

Grundlagen

Idee

- Cluster als Gebiete im d -dimensionalen Raum, in denen die Objekte dicht beieinander liegen
- getrennt durch Gebiete, in denen die Objekte weniger dicht liegen

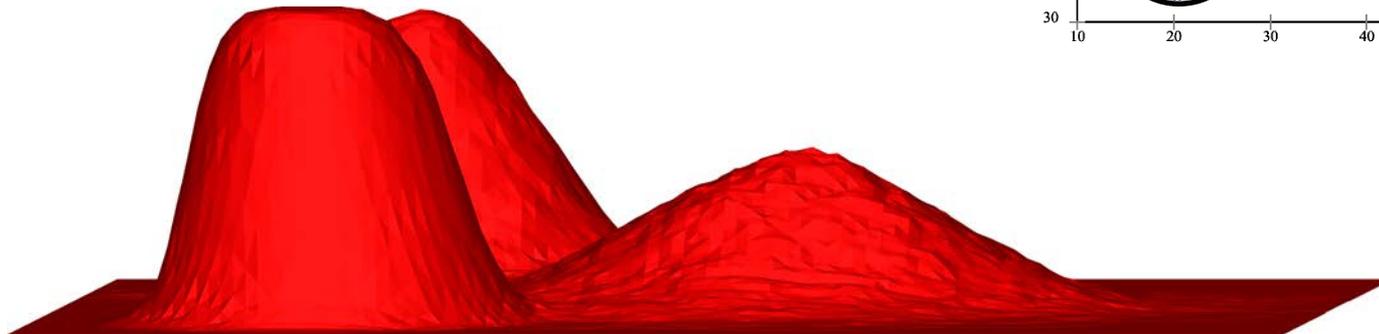
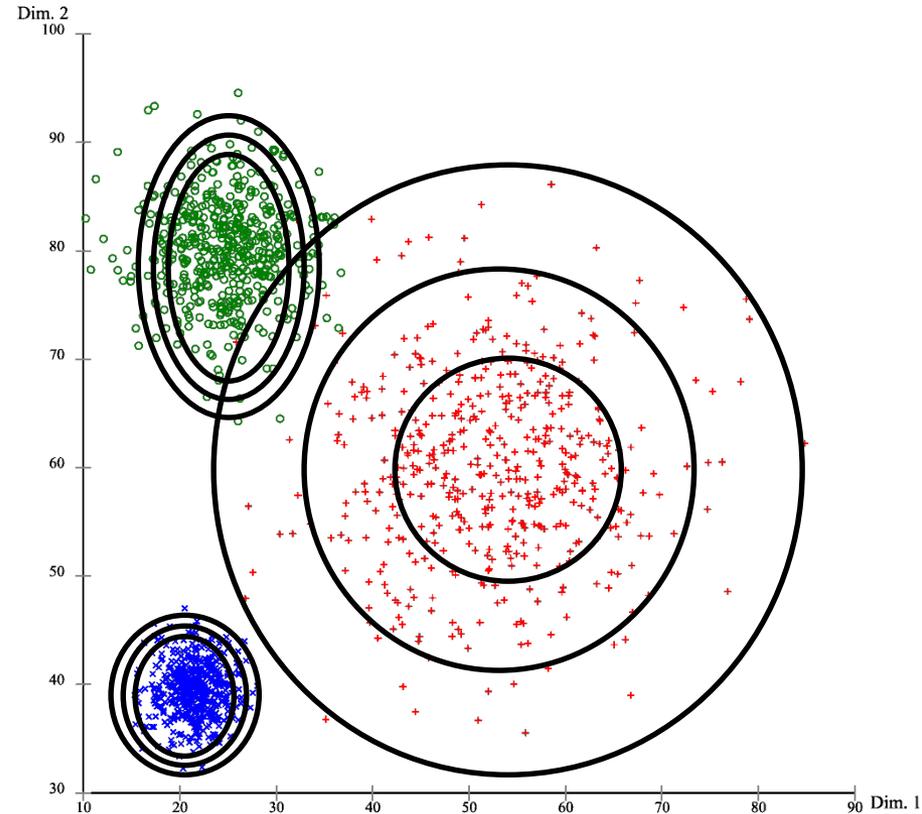
Zentrale Annahmen

- für jedes Objekt eines Clusters überschreitet die lokale Punktdichte einen gegebenen Grenzwert
- die Menge von Objekten, die den Cluster ausmacht, ist räumlich zusammenhängend

Intuition

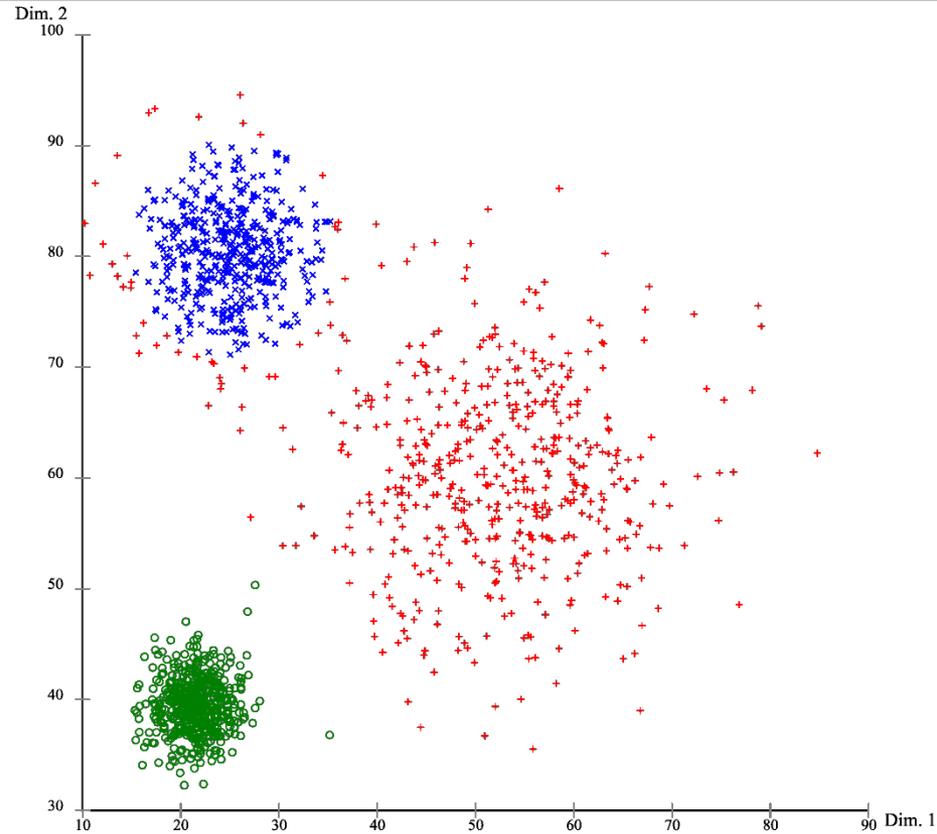
(nach: [Kriegel, Kröger, Sander, Zimek 2011])

- Partitionierend: approximiere unterschiedliche Dichtefunktionen
 - „parametrisch“
- Dichte-basiert: unabhängig von zugrundeliegender Dichtefunktion: selektiere Bereiche, die mit Mindest-Dichte verbunden sind
 - „nicht-parametrisch“



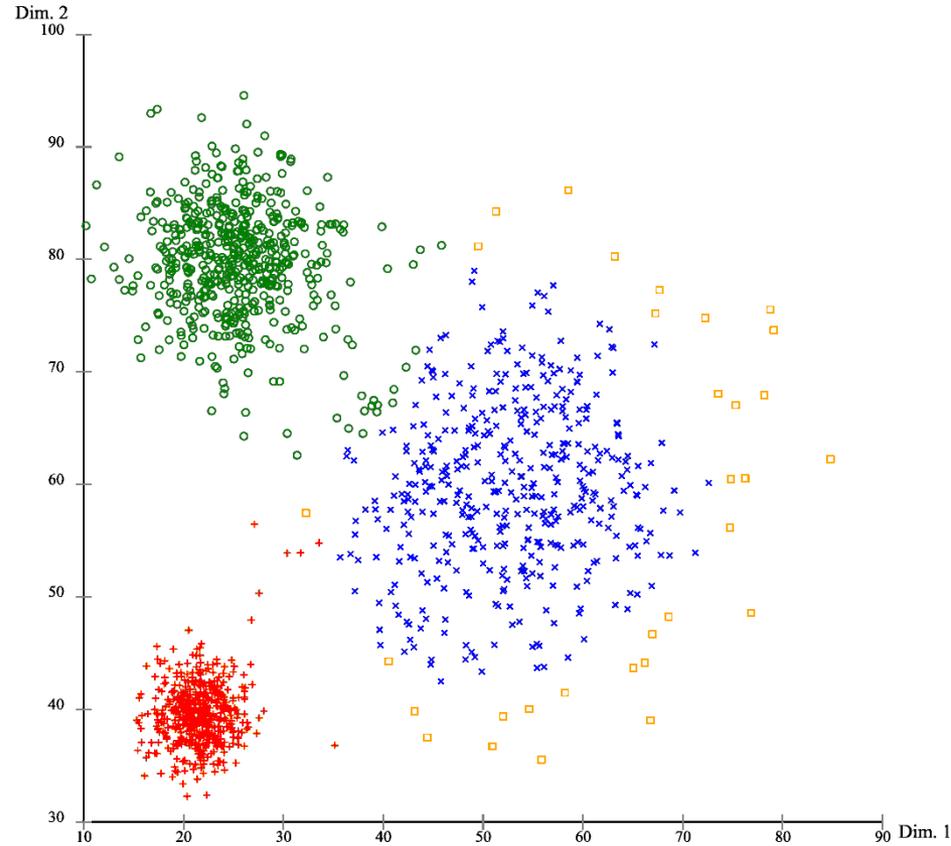
Intuition

hohes Dichte-Niveau: Cluster
geringer Dichte gilt als Noise



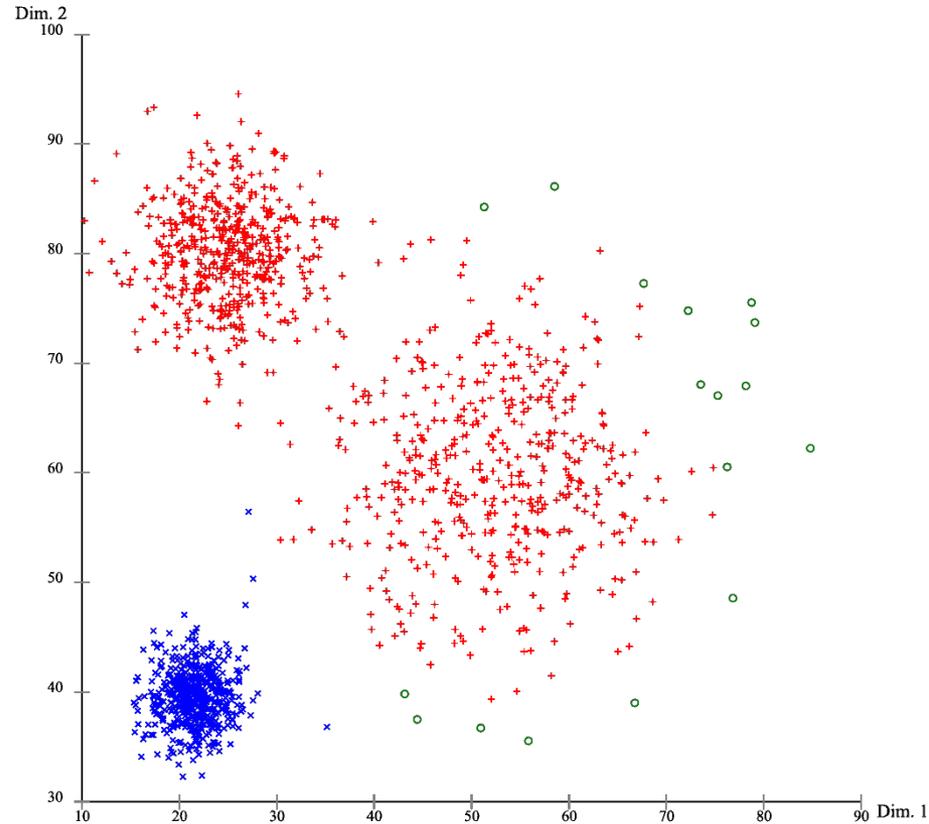
Intuition

mittleres Dichte-Niveau:
Identifikation von drei Clustern,
einige Punkte in den Ausläufern
der Verteilungen werden zu
Noise



Intuition

geringes Dichte-Niveau: zwei
Cluster verschmelzen



Intuition

Partitionierend: Optimierungsproblem

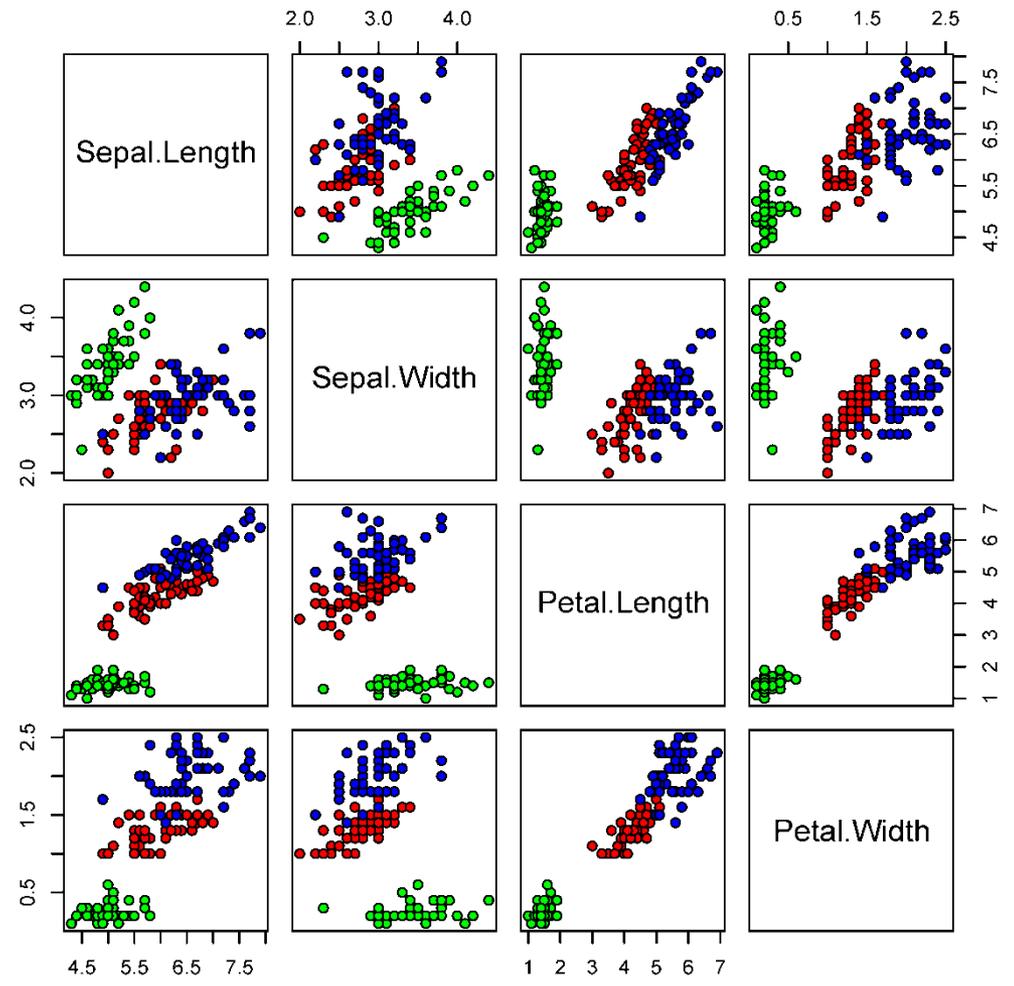
Dichte-basiert: „natürliche“ Strukturen (oft: geographisch/
topographisch, oder biologisch)

Beispiel: Höhenprofil (Vulkan Mt. Eden, Auckland)



Intuition

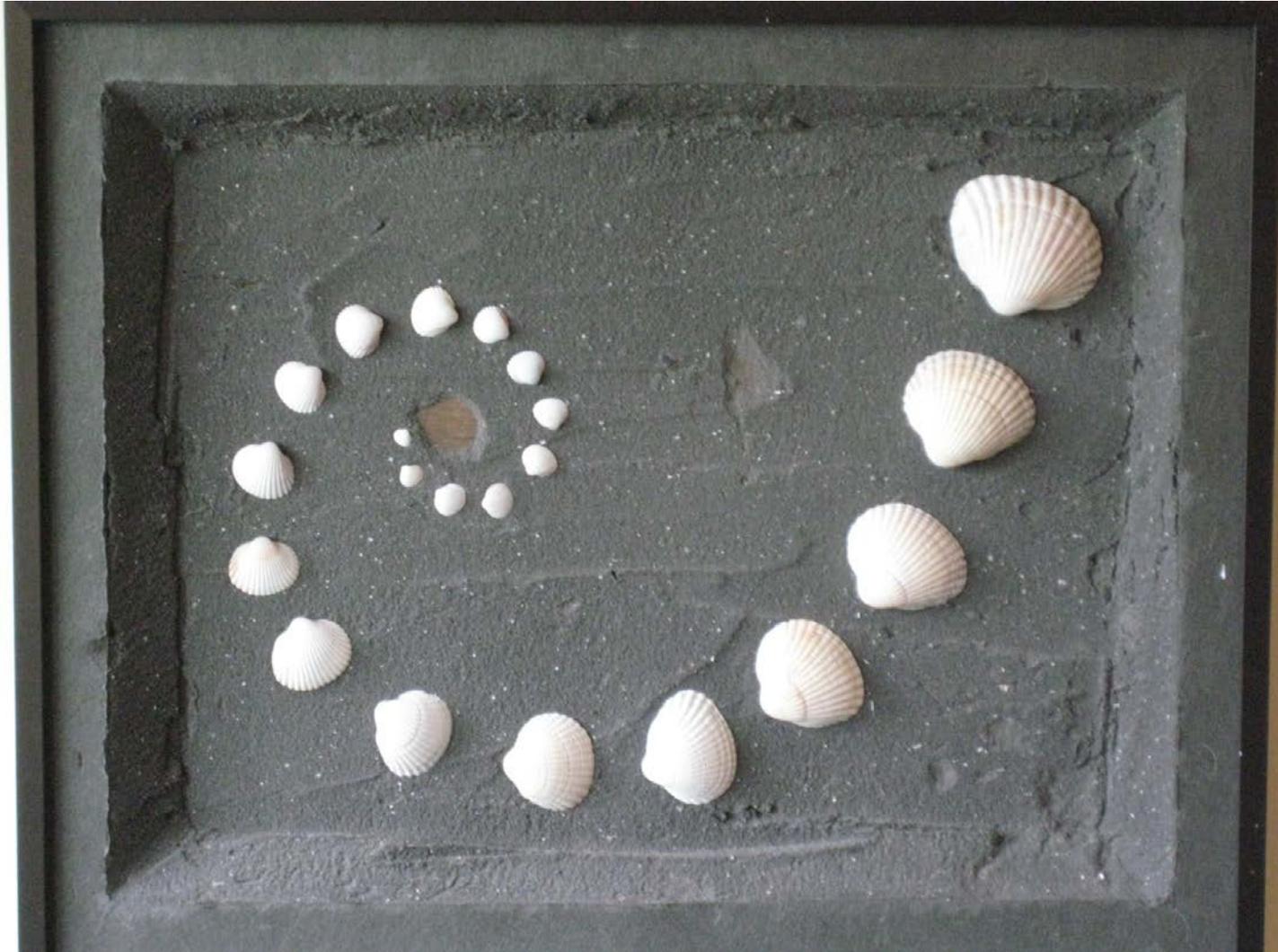
Beispiel: Iris-Arten – innerhalb einer Art gewisse Bandbreite, aber keine Sprünge



Intuition

Beispiel:

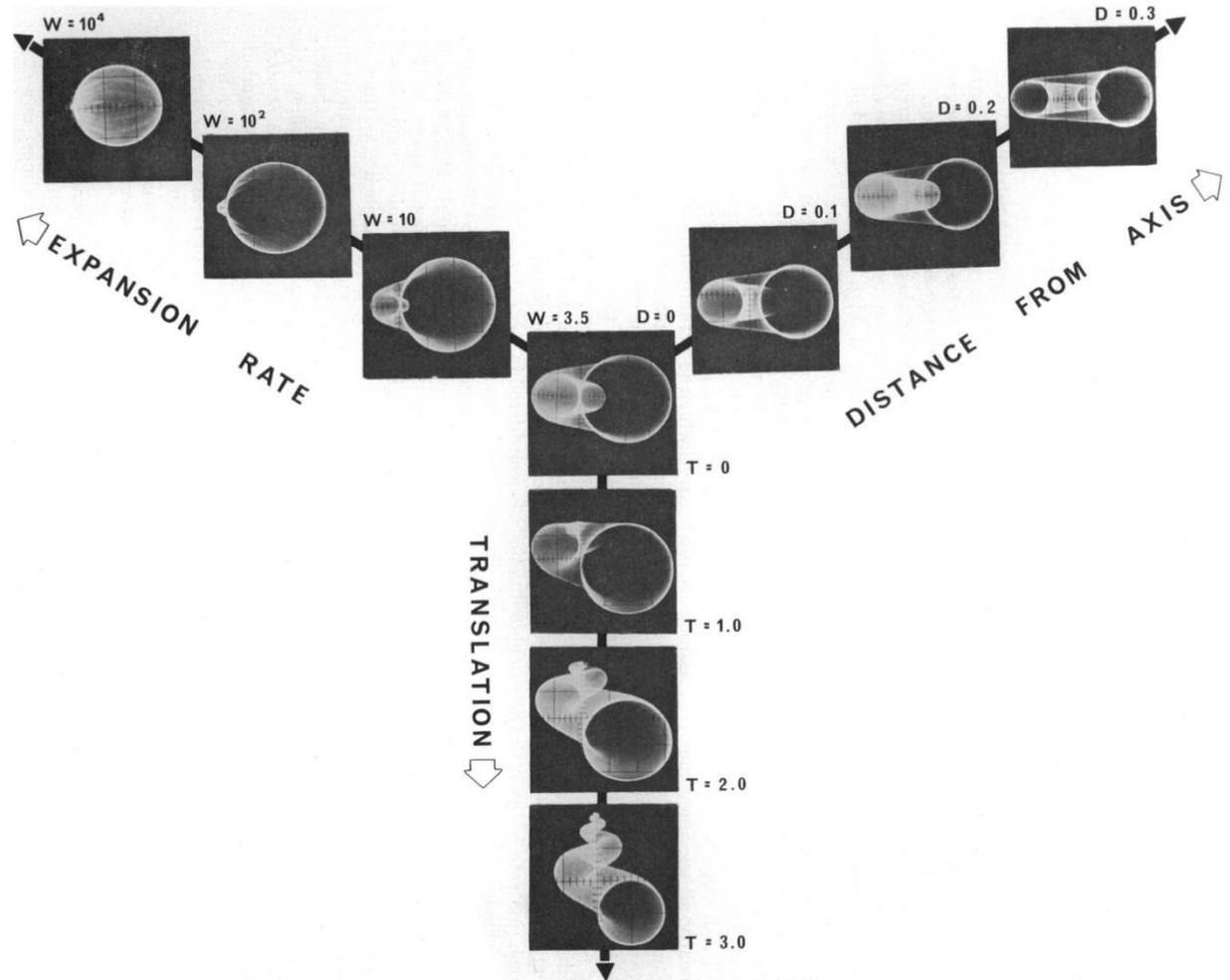
Wachstum –
ähnliche Form
aber
unterschiedliche
Größe innerhalb
einer Art



Intuition

Beispiel:
 Formparameter von
 Schnecken

D. M. Raup. Geometric analysis of shell coiling: General problems. Journal of Paleontology, 40(5):1178–1190, 1966

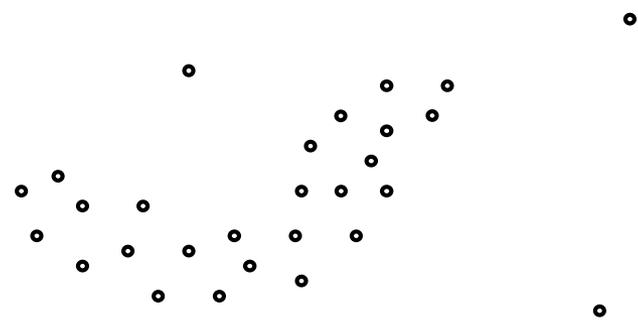


TEXT-FIG. 2—Computer produced shell forms illustrating the morphological effects of change in whorl expansion rate (W), translation rate (T), and the distance from the generating curve to the axis (D),

Idee zur effizienten Suche

Parameter $\varepsilon \in \mathbb{R}$ und $MinPts \in \mathbb{N}$ spezifizieren Dichtegrenzwert

ε $MinPts = 4$

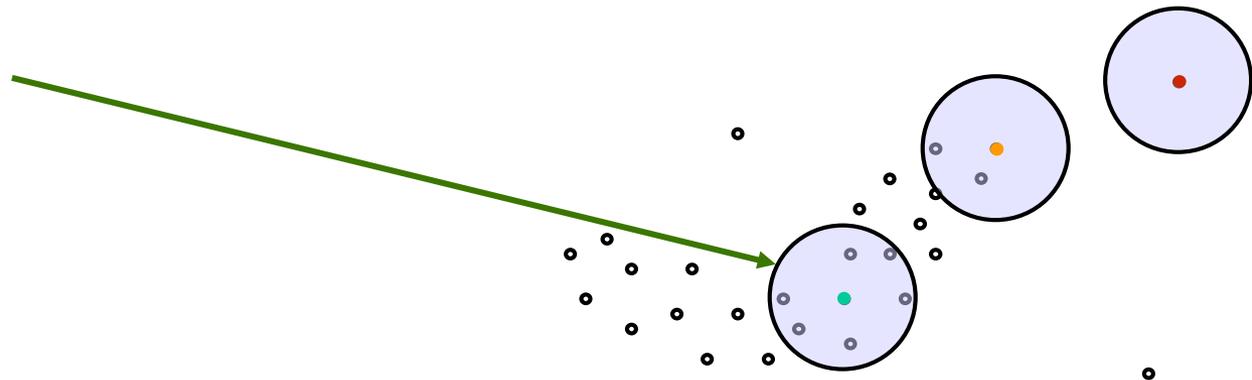


Idee zur effizienten Suche

Parameter $\varepsilon \in \mathbb{R}$ und $MinPts \in \mathbb{N}$ spezifizieren Dichtegrenzwert

ε $MinPts = 4$

- *Kernpunkt*

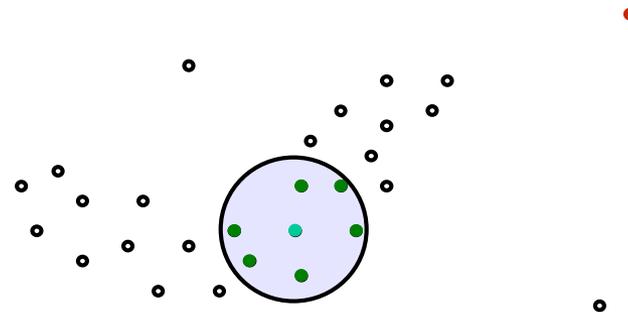


Idee zur effizienten Suche

Parameter $\varepsilon \in \mathbb{R}$ und $MinPts \in \mathbb{N}$ spezifizieren Dichtegrenzwert

ε $MinPts = 4$

- *Kernpunkt*
- *Direkte Dichte-Erreichbarkeit*

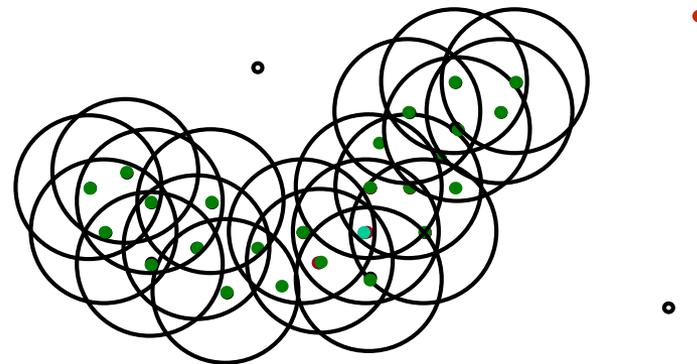


Idee zur effizienten Suche

Parameter $\varepsilon \in \mathbb{R}$ und $MinPts \in \mathbb{N}$ spezifizieren Dichtegrenzwert

ε $MinPts = 4$

- *Kernpunkt*
- *Direkte Dichte-Erreichbarkeit*
- *Dichte-Erreichbarkeit*

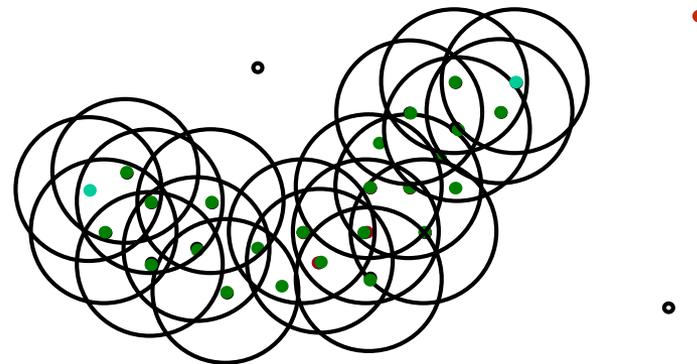


Idee zur effizienten Suche

Parameter $\varepsilon \in \mathbb{R}$ und $MinPts \in \mathbb{N}$ spezifizieren Dichtegrenzwert

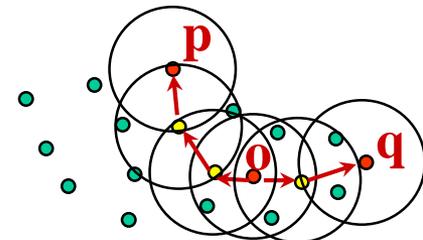
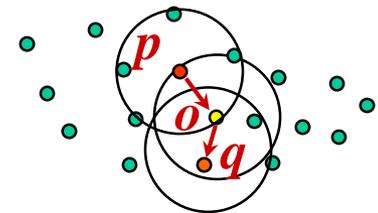
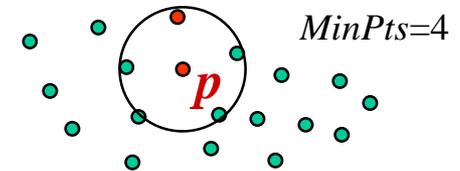
ε $MinPts = 4$

- *Kernpunkt*
- *Direkte Dichte-Erreichbarkeit*
- *Dichte-Erreichbarkeit*
- *Dichte-Verbundenheit*



Formalisierung [Ester, Kriegel, Sander & Xu 1996]

- Ein Objekt $p \in DB$ heißt *Kernobjekt*, wenn gilt:
 $|RQ(p, \varepsilon)| \geq MinPts$
 $RQ(p, \varepsilon) = \{o \in DB \mid dist(p, o) \leq \varepsilon\}$
- Ein Objekt $q \in DB$ ist *direkt dichte-erreichbar* von $p \in DB$ bzgl. ε und $MinPts$, wenn gilt:
 $q \in RQ(p, \varepsilon)$ und p ist ein Kernobjekt in DB .
- Ein Objekt p ist *dichte-erreichbar* von q , wenn es eine Kette von direkt erreichbaren Objekten von q nach p gibt.
- Zwei Objekte p und q sind *dichte-verbunden*, wenn sie beide von einem dritten Objekt o aus dichte-erreichbar sind.



Formalisierung

Ein (*dichtebasierter*) Cluster C bzgl. ε und $MinPts$ ist eine nicht-leere Teilmenge von DB für die die folgenden Bedingungen erfüllt sind:

Maximalität: $\forall p, q \in DB$: wenn $p \in C$ und q dichte-erreichbar von p ist, dann ist auch $q \in C$.

Verbundenheit: $\forall p, q \in C$: p ist dichte-verbunden mit q .

Formalisierung

Definition Clustering

Ein *dichtebasiertes Clustering* CL der Menge DB bzgl. ε und $MinPts$ ist eine „vollständige“ Menge von dichtebasierten Clustern bzgl. ε und $MinPts$ in DB .

Definition Noise

Die Menge $Noise_{CL}$ („*Rauschen*“) ist definiert als die Menge aller Objekte aus DB , die nicht zu einem der dichtebasierten Cluster $C \in CL$ gehören.

Grundlegende Eigenschaft

Sei C ein dichtebasierter Cluster und sei $p \in C$ ein Kernobjekt.

Dann gilt:

$$C = \{o \in DB \mid o \text{ dichte-erreichbar von } p \text{ bzgl. } \varepsilon \text{ und } MinPts\}.$$

=> ermöglicht effiziente Suche (WARUM?)

Achtung: Cluster enthält nicht notwendigerweise nur Kernpunkte

Algorithmus DBSCAN

```

DBSCAN(Punktmenge DB, Real  $\epsilon$ , Integer MinPts)
  // Zu Beginn sind alle Objekte unklassifiziert,
  // o.ClId = UNKLASSIFIZIERT für alle o  $\in$  DB

  ClusterId := nextId(NOISE);
  for i from 1 to |DB| do
    Objekt := DB.get(i);
    if Objekt.ClId = UNKLASSIFIZIERT then
      if ExpandiereCluster(DB, Objekt, ClusterId,  $\epsilon$ , MinPts)
      then ClusterId:=nextId(ClusterId);
  
```

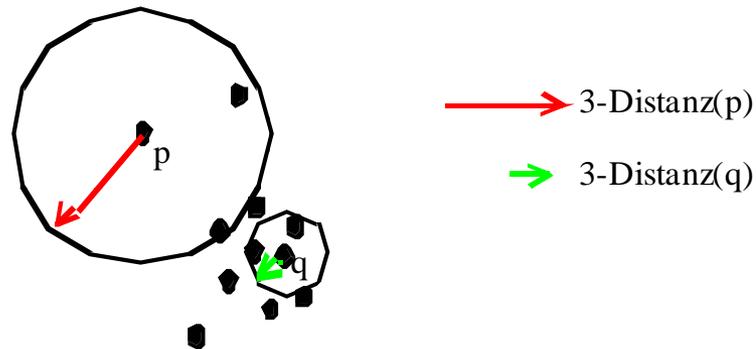
```
ExpandiereCluster(DB, StartObjekt, ClusterId,  $\epsilon$ , MinPts): Boolean
seeds := RQ(StartObjekt,  $\epsilon$ );
if |seeds| < MinPts then // StartObjekt ist kein Kernobjekt
    StartObjekt.ClId := NOISE;
    return false;
// sonst: StartObjekt ist ein Kernobjekt
forall o  $\in$  seeds do o.ClId := ClusterId;
entferne StartObjekt aus seeds;
while seeds  $\neq$  Empty do
    wähle ein Objekt o aus der Menge seeds;
    Nachbarschaft := RQ(o,  $\epsilon$ );
    if |Nachbarschaft|  $\geq$  MinPts then // o ist ein Kernobjekt
        for i from 1 to |Nachbarschaft| do
            p := Nachbarschaft.get(i);
            if p.ClId in {UNCLASSIFIED, NOISE} then
                if p.ClId = UNCLASSIFIED then
                    füge p zur Menge seeds hinzu;
                p.ClId := ClusterId;
            entferne o aus der Menge seeds;
return true;
```

Parameterbestimmung

Cluster: Dichte größer als die durch ε und *MinPts* spezifizierte „Grenzdichte“

Gesucht: der am wenigsten dichte Cluster in der Datenmenge

Heuristische Methode: betrachte die Distanzen zum *k*-nächsten Nachbarn.

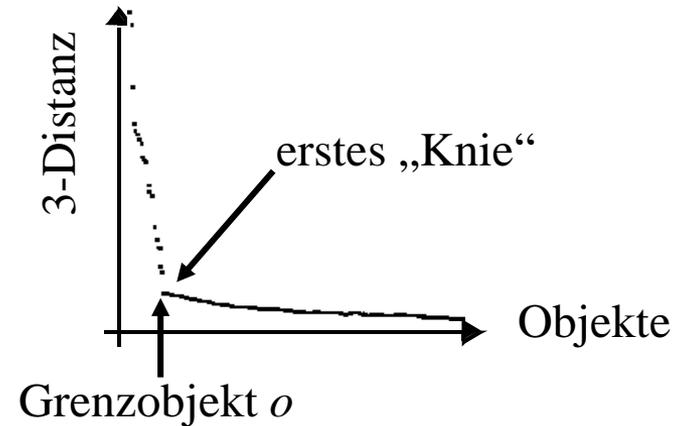


Funktion *k-Distanz*: Distanz eines Objekts zu seinem *k*-nächsten Nachbarn

k-Distanz-Diagramm: die *k*-Distanzen aller Objekte absteigend sortiert

Parameterbestimmung

Beispiel eines k -Distanz-Diagramms

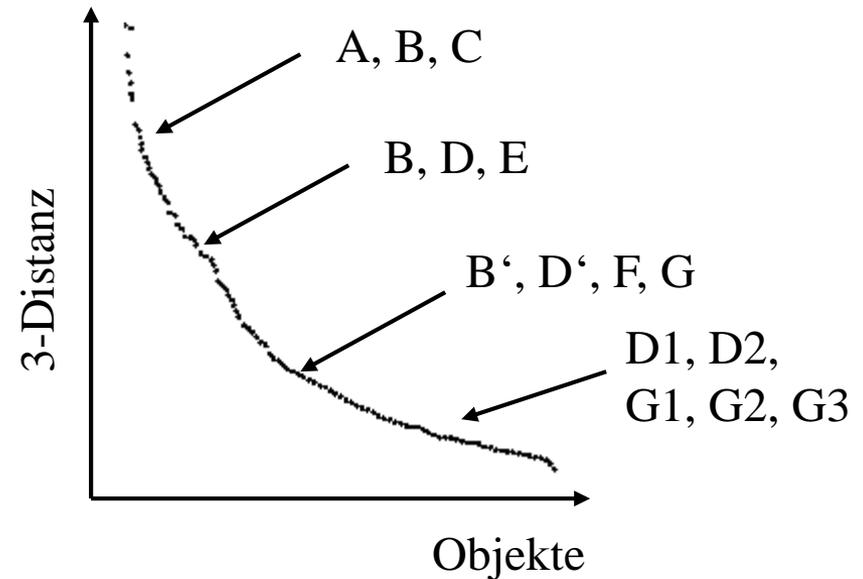
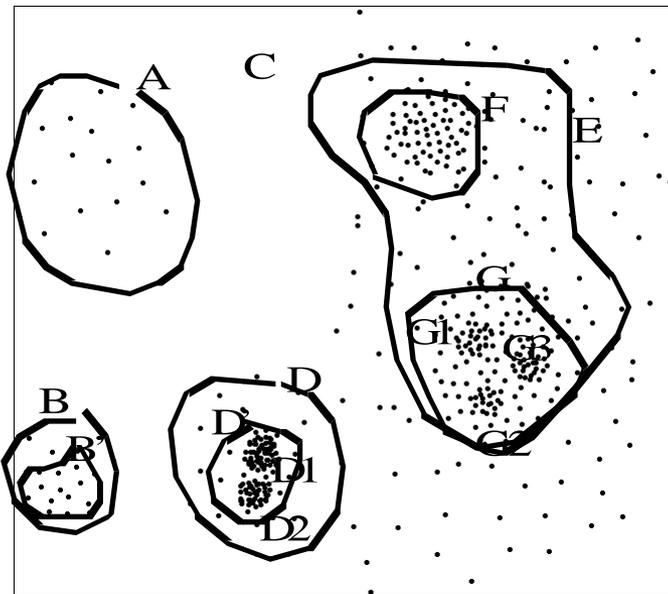


Heuristische Methode

- Benutzer gibt einen Wert für k vor (Default ist $k = 2 \cdot d - 1$), $MinPts := k + 1$.
- System berechnet das k -Distanz-Diagramm und zeigt das Diagramm an.
- Der Benutzer wählt ein Objekt o im k -Distanz-Diagramm als Grenzobjekt aus, $\varepsilon := k\text{-Distanz}(o)$.

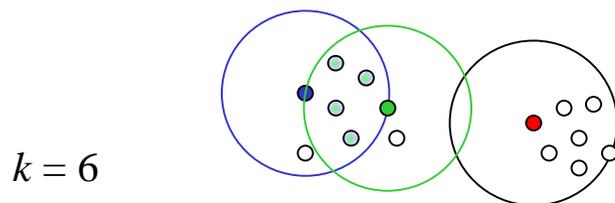
Probleme der Parameterbestimmung

- hierarchische Cluster
- stark unterschiedliche Dichte in verschiedenen Bereichen des Raumes
- Cluster und Rauschen sind nicht gut getrennt



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:
 - $SNN_k\text{-similarity}(p, q) = |INN(p, k) \cap NN(q, k)|$
 - $NN(o, k) =$ Menge der k -nächsten Nachbarn von o

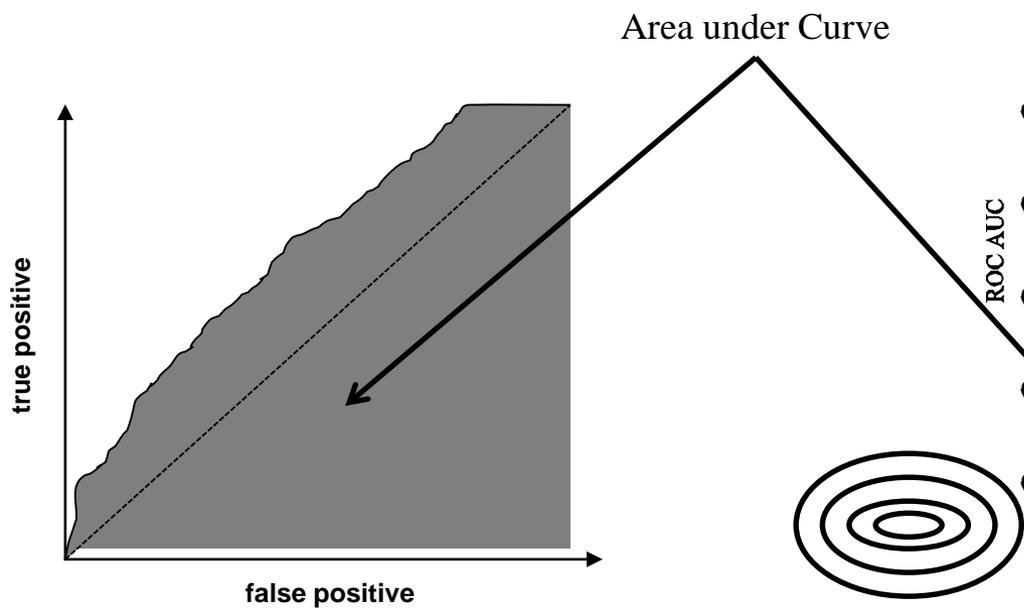


$$SNN_6\text{-similarity}(\bullet, \bullet) = 4$$

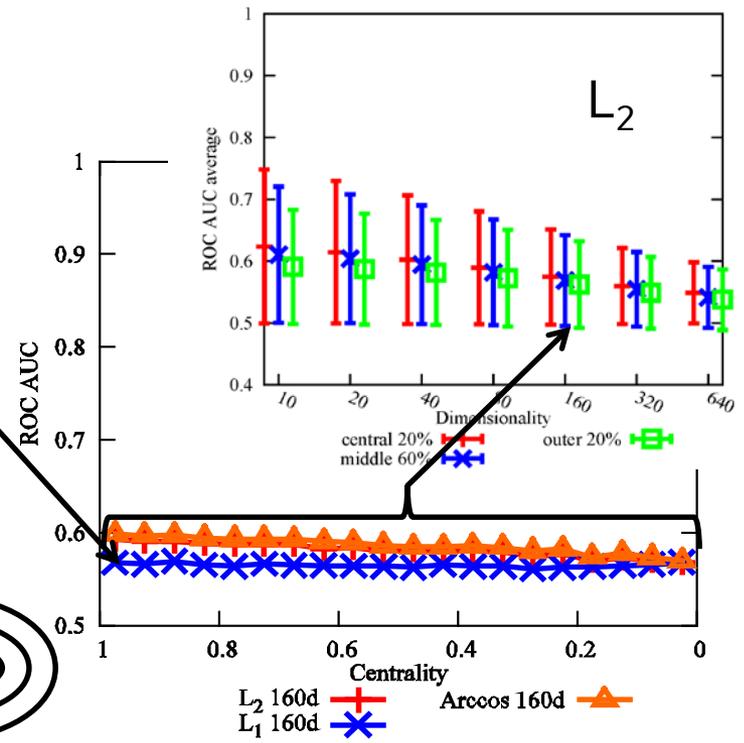
$$SNN_6\text{-similarity}(\bullet, \bullet) = 0$$

Studie: Stabilität von SNN [Houle, Kriegel, Kröger, Schubert, Zimek 2010]

Experiment: Jedes Datenbankobjekt wird einmal als Anfrageobjekt verwendet, Nachbarschaftsranking: Area under curve (AUC) of the Receiver Operating Characteristic (ROC)



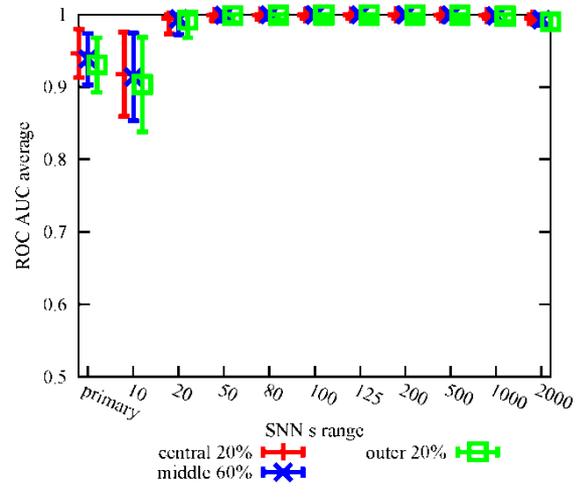
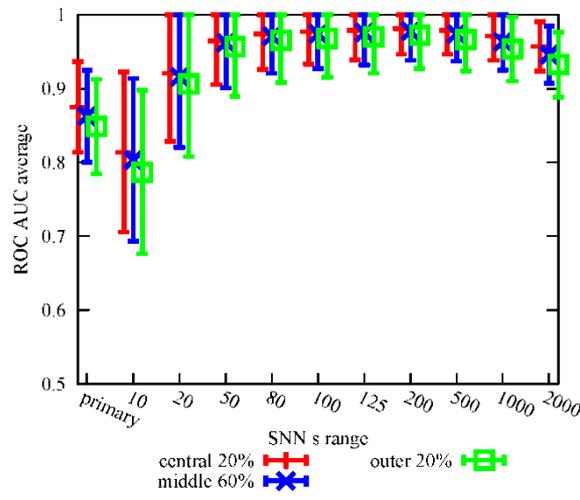
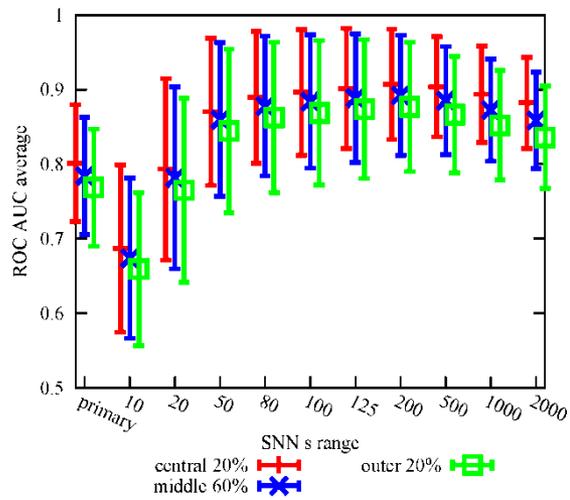
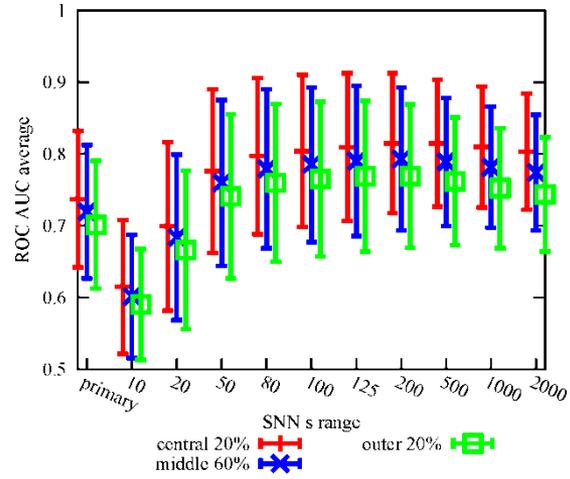
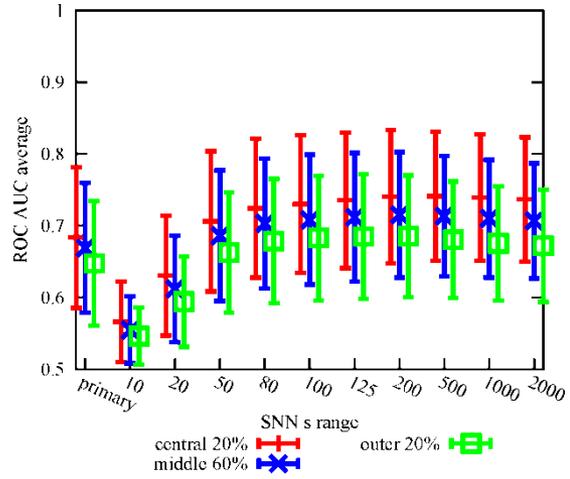
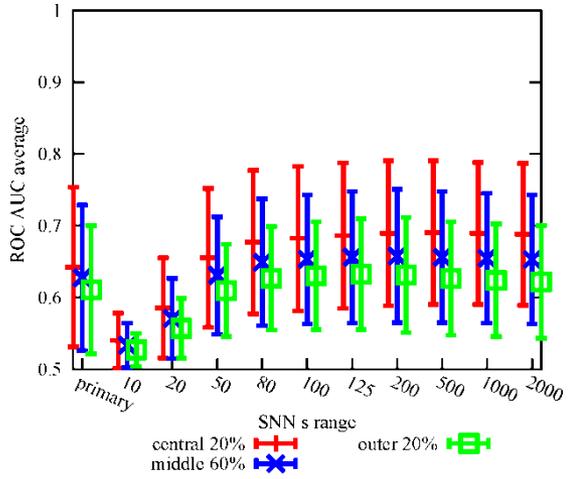
eine Anfrage (Nachbarn gleicher vs. anderer Cluster)



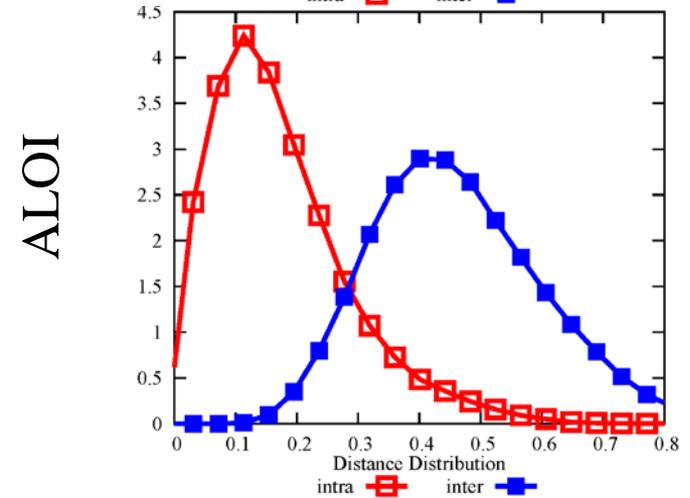
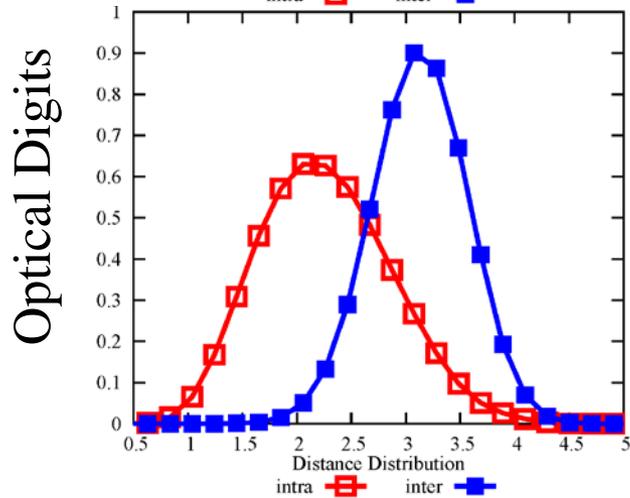
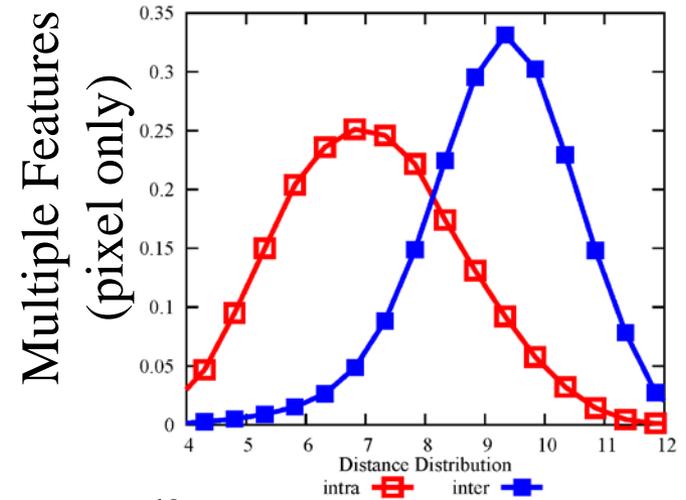
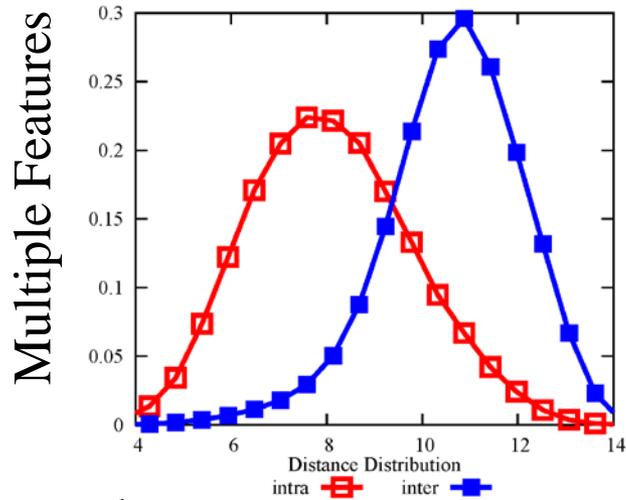
Durchschnitt aller Anfragen

Dichtebasiertes Clustering

SNN basierend auf Euklidischer Distanz, künstliche Datensätze, alle Attribute relevant, 20/40/80/160/320/640 Dimensionen

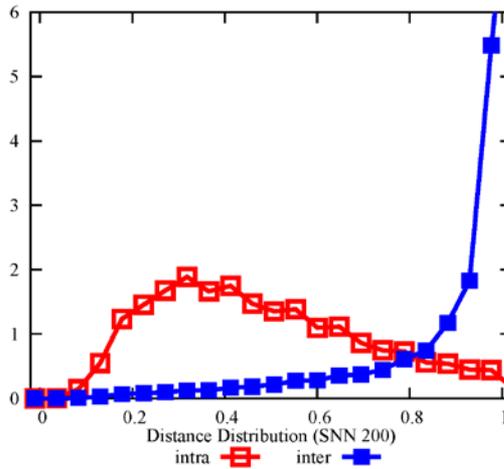


Verteilungen Euklidischer Distanz (intra-cluster vs. inter-cluster Distanzen)

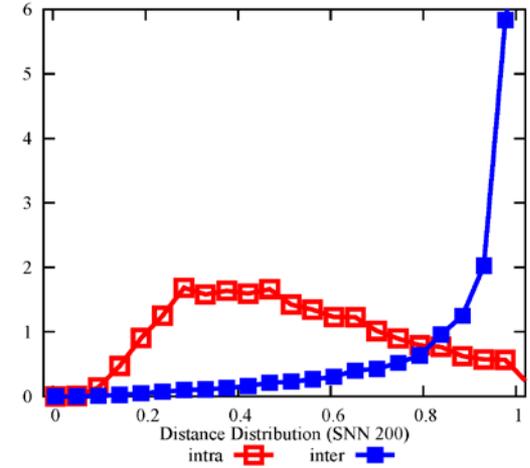


Verteilungen SNN Distanz (intra-cluster vs. inter-cluster Distanzen)

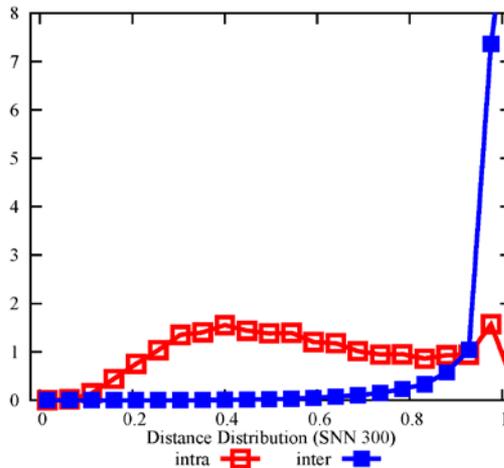
Multiple Features



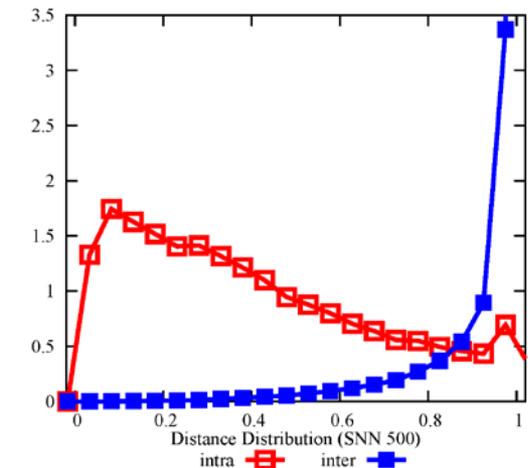
Multiple Features
(pixel only)



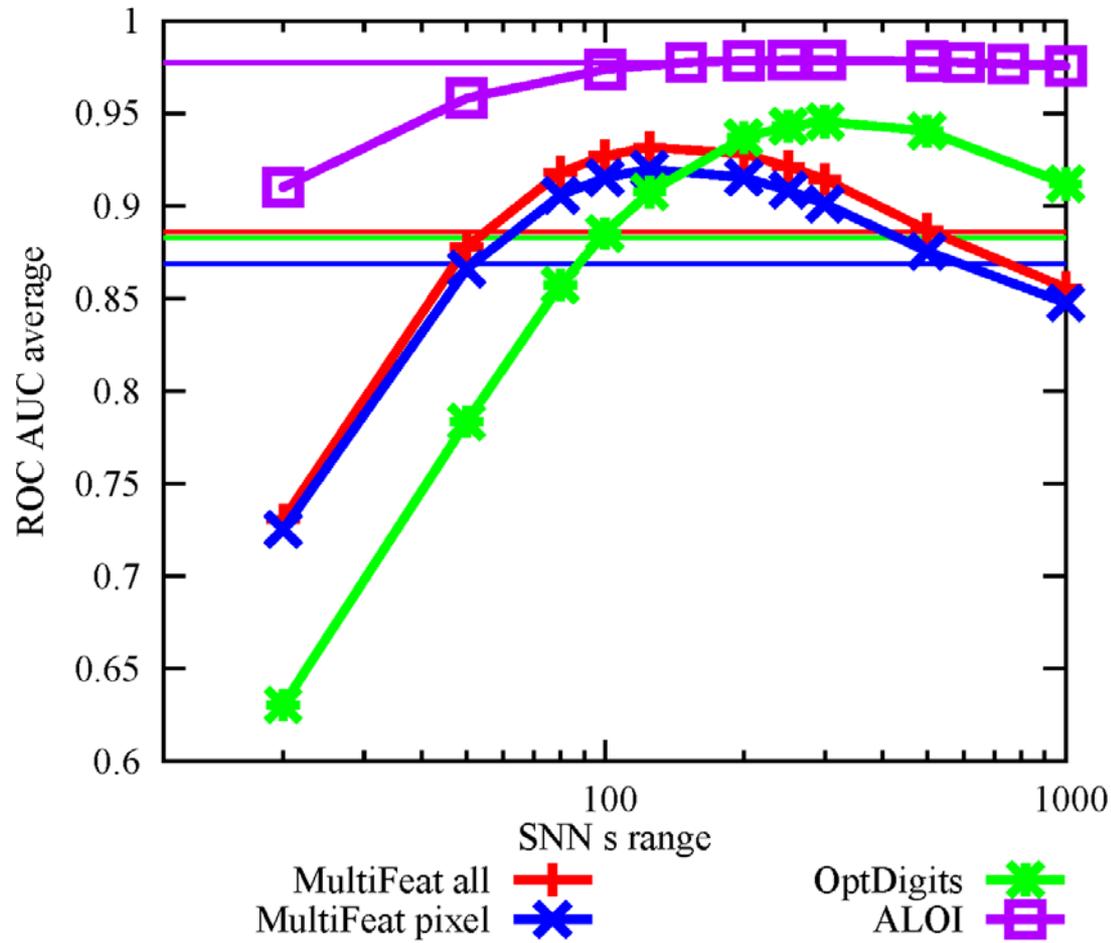
Optical Digits



ALOI



Ranking-Qualität



mehr Ergebnisse auf:

<http://www.dbs.ifi.lmu.de/research/SNN/>

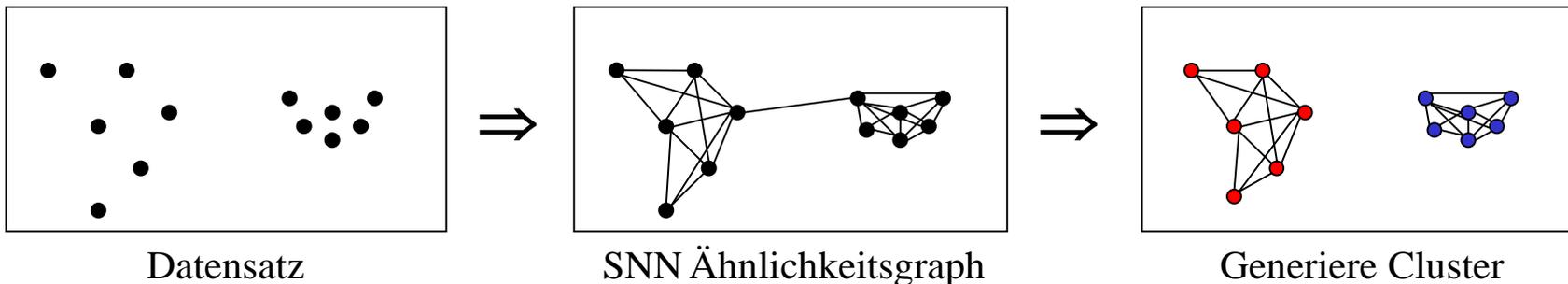
Einfaches SNN-Clustering [Jarvis, Patrick 73]:

1. Berechnung der Ähnlichkeitsmatrix und des Ähnlichkeitsgraphen

- für alle Objekt-Paare $p, q \in DB$: berechne SNN_k -similarity(p, q)
- SNN_k -Ähnlichkeitsgraph:
 - Knoten = Objekte
 - Kante zwischen jedem Objektpaar p, q mit Gewicht SNN_k -similarity(p, q)
- Keine Kanten mit Gewicht 0

2. Generiere Cluster

- Lösche alle Kanten, deren Gewicht unterhalb eines Grenzwerts τ liegen
- Cluster = verbundene Komponenten im resultierenden Graphen



Problem:

- Threshold τ schwer zu bestimmen
- kleine Variationen führen zu stark unterschiedlichen Ergebnissen

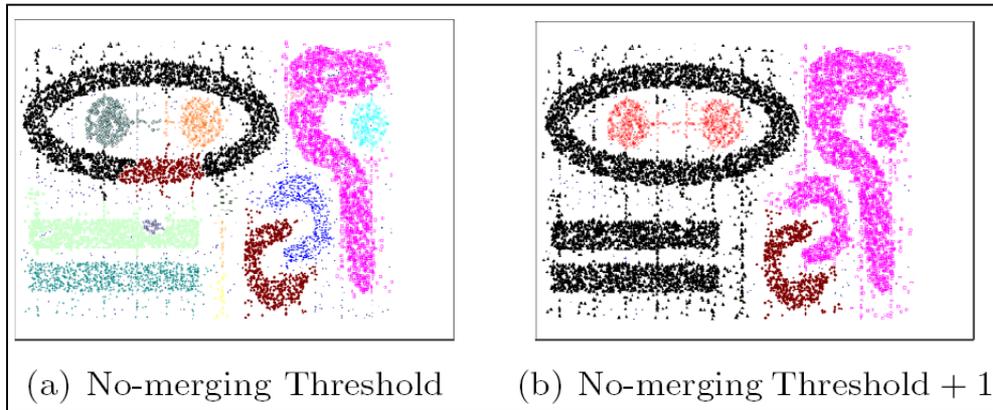


Bild aus:

[Ertöz, Steinbach, Kumar 03]

Lösung [Ertöz, Steinbach, Kumar 03]

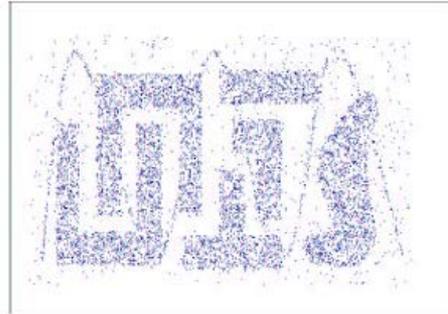
- Kombiniere SNN-Ähnlichkeit mit dichtebasierten Konzepten
- SNN-Dichte:

Anzahl der Punkte innerhalb eines spezifizierten Radius ε bzgl. SNN-Ähnlichkeit

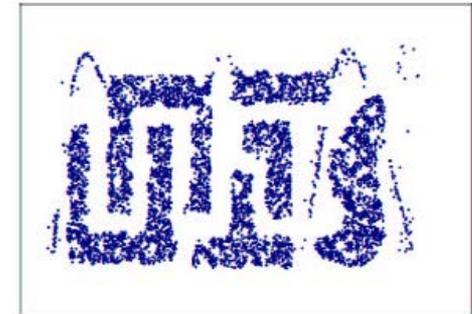
$$\text{SNN}_k\text{-density}(p, \varepsilon) = |\{q \mid \text{SNN}_k\text{-similarity}(p, q) \geq \varepsilon\}|$$

SNN-Dichte

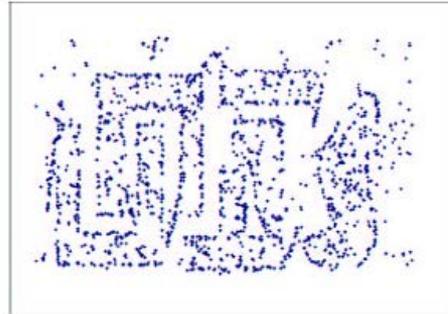
- Beispiel:
 - 10 000 Daten (Bild (a))
 $k = 50, \varepsilon = 20$
 - Bild (b): "Kernpunkte"
alle Punkte mit SNN-Dichte ≥ 34
 - Bild (c): "Randpunkte"
alle Punkte mit SNN-Dichte ≥ 17
 - Bild (d): "Rauschen"
alle Punkte mit SNN-Dichte < 17



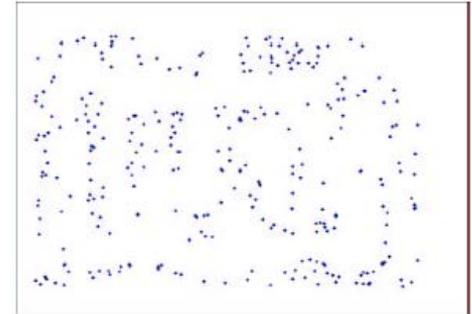
(a) All Points



(b) High SNN Density



(c) Medium SNN Density



(d) Low SNN Density

Bild aus: [Ertöz, Steinbach, Kumar 03]

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

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

Eingabe: k , ε , $minPts$

1. Berechne Ähnlichkeitsmatrix und ϵ -graph
(siehe einfaches SNN-Clustering)
2. Berechne die SNN_k -Dichte für jeden Punkt bzgl. ε
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 einen Kernpunkt q gibt, mit $SNN_k\text{-similarity}(q, p) \geq \varepsilon$
6. Alle anderen Nicht-Kernpunkte sind Rauschen

→ **DBSCAN mit SNN-Ähnlichkeit**

Diskussion

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

Generalisierung von DBSCAN [Sander, Ester, Kriegel & Xu 1998]

Grundidee für dichte-basierte Cluster :

“ ε -Nachbarschaft enthält mindestens $MinPts$ Punkte”

“Distanz $\leq \varepsilon$ ”

$NPred(o,p)$

reflexiv, symmetrisch
für Paare von Objekten

Verallgemeinerte Nachbarschaft

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

“ $|N_\varepsilon| \geq MinPts$ ”

$MinWeight(N)$

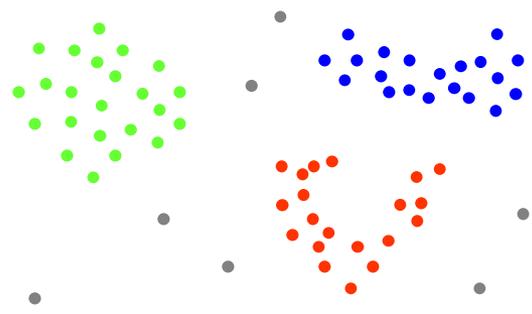
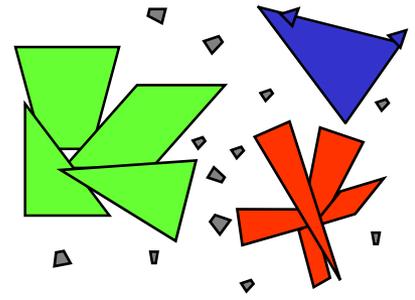
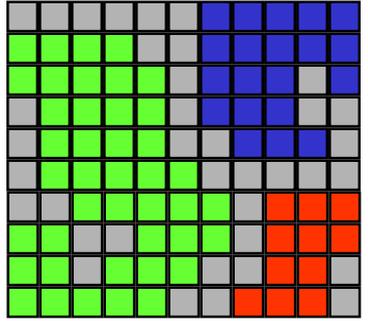
beliebiges Prädikat für
Mengen von Objekten

Verallgemeinerte minimale Kardinalität

$MinWeight(N_{NPred}(o))$

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

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

Algorithmus GDBSCAN

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

Literatur

- L. Ertöz, M. Steinbach, V. Kumar: Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data. *SDM* 2003.
- M. Ester, H.-P. Kriegel, J. Sander, X. Xu: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *KDD* 1996: 226-231.
- M. E. Houle, H.-P. Kriegel, P. Kröger, E. Schubert, A. Zimek: Can Shared-Neighbor Distances Defeat the Curse of Dimensionality? *SSDBM* 2010: 482-500.
- R. A. Jarvis, E. A. Patrick: Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on Computers* 100(11) : 1025-1034 (1973).
- H.-P. Kriegel, P. Kröger, J. Sander, A. Zimek: Density-based clustering. *Wiley Interdisc. Rev.: Data Mining and Knowledge Discovery* 1(3): 231-240 (2011).
- J. Sander, M. Ester, H.-P. Kriegel, X. Xu: Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. *Data Min. Knowl. Discov.* 2(2): 169-194 (1998).

Dichte-basierte Verfahren: Was haben Sie gelernt?

- Parametrisches vs. Nicht-Parametrisches Clustering
- Algorithmus DBSCAN: Intuition und Formalisierung
 - Kernpunkte
 - (direkt) dichte-erreichbar
 - dichte-verbundene Menge
 - Randpunkte
 - Rauschpunkte
- Heuristiken zur Bestimmung geeigneter Dichte-Grenzwert-Parameter
- Shared-Neighbor Distanzen
- (dichte-basiertes) Clustering mit SNN
- Generalisierung von DBSCAN