes 1988]

#### Agglomeratives hierarchisches Clustering

1. Bilde initiale Cluster, die jeweils aus einem Objekt bestehen, und bestimme die Distanzen zwischen allen Paaren dieser Cluster.
2. Bilde einen neuen Cluster aus den zwei Clustern, welche die geringste Distanz zueinander haben.
3. Bestimme die Distanz zwischen dem neuen Cluster und allen anderen Clustern.
4. Wenn alle Objekte in einem einzigen Cluster befinden: Fertig, andernfalls wiederhole ab Schritt 2.

230

### *Distanzfunktionen für Cluster*

- Die Verfahren unterscheiden sich anhand ihrer Distanzfunktionen für Cluster
  - Single Link
  - Complete Link
  - Average Link
- Sei eine Distanzfunktion  $dist(x,y)$  für Paare von Objekten gegeben
- Seien  $X, Y$  Cluster, d.h. Mengen von Objekten.

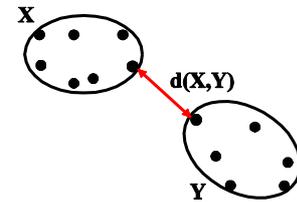
231

## 5.4 Hierarchische Verfahren

### Single-Link Distanz

- Definition:

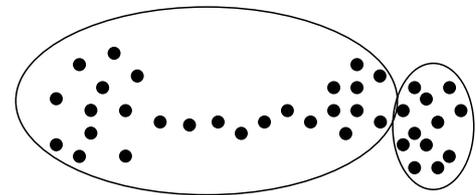
$$\text{distSL}(X, Y) = \min_{x \in X, y \in Y} \text{dist}(x, y)$$



Single-Link

- Eigenschaften:

- Effiziente Implementierung (z.B. SLINK):  $O(n^2)$
- Single-Link Effekt: „kettenförmige“ Cluster, Cluster werden durch wenige, kettenförmig verteilte Objekte vereinigt
  - Cluster mit starker Streuung
  - Cluster mit langgezogener Struktur



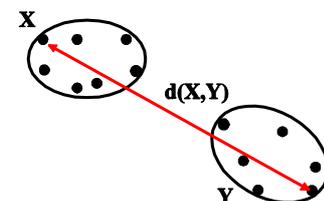
232

## 5.4 Hierarchische Verfahren

### Complete-Link Distanz

- Definition:

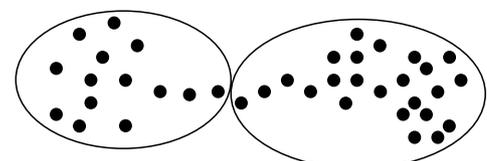
$$\text{distCL}(X, Y) = \max_{x \in X, y \in Y} \text{dist}(x, y)$$



Complete-Link

- Eigenschaften:

- Effiziente Implementierung (z.B. CLINK):  $O(n^2)$
- Complete-Link Effekt
  - Kleine, stark abgegrenzte Cluster
  - Gleichgroße, konvexe Cluster



233

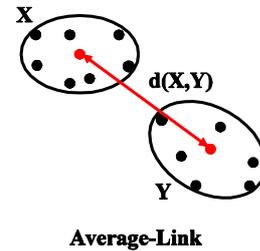
## 5.4 Hierarchische Verfahren

---

### *Average-Link Distanz*

- Definition:

$$distAL(X, Y) = \frac{1}{|X| \cdot |Y|} \cdot \sum_{x \in X, y \in Y} dist(x, y)$$



- Eigenschaften:
  - Keine effiziente Implementierung
  - Kompromiss zwischen Single- und Complete-Link Ansatz

234

## 5.4 Hierarchische Verfahren

---

### *Diskussion*

- + erfordert keine Kenntnis der Anzahl  $k$  der Cluster
- + findet nicht nur ein flaches Clustering, sondern eine ganze Hierarchie
- + ein einzelnes Clustering kann aus dem Dendrogramm gewonnen werden, z.B. mit Hilfe eines horizontalen Schnitts durch das Dendrogramm (erfordert aber wieder Anwendungswissen)
- Entscheidungen können nicht zurückgenommen werden
- Single-Link-Effekte, Complete-Link-Effekte
- Ineffizienz
  - Laufzeitkomplexität von mindestens  $O(n^2)$  für  $n$  Objekte

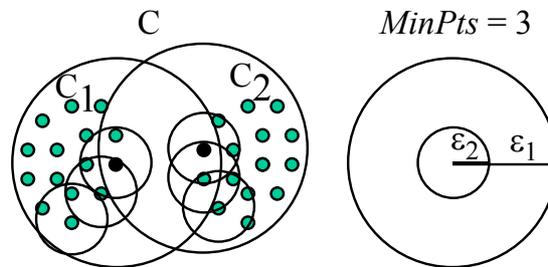
235

## 5.4 Hierarchische Verfahren

### *Dichtebasiertes hierarchisches Clustering*

[Ankerst, Breunig, Kriegel & Sander 1999]

- für einen konstanten *MinPts*-Wert sind dichte-basierte Cluster bzgl. eines kleineren  $\varepsilon$  vollständig in Clustern bzgl. eines größeren  $\varepsilon$  enthalten



- in einem DBSCAN-ähnlichen Durchlauf gleichzeitig das Clustering für verschiedene Dichte-Parameter bestimmen
  - zuerst die dichteren Teil-Cluster, dann den dünneren Rest-Cluster
- kein Dendrogramm, sondern eine auch noch bei sehr großen Datenmengen übersichtliche Darstellung der Cluster-Hierarchie

236

## 5.4 Hierarchische Verfahren

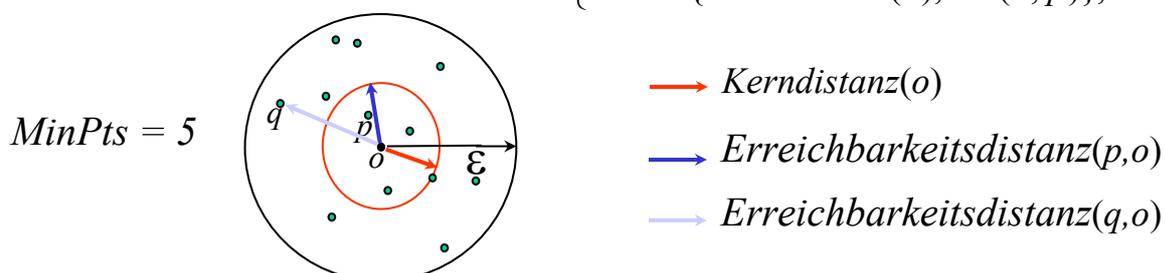
### *Grundbegriffe*

*Kerndistanz* eines Objekts  $p$  bzgl.  $\varepsilon$  und *MinPts*

$$\text{Kerndistanz}_{\varepsilon, \text{MinPts}}(o) = \begin{cases} \text{UNDEFINIERT, wenn } |RQ(o, \varepsilon)| < \text{MinPts} \\ \text{MinPtsDistanz}(o), \text{sonst} \end{cases}$$

*Erreichbarkeitsdistanz* eines Objekts  $p$  relativ zu einem Objekt  $o$

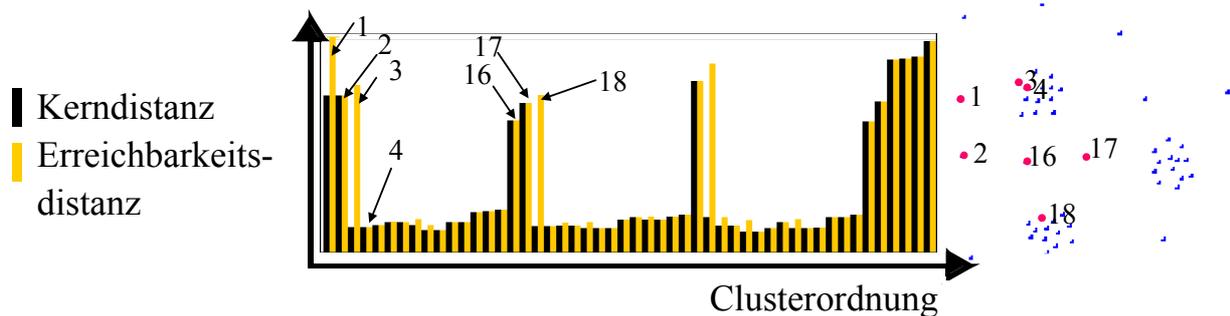
$$\text{Erreichbarkeitsdistanz}_{\varepsilon, \text{MinPts}}(p, o) = \begin{cases} \text{UNDEFINIERT, wenn } |RQ(o, \varepsilon)| < \text{MinPts} \\ \max\{\text{Kerndistanz}(o), \text{dist}(o, p)\}, \text{sonst} \end{cases}$$



237

### Clusterordnung

- OPTICS liefert nicht direkt ein (hierarchisches) Clustering, sondern eine „Clusterordnung“ bzgl.  $\varepsilon$  und  $MinPts$
- Clusterordnung bzgl.  $\varepsilon$  und  $MinPts$ 
  - beginnt mit einem beliebigen Objekt
  - als nächstes wird das Objekt besucht, das zur Menge der bisher besuchten Objekte die minimale Erreichbarkeitsdistanz besitzt



238

### Algorithmus OPTICS

- Datenstrukturen
  - SeedList
    - speichert Punkte mit „aktueller“ Erreichbarkeitsdistanz aufsteigend sortiert
  - ClusterOrder
    - resultierende Clusterordnung wird schrittweise aufgebaut

- Hauptschleife:

```
SeedList =  $\emptyset$ ;
```

```
WHILE es gibt noch unmarkierte Objekte in DB DO
```

```
  IF SeedList =  $\emptyset$ 
```

```
    THEN füge beliebiges noch unmarkiertes Objekt in ClusterOrder ein mit Erreichbarkeitsdistanz  $\infty$ ;
```

```
    ELSE füge erstes Objekt aus der SeedList mit aktueller Erreichbarkeitsdistanz in ClusterOrder ein;
```

```
    // sei obj das zuletzt in ClusterOrder eingefügte Objekt
```

```
    markiere obj als bearbeitet;
```

```
    FOR ALL neighbor  $\in$  RQ(obj,  $\varepsilon$ ) DO
```

```
      SeedList.update(neighbor, obj); // insert/update neighbor in SeedList mit referenzobjekt obj;
```

239

## 5.4 Hierarchische Verfahren

### *Algorithmus OPTICS*

- Einfügen/Updaten eines Objekts  $o$  in SeedList
  - Beachte: Für alle Objekte  $p$  in SeedList ist die „aktuelle“ Erreichbarkeitsdistanz  $p.rdist$  gespeichert.
  - SeedList ist nach  $p.rdist$  aufsteigend sortiert (als Heap organisiert)
  - Referenzobjekt:  $obj$

SeedList :: update( $o$ ,  $obj$ )

Berechne Erreichbarkeitsdistanz $_{\epsilon, MinPts}(o, obj) =: rdistneu\_o$ ;

**IF**  $o$  ist bereits in SeedList **THEN**

**IF**  $rdistneu\_o \leq o.rdist$  **THEN**

$o.rdist := rdistneu\_o$ ;

        verschiebe  $o$  in SeedList (nach vorne);      // aufsteigen im Heap

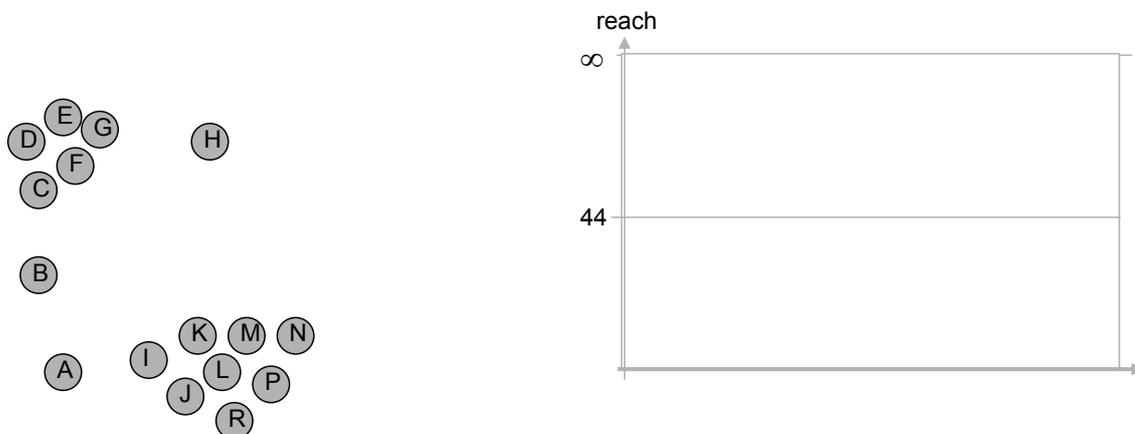
**ELSE**      //  $o$  ist noch nicht in SeedList

    füge  $o$  mit  $o.rdist := rdistneu\_o$  in SeedList ein;      // normales Einfügen in Heap

240

## 5.4 Hierarchische Verfahren

- Example Database (2-dimensional, 16 points)
- $\epsilon = 44$ ,  $MinPts = 3$

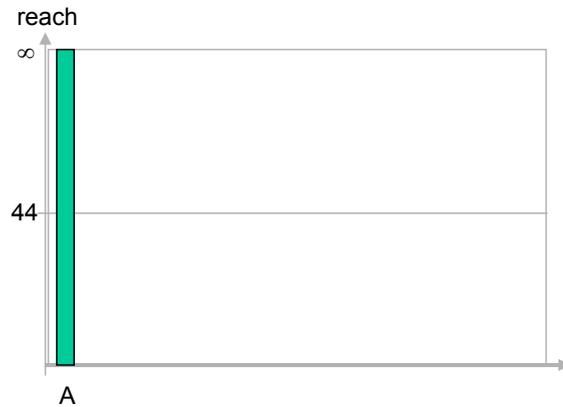
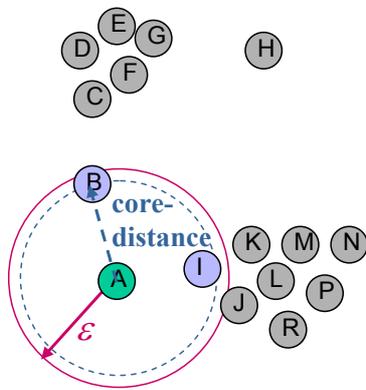


seed list:

241

## 5.4 Hierarchische Verfahren

- Example Database (2-dimensional, 16 points)
- $\epsilon = 44$ ,  $MinPts = 3$

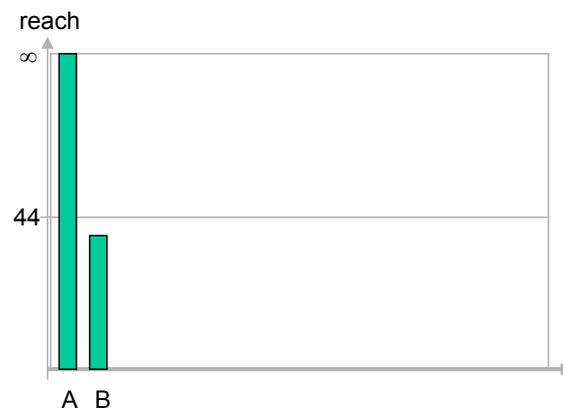
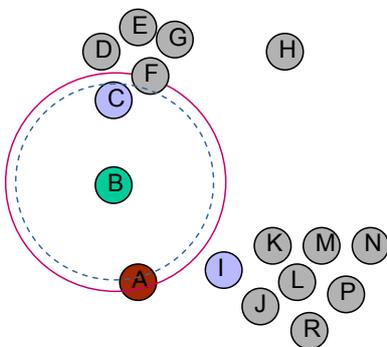


seed list: (B,40) (I, 40)

242

## 5.4 Hierarchische Verfahren

- Example Database (2-dimensional, 16 points)
- $\epsilon = 44$ ,  $MinPts = 3$

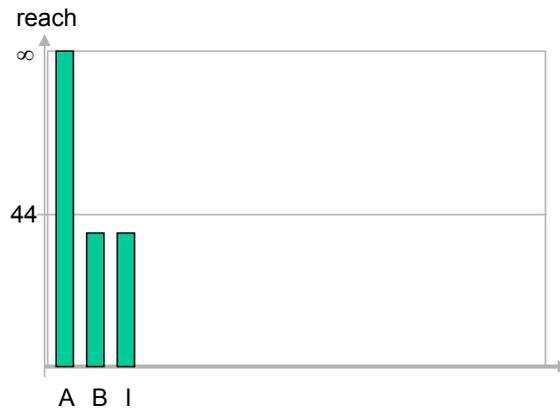
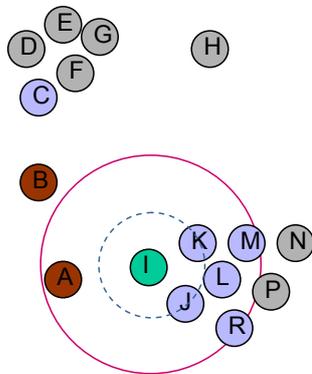


seed list: (I, 40) (C, 40)

243

## 5.4 Hierarchische Verfahren

- Example Database (2-dimensional, 16 points)
- $\epsilon = 44$ ,  $MinPts = 3$

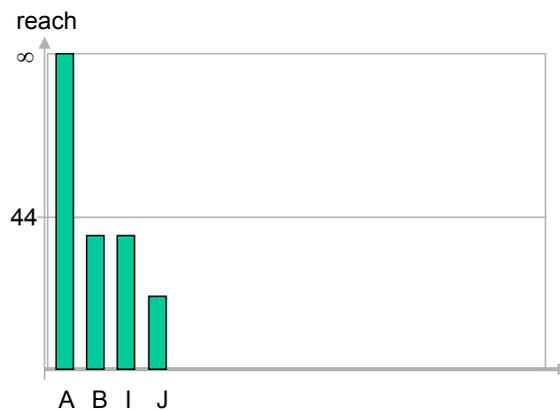
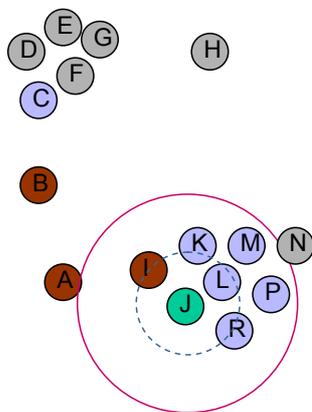


seed list: (J, 20) (K, 20) (L, 31) (C, 40) (M, 40) (R, 43)

244

## 5.4 Hierarchische Verfahren

- Example Database (2-dimensional, 16 points)
- $\epsilon = 44$ ,  $MinPts = 3$

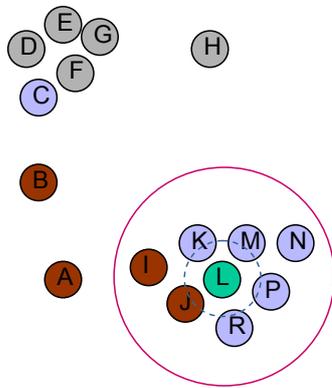


seed list: (L, 19) (K, 20) (R, 21) (M, 30) (P, 31) (C, 40)

245

## 5.4 Hierarchische Verfahren

- Example Database (2-dimensional, 16 points)
- $\epsilon = 44$ ,  $MinPts = 3$

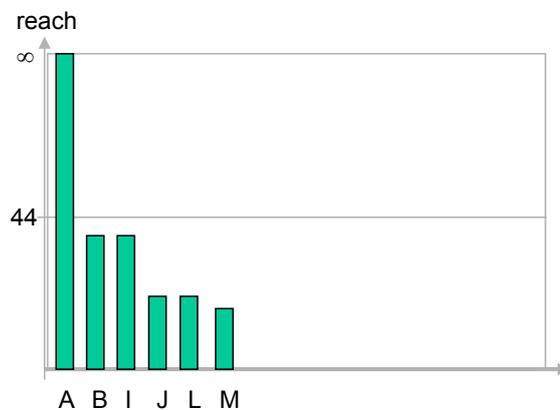
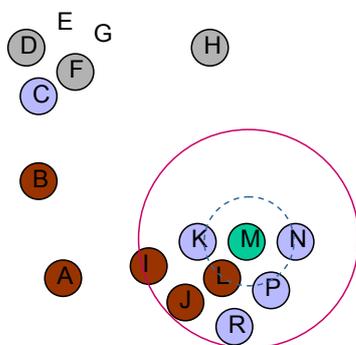


seed list: (M, 18) (K, 18) (R, 20) (P, 21) (N, 35) (C, 40)

246

## 5.4 Hierarchische Verfahren

- Example Database (2-dimensional, 16 points)
- $\epsilon = 44$ ,  $MinPts = 3$

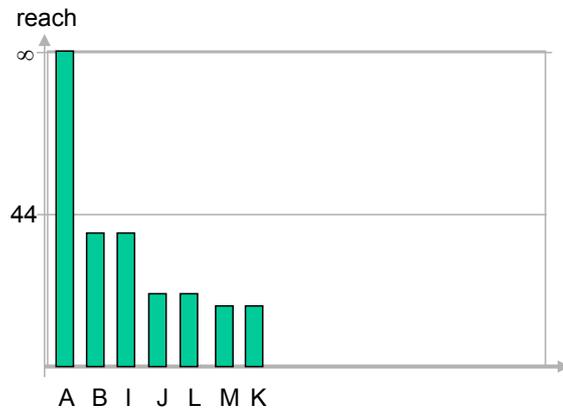
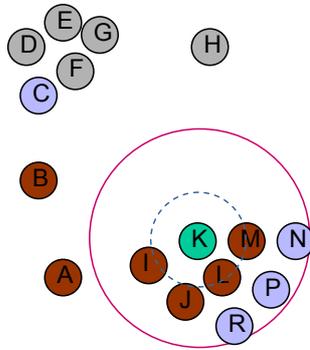


seed list: (K, 18) (N, 19) (R, 20) (P, 21) (C, 40)

247

## 5.4 Hierarchische Verfahren

- Example Database (2-dimensional, 16 points)
- $\epsilon = 44$ ,  $MinPts = 3$

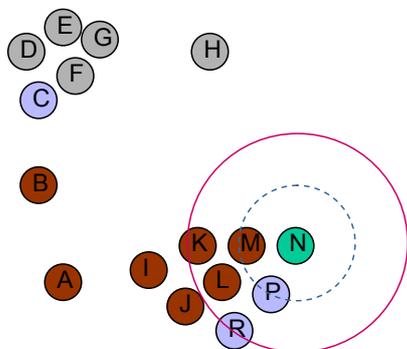


seed list: (N, 19) (R, 20) (P, 21) (C, 40)

248

## 5.4 Hierarchische Verfahren

- Example Database (2-dimensional, 16 points)
- $\epsilon = 44$ ,  $MinPts = 3$

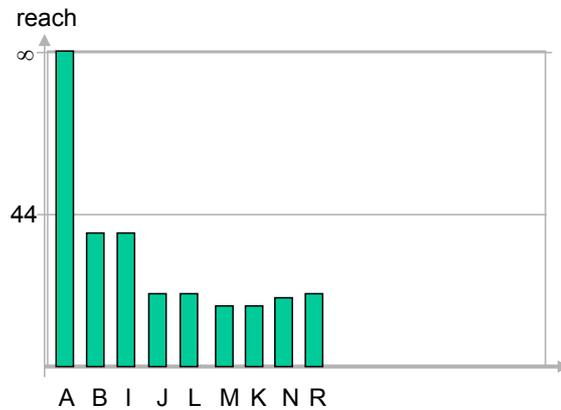
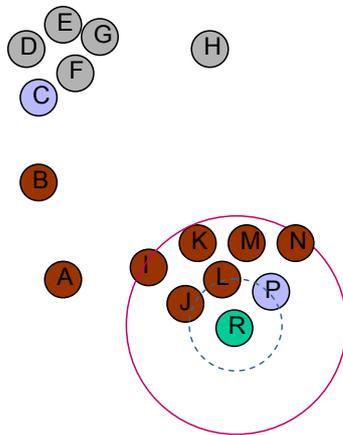


seed list: (R, 20) (P, 21) (C, 40)

249

## 5.4 Hierarchische Verfahren

- Example Database (2-dimensional, 16 points)
- $\epsilon = 44$ ,  $MinPts = 3$

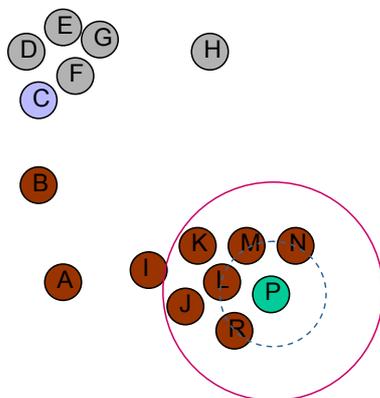


seed list: (P, 21) (C, 40)

250

## 5.4 Hierarchische Verfahren

- Example Database (2-dimensional, 16 points)
- $\epsilon = 44$ ,  $MinPts = 3$

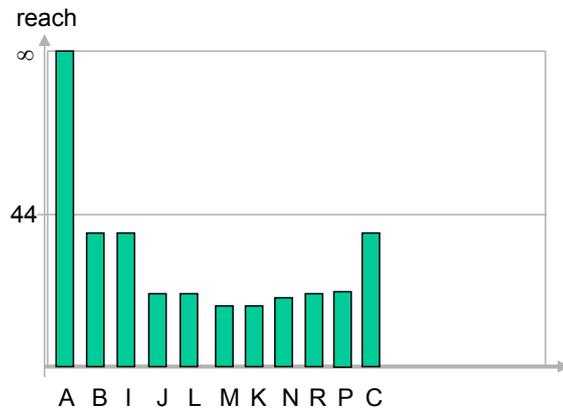
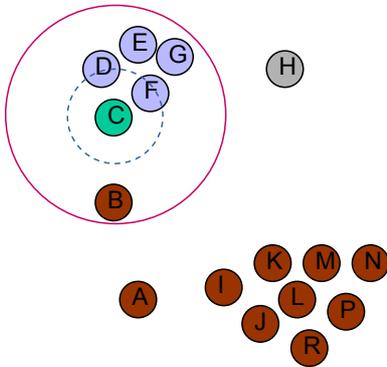


seed list: (C, 40)

251

## 5.4 Hierarchische Verfahren

- Example Database (2-dimensional, 16 points)
- $\epsilon = 44$ ,  $MinPts = 3$

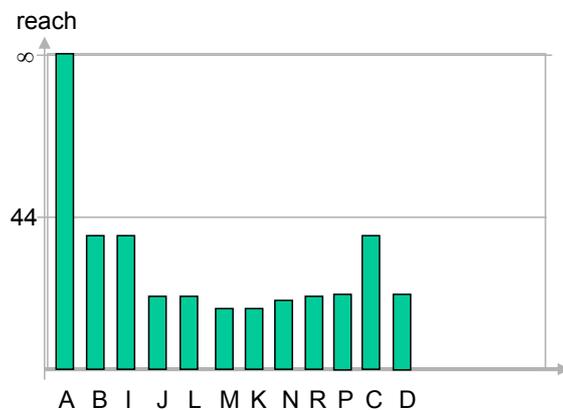
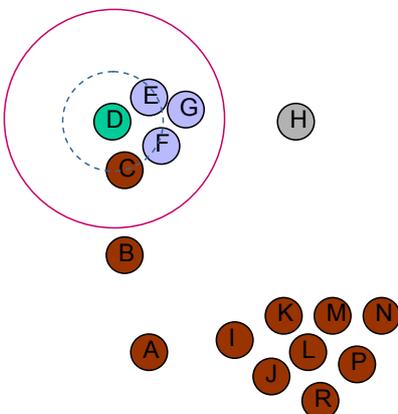


seed list: (D, 22) (F, 22) (E, 30) (G, 35)

252

## 5.4 Hierarchische Verfahren

- Example Database (2-dimensional, 16 points)
- $\epsilon = 44$ ,  $MinPts = 3$

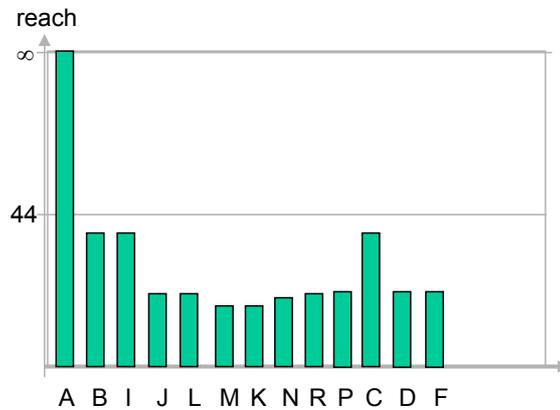
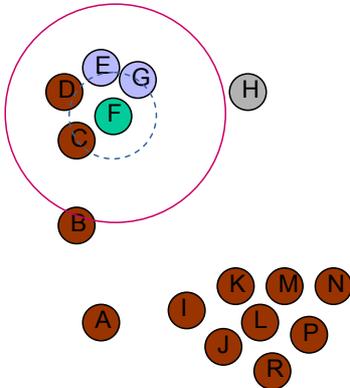


seed list: (F, 22) (E, 22) (G, 32)

253

## 5.4 Hierarchische Verfahren

- Example Database (2-dimensional, 16 points)
- $\epsilon = 44$ ,  $MinPts = 3$

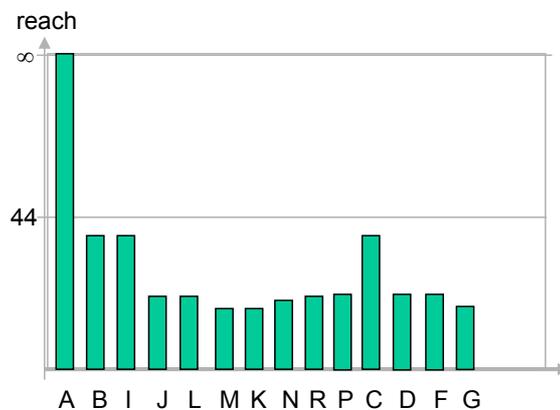
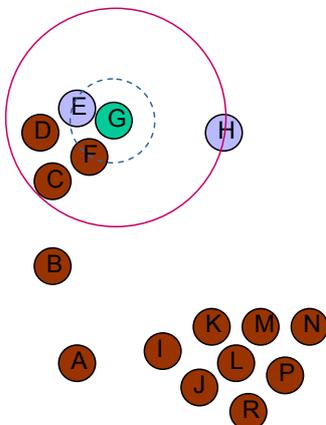


seed list: (G, 17) (E, 22)

254

## 5.4 Hierarchische Verfahren

- Example Database (2-dimensional, 16 points)
- $\epsilon = 44$ ,  $MinPts = 3$

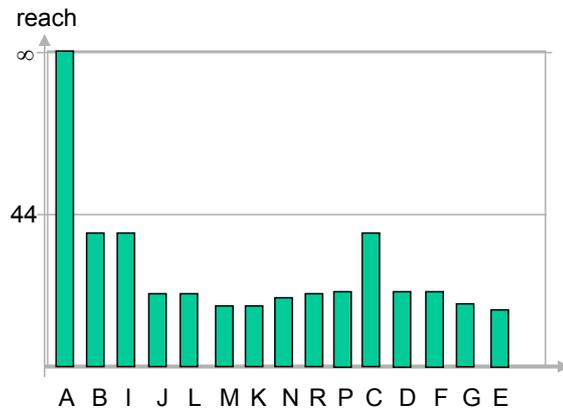
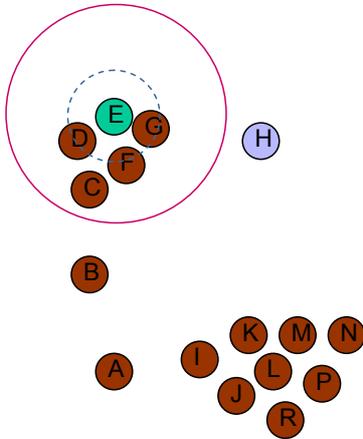


seed list: (E, 15) (H, 43)

255

## 5.4 Hierarchische Verfahren

- Example Database (2-dimensional, 16 points)
- $\epsilon = 44$ ,  $MinPts = 3$

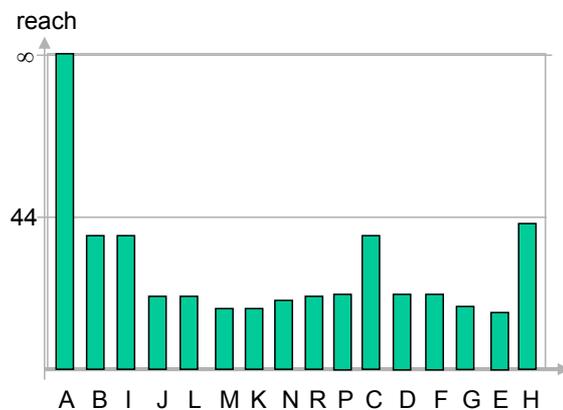
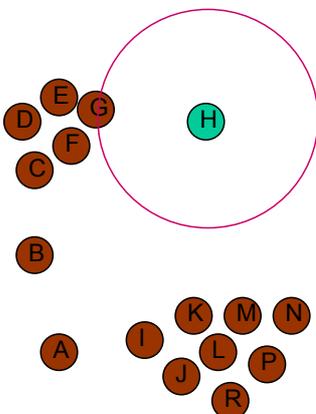


seed list: (H, 43)

256

## 5.4 Hierarchische Verfahren

- Example Database (2-dimensional, 16 points)
- $\epsilon = 44$ ,  $MinPts = 3$

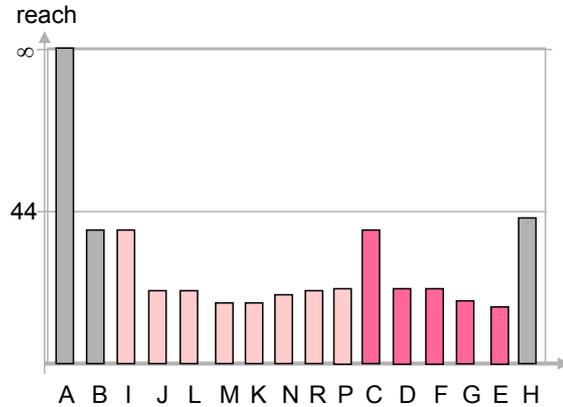
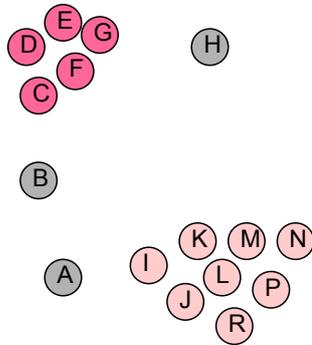


seed list: -

257

## 5.4 Hierarchische Verfahren

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$ ,  $MinPts = 3$

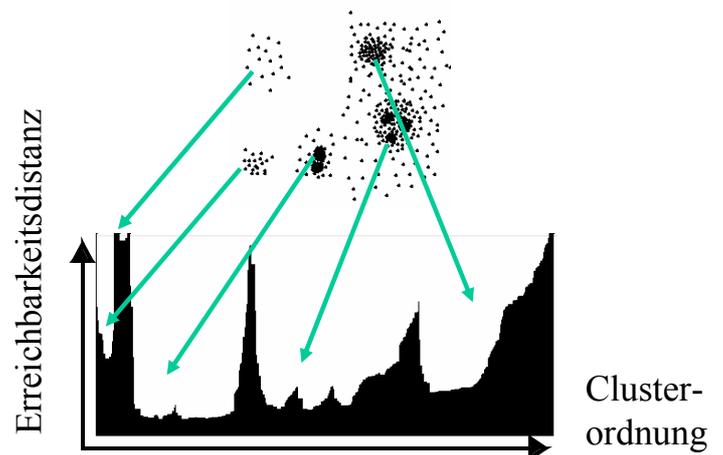
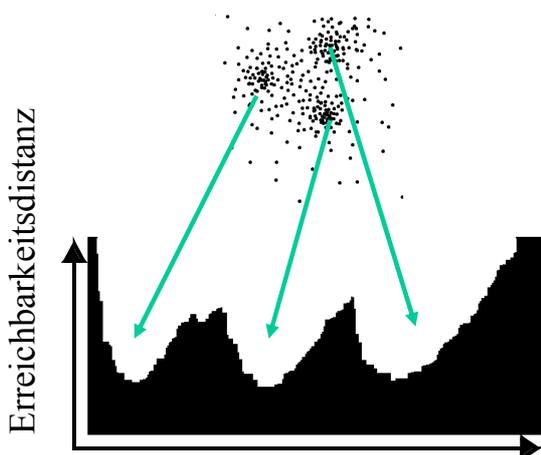


258

## 5.4 Hierarchische Verfahren

### *Erreichbarkeits-Diagramm*

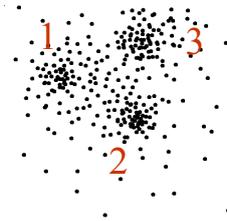
- Zeigt die Erreichbarkeitsdistanzen (bzgl.  $\varepsilon$  und  $MinPts$ ) der Objekte als senkrechte, nebeneinanderliegende Balken
- in der durch die Clusterordnung der Objekte gegebenen Reihenfolge



259

## 5.4 Hierarchische Verfahren

### *Parameter-Sensitivität*



$MinPts = 10, \epsilon = 10$



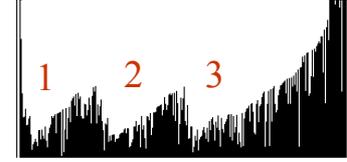
optimale Parameter

$MinPts = 10, \epsilon = 5$



kleineres  $\epsilon$

$MinPts = 2, \epsilon = 10$



kleineres  $MinPts$



Clusterordnung ist robust gegenüber den Parameterwerten  
gute Resultate wenn Parameterwerte „groß genug“

260

## 5.4 Hierarchische Verfahren

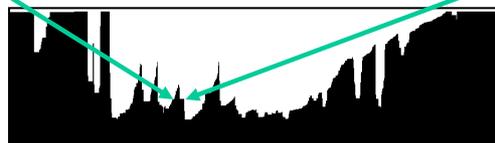
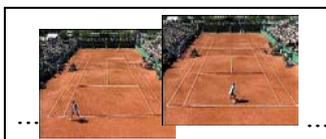
### *Heuristische Parameter-Bestimmung*

$\epsilon$

- wähle größte  $MinPts$ -Distanz aus einem Sample oder
- berechne durchschnittliche  $MinPts$ -Distanz für gleichverteilte Daten

$MinPts$

- glätte Erreichbarkeits-Diagramm
- vermeide “single-” bzw. “ $MinPts$ -link” Effekt

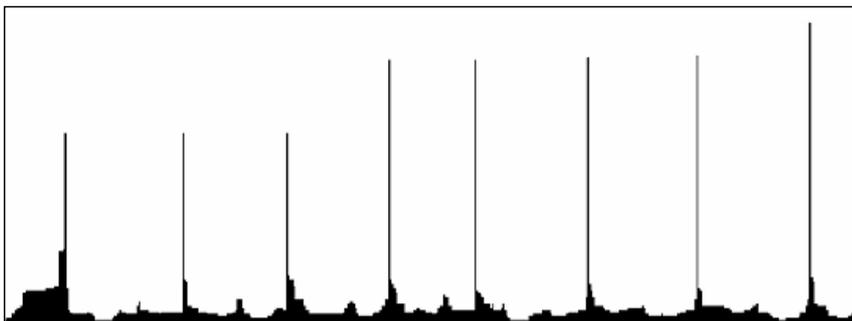


261

### *Manuelle Analyse der Cluster*

#### Mit Erreichbarkeits-Diagramm

- gibt es Cluster?
- wieviele Cluster?
- sind die Cluster hierarchisch geschachtelt?
- wie groß sind die Cluster?

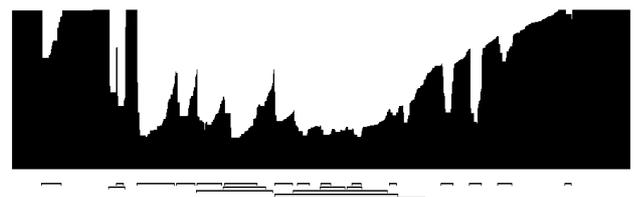
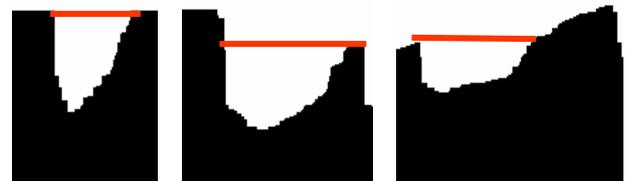


Erreichbarkeits-Diagramm

### *Automatisches Entdecken von Clustern*

$\xi$ -Cluster [Ankerst, Breunig, Kriegel, Sander 99]

- Teilsequenz der Clusterordnung
- beginnt in einem Gebiet  $\xi$ -steil *abfallender* Erreichbarkeitsdistanzen
- endet in einem Gebiet  $\xi$ -steil *steigender* Erreichbarkeitsdistanzen bei etwa demselben absoluten Wert
- enthält mindestens *MinPts* Punkte



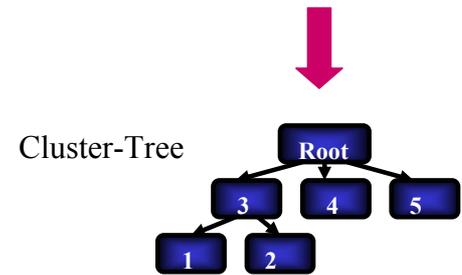
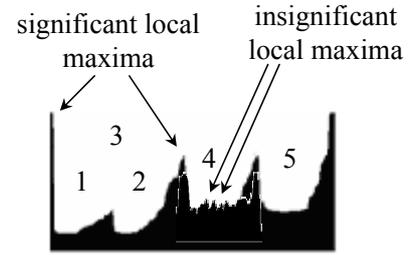
## 5.4 Hierarchische Verfahren

### Automatisches Entdecken von Clustern

#### ClusterTree [Sander, Qin, Lu, Niu, Kovarsky 02]

• Cluster sind geteilt durch “signifikante” lokale Maxima:

- *Minimale Anzahl der Punkte zwischen 2 Maxima*  
Richtwert: 0.5 % der Datenbank
- *Verhältnis zwischen Erreichbarkeitsdistanz des lokalen Maximas und der durchschnittlichen Erreichbarkeitsdistanzen links und rechts des Maximas in der Clusterordnung*  
Richtwert: 0.75



264

## 5.4 Hierarchische Verfahren

### Automatisches Entdecken von Clustern

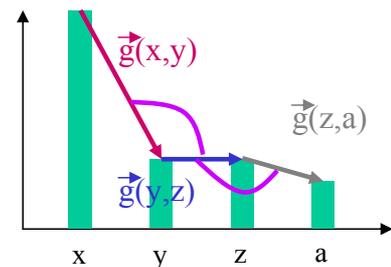
#### GradientClustering

[Brecheisen, Kriegel, Kröger, Pfeifle 04]

- Teilsequenzen der Clusterordnung
- beginnt/endet mit “Inflexion Point”
  - Gradientvektor
  - Inflexion Index
  - Inflexion Point, wenn  $II(o) > t$
  - Gradient Determinante
  - Fallunterscheidung:

$II(o) > t$  and  $GD(o) > 0$   
 $\Rightarrow$  Startpunkt oder erster Punkt  
 außerhalb des Clusters

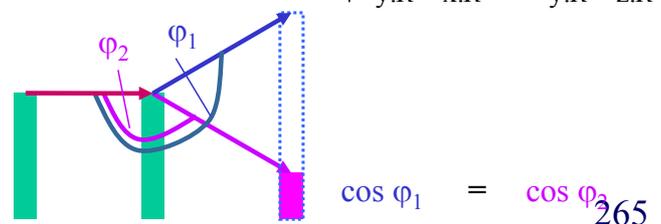
$II(o) > t$  and  $GD(o) < 0$   
 $\Rightarrow$  Endpunkt oder zweiter Punkt  
 innerhalb des Clusters



$$\vec{g}(x,y) = \begin{pmatrix} \text{width} \\ y.R - x.R \end{pmatrix}$$

$$II(y) = \cos \varphi(\vec{g}(x,y), \vec{g}(y,z))$$

$$GD(\vec{g}(x,y), \vec{g}(y,z)) = \begin{vmatrix} \text{width} & -\text{width} \\ y.R - x.R & y.R - z.R \end{vmatrix}$$



265

### Übersicht

#### 5.5.1 Clustering mit kategorischer Attribute

Grundlagen, Algorithmus  $k$ -modes

#### 5.5.2 Verallgemeinertes dichtebasiertes Clustering

Grundlagen, Algorithmus GDBSCAN, Beispiele

266

---

### 5.5.1 Clustering mit kategorischen Attributen

---

#### *Grundlagen* [Huang 1997]

- $k$ -medoid-Algorithmus wesentlich langsamer als  $k$ -means- Algorithmus
- $k$ -means-Verfahren nicht direkt für kategorische Attribute anwendbar



gesucht ist ein Analogon zum Centroid eines Clusters

- Numerische Attribute

Centroid  $\bar{x}$  einer Menge  $C$  von Objekten minimiert  $TD(C, \bar{x}) = \sum_{p \in C} dist(p, \bar{x})$

- Kategorische Attribute

Mode  $m$  einer Menge  $C$  von Objekten minimiert  $TD(C, m) = \sum_{p \in C} dist(p, m)$

(  $m$  ist nicht unbedingt ein Element der Menge  $C$  )

$m = (m_1, \dots, m_d)$ ,  $dist$  eine Distanzfunktion für kategorische Attribute, z.B.

$$dist(x, y) = \sum_{i=1}^d \delta(x_i, y_i) \text{ mit } \delta(x_i, y_i) = \begin{cases} 0 & \text{falls } x_i = y_i \\ 1 & \text{sonst} \end{cases}$$

267

## 5.5.1 Clustering mit kategorischen Attributen

---

### *Bestimmung des Modes*

- Die Funktion  $TD(C, m) = \sum_{p \in C} dist(p, m)$  wird minimiert genau dann, wenn für  $m = (m_1, \dots, m_d)$  und für alle Attribute  $A_i, i = 1, \dots, d$ , gilt:
  - es gibt in  $A_i$  keinen häufigeren Attributwert als  $m_i$
- Der Mode einer Menge von Objekten ist nicht eindeutig bestimmt.
- Beispiel
  - Objektmenge  $\{(a, b), (a, c), (c, b), (b, c)\}$
  - (a, b) ist ein Mode
  - (a, c) ist ein Mode

268

## 5.5.1 Clustering mit kategorischen Attributen

---

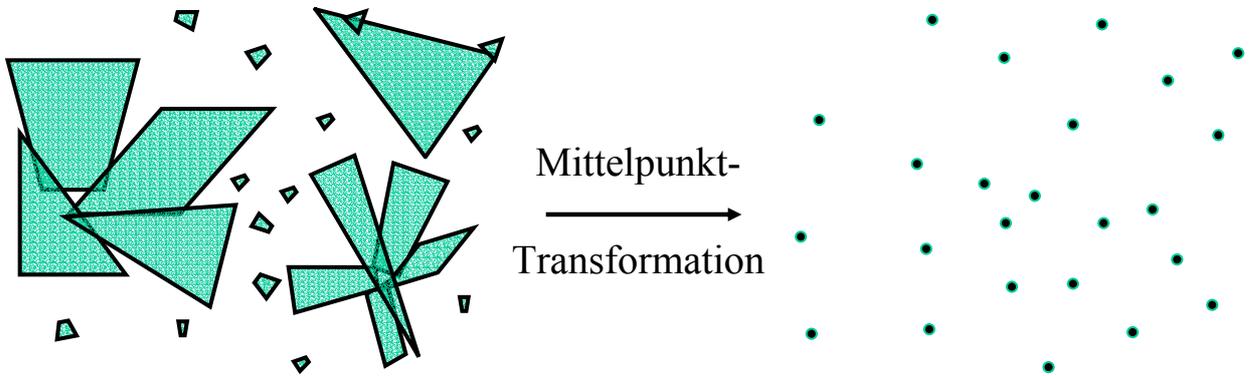
### *Algorithmus k-modes*

- Initialisierung
  - keine zufällige Partitionierung
  - sondern  $k$  Objekte aus der Datenmenge als initiale *Modes*
- Cluster-Repräsentanten
  - Mode anstelle des Centroids
- Distanzfunktion
  - anstelle der quadrierten euklidischen Distanz
  - Distanzfunktion für Datensätze mit kategorischen Attributen

269

## 5.5.2 Verallgemeinertes dichtebasiertes Clustering

### Clustering ausgedehnter Objekte



Berücksichtigung der Fläche und nicht-räumlicher Attribute  
natürlicher Begriff der Verbundenheit

270

## 5.5.2 Verallgemeinertes dichtebasiertes Clustering

### Algorithmus GDBSCAN [Sander, Ester, Kriegel & Xu 1998]

Grundidee für dichte-basierte Cluster :

“ $\varepsilon$ -Nachbarschaft enthält mindestens **MinPts** Punkte”

“Distanz  $\leq \varepsilon$ ”

$NPred(o,p)$   
reflexiv, symmetrisch  
für Paare von Objekten

**Verallgemeinerte Nachbarschaft**  
 $N_{NPred}(o) = \{p \mid NPred(o, p)\}$

“ $|N_\varepsilon| \geq MinPts$ ”

$MinWeight(N)$   
beliebiges Prädikat für  
Mengen von Objekten

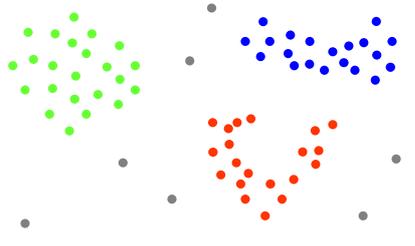
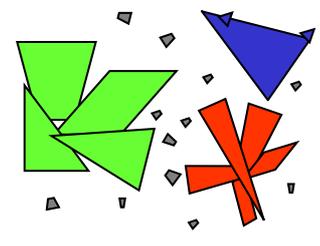
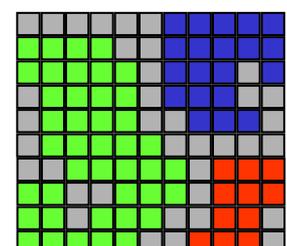
**Verallgemeinerte minimale Kardinalität**  
 $MinWeight(N_{NPred}(o))$

“ $NPred$ -Nachbarschaft hat mindestens das “Gewicht”  $MinWeight$ ”

271

## 5.5.2 Verallgemeinertes dichte-basiertes Clustering

### Beispiele

			
<i>NPred</i>	$\text{dist}(p,q) \leq \epsilon$	$\text{intersect}(p,q)$	Nachbarzelle und ähnliche Farbe
<i>MinWeight</i>	$\text{cardinality}(\dots) \geq \text{MinPoints}$	Summe der Flächen $\geq$ 5 % der Gesamtfläche	true

272

## 5.5.2 Verallgemeinertes dichte-basiertes Clustering

### Algorithmus GDBSCAN

- dasselbe algorithmische Schema wie DBSCAN
- anstelle einer  $\text{RQ}(o,\epsilon)$ -Anfrage eine  $N_{NPred}$ -Anfrage
- anstelle der Bedingung  $|\text{RQ}(o,\epsilon)| \geq \text{MinPts}$  das *MinWeight*-Prädikat auswerten
- Laufzeitkomplexität  $O(n \log n)$  bei geeigneter Unterstützung der  $N_{NPred}$ -Anfrage
- Beliebige Nachbarschaftsprädikate denkbar

273