Einführung in Web- und Data-Science

Clustering

Prof. Dr. Ralf Möller

Universität zu Lübeck

Institut für Informationssysteme

Tanya Braun (Übungen)

Danksagung

- Zur Vorbereitung dieser Präsentationen wurden Materialien verwendet von
  - Eamonn Keogh (University of California – Riverside) und
  - Sascha Szott (HPI Potsdam)

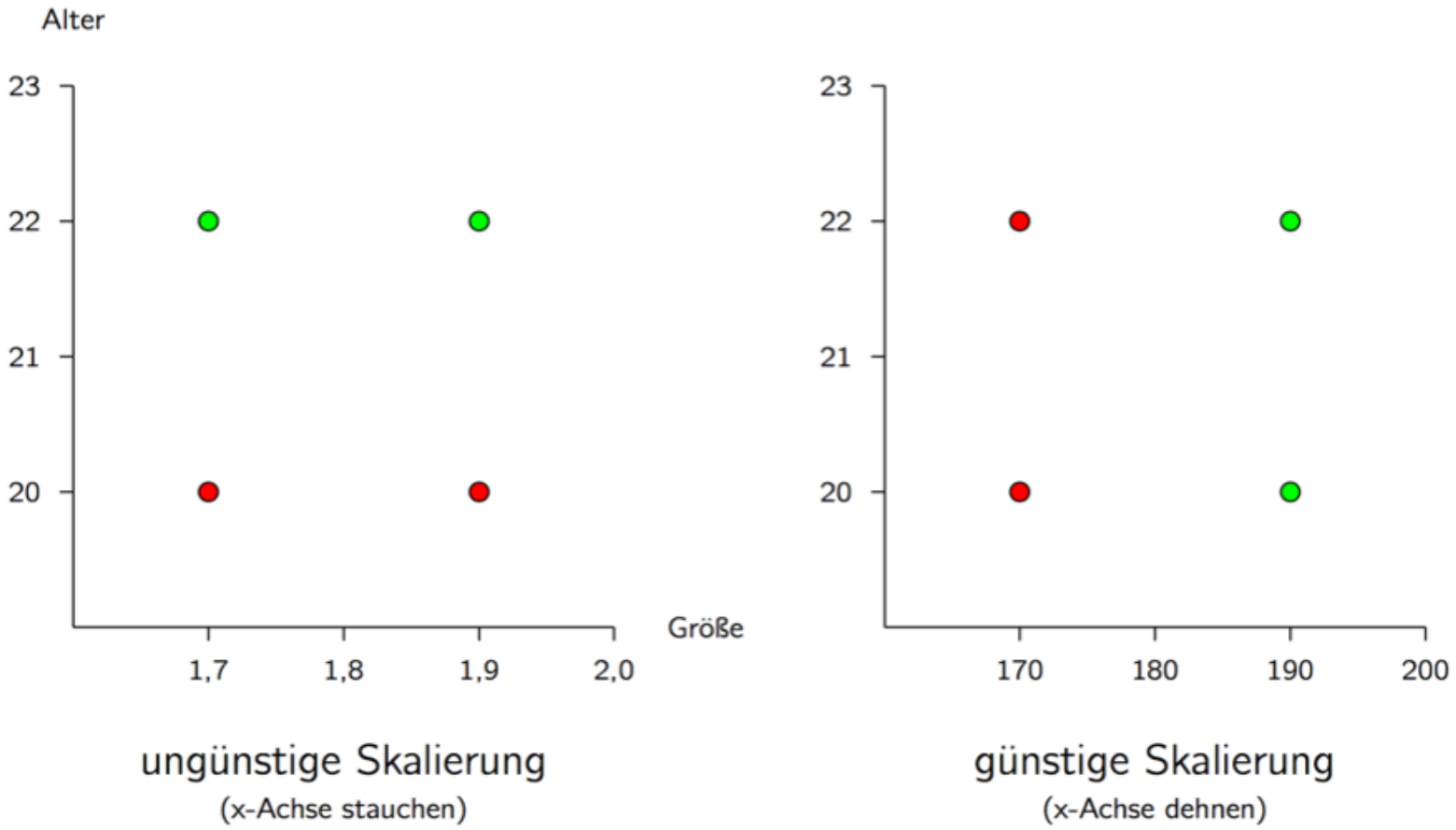
Clustering

- Form des unüberwachten Lernens
- Suche nach natürlichen Gruppierungen von Objekten
  - Klassen direkt aus Daten bestimmen
    - Hohe Intra-Klassen-Ähnlichkeit
    - Kleine Inter-Klassen-Ähnlichkeit
  - Ggs.: Klassifikation
- Distanzmaße
  - z. B. Minkowski Distanz (im $\mathbb{R}^n$):

$$d_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} = ||\mathbf{x} - \mathbf{y}||_p$$

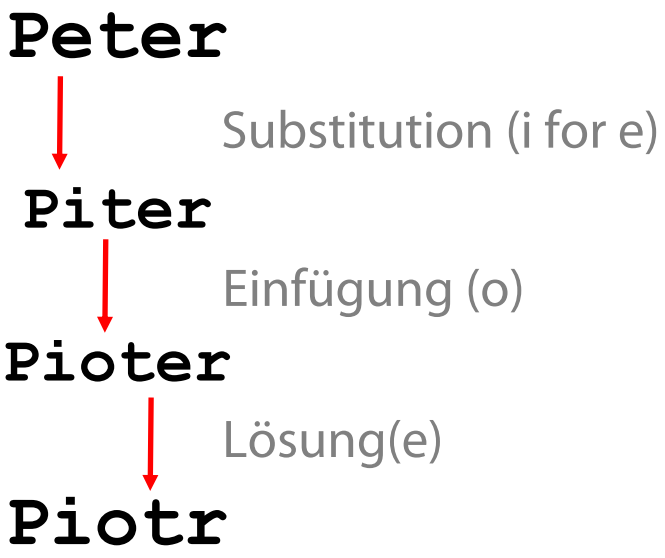
  - für $p = 1$: Manhattan Distanz
  - für $p = 2$: Euklidische Distanz

Einflüsse des Distanzmaßes auf Clusterbildung

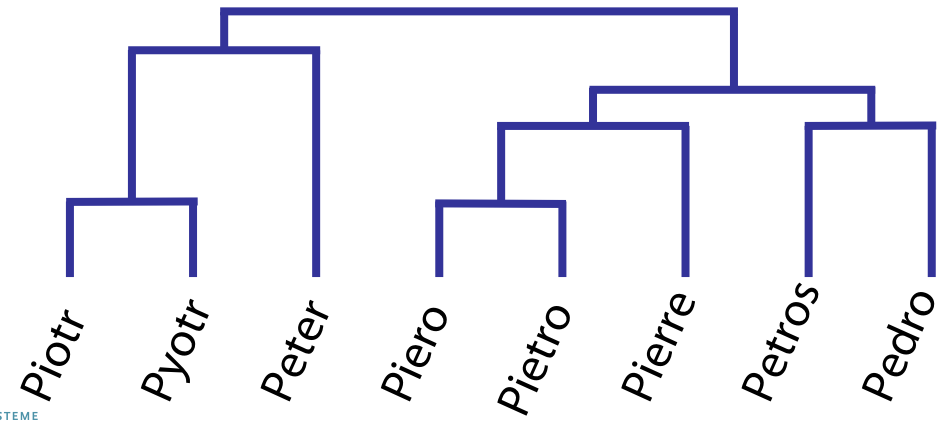


Abhilfe: Gewichtung, z. B. durch Normalisierung

Hierarchisches Clustering



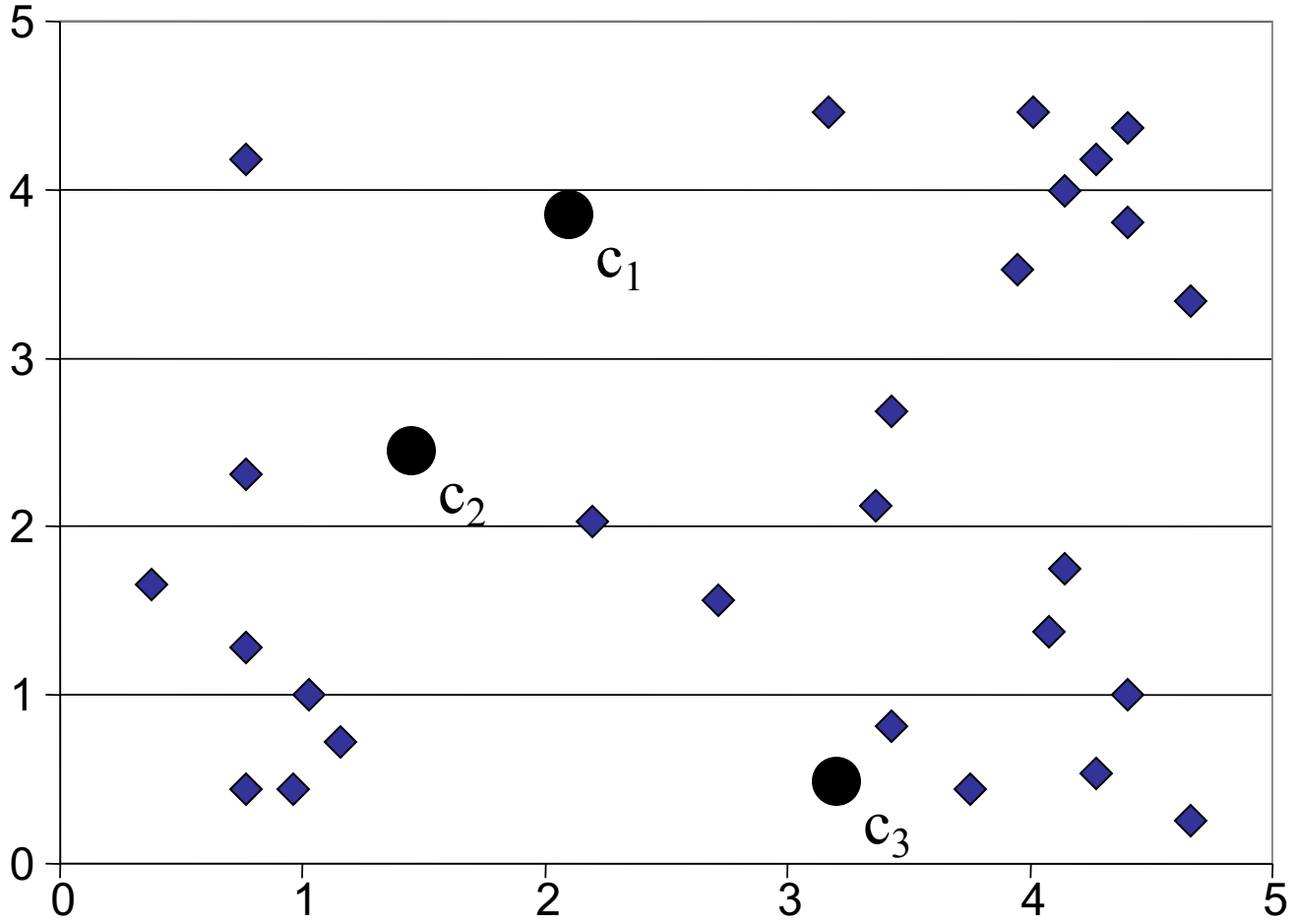
- Kostenfunktion
  - Substitution 1
  - Einfügung 1
  - Lösung 1
- Dist(Peter, Piotr) = 3
- $Dist(C_i, C_j) = \frac{1}{|C_i| \cdot |C_j|} \sum_{c \in C_i} \sum_{d \in C_j} Dist(c, d)$



Dendrogramm

Partitionierung: K-means Clustering (1)

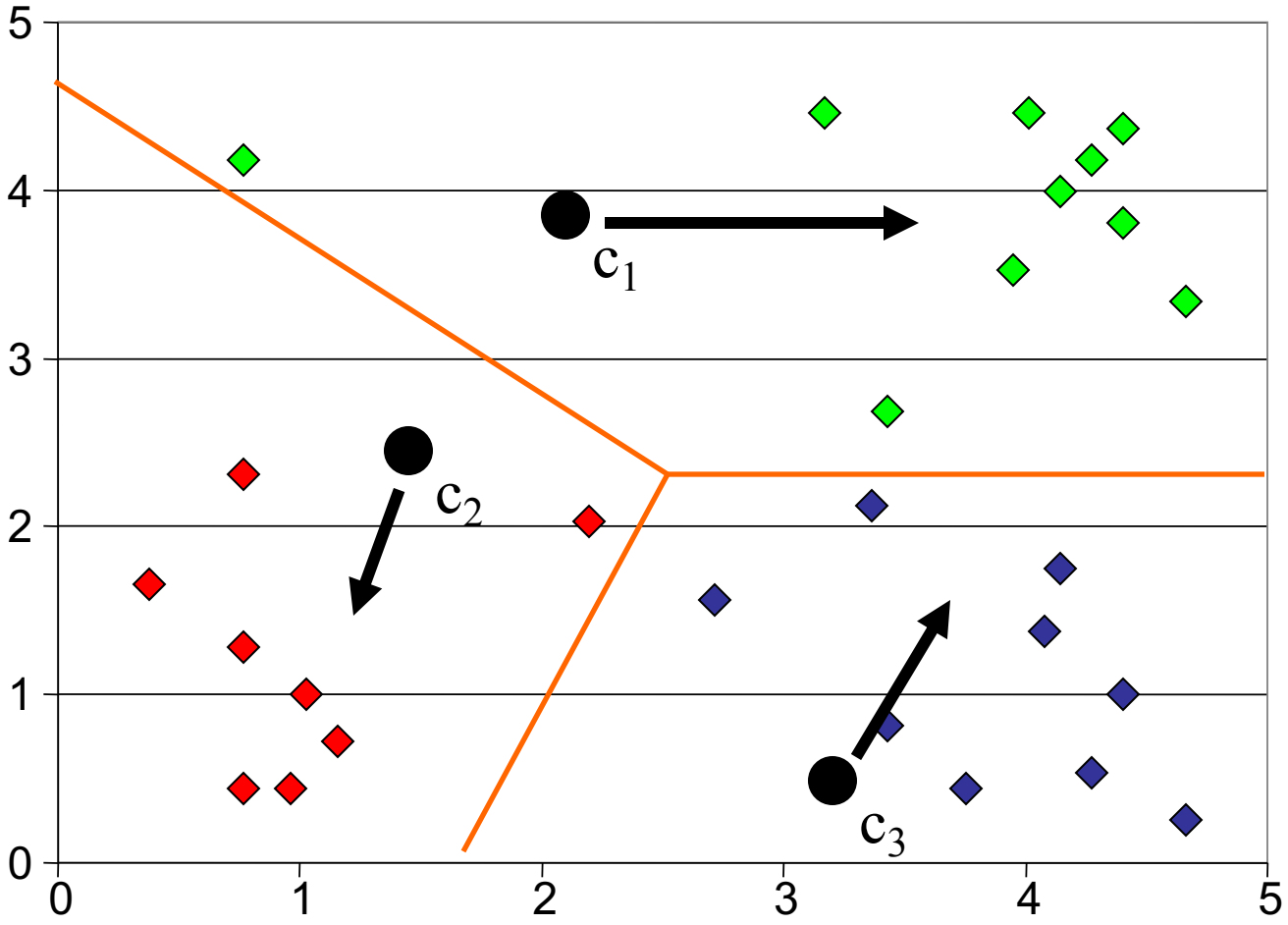
Distanzmaß: Euklidische Distanz



$$C_i^t = \left\{ x_j : \left\| x_j - c_i^t \right\|_2 \leq \left\| x_j - c_r^t \right\|_2 \text{ for all } r = 1 \ldots k, r \neq i \right\}$$

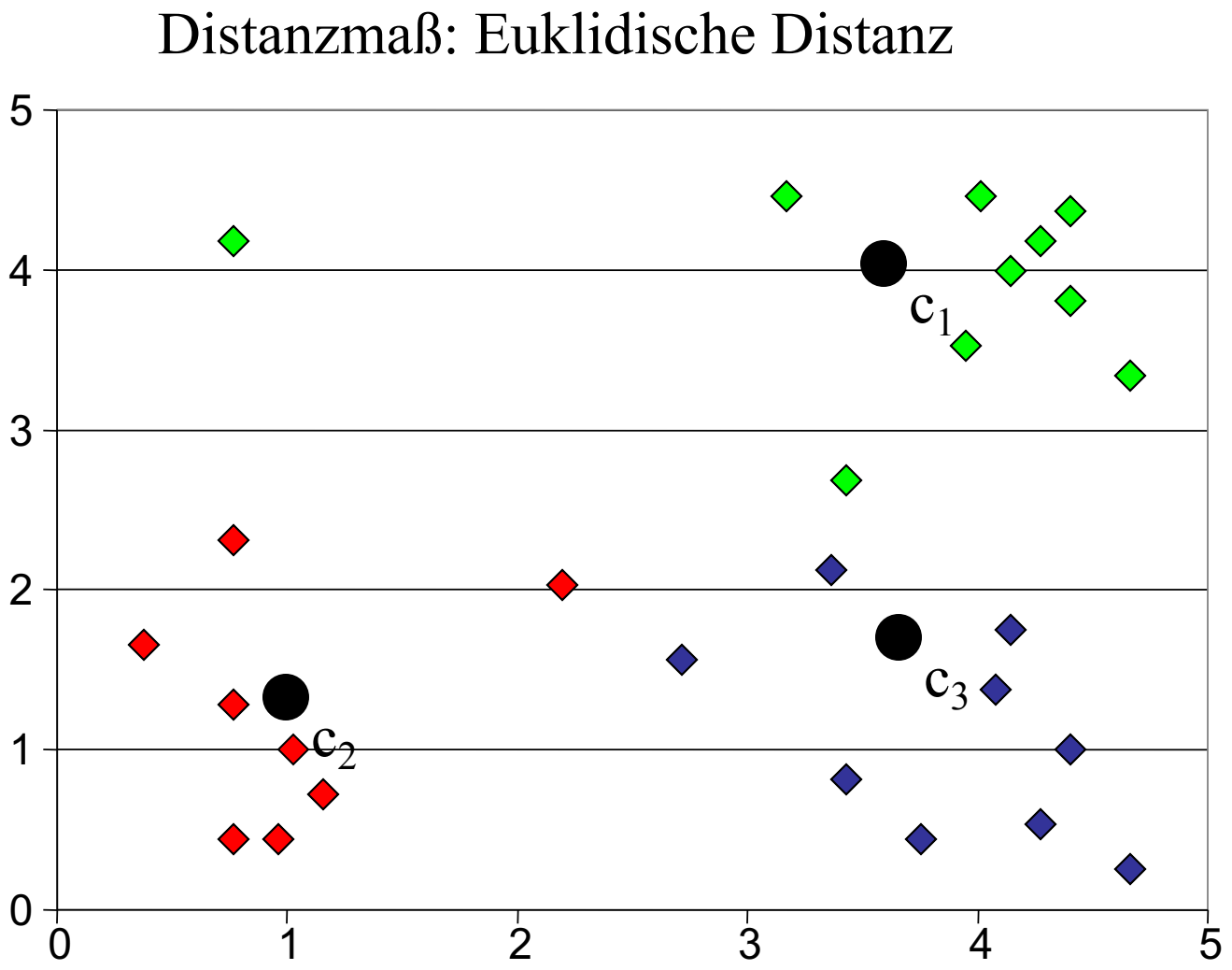
K-means Clustering (2)

Distanzmaß: Euklidische Distanz

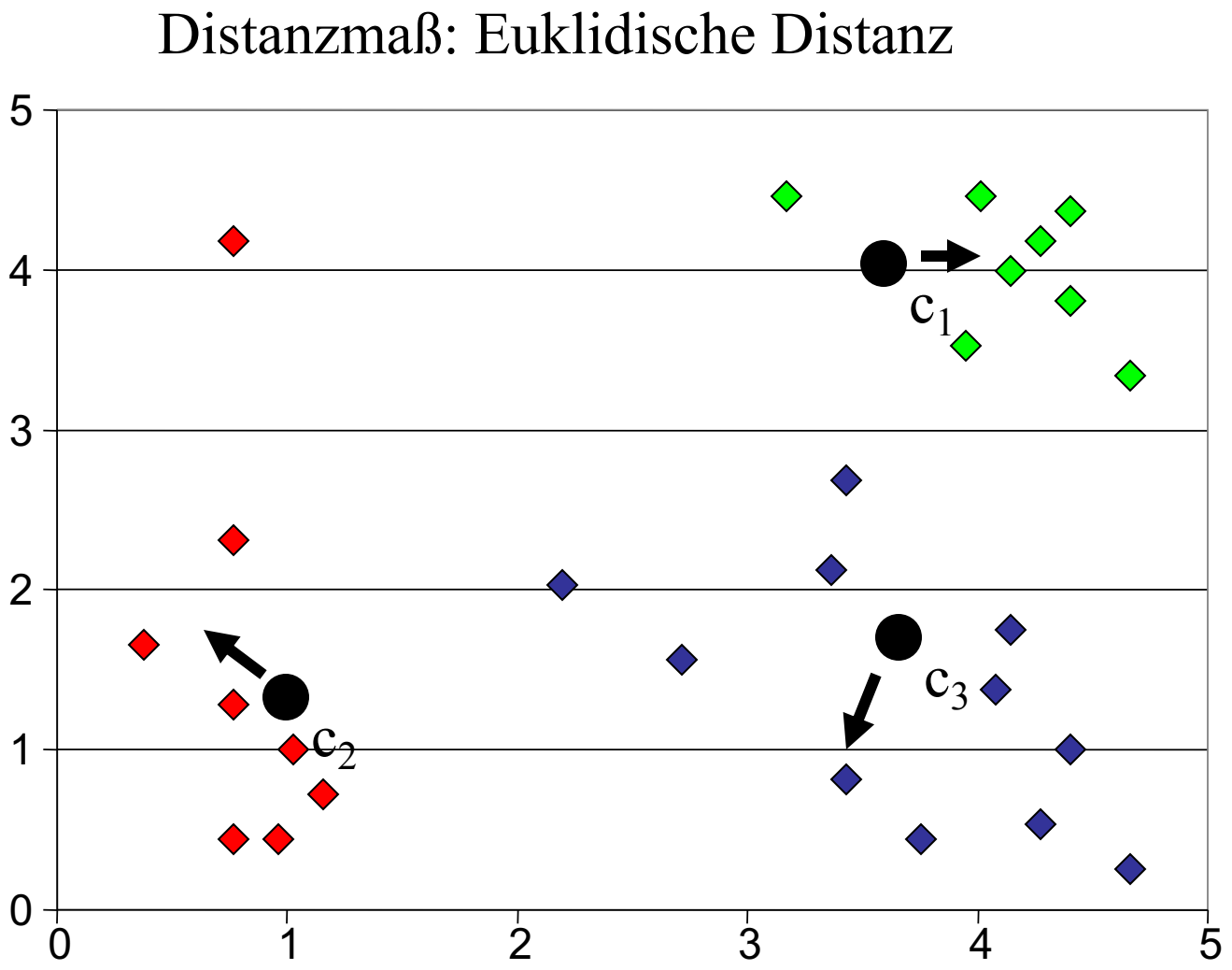


$$c_i^{t+1} = \frac{1}{|C_i^t|} \sum_{x_j \in C_i^t} x_j$$

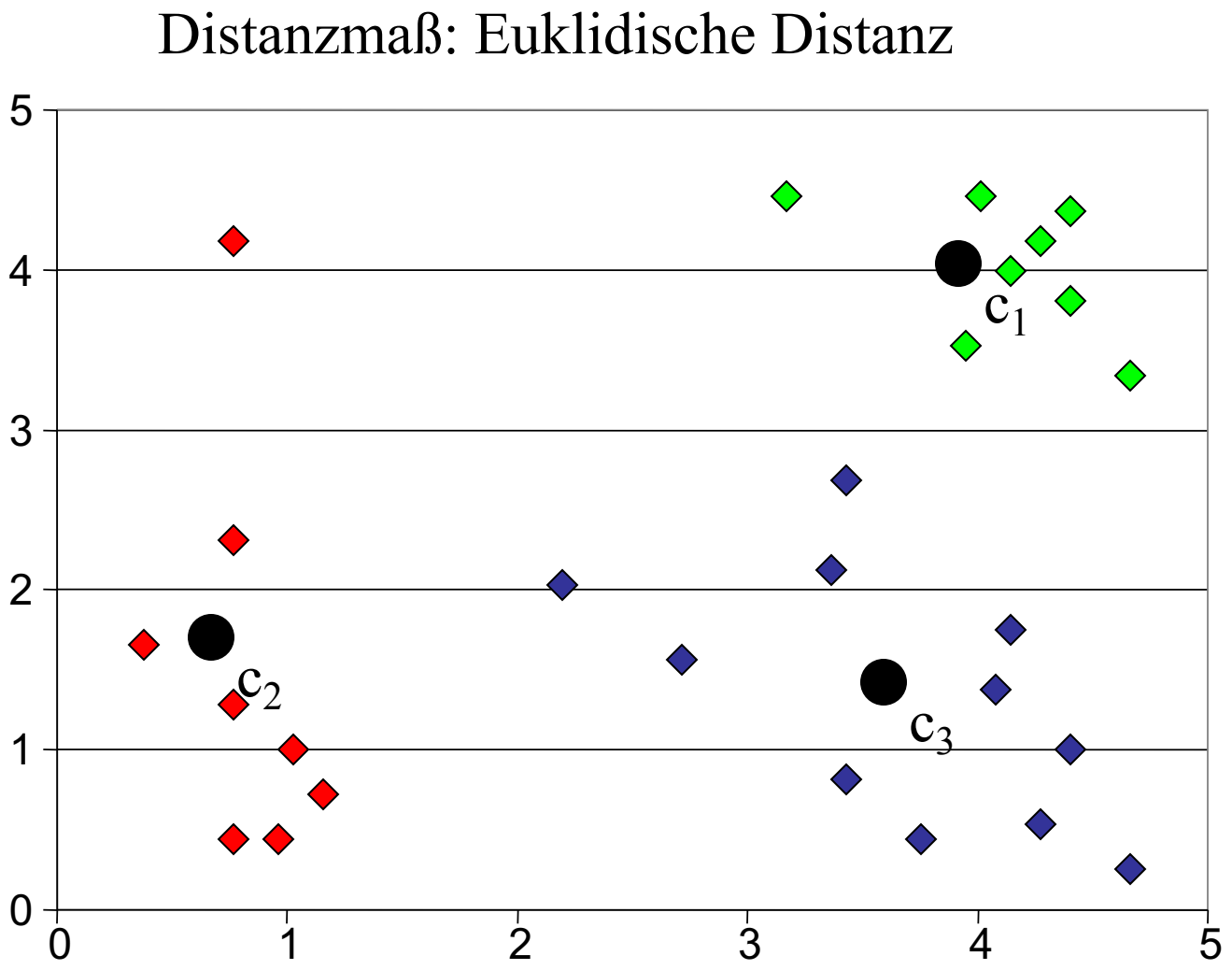
K-means Clustering (3)



K-means Clustering (4)



K-means Clustering (5)



K-Means: Cluster-Repräsentation

- Parameter $k \in \mathbb{N}$ bestimmt Anzahl der Cluster (woher?)
- Jedes Cluster C_i durch Zentroid $c_i \in \mathbb{R}^n$ repräsentiert Mittelwert bezüglich aller in C_i enthaltenen Punkte, d.h.,

$$c_i = \left(\frac{1}{|C_i|} \sum_{x_j \in C_i} x_j^1 , \dots , \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j^n \right)$$

  x_j^l l'te Kompontente

- Ziel: wähle Cluster $C_1, \dots, C_k \subseteq \mathcal{X}$ (alle Datenpunkte), so dass $\{C_1, \dots, C_k\}$ eine Partition von $\mathcal{X}$ ist und

$$E(C_1, \dots, C_k) = \sum_{i=1}^{k} \sum_{x_j \in C_i} \left\| x_j - c_i \right\|_2^2$$

  (intra-cluster Varianz) minimiert wird

K-Means: Algorithmus

1. Wähle k zufällige Punkte $c_1, \dots, c_k \in \mathbb{R}^n$
2. $\forall\, x_j \in \mathcal{X}$: ordne x_j dem nähesten Zentroid zu, d.h., x_j wird c_i zugeordnet, falls
$$d(x_j, c_i) = \min_{1 \le i \le k} d(x_j, c_i)$$
wobei $d(.)$ eine Distanzfunktion ist (z.B. $\|.\|_2$)
3. Sei C_i die Menge aller Objekte, die c_i zugeordnet sind. Berechne ausgehend von C_i den Zentroid c_i neu.
4. Falls sich im vorherigen Schritt mindestens ein Zentroid geändert hat, gehe zu 2.
Andernfalls: Stop
   - $C_1, \dots, C_k$ ist eine Partitionierung von $\mathcal{X}$

K-Means-Ergebnis hängt vom Startwert ab

gutes Resultat:

schlechtes Resultat:

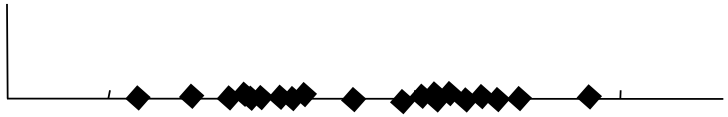
Diskussion

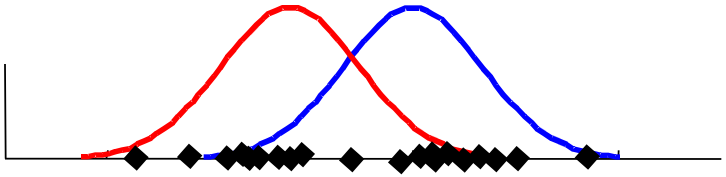
- Meist relativ wenige Schritte notwendig
  - Findet aber ggf. nur lokales Optimum
- Nur anwendbar, wenn Mittel definiert
  - Erweiterungen für kategoriale Daten existieren
- Basiert auf vorgegebener Clusteranzahl k
- Cluster haben meist gleiche Größe
- Probleme bei nichtkonvexen Formen
  - Varianten von K-Means (z.B. K-Medoid)

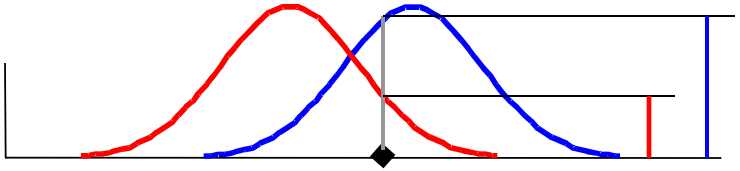
Trend

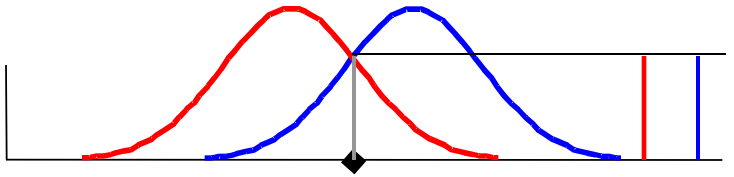
Wunsch

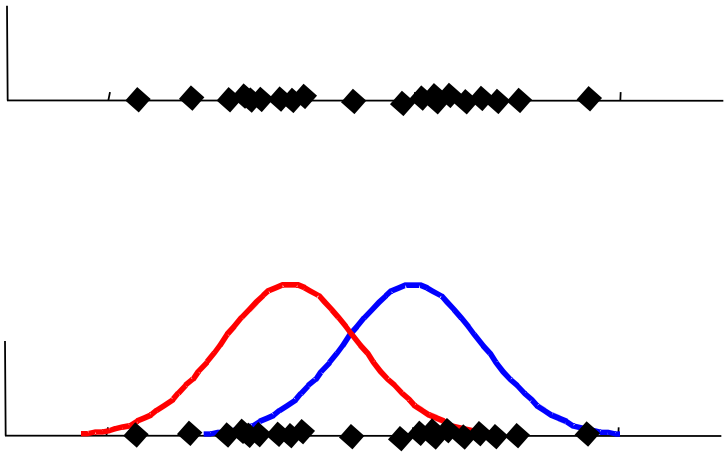
Anpassung von Gauß-Funktionen

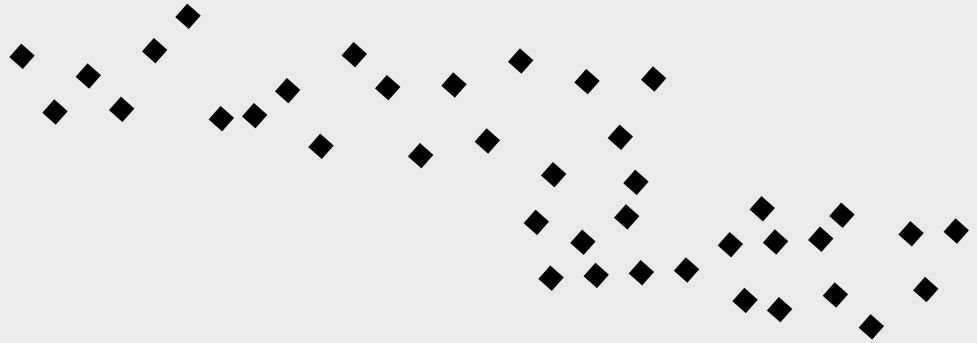


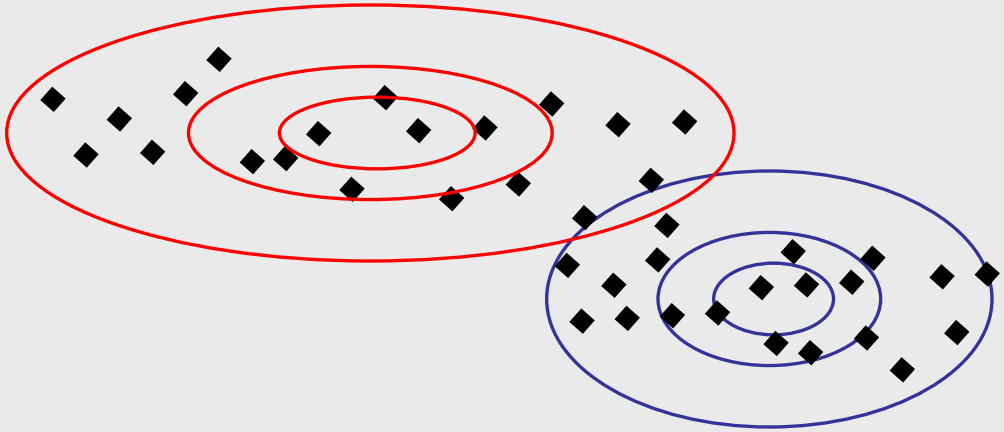












Anpassung über Expectation-Minimization

- **Initialisierung:** Wähle k zufällige Mittelwerte, etc.
- **E Schritt:** $\forall x_j \in \mathcal{X}$:

$$P(c_i|x_j) = \frac{P(c_i)P(x_j|c_i)}{P(x_i)} = \frac{P(c_i)P(x_j|c_i)}{\sum_{i'} P(c_{i'})P(x_k|c_{i'})}$$

$\mathcal{N}(x_j, \mu, \sigma^2)$

- **M Schritt:** $\forall c_i$:

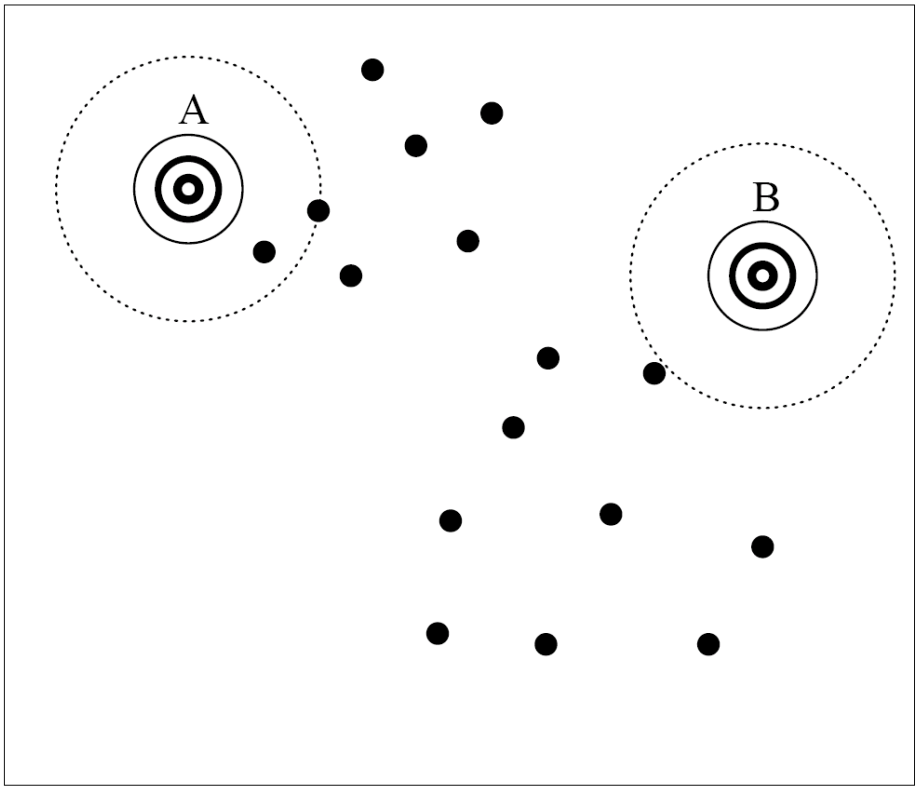
$$P(c_i) = \frac{1}{|\mathcal{X}|} \sum_{x_j \in \mathcal{X}} P(c_i|x_j)$$

$$\sigma_i^2 = \frac{\sum_{x_j \in \mathcal{X}} \left(x_j - \mu^2\right)^2 P(c_i|x_j)}{\sum_{x_j \in \mathcal{X}} P(\mu_i|c_i)}$$

$$\mu_i = \frac{\sum_{x_j \in \mathcal{X}} x_j P(c_i|x_j)}{\sum_{x_j \in \mathcal{X}} P(c_i|x_j)}$$

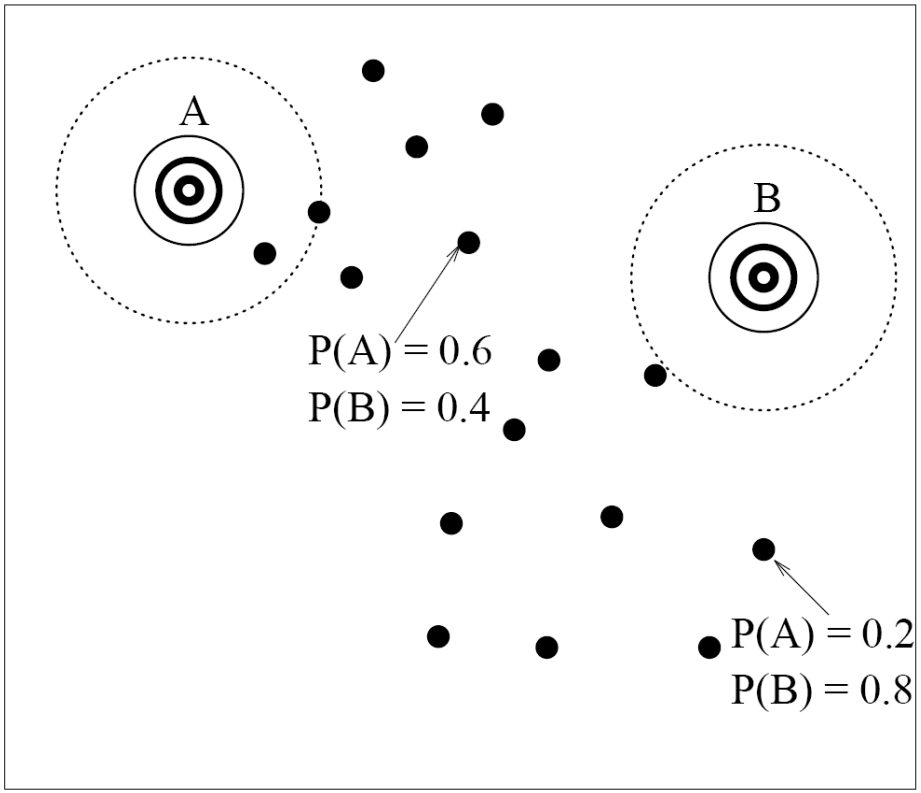
Initialisierung

- Weise Parametern zufällige Werte zu



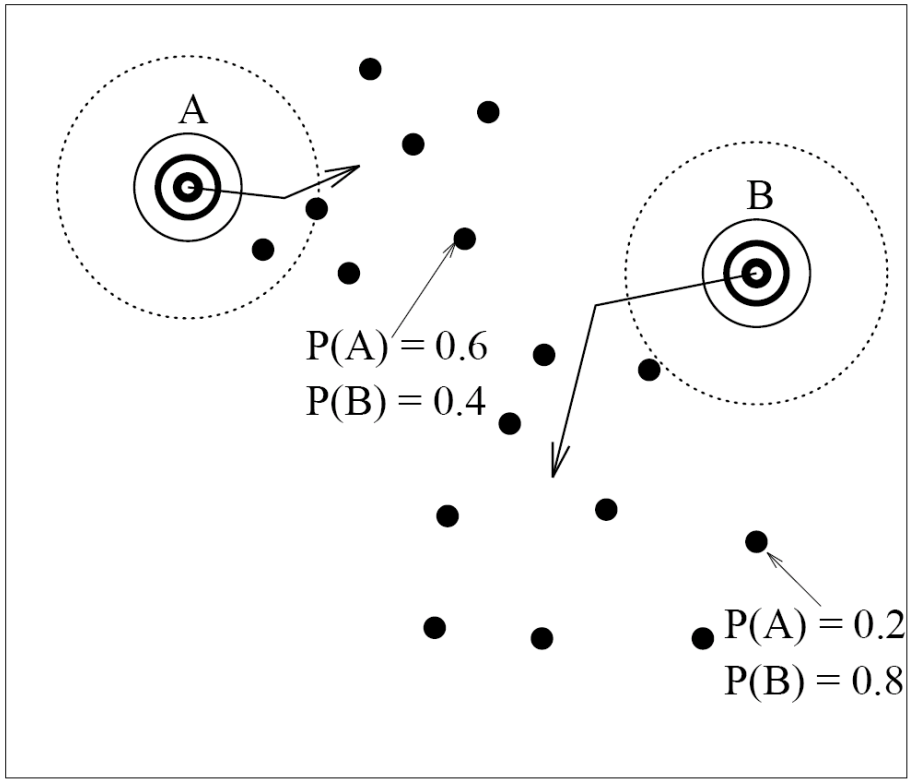
E-Schritt

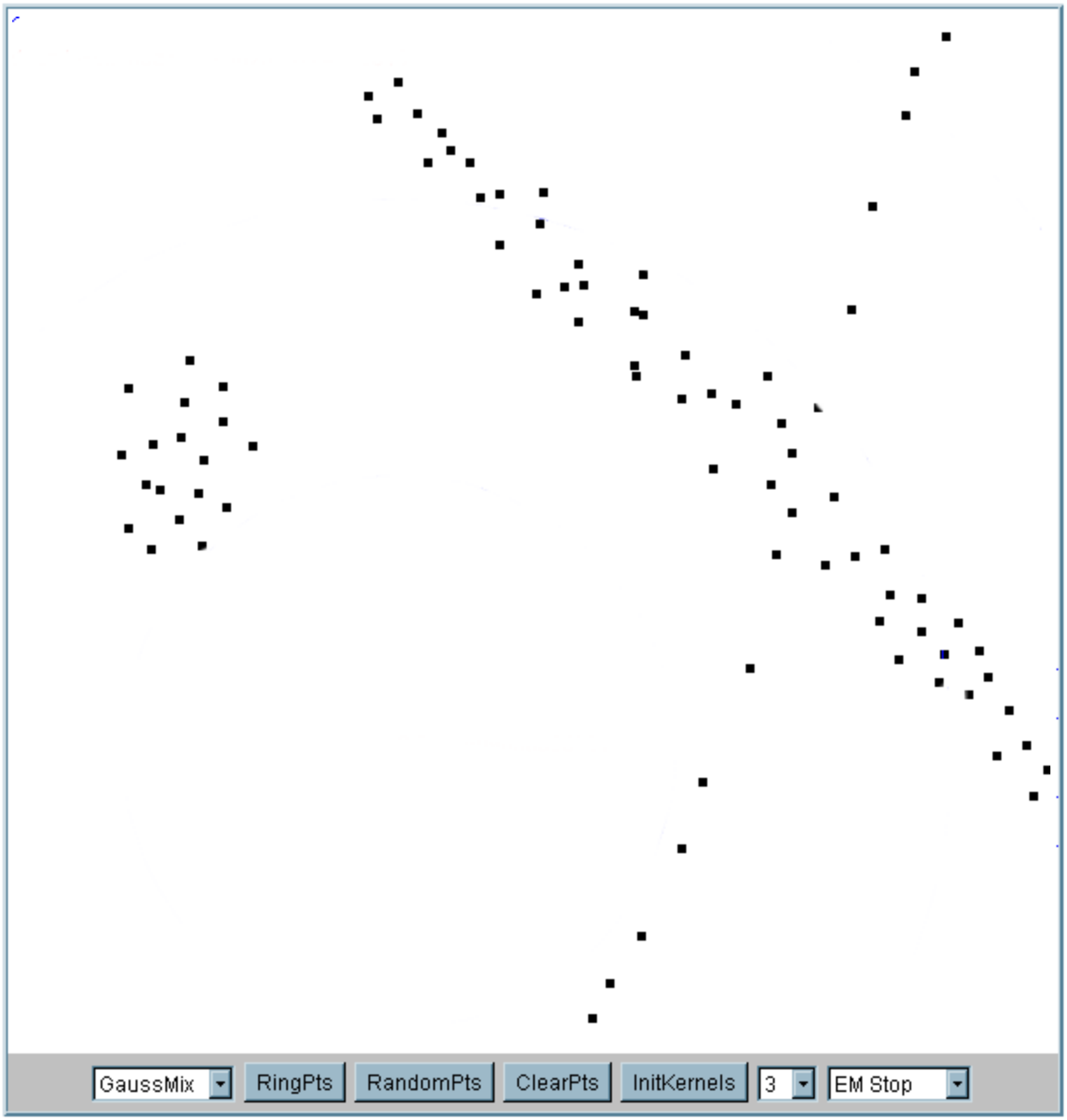
- Nehme an, Parameter sind bekannt
- Weise Daten zu



M-Schritt

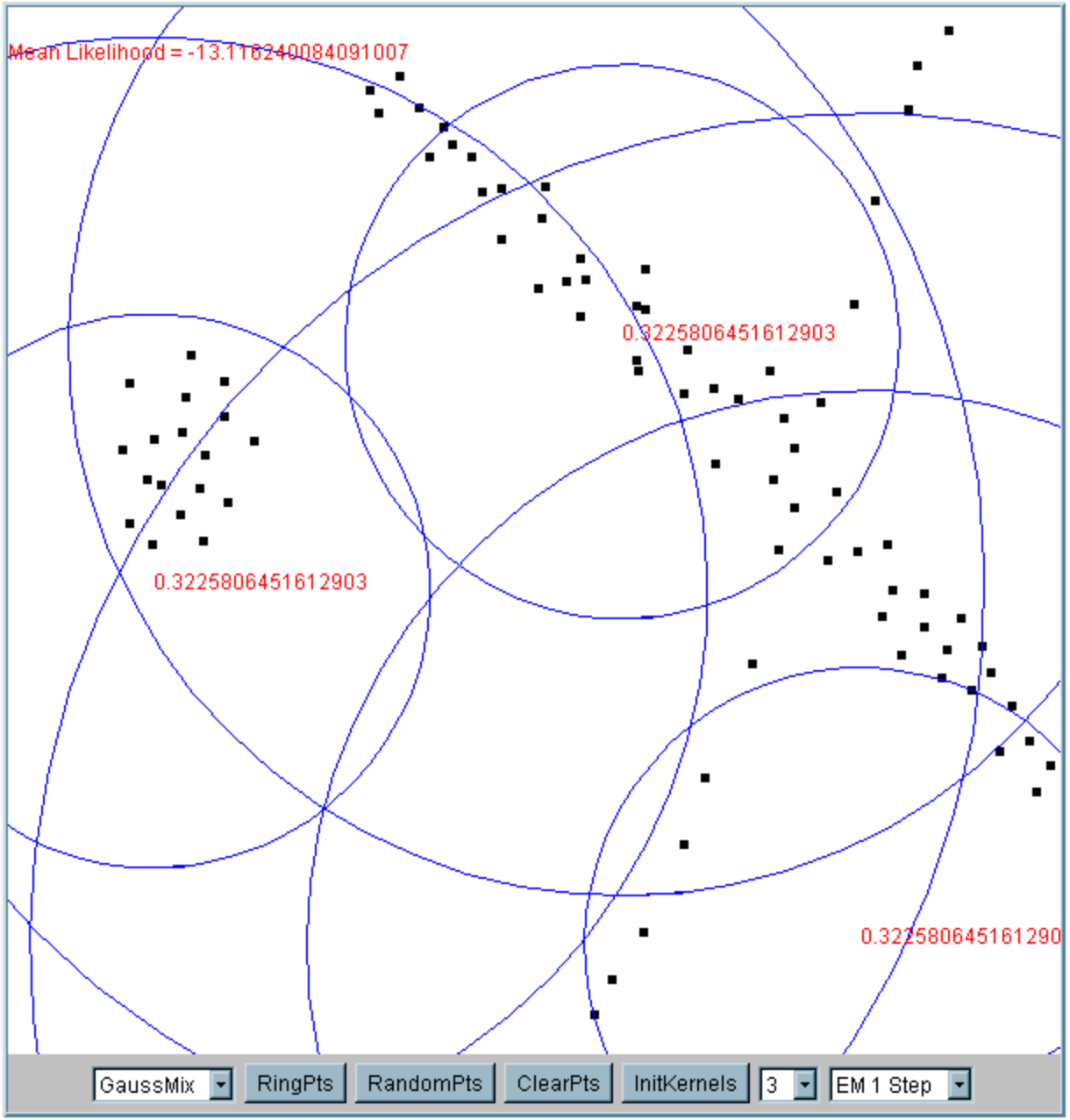
- Passe Parameter über zugeordnete Punktmenge an



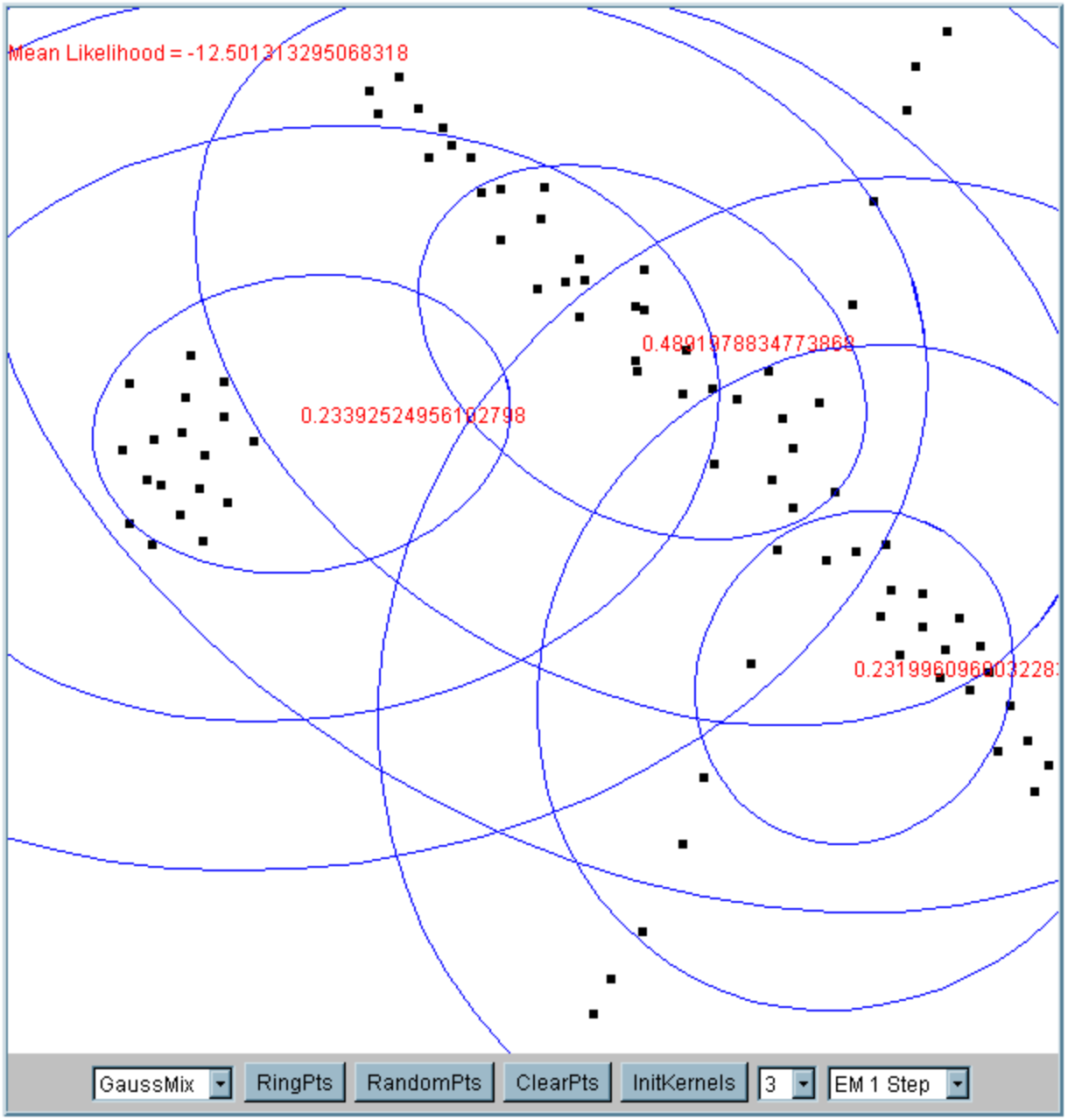

GaussMix
RingPts
RandomPts
ClearPts
InitKernels
3
EM Stop

Iteration 1

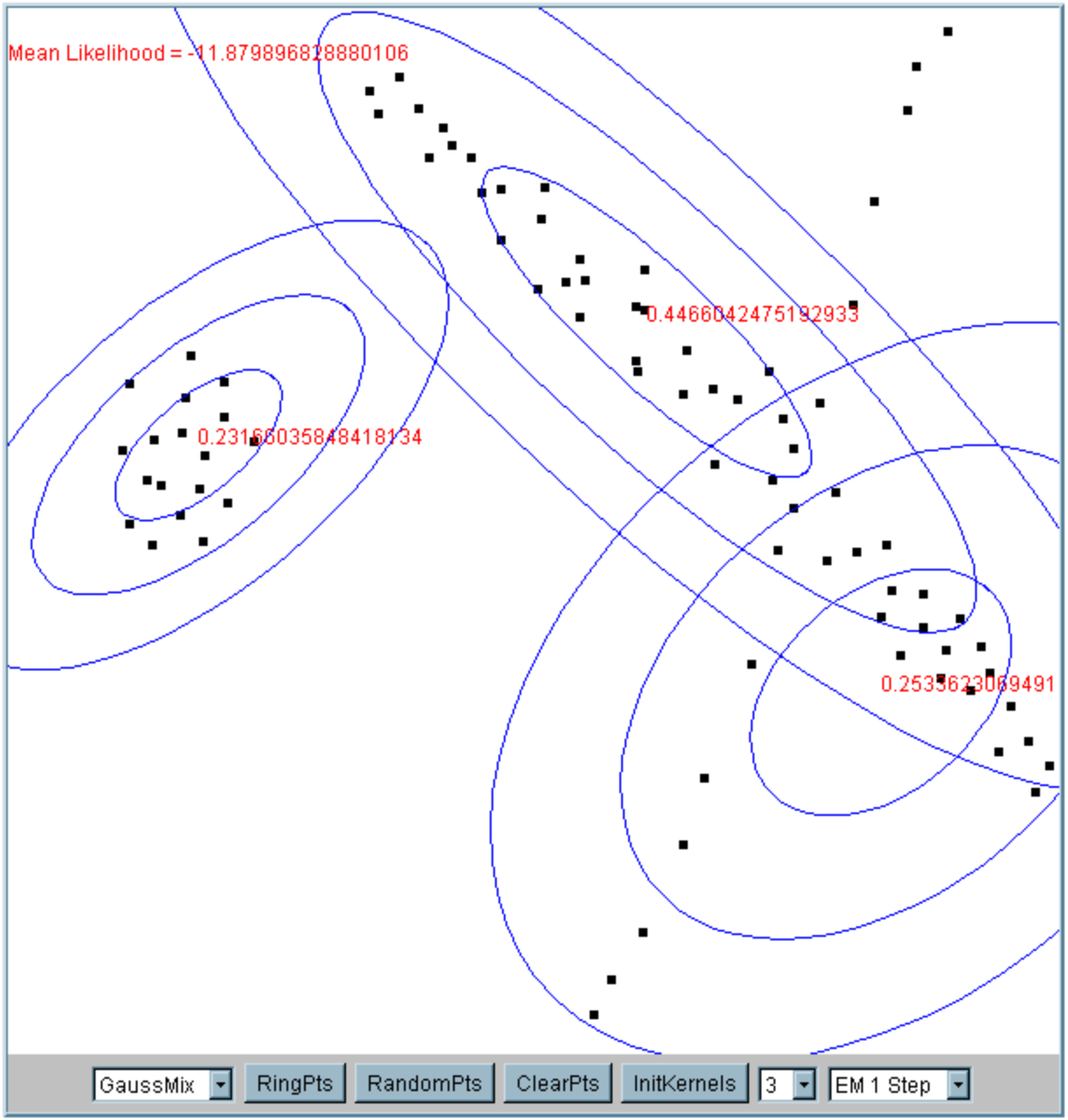
Die Cluster-Mittelwerte werden zufällig gewählt



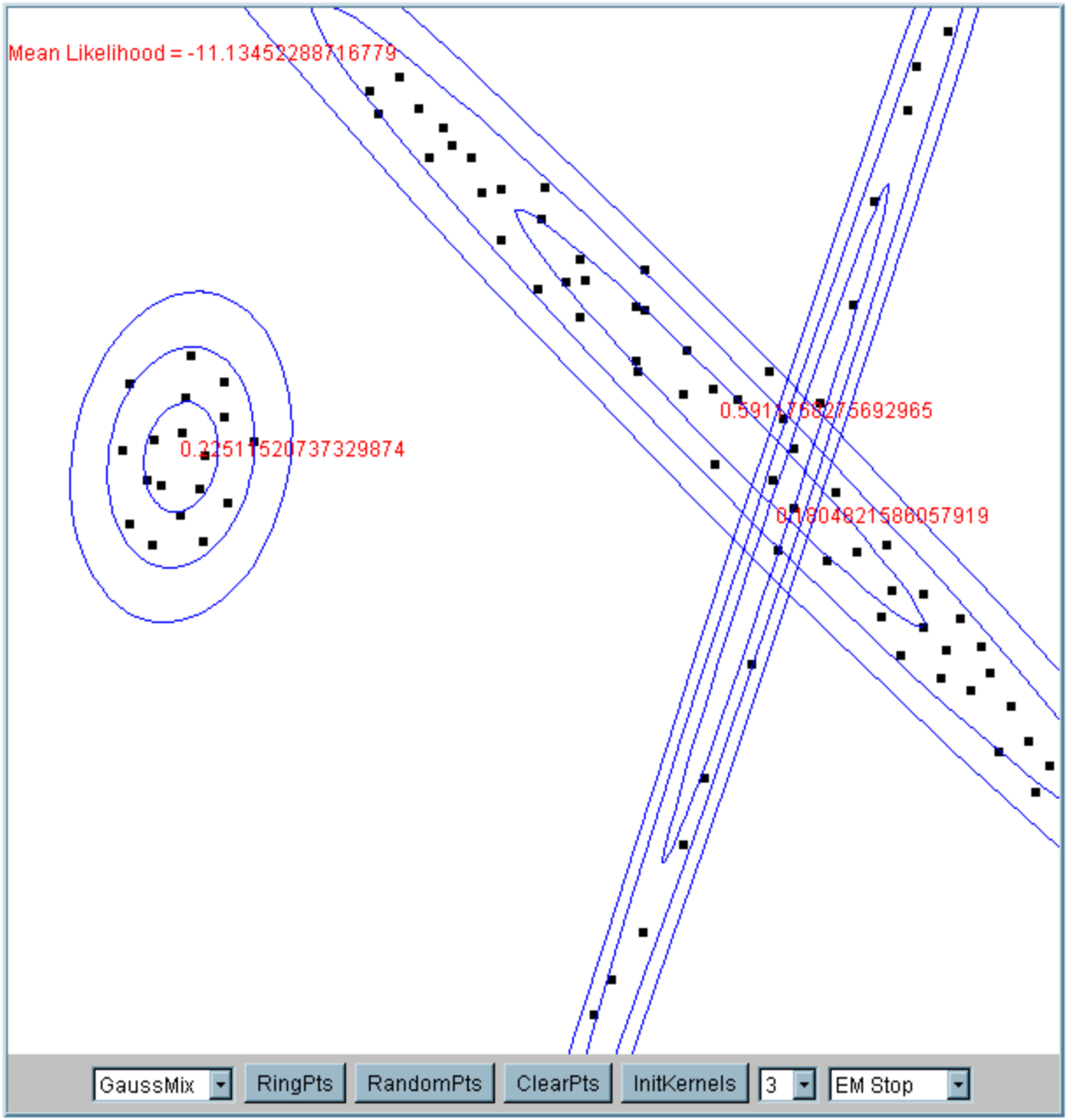
Iteration 2


Mean Likelihood = -12.501313295068318
0.48918788834773868
0.233925249561102798
0.2319960960032283
GaussMix
RingPts
RandomPts
ClearPts
InitKernels
3
EM 1 Step

Iteration 5

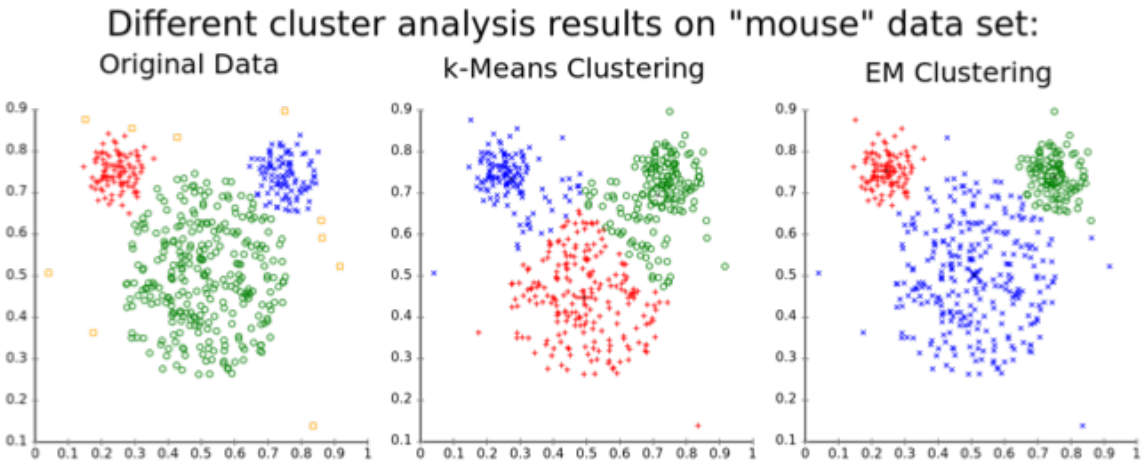

Mean Likelihood = -11.8798968288880106
0.4466042475192933
0.2316803584841813 4
0.2535623069491
GaussMix
RingPts
RandomPts
ClearPts
InitKernels
3
EM 1 Step

Iteration 25



Diskussion EM

- Bestimmung einer Mixtur von multivariaten Gauss-Kurven (Gaussian mixture)
  - K-Means ist spezielle Form des EM-Verfahrens
- Wahrscheinlichkeitsbasierte Zuordnung zu Clustern anstelle einer deterministischen Zuordnung
  - Cluster können verschiedene Größen haben (Varianz)



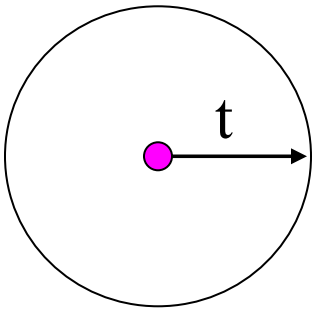
Wikipedia

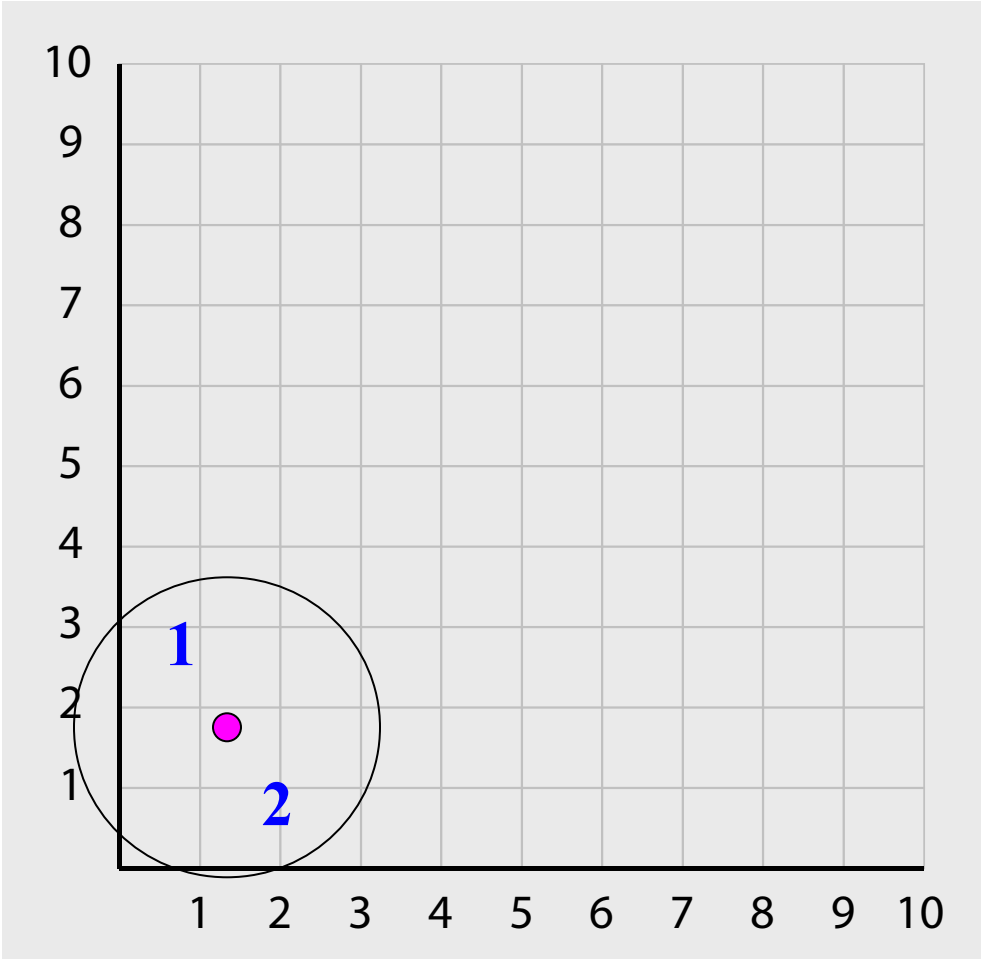
Inkrementelle Clusterbildung

- Nächste-Nachbarn-Clusterbildung
  - Nicht verwechseln mit Nächsten-Nachbarn-Klassifikation
- Neue Datenpunkte inkrementell in bestehende Cluster integriert, so dass Distanz minimiert
- Schwellwert t, um zu bestimmen, ob neues Cluster aufgemacht werden soll

Inkrementelle Clusterbildung

Schwellwert t

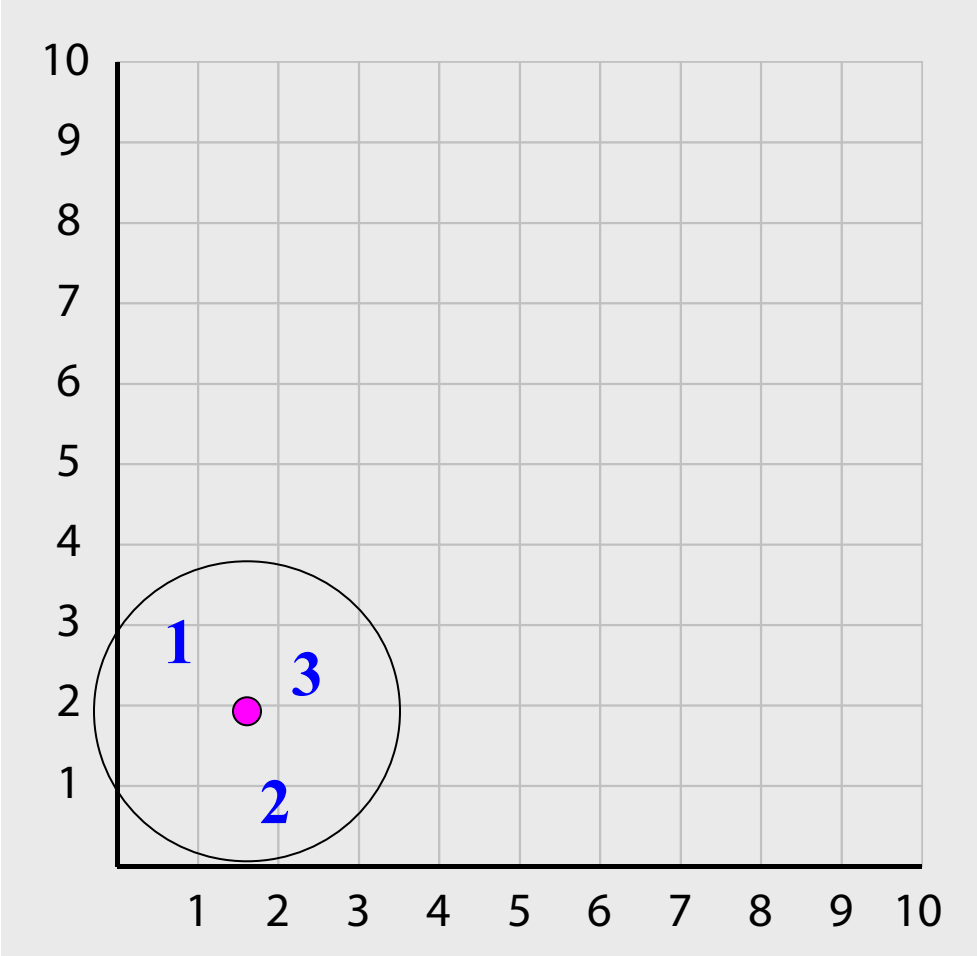




Inkrementelle Clusterbildung

Neuer Datenpunkt …

… ist im Schwellwertbereich des Cluster 1, also fügen wir ihn hin und aktualisieren den Clustermittelpunkt



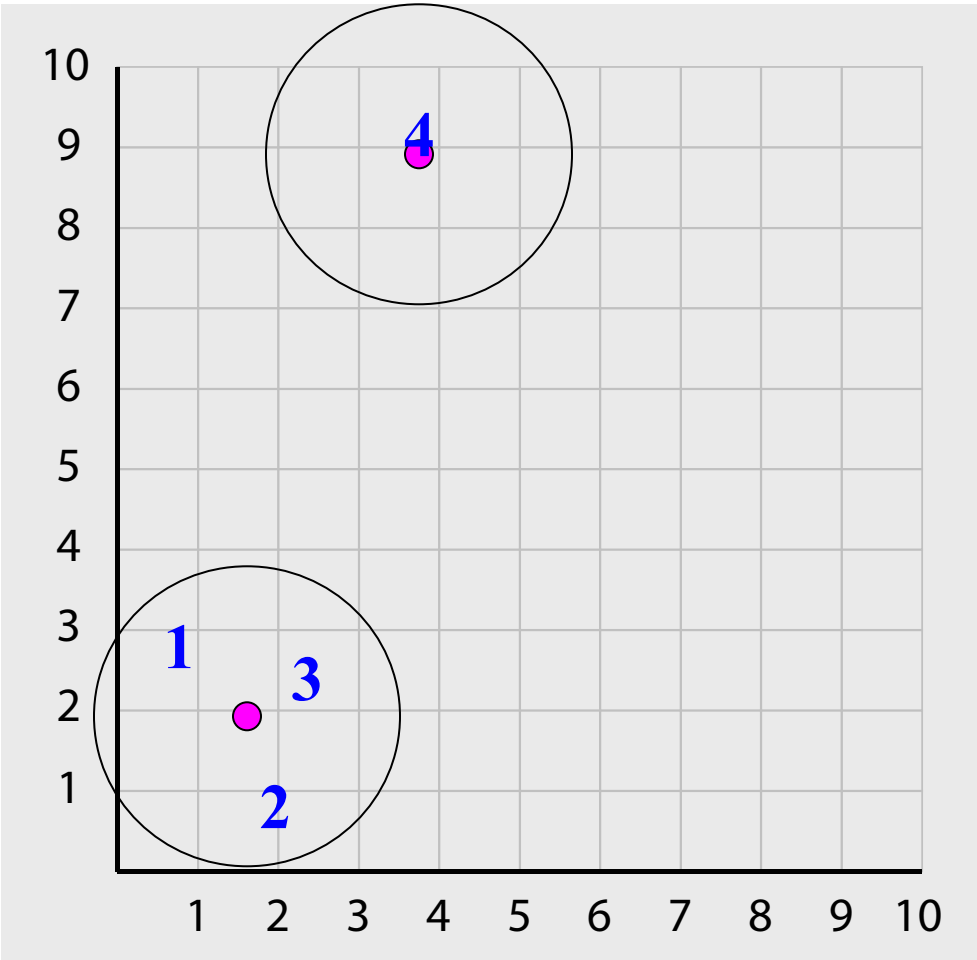
Inkrementelle Clusterbildung

Neuer Datenpunkt …

… ist nicht im Schwerwertbereich von Cluster 1, also erzeugen wir ein neues Cluster, und so weiter ...

Ergebnis des Verfahrens ist klar von der Reichenfolge abhängig…

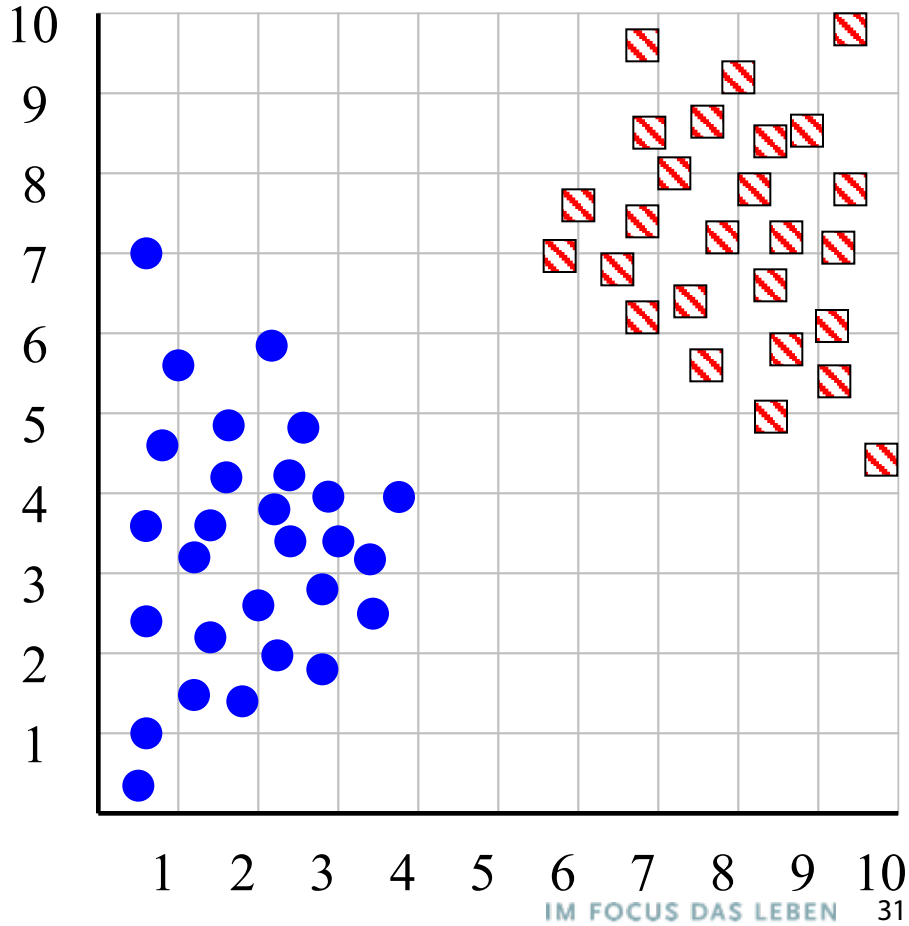
Es ist nicht einfach, den Schwellwert t zu bestimmen …



Was ist die richtige Anzahl von Clustern?

- Offenes Problem
- Viele Approximierungs-methoden
  - z.B. intra-cluster Varianz

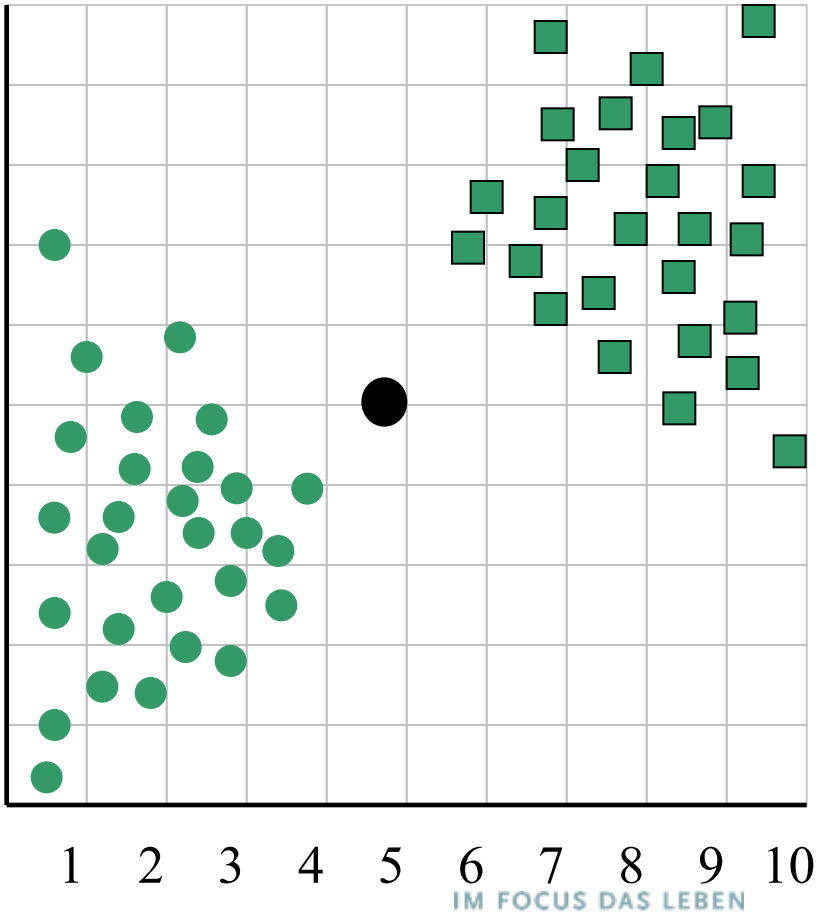
$$E(C_1, \ldots, C_k) = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - c_i\|_2^2$$



Was ist die richtige Anzahl von Clustern?

k = 1: Zielfunktion liefert 873.0

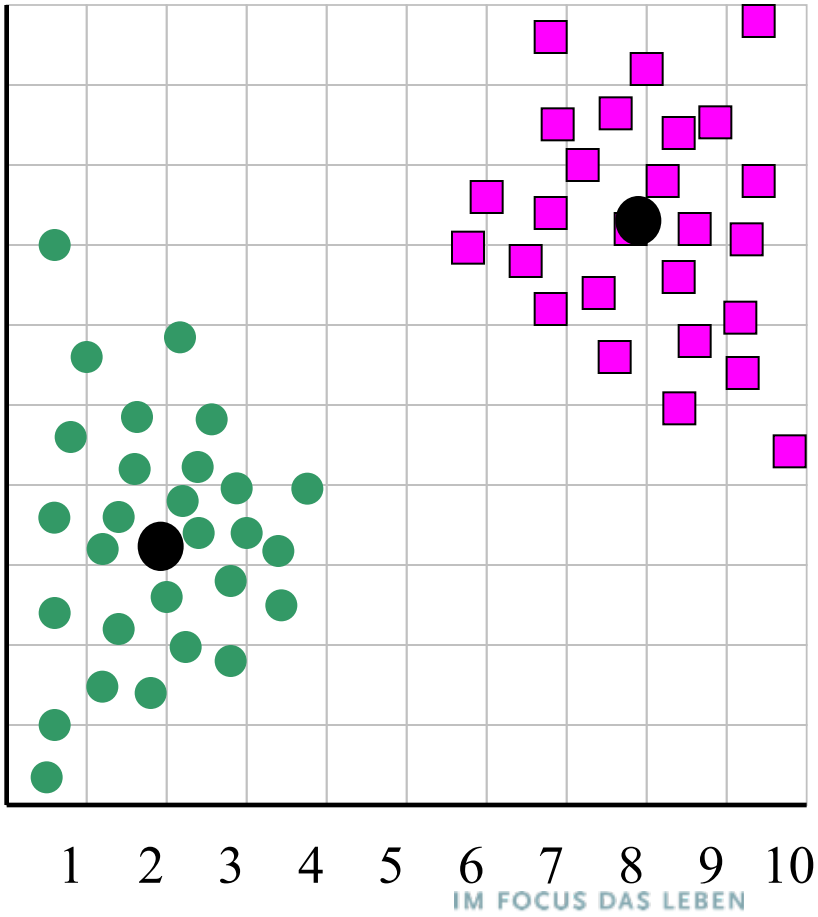
$$E(C_1, \ldots, C_k) = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - c_i\|_2^2$$



Was ist die richtige Anzahl von Clustern?

k = 2: Zielfunktion liefert 173.1

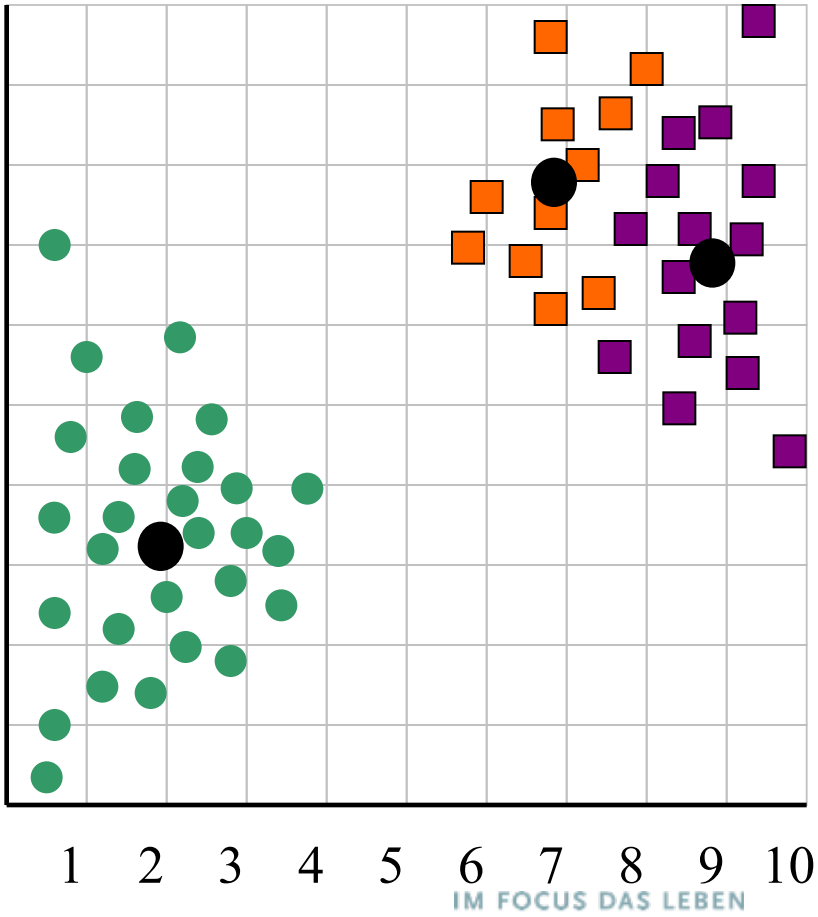
$$E(C_1, \ldots, C_k) = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - c_i\|_2^2$$



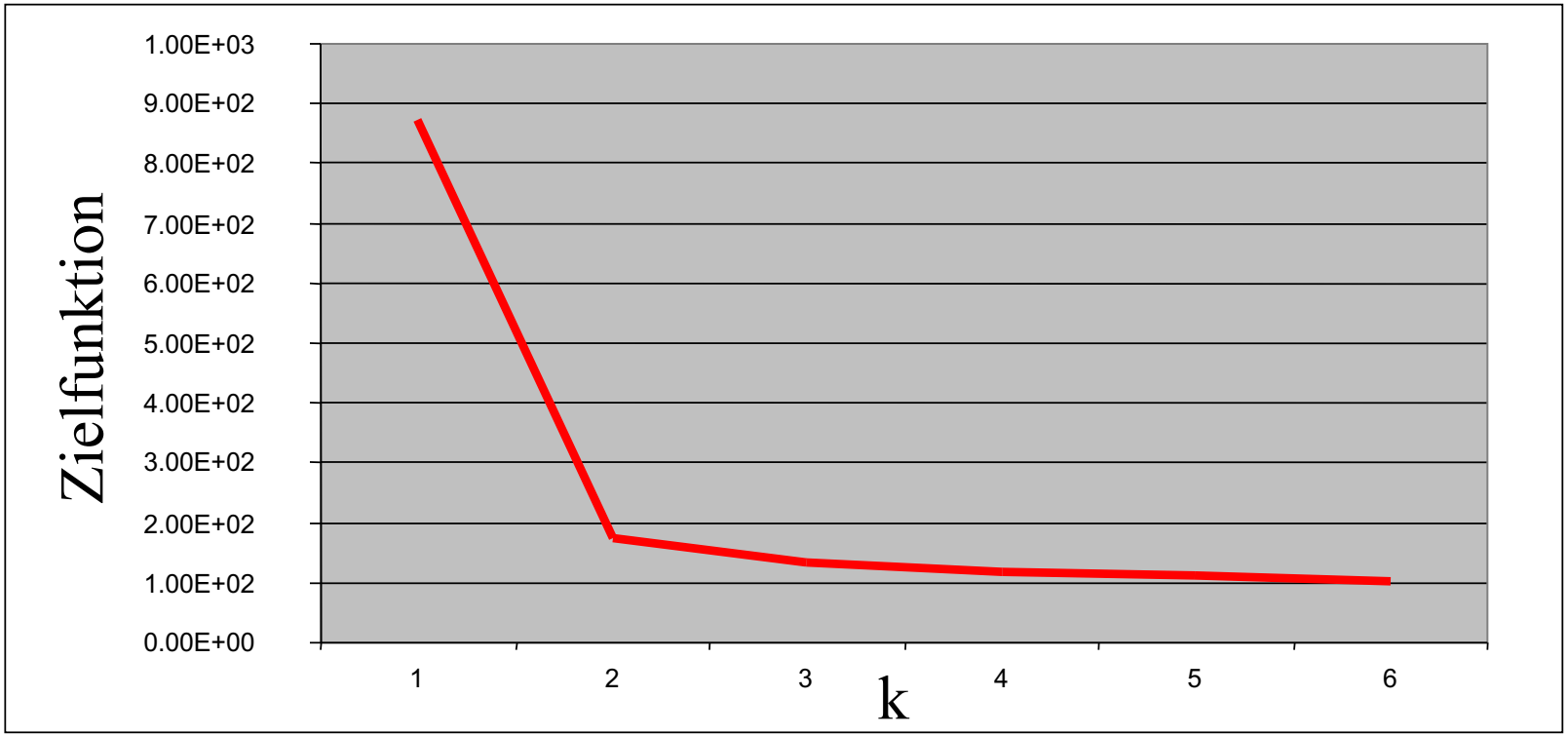
Was ist die richtige Anzahl von Clustern?

k = 3: Zielfunktion liefert 133.6

$$E(C_1, \ldots, C_k) = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - c_i\|_2^2$$



Was ist die richtige Anzahl von Clustern?



Dichtebasierendes partitionierendes Clustering

- DBSCAN-Verfahren (Density Based Spatial Clustering of Applications with Noise)
- Motivation: Punktdichte innerhalb eines Clusters höher als außerhalb des Clusters
- Resultierende Cluster können beliebige Form haben
  - Bei distanzbasierten Methoden ausschließlich konvexe Cluster
- Clusteranzahl k muss nicht initial vorgegeben werden

DBSCAN – Definitionen

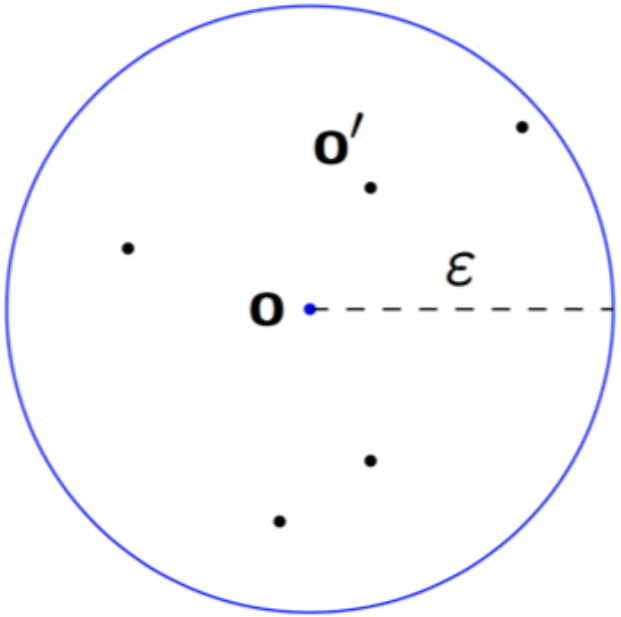
- *ε-Nachbarschaft* eines Objektes $\mathbf{o} \in O$:

$$N_\varepsilon(\mathbf{o}) := \{\mathbf{o}' \in O : d(\mathbf{o}, \mathbf{o}') \leq \varepsilon\}$$

- Aufteilung der Objekte in O
  - $\mathbf{o} \in O$ heißt *Kernobjekt* $:\Longleftrightarrow |N_\varepsilon(\mathbf{o})| \geq m$
  - $\mathbf{o} \in O$ heißt *Randobjekt* $:\Longleftrightarrow$ $\mathbf{o}$ ist kein Kernobjekt
- Parameter $\varepsilon \in \mathbb{R}^+$ und $m \in \mathbb{N}$ müssen initial vorgegeben werden (Heuristik zur Bestimmung der Parameter basierend auf der Dichte des „dünnsten“ Clusters)
- im Folgenden sei core(O) die Menge aller Kernobjekte in O
- in den folgenden Beispielen: $m = 4$

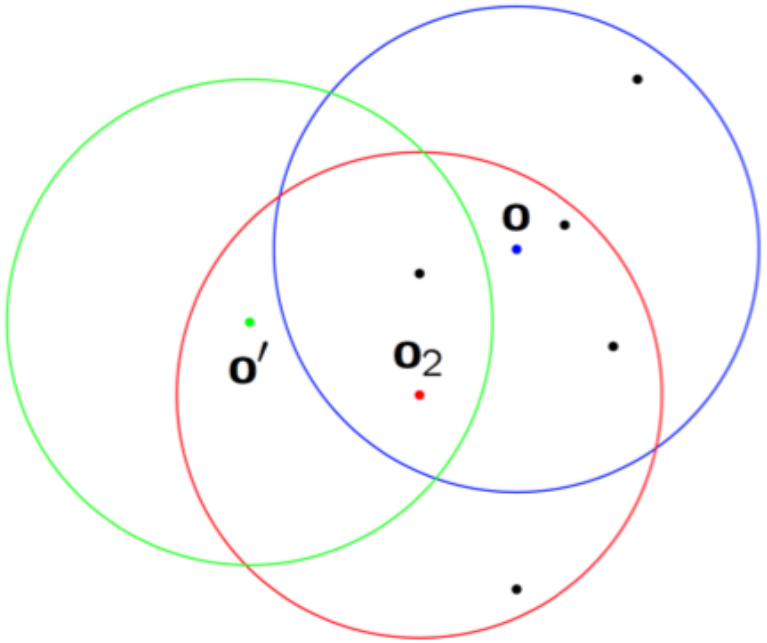
DBSCAN – Definitionen

$\mathbf{o}' \in O$ ist *direkt dichte-erreichbar* von $\mathbf{o} \in O :\Longleftrightarrow$
$\mathbf{o}' \in N_\varepsilon(\mathbf{o}) \wedge \mathbf{o} \in \text{core}(O)$



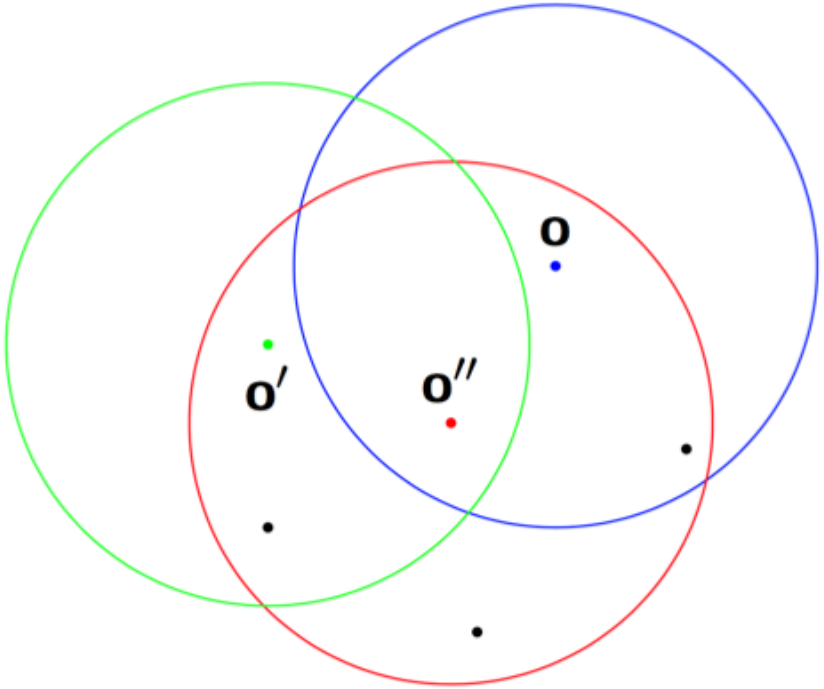
DBSCAN – Definitionen

$\mathbf{o}'$ ist *dichte-erreichbar* von $\mathbf{o}$ $:\Longleftrightarrow$
$\exists \mathbf{o}_1, \ldots, \mathbf{o}_i \in O : \mathbf{o}_1 = \mathbf{o} \wedge \mathbf{o}_i = \mathbf{o}' \wedge \forall j \in \{1, \ldots, i-1\} :$
$\mathbf{o}_{j+1}$ direkt dichte-erreichbar von $\mathbf{o}_j$



DBSCAN – Definitionen

$\mathbf{o}, \mathbf{o}' \in O$ sind *dichte-verbunden* $:\Longleftrightarrow \exists \mathbf{o}'' \in O$: $\mathbf{o}$ und $\mathbf{o}'$ sind von $\mathbf{o}''$ aus dichte-erreichbar



DBSCAN – Definitionen

Ein *Cluster* C ist eine nichtleere Teilmenge von O, die folgende Bedingungen erfüllt:

1. $\forall \mathbf{o}, \mathbf{o}' \in O$: ist $\mathbf{o} \in C$ und $\mathbf{o}'$ dichte-erreichbar von $\mathbf{o}$, dann ist $\mathbf{o}' \in C$ (Maximalität)
2. $\forall \mathbf{o}, \mathbf{o}' \in C$: $\mathbf{o}$ ist dichte-verbunden mit $\mathbf{o}'$ (Konnektivität)

Seien $C_1, \ldots, C_k$ Cluster bezüglich der Parameter (ε_i, m_i) mit $1 \leq i \leq k$. Dann ist die Menge N (*noise*) definiert als:

$$N := \{\mathbf{o} \in O : \forall i \in \{1, \ldots, k\}(\mathbf{o} \notin C_i)\}$$

N enthält also die Punkte, die keinem Cluster zugeordnet sind.

DBSCAN – Lemma 1

Lemma 1

Sei $\mathbf{o} \in \text{core}(O)$, dann ist die Menge
$\{\mathbf{o}' \in O : \mathbf{o}' \text{ ist dichte-erreichbar von } \mathbf{o}\}$ ein Cluster.

Bestimmung eines Clusters C in zwei Schritten

1. wähle einen beliebigen Punkt $\mathbf{o} \in \text{core}(O)$
2. ermittle die Menge P aller Objekte, die von $\mathbf{o}$ aus dichte-erreichbar sind

Dann ist $C = P \cup \{\mathbf{o}\}$.

DBSCAN – Lemma 2

Lemma 2

Sei C ein Cluster und $\mathbf{o} \in C$ ein Kernobjekt. Dann gilt folgende Gleichung

$$C = \{\mathbf{o}' \in O : \mathbf{o}' \text{ ist dichte-erreichbar von } \mathbf{o}\}.$$

Damit folgt, dass ein Cluster durch *jedes* beliebige seiner Kernobjekte eindeutig bestimmt ist.

DBSCAN

Eingabe: O, ε, m
Ausgabe: Funktion $c : O \to \mathbb{N}$, die jedem Objekt eine Clusternummer zuordnet

$c\_id := 1$ // -1: unclassified, -2: noise
$\forall \mathbf{o} \in O : c(\mathbf{o}) := -1$
$\forall \mathbf{o} \in O$ do
  if $c(\mathbf{o}) = -1$ then
    if ExpandCluster$(O, \mathbf{o}, c_id, \varepsilon, m)$ then
      $c\_id := c\_id + 1$
    fi
  fi
od

ExpandCluster

Eingabe: $O, \mathbf{o} \in O, c\_id, \varepsilon, m$
Ausgabe: Wahrheitswert true oder false

$S := neighborhood(O, \mathbf{o}, \varepsilon)$
if $|S| < m$ then // $\mathbf{o}$ ist ein Randobjekt
  $c(\mathbf{o}) := -2$
  return false
else // $\mathbf{o}$ ist ein Kernobiekt
  // bestimme alle Objekte, die von $\mathbf{o}$ aus dichte-erreichbar sind
  $\forall \mathbf{o}' \in S : c(\mathbf{o}') := c\_id$
  $S := S - \{\mathbf{o}\}$
  while $S \neq \emptyset$ do
    $\mathbf{o}' := S.getElement()$
    $R := neighborhood(O, \mathbf{o}', \varepsilon)$
    if $|R| \geq m$ then
      $\forall \mathbf{o}'' \in R$ do
        if $c(\mathbf{o}'') \in \{-1, -2\}$ then
          if $c(\mathbf{o}'') = -1$ then
            $S := S \cup \{\mathbf{o}''\}$
          endif
          $c(\mathbf{o}'') := c\_id$
        fi
      od
    fi
    $S := S - \{\mathbf{o}'\}$
  od
  return true
fi