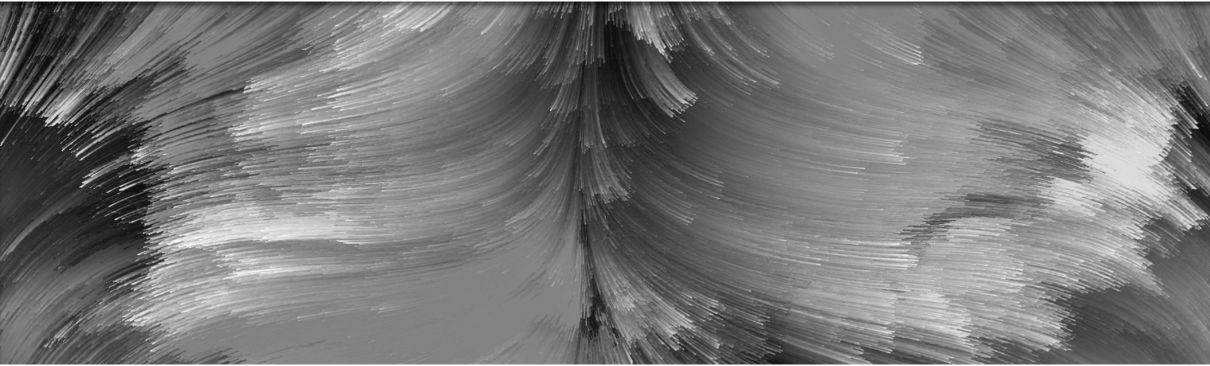


Künstliche Intelligenz (Sommersemester 2024)

Kapitel 05: Clustering

Prof. Dr. Adrian Ulges

1. ML Methoden durch Lernsignale

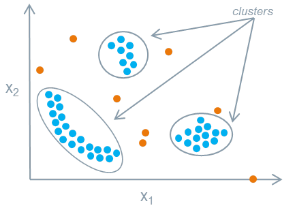
Was ist das Lernsignal?

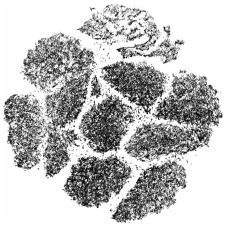
	supervised (dt. „überwachtes Lernen“)	semi-supervised (dt. „halbüberwacht“)	self-supervised (dt. „selbstüberwacht“)	unsupervised (dt. „unüberwacht“)	reinforcement learning (dt. „verstärkendes Lernen“)
Trainings-daten	Labels sind **bekannt** $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$	**Einige** Labels bekannt $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m), x_{m+1}, x_{m+2}, \ldots, x_n$	Labels sind **unbekannt** $\mathbf{x}_1, \ldots, \mathbf{x}_n$	Labels sind **unbekannt** $\mathbf{x}_1, \ldots, \mathbf{x}_n$	Eine Umgebung gibt eine (verzögerte) **Belohnung (engl. „reward“) R**
Ziel-setzung	Lerne eine Abbildung von Input zu Labels $f_\theta : \mathbf{x} \mapsto \hat{y}$	*siehe „überwacht“*	Definiere **Pseudo-Labels** y' für ein überwachtes „Hilfsproblem“ $f_\theta : \mathbf{x} \mapsto \hat{y}'$	Lerne eine Struktur / Repräsentation der Daten **x'**, z.B. $f_\theta : \mathbf{x} \mapsto \mathbf{x}'$,	Lerne das Verhalten eines Agenten (engl. **“Policy“**): Wenn in Zustand s, wähle Aktion a: $f_\theta : s \mapsto a$
Loss ℓ	Basiert auf Label-Vergleich $\ell(\hat{y}_i, y_i)$	Basiert auf Labels und Goodness-of-fit $\ell(\hat{y}_i, y_i, \mathbf{x}_i, \mathbf{x}'_i)$	Basiert auf **Pseudo-Labels** $\ell(\hat{y}'_i, y'_i)$	Basiert auf **Goodness-of-fit**, z.B. $\ell(\mathbf{x}_i, \mathbf{x}'_i)$	Basiert auf Reward $\sum_j R_j$

- Bisher: Überwachtes Lernen.
- Jetzt: Unüberwachtes Lernen.

Unüberwachtes Lernen = Lernen ohne Labels Bilder: [2], [1]

- Clustering: Entdecken kohärenter Gruppen von Proben
- Anomalieerkennung: Erkennen von Ausreißern / ungewöhnlichen Proben
- Dimensionsreduktion: Komprimierung von Proben
- Itemset-Mining: Finden häufiger Unterstrukturen in den Daten



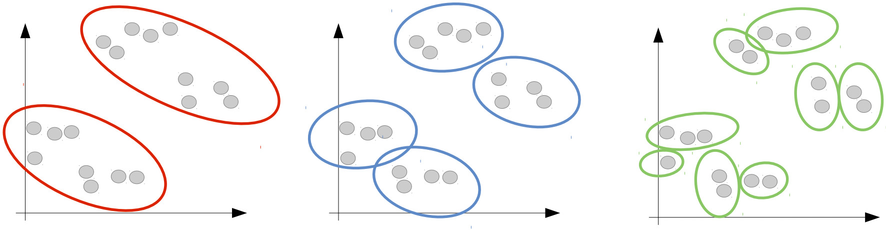


Clustering: Definition

Clustering = Entdeckung kohärenter Untergruppen (*Cluster*) in Stichproben.

Bemerkungen

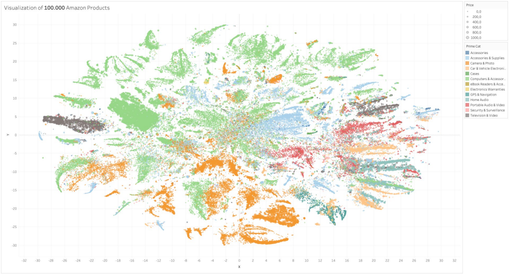
- **Herausforderung 1**: "Klassen" sind ($\neq$ überwachter Klassifikation) unbekannt.
- **Herausforderung 2**: Die Granularität der Cluster ist a priori unklar.

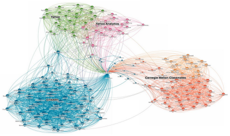


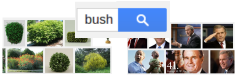
Clustering: Anwendungen Bilder: [4], [3]

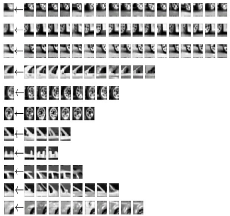
Clustering hat zahlreiche Anwendungen in verschiedenen Bereichen:

- Marktforschung
- Information Retrieval
- Computer Vision
- Soziale Netzwerke
- ...









Outline

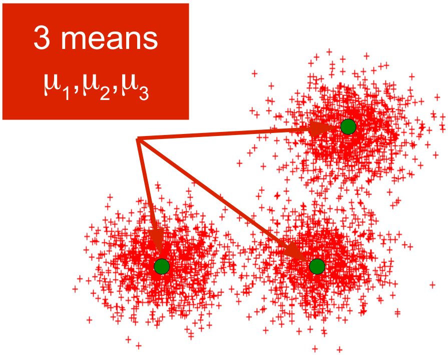
1. K-Means

2. Model Selection

Clustering: K-Means

K-Means ist ein gängiger "First-Choice"-Algorithmus für Clustering:

- Gegeben: Trainingsmenge $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$.
- Wir nehmen an, dass sich die Objekte um K unbekannte Zentren *(die "K Means")* $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K \in \mathbb{R}^d$ herum gruppieren.
- Jedes Objekt $\mathbf{x}_i$ gehört zu einem Zentrum $\boldsymbol{\mu}_{k(i)}$.
- Grundannahme: Die Cluster sind Hypersphären von identischer Größe.



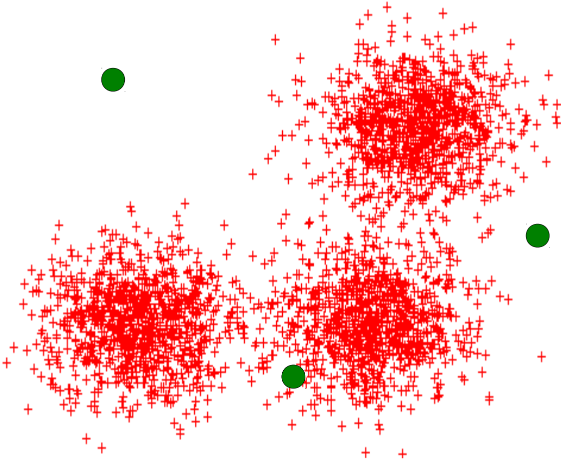
K-Means: Ansatz

Wir stehen vor einem Henne-Ei-Problem:

- Wenn wir die Zentren $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K$ kennen würden, könnten wir die Cluster bestimmen *(indem wir jedes Objekt seinem nächstgelegenen Zentrum zuweisen)*.
- Wenn wir die Clusterzugehörigkeiten $k(i)$ kennen würden, könnten wir leicht die Zentren bestimmen *(indem wir alle Objekte eines Zentrums mitteln)*.
- Ansatz (alternierende Optimierung): Wir fixieren abwechselnd die Clusterzugehörigkeiten/Zentren und schätzen die Zentren/Clusterzugehörigkeiten:

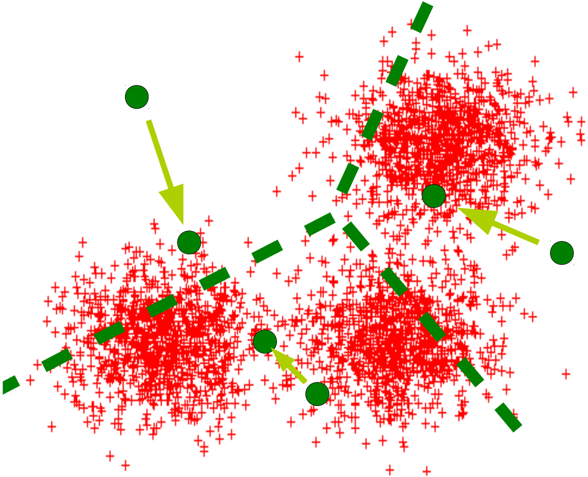
```
function KMEANS($\mathbf{x}_1, ..., \mathbf{x}_n$, $K$)
    initialize $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K$ by random sampling from $\mathbf{x}_1, ..., \mathbf{x}_n$
    repeat
        for $i = 1, .., n$:          // assign samples to the closest center
            $k(i) := \arg\min_{k=1,...,K} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|$
        for $k = 1, ...K$:           // re-estimate each cluster's center
            $X_k := \{\mathbf{x}_i \mid k(i) = k\}$
            $\boldsymbol{\mu}_k := \frac{1}{|X_k|} \sum_{\mathbf{x} \in X_k} \mathbf{x}$
    until $k(1), ..., k(n)$ do not change
    return $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K$
```

Beispiel: Schritt 0



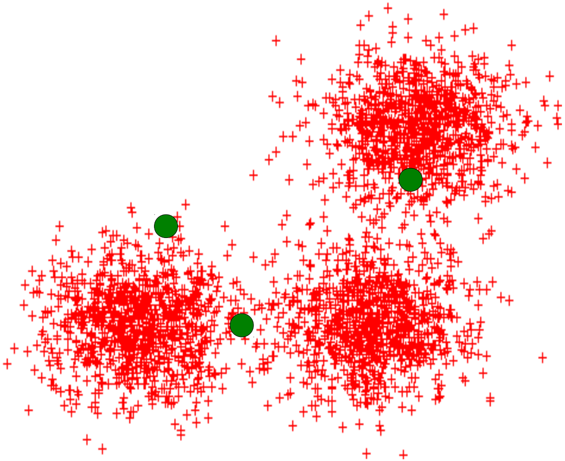
function KMEANS($\mathbf{x}_1, \dots, \mathbf{x}_n$, K)
  initialize $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K$ by random sampling from $\mathbf{x}_1, \dots, \mathbf{x}_n$
  repeat
    for $i = 1, .., n$:   // assign samples to the closest mean
      $k(i) := \arg\min_{k=1,\dots,K} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|$
    for $k = 1, \dots K$:   // re-estimate each cluster's mean
      $X_k := \{\mathbf{x}_i \mid k(i) = k\}$
      $\boldsymbol{\mu}_k := \frac{1}{|X_k|} \sum_{\mathbf{x} \in X_k} \mathbf{x}$
  until $k(1), \dots, k(n)$ do not change
  return $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K$

Beispiel: Schritt 1

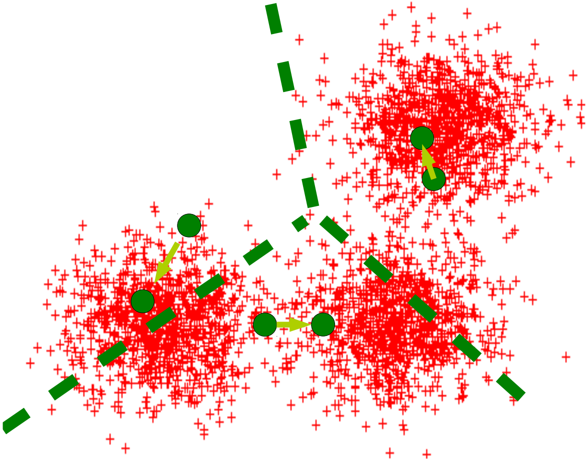


```
function KMEANS(x_1,...,x_n, K)
    initialize μ_1,...,μ_K by random sampling from x_1,...,x_n
    repeat
        for i = 1,..,n:        // assign samples to the closest mean
            k(i) := arg min_{k=1,...,K} ||x_i − μ_k||
        for k = 1,...K:        // re-estimate each cluster's mean
            X_k := {x_i | k(i) = k}
            μ_k := 1/|X_k| Σ_{x∈X_k} x
    until k(1),...,k(n) do not change
    return μ_1,...,μ_K
```

Beispiel: Schritt 1 abgeschlossen...

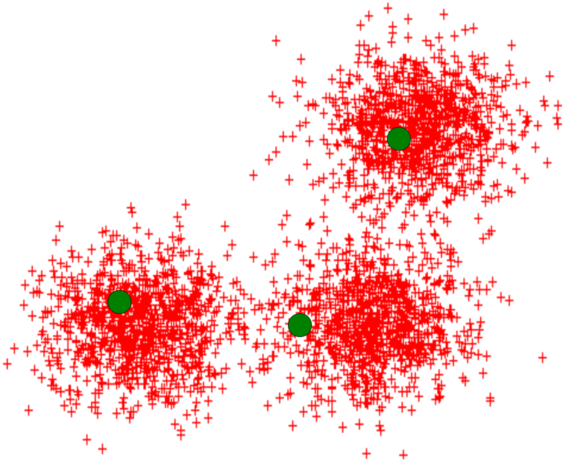


Beispiel: Schritt 2

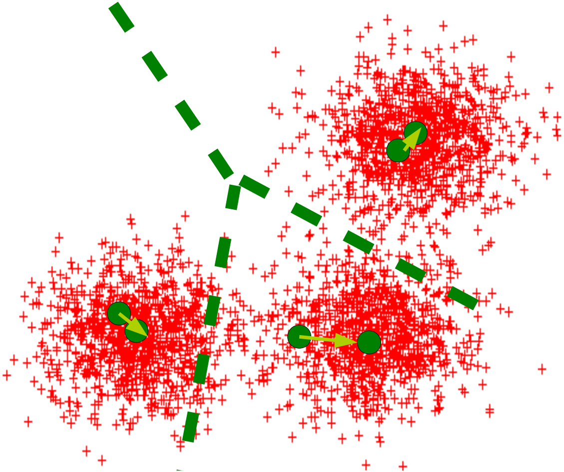


```
function KMEANS(x_1, ..., x_n, K)
    initialize μ_1, ..., μ_K by random sampling from x_1, ..., x_n
    repeat
        for i = 1, .., n:        // assign samples to the closest mean
            k(i) := arg min_{k=1,...,K} ||x_i − μ_k||
        for k = 1, ...K:         // re-estimate each cluster's mean
            X_k := {x_i | k(i) = k}
            μ_k := 1/|X_k| Σ_{x∈X_k} x
    until k(1), ..., k(n) do not change
    return μ_1, ..., μ_K
```

Beispiel: Schritt 2 abgeschlossen...

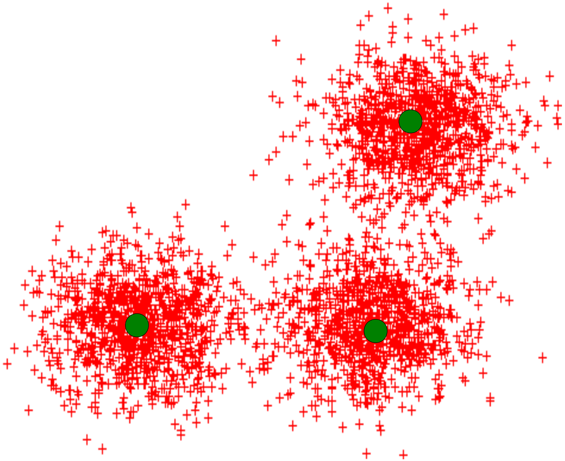


Beispiel: Schritt 3



```
function KMEANS(x_1, ..., x_n, K)
    initialize μ_1, ..., μ_K by random sampling from x_1, ..., x_n
    repeat
        for i = 1, .., n:        // assign samples to the closest mean
            k(i) := arg min_{k=1,...,K} ‖x_i − μ_k‖
        for k = 1, ...K:         // re-estimate each cluster's mean
            X_k := {x_i | k(i) = k}
            μ_k := 1/|X_k| Σ_{x∈X_k} x
    until k(1), ..., k(n) do not change
    return μ_1, ..., μ_K
```

Beispiel: Endergebnis



K-Means: Eigenschaften

Effizienz?
Die Zeitkomplexität beträgt $O(K \cdot n \cdot d)$ pro Iteration. Oft konvergieren wir in < 100 Iterationen. ☺

Konvergenz?
K-Means minimiert die Summe der quadratischen Fehler $\mathcal{L}$:

$$\mathcal{L}(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K) = \sum_{i=1}^{n} \left\| \mathbf{x}_i - \boldsymbol{\mu}_{k(i)} \right\|^2$$

...und Konvergenz ist garantiert!

Beweis

→ Die Fehler $\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_2, \ldots$ nach jedem Schritt nehmen monoton ab.
Außerdem haben die Fehler eine untere Grenze: $\mathcal{L}_k \geq 0$ für alle k.

→ Die Sequenz der Fehler $\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_2, \ldots$ konvergiert.

→ $\mathcal{L}$ hört irgendwann auf sich zu ändern (daher stoppt der Algorithmus).

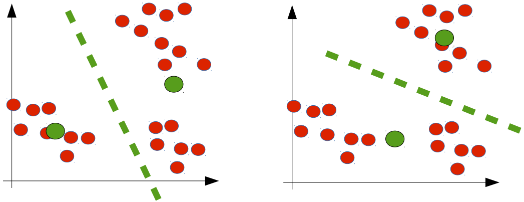
K-Means: Eigenschaften (cont'd)

Führt K-Means bei gleichen Eingabedaten immer zum gleichen Ergebnis?
Nein: Es handelt sich um ein lokales Suchverfahren!

- **Grund 1**: Die Reihenfolge der Zentren kann vertauscht sein:

$$\boldsymbol{\mu}_1 = (0,0), \boldsymbol{\mu}_2 = (1,1), \boldsymbol{\mu}_3 = (5,3)$$
$$\boldsymbol{\mu}_1 = (5,3), \boldsymbol{\mu}_2 = (0,0), \boldsymbol{\mu}_3 = (1,1)$$

- **Grund 2**: Die resultierenden Zentren können komplett unterschiedlich sein:



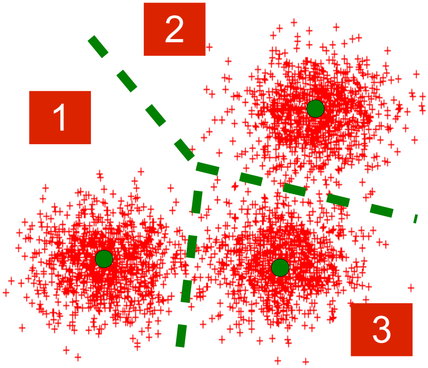
Vorgehensweise

Wir starten K-Means mehrmals neu mit verschiedenen Initialisierungen und behalten das Ergebnis mit minimalem Fehler $\mathcal{L}$.

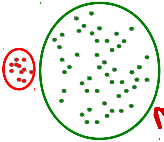
K-Means: Eigenschaften (Fortsetzung)

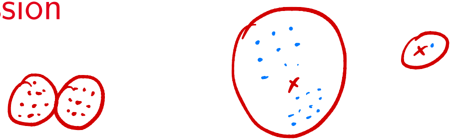
Gegeben ein Clustering-Ergebnis $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$, können wir neue Datenpunkte $\mathbf{x}$ dem passendsten Cluster zuweisen *(dies wird als Vektorquantisierung bezeichnet)*:

$$k(\mathbf{x}) = \arg\min_k ||\mathbf{x} - \boldsymbol{\mu}_k||$$

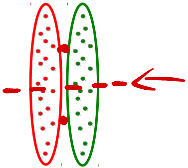


K-Means: Diskussion

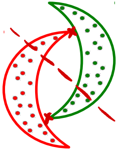




unterschiedliche Varianz ↯



anisotrope Cluster ↯



nicht normalverteilte Cluster ↯

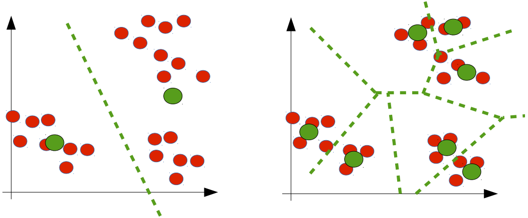
Outline

Wahl von K: Model Selection

> *"Model selection is the task of selecting a statistical model from a set of candidate models, given data."*
>
> (en.wikipedia.org)

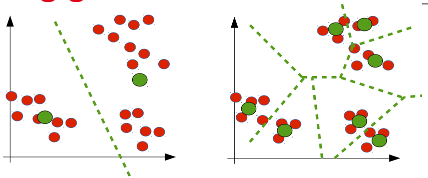
Hier: Model Selection = Wahl der Clusteranzahl K

- K zu klein (*Untersegmentierung*): Cluster sind zu heterogen.
- K zu hoch (*Übersegmentierung*): zu viele Parameter, Cluster zu feingranular.
- Die Auswahl des 'falschen' K führt auch zu **instabilen** Ergebnissen.



Ein Ansatz für Model Selection: BIC

Ziel: Messung der **Anpassungsgüte eines Modells** ohne Labels



Beispiel: Das **Bayes'sche Informationskriterium (BIC)**

1. Die Cluster sollten kompakt sein *(geringer Fehler $\mathcal{L}$)*.
2. Das Modell sollte einfach sein, d.h. nur wenige Parameter haben.

Länge der Vektoren

- Sei $\#\theta$ die Anzahl der Parameter *(hier: $\#\theta = K \cdot d$)*.

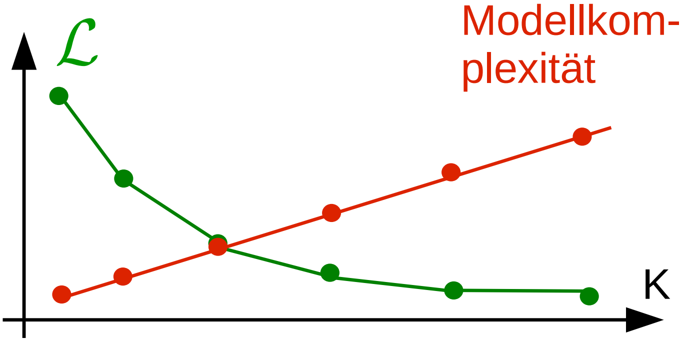
Klusterzentren

- Wir testen verschiedene Werte von K und wählen den besten:

$$K^* = \arg\min_K \underbrace{\sum_{i=1}^{n} \left(\mathbf{x}_i - \boldsymbol{\mu}_{k(i)}\right)^2}_{\mathcal{L}(K)} + \underbrace{\#\theta \cdot \ln(n)}_{\text{Modellkomplexität}}$$

Das Bayes-Informationskriterium

$$K^* = \arg\min_K \underbrace{\sum_{i=1}^{n} \left(\mathbf{x}_i - \boldsymbol{\mu}_{k(i)}\right)^2}_{\mathcal{L}(K)} + \underbrace{\#\theta \cdot ln(n)}_{\text{Modellkomplexität}}$$



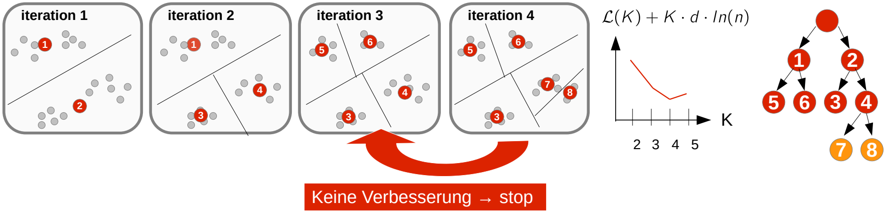
Auswahl von K: Suchstrategien

Es ist sehr teuer, verschiedene Werte von K zu testen:
Für jeden müssten wir eine vollständige Clusteranalyse durchführen!

Verbesserung: **Hierarchisches Clustering** *(effizienter)*

1. ... wähle das größte Cluster aus.
2. ... wende K-Means auf die Objekte des Clusters an und erhalte K neue Cluster.
3. ... wenn die Qualität *(d.h., BIC)* nicht mehr verbessert wird, breche ab.
4. ... gehe zu Schritt 1.

Wir erhalten einen Clusterbaum.



References I

[1] M. Dukia: DoS(Denial of Service) Attacks.
http://www.securitykiller.org/2015/12/dosdenial-of-service-attacks.html (retrieved: Nov 2016).

[2] P. Ipeirotis: A Plea to Amazon: Fix Mechanical Turk! .
http://www.behind-the-enemy-lines.com/2010/10/plea-to-amazon-fix-mechanical-turk.html (retrieved: Nov 2016).

[3] The Value of a Professional Network?
https://www.linkedin.com/pulse/value-professional-network-daniel-tunkelang (retrieved: Oct 2016).

[4] University of Vermont: Complex Networks (Course Page).
http://www.uvm.edu/pdodds/teaching/courses/2010-01UVM-303/content/pictures.html (retrieved: Nov 2016).