Wissensentdeckung in Datenbanken

Clustering (II)

Nico Piatkowski und Uwe Ligges

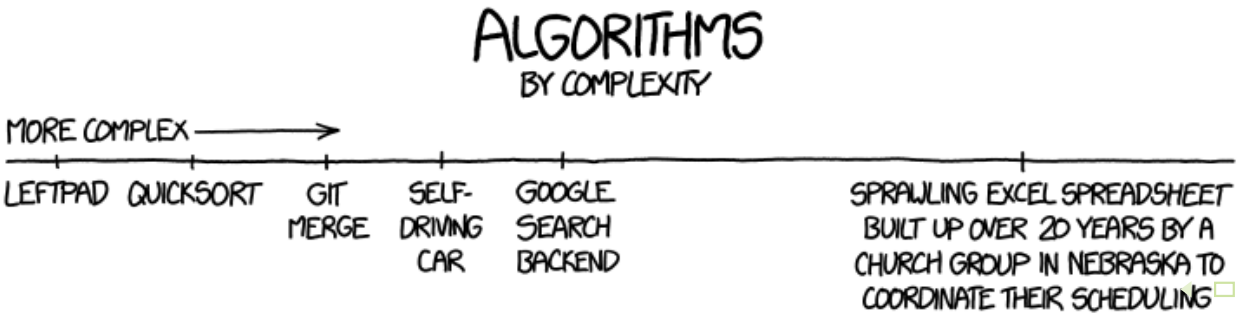
Informatik—Künstliche Intelligenz
Computergestützte Statistik
Technische Universität Dortmund

11.07.2017

Zusammenfassung: Merkmalsauswahl

Sowohl bei Regressions- als auch bei Klassifikationsproblemen, kann es helfen unwichtige Variablen **vor dem Lernen** auszuschließen.

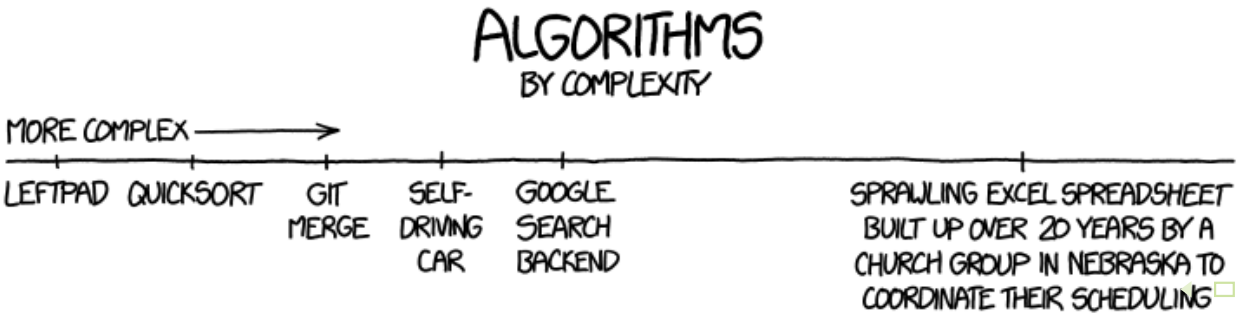
- Eine Greedy-Merkmalsauswahl verwaltet eine Menge von Variablen M
- Das Verfahren paßt die Menge M iterativ an indem Variablen (basierend auf einer **Verlustfunktion**) hinzugeügt/entfernt werden
  - **Forward-Selection**: Die Menge M ist zu Anfang leer und wird in jeder Iteration großer
  - **Backward-Selection**: Die Menge M enthält zu Anfang alle Variablen und wird in jeder Iteration kleiner



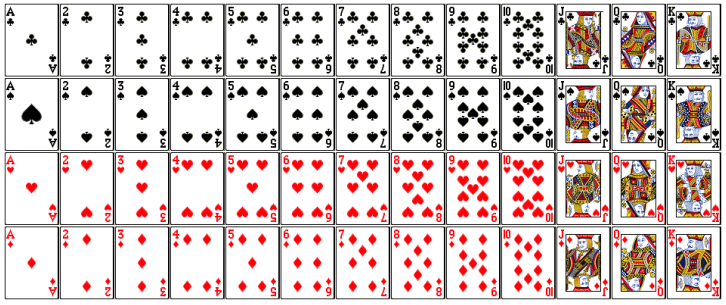
Zusammenfassung: Merkmalsauswahl (II)

Sowohl bei Regressions- als auch bei Klassifikationsproblemen, kann es helfen unwichtige Variablen **vor dem Lernen** auszuschließen.

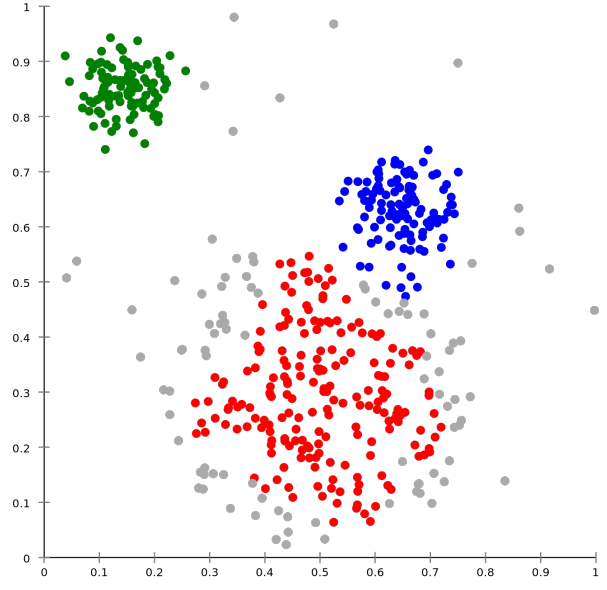
- Eine regularisierungsbasierte Merkmalsauswahl bevorzugt Modelle mit kleiner Norm
- Im Falle der l_1-Norm (LASSO regression) werden Modelle bevorzugt bei denen einige Modellparameter $= 0$ sind
- Im Falle linearer Modelle entspricht dies direkt einer Auswahl an Variablen
- **Nachdem** die Merkmale ausgewählt wurden, wird das eigentliche Modell gelernt



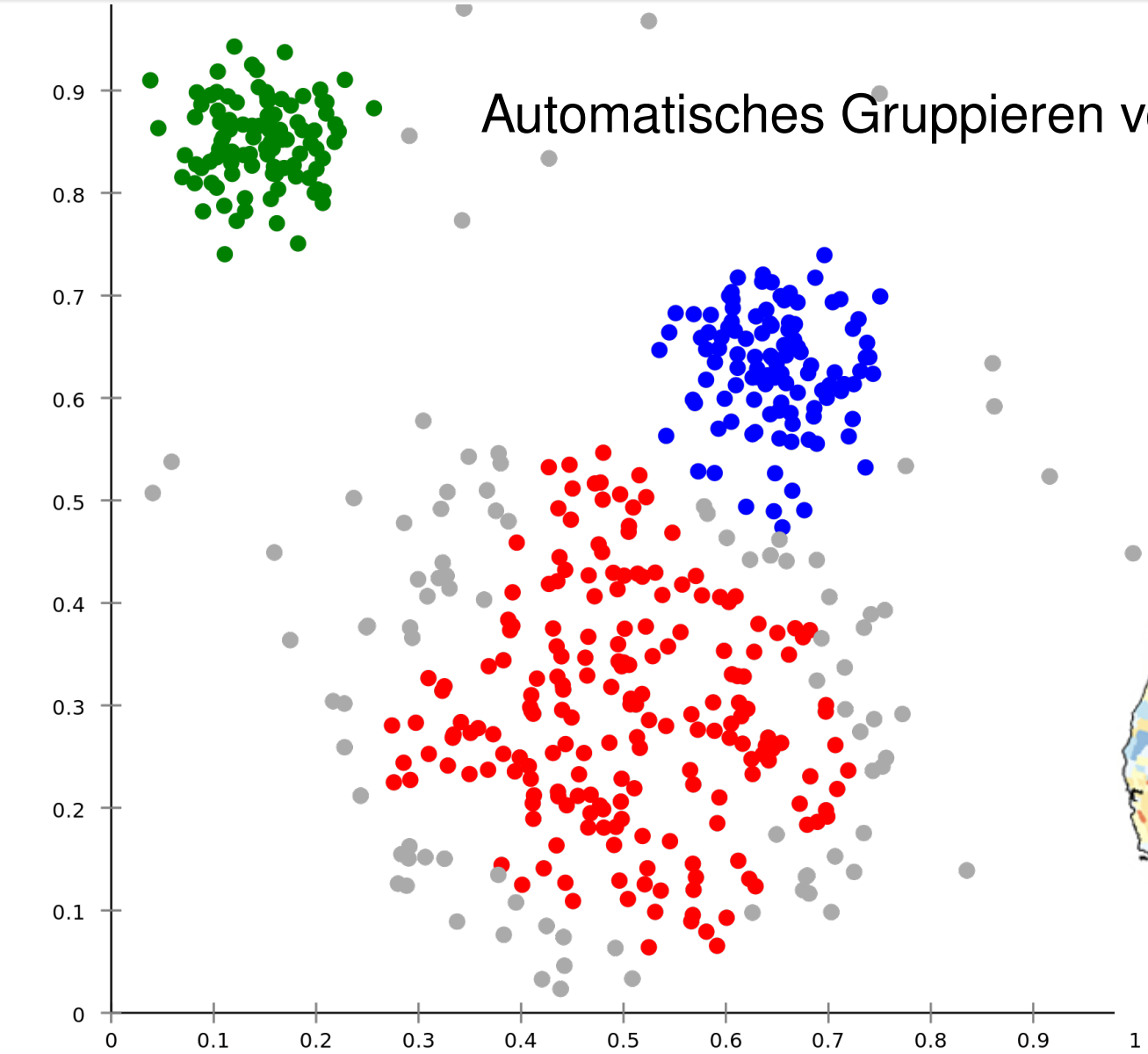
Überblick

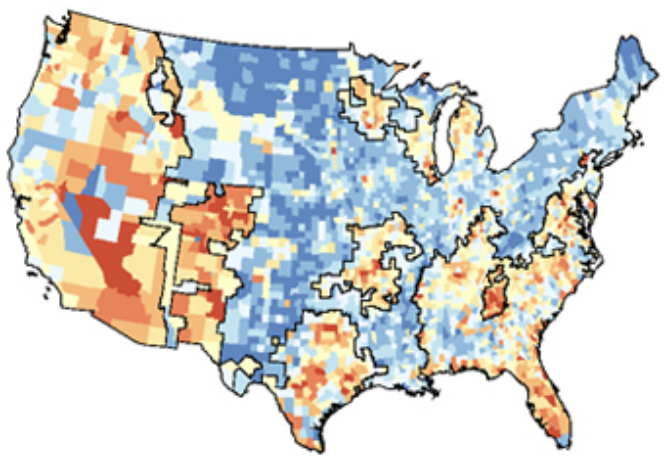


- Clustering
  - Wiederholung: Problemstellung
  - Wiederholung: k-Means
  - DBSCAN
  - LDA



Clusteranalyse





k-Means / Lloyd's Algorithmus

Eingabe: Daten $\mathcal{D}$, Anzahl Cluster k, Metrik/Distanzmaß $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+ \cup \{0\}$

(1) Weise jedem Punkt in D einen zufälligen Cluster zu

(2) Bestimme Clusterzentrum c ("Mittelpunkt") jedes Clusters

(3) Weise jedem Punkt x den Cluster zu, dessen Mittelpunkt c am nächsten zu x ist (mittels f)

(4) Wiederhole Schritte 2 und 3 so lange, bis sich die Clusterzuweisung nicht mehr ändert oder Zeit aufgebraucht

Man kann zeigen: Minimiert Distanzen innerhalb der Cluster und maximiert Distanz zwischen den Clustern

k-Means / Lloyd's Algorithmus

Eingabe: Daten $\mathcal{D}$, Anzahl Cluster k, Metrik/Distanzmaß $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+ \cup \{0\}$

(1) Weise jedem Punkt in D einen zufälligen Cluster zu

(2) Bestimme Clusterzentrum c ("Mittelpunkt") jedes Clusters

(3) Weise jedem Punkt x den Cluster zu, dessen Mittelpunkt c am nächsten zu x ist (mittels f)

(4) Wiederhole Schritte 2 und 3 so lange, bis sich die Clusterzuweisung nicht mehr ändert oder Zeit aufgebraucht

Man kann zeigen: Minimiert Distanzen innerhalb der Cluster und maximiert Distanz zwischen den Clustern

k-Means / Lloyd's Algorithmus

Eingabe: Daten $\mathcal{D}$, Anzahl Cluster k, Metrik/Distanzmaß $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+ \cup \{0\}$

(1) Weise jedem Punkt in D einen zufälligen Cluster zu

(2) Bestimme Clusterzentrum c ("Mittelpunkt") jedes Clusters

(3) Weise jedem Punkt x den Cluster zu, dessen Mittelpunkt c am nächsten zu x ist (mittels f)

(4) Wiederhole Schritte 2 und 3 so lange, bis sich die Clusterzuweisung nicht mehr ändert oder Zeit aufgebraucht

Man kann zeigen: Minimiert Distanzen innerhalb der Cluster und maximiert Distanz zwischen den Clustern

k-Means / Lloyd's Algorithmus

Eingabe: Daten $\mathcal{D}$, Anzahl Cluster k, Metrik/Distanzmaß $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+ \cup \{0\}$

(1) Weise jedem Punkt in D einen zufälligen Cluster zu

(2) Bestimme Clusterzentrum $\boldsymbol{c}$ ("Mittelpunkt") jedes Clusters

(3) Weise jedem Punkt $\boldsymbol{x}$ den Cluster zu, dessen Mittelpunkt $\boldsymbol{c}$ am nächsten zu $\boldsymbol{x}$ ist (mittels f)

(4) Wiederhole Schritte 2 und 3 so lange, bis sich die Clusterzuweisung nicht mehr ändert oder Zeit aufgebraucht

Man kann zeigen: Minimiert Distanzen innerhalb der Cluster und maximiert Distanz zwischen den Clustern

k-Means / Lloyd's Algorithmus

Eingabe: Daten $\mathcal{D}$, Anzahl Cluster k, Metrik/Distanzmaß $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+ \cup \{0\}$

(1) Weise jedem Punkt in D einen zufälligen Cluster zu

(2) Bestimme Clusterzentrum $\boldsymbol{c}$ ("Mittelpunkt") jedes Clusters

(3) Weise jedem Punkt $\boldsymbol{x}$ den Cluster zu, dessen Mittelpunkt $\boldsymbol{c}$ am nächsten zu $\boldsymbol{x}$ ist (mittels f)

(4) Wiederhole Schritte 2 und 3 so lange, bis sich die Clusterzuweisung nicht mehr ändert oder Zeit aufgebraucht

Man kann zeigen: Minimiert Distanzen innerhalb der Cluster und maximiert Distanz zwischen den Clustern

k-Means / Lloyd's Algorithmus

Eingabe: Daten $\mathcal{D}$, Anzahl Cluster k, Metrik/Distanzmaß $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+ \cup \{0\}$

(1) Weise jedem Punkt in D einen zufälligen Cluster zu

(2) Bestimme Clusterzentrum $\boldsymbol{c}$ ("Mittelpunkt") jedes Clusters

(3) Weise jedem Punkt $\boldsymbol{x}$ den Cluster zu, dessen Mittelpunkt $\boldsymbol{c}$ am nächsten zu $\boldsymbol{x}$ ist (mittels f)

(4) Wiederhole Schritte 2 und 3 so lange, bis sich die Clusterzuweisung nicht mehr ändert oder Zeit aufgebraucht

Man kann zeigen: Minimiert Distanzen innerhalb der Cluster und maximiert Distanz zwischen den Clustern

k-Means Verlustfunktion

Notation:

- Datensatz $\mathcal{D}$ mit $|\mathcal{D}| = N$, n-dimensionalen Datenpunkten
- Metrik/Distanzmaß $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+ \cup \{0\}$
- Die Menge $C = \{\boldsymbol{c}^{(1)}, \boldsymbol{c}^{(2)}, \ldots, \boldsymbol{c}^{(k)}\}$ enthält die k Clusterzentren

Optimierungsproblem:

$$\min_{C \subset \mathbb{R}^n, |C|=k} \ell(C; \mathcal{D}) = \min_{C \subset \mathbb{R}^n, |C|=k} \sum_{\boldsymbol{x} \in \mathcal{D}} \min_{\boldsymbol{c} \in C} f(\boldsymbol{x}, \boldsymbol{c})$$

k-Means Verlustfunktion (mit Euklidischem Abstand)

Jetzt:

- Metrik/Distanzmaß $f(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_2^2$

Verlustfunktion:

$$\ell(C; \mathcal{D}) = \sum_{\boldsymbol{x} \in \mathcal{D}} \min_{\boldsymbol{c} \in C} \|\boldsymbol{x} - \boldsymbol{c}\|_2^2$$

$$\frac{\partial}{\partial \boldsymbol{c}_j^{(i)}} \ell(C; \mathcal{D}) = \sum_{\boldsymbol{x} \in \mathcal{D}} \frac{\partial}{\partial \boldsymbol{c}_j^{(i)}} \min_{\boldsymbol{c} \in C} \sum_{l=1}^{n} (\boldsymbol{x}_l - \boldsymbol{c}_l)^2$$

$$= \sum_{\boldsymbol{x} \in \mathcal{D}_i} \frac{\partial}{\partial \boldsymbol{c}_j^{(i)}} (\boldsymbol{x}_j - \boldsymbol{c}_j^{(i)})^2$$

$$= \sum_{\boldsymbol{x} \in \mathcal{D}_i} 2(\boldsymbol{c}_j^{(i)} - \boldsymbol{x}_j)$$

Kurzschreibweise: $\mathcal{D}_i$ enthält Datenpunkt mit minimalem Abstand zu (Cluster) $\boldsymbol{c}_i$

k-Means Verlustfunktion (mit Euklidischem Abstand)

Jetzt:

- Metrik/Distanzmaß $f(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_2^2$

Verlustfunktion:

$$\ell(C; \mathcal{D}) = \sum_{\boldsymbol{x} \in \mathcal{D}} \min_{\boldsymbol{c} \in C} \|\boldsymbol{x} - \boldsymbol{c}\|_2^2$$

$$\frac{\partial}{\partial \boldsymbol{c}_j^{(i)}} \ell(C; \mathcal{D}) = \sum_{\boldsymbol{x} \in \mathcal{D}} \frac{\partial}{\partial \boldsymbol{c}_j^{(i)}} \min_{\boldsymbol{c} \in C} \sum_{l=1}^{n} (\boldsymbol{x}_l - \boldsymbol{c}_l)^2$$

$$= \sum_{\boldsymbol{x} \in \mathcal{D}_i} \frac{\partial}{\partial \boldsymbol{c}_j^{(i)}} (\boldsymbol{x}_j - \boldsymbol{c}_j^{(i)})^2$$

$$= \sum_{\boldsymbol{x} \in \mathcal{D}_i} 2(\boldsymbol{c}_j^{(i)} - \boldsymbol{x}_j)$$

Kurzschreibweise: $\mathcal{D}_i$ enthält Datenpunkt mit minimalem Abstand zu (Cluster) $\boldsymbol{c}_i$

k-Means Verlustfunktion (mit Euklidischem Abstand) (II)

Es gilt **im Optimum**:

$$\ell(C;\mathcal{D}) = 0 = \sum_{\boldsymbol{x}\in\mathcal{D}_i} 2(\boldsymbol{c}_j^{(i)} - \boldsymbol{x}_j)$$

$$\Leftrightarrow 0 = |\mathcal{D}_i| 2\boldsymbol{c}_j^{(i)} - 2\sum_{\boldsymbol{x}\in\mathcal{D}_i} \boldsymbol{x}_j$$

$$\frac{1}{|\mathcal{D}_i|}\sum_{\boldsymbol{x}\in\mathcal{D}_i} \boldsymbol{x}_j = \boldsymbol{c}_j^{(i)}$$

Also:

- k-Means ist ein Optimierungsverfahren erster Ordnung (wie Gradientenabsteig!)
- **Aber**: Zielfunktion ist nicht-konvex. Keine Konvergenz zum globalen Optimum!

k-Means Verlustfunktion (mit Euklidischem Abstand) (II)

Es gilt **im Optimum**:

$$\ell(C;\mathcal{D}) = 0 = \sum_{\boldsymbol{x}\in\mathcal{D}_i} 2(\boldsymbol{c}_j^{(i)} - \boldsymbol{x}_j)$$

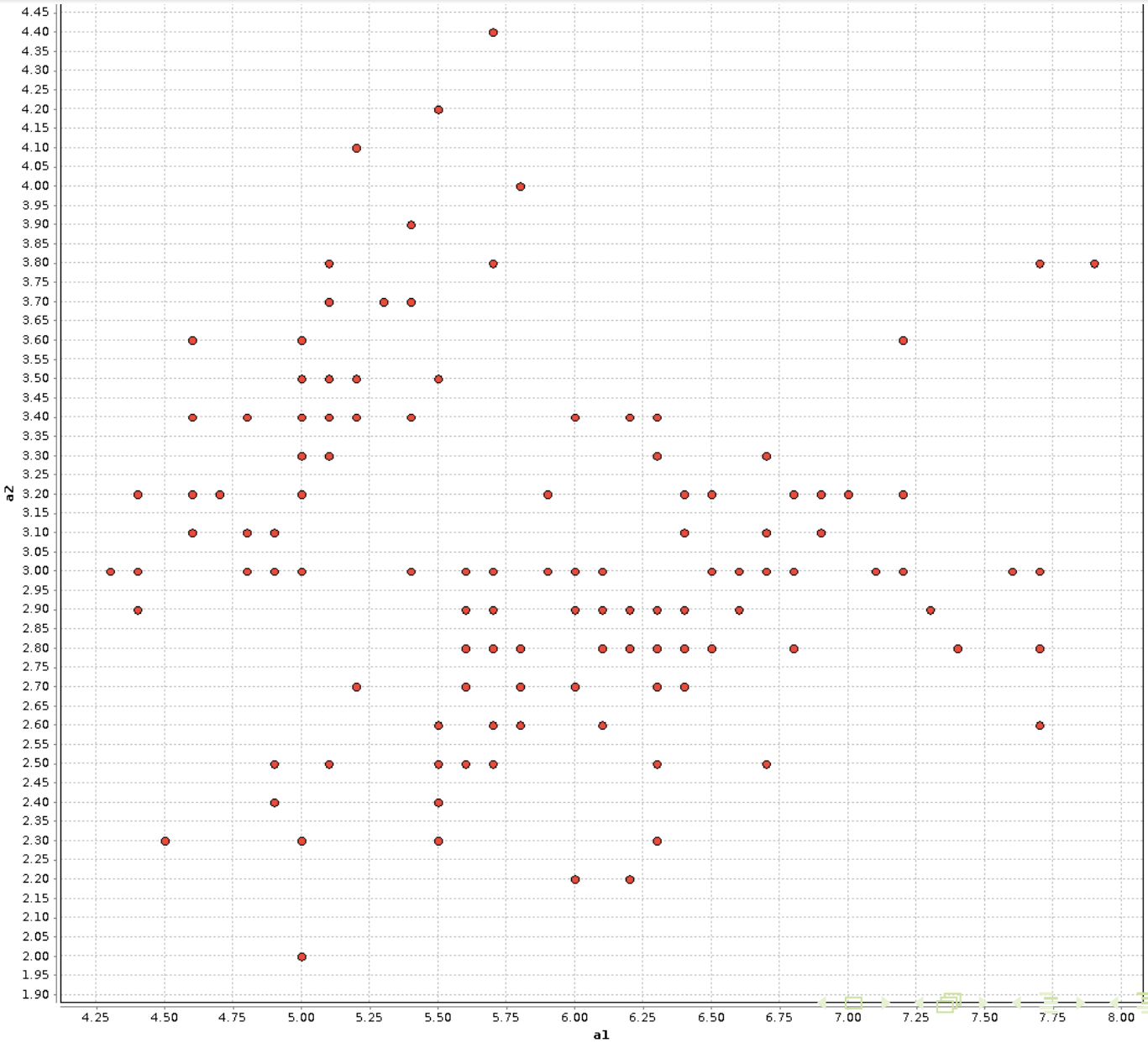
$$\Leftrightarrow 0 = |\mathcal{D}_i| 2\boldsymbol{c}_j^{(i)} - 2\sum_{\boldsymbol{x}\in\mathcal{D}_i} \boldsymbol{x}_j$$

$$\frac{1}{|\mathcal{D}_i|}\sum_{\boldsymbol{x}\in\mathcal{D}_i} \boldsymbol{x}_j = \boldsymbol{c}_j^{(i)}$$

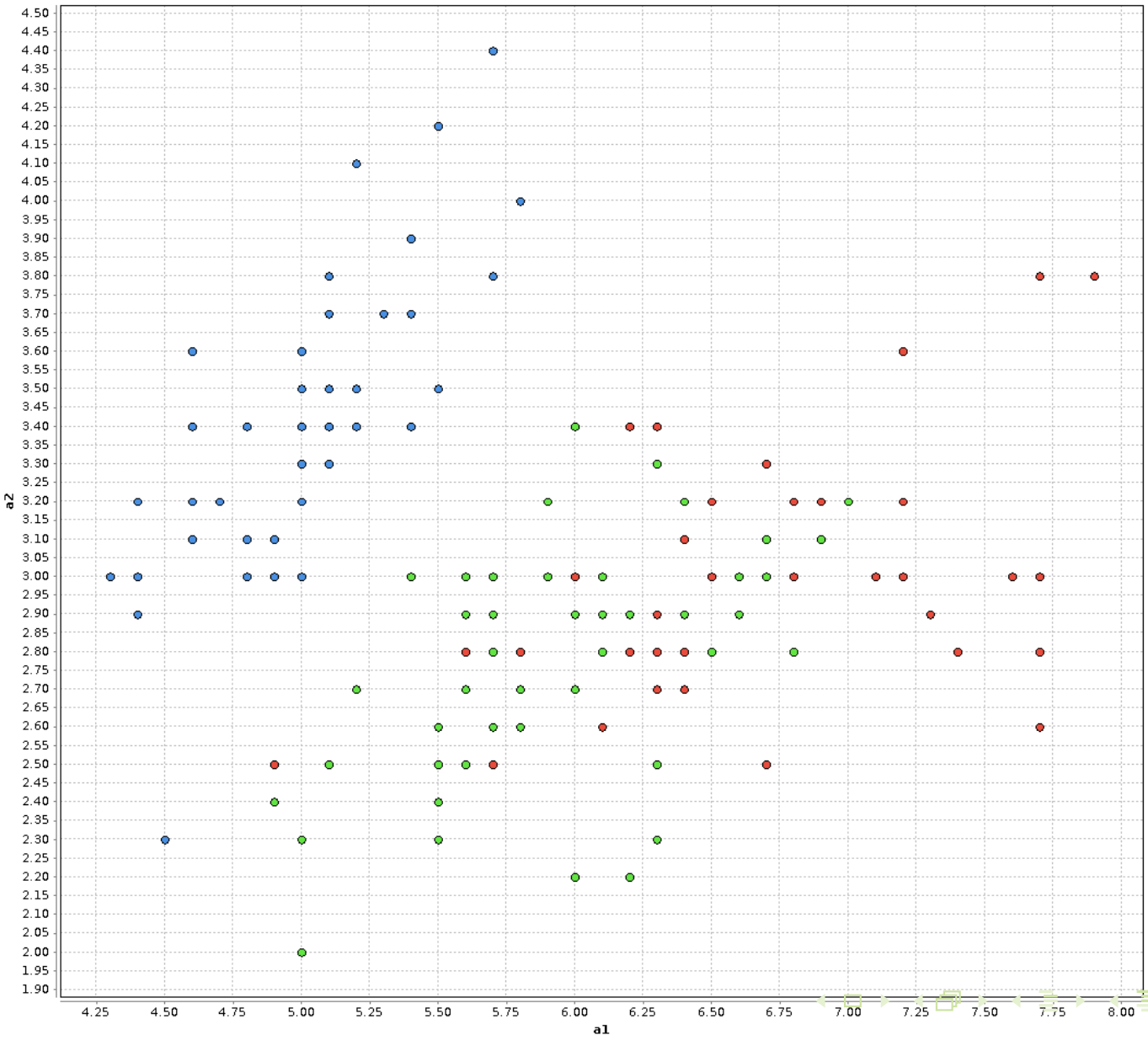
Also:

- k-Means ist ein Optimierungsverfahren erster Ordnung (wie Gradientenabsteig!)
- **Aber**: Zielfunktion ist nicht-konvex. Keine Konvergenz zum globalen Optimum!

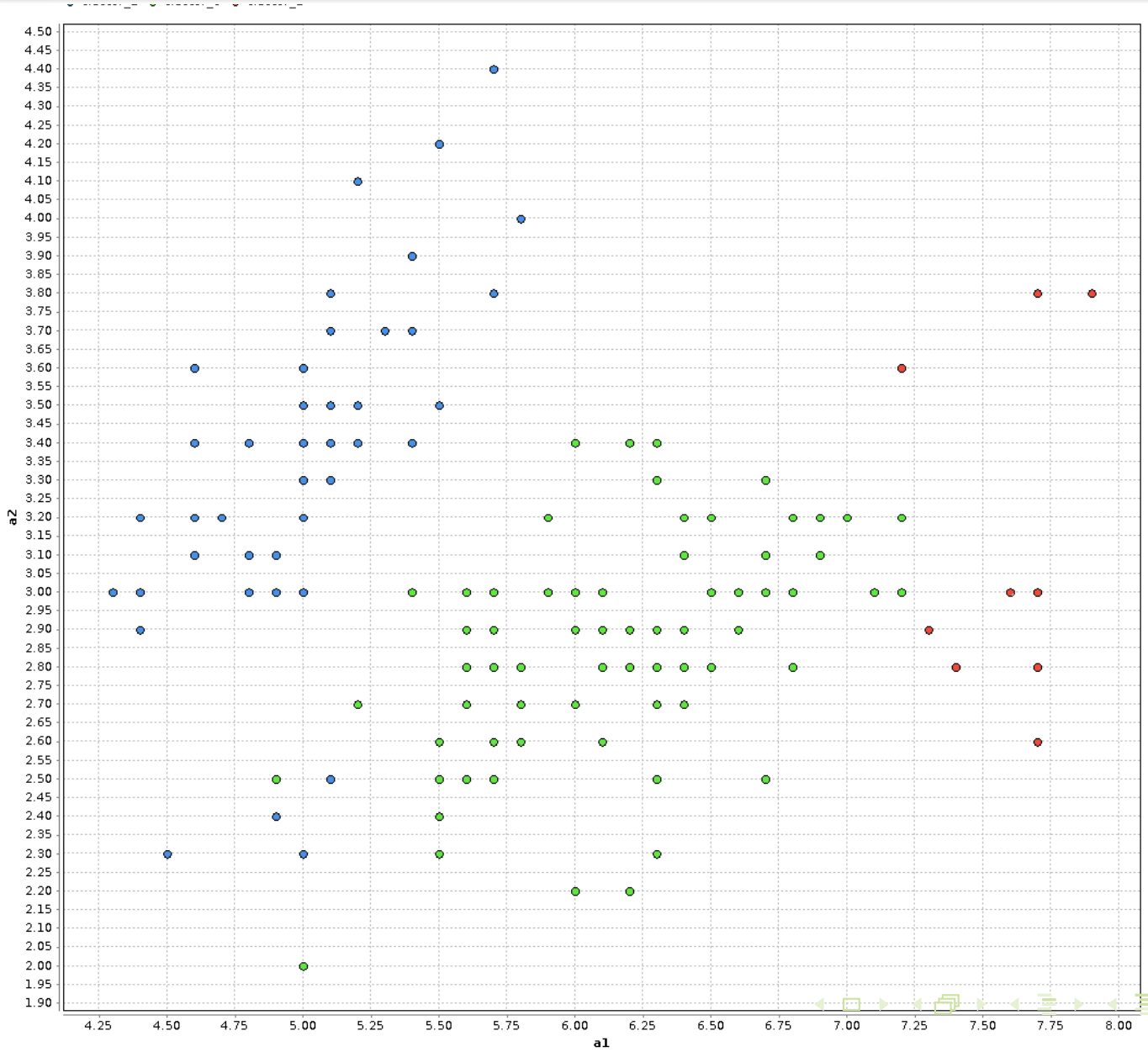
Beispiel: Daten (x)



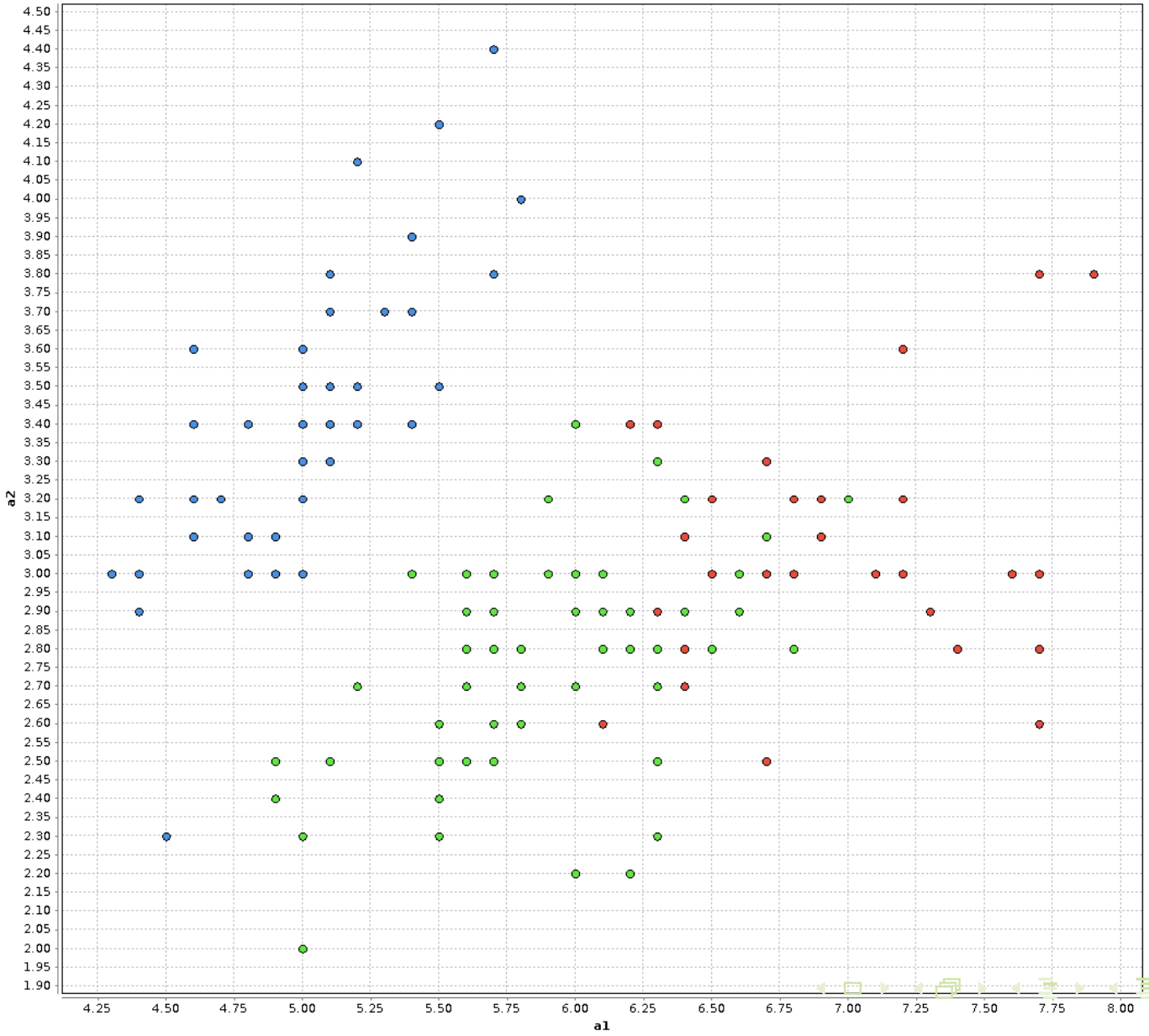
Beispiel: Daten mit Klassen (x, y)



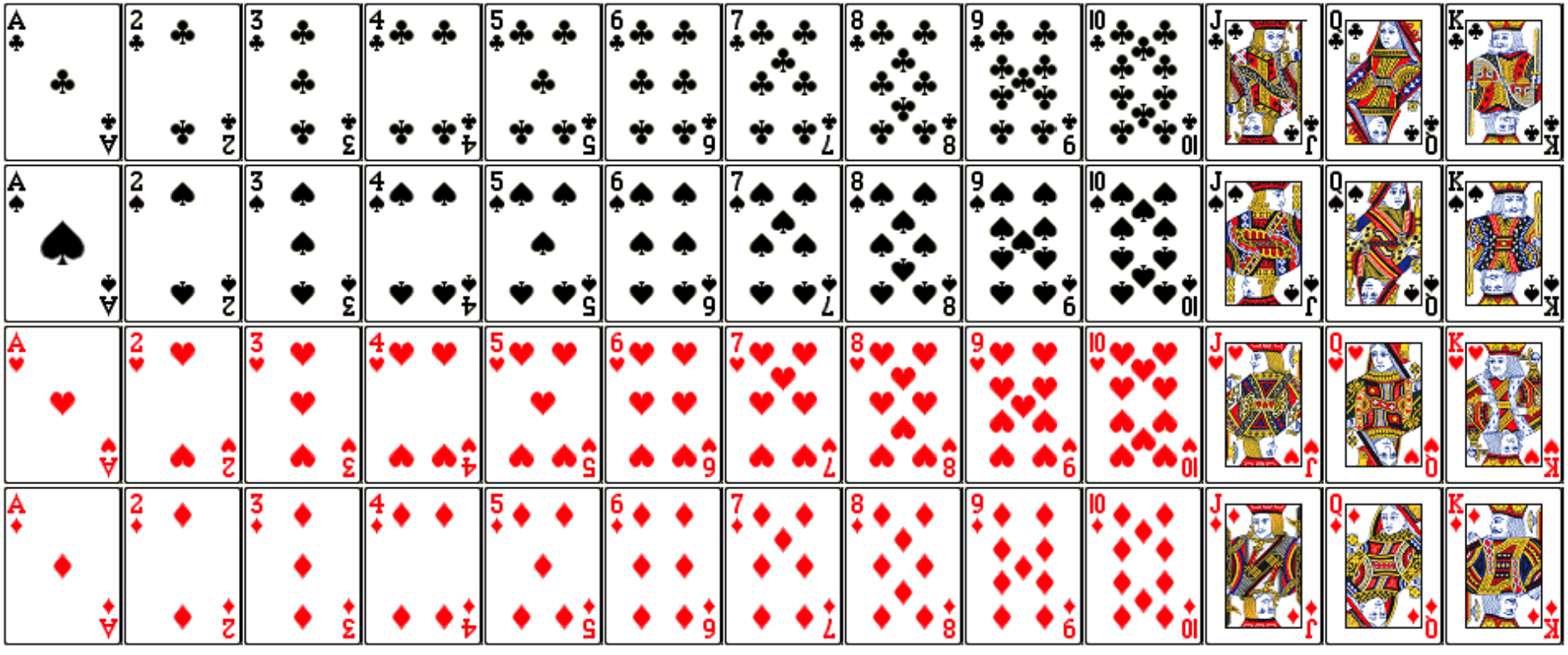
Beispiel: 3-means, 1 Iteration



Beispiel: 3-means, 10 Iterationen



Intuition: Auswahl von k und Distanzmaß (II)



DBSCAN: Vermeidung von k

Wie kann die Wahl von k vermieden werden?

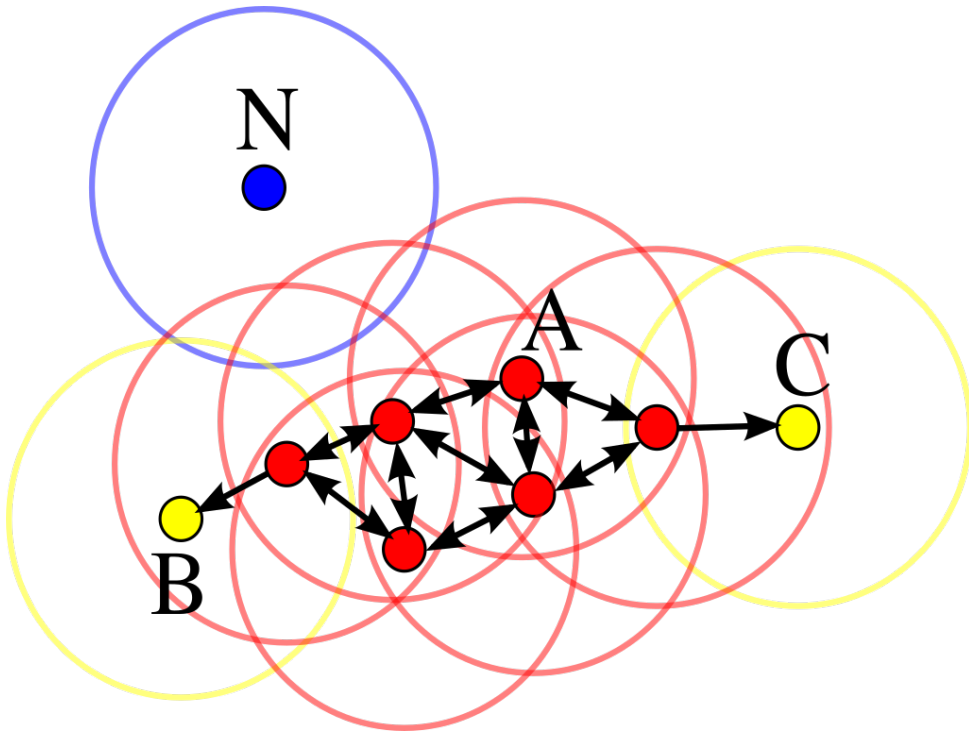
- Neue Verlustfunktion(?) = neuer Algorithmus

Kategorisierung der Punkte eines Datensatzes:

- Kernpunkte
- Dichte-erreichbar
- Rauschen

Problemstellung mittels

- Nachbarschaftsgröße ϵ
- Mindestanzahl an Nachbarn $minPts$



DBSCAN: Vermeidung von k

Wie kann die Wahl von k vermieden werden?

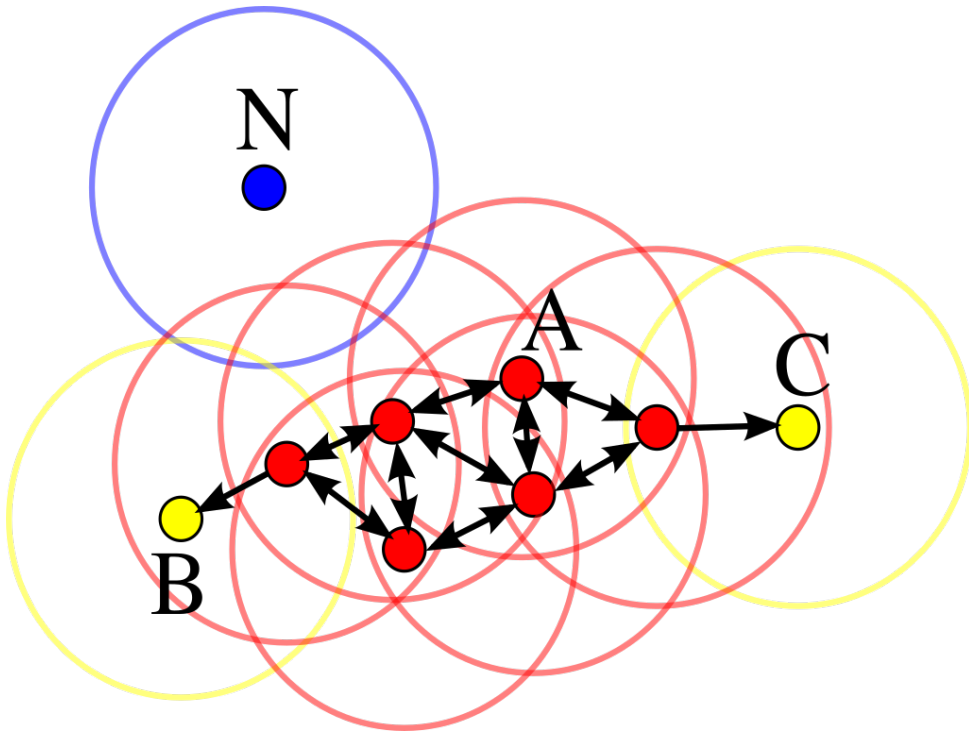
- Neue Verlustfunktion(?) = neuer Algorithmus

Kategorisierung der Punkte eines Datensatzes:

- Kernpunkte
- Dichte-erreichbar
- Rauschen

Problemstellung mittels

- Nachbarschaftsgröße ϵ
- Mindestanzahl an Nachbarn $minPts$



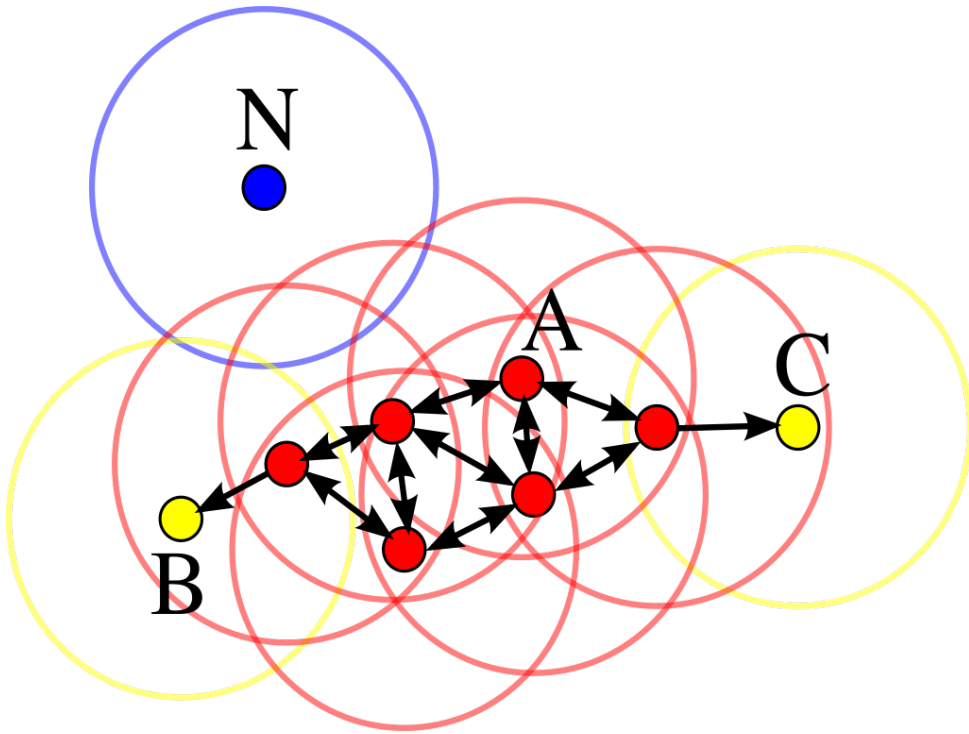
DBSCAN: Vermeidung von k

Wie kann die Wahl von k vermieden werden?

- Neue Verlustfunktion(?) = neuer Algorithmus

Kategorisierung der Punkte eines Datensatzes:

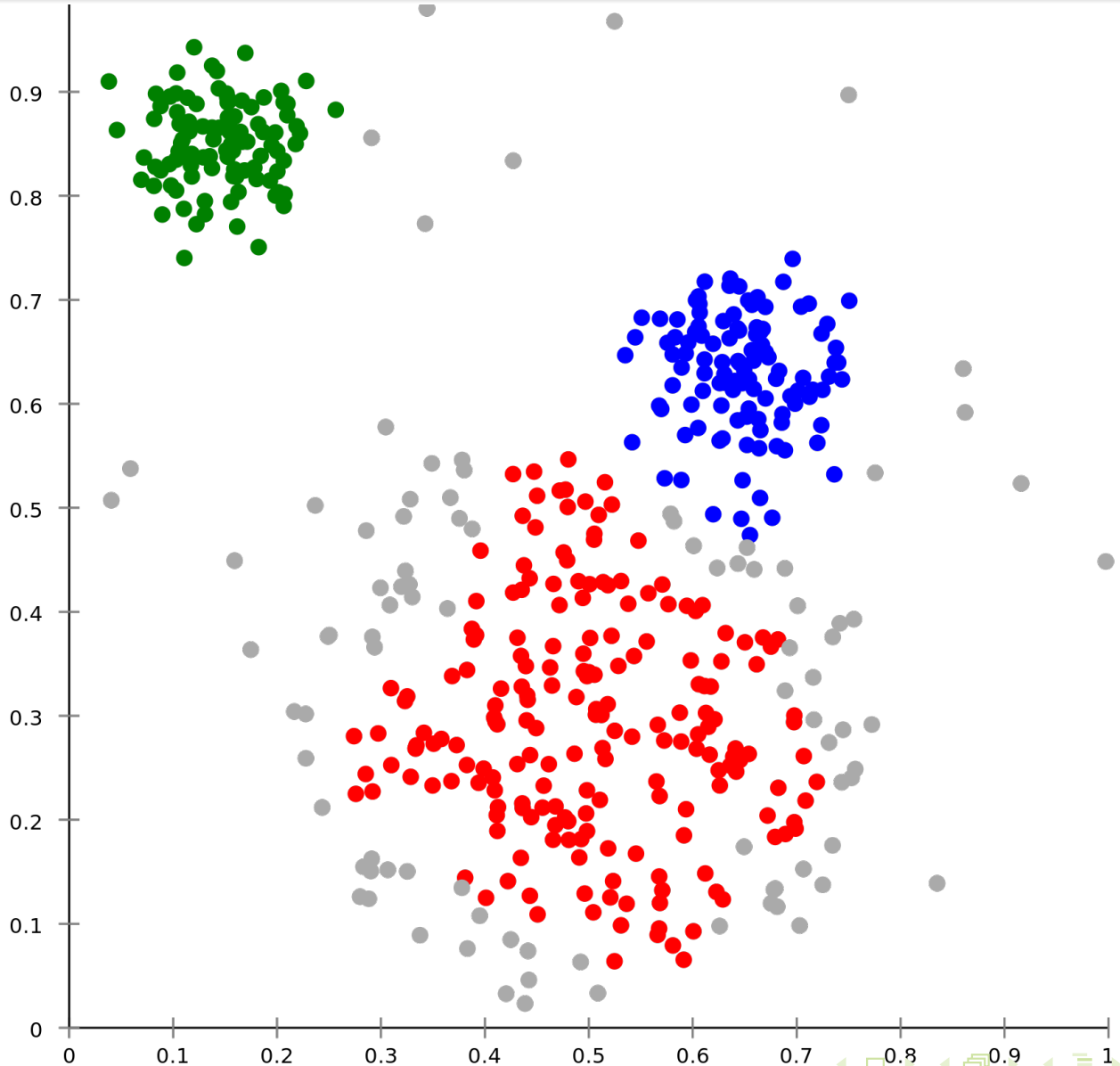
- Kernpunkte
- Dichte-erreichbar
- Rauschen



Problemstellung mittels

- Nachbarschaftsgröße ϵ
- Mindestanzahl an Nachbarn $minPts$

DBSCAN Clustering



Clustern von Text

Menge von Dokumenten = Korpus
Verschiedene Darstellungen denkbar:

- Vorhandensein von Worten (Binärvektoren, Mengen)
- Anzahl von Worten (Bag-of-Words)
- Term-Frequency-Inverse-Document-Frequency (TF-IDF)

$$tfidf_{w,d} = \frac{\text{Häufigkeit Wort } w \text{ in Dokument } d}{\max_{d'} \text{Häufigkeit Wort } w \text{ in Dokument } d'} \log \frac{N}{N_w}$$

N_w ist Anzahl Dok. mit Wort w; N ist Anzahl aller Dok.

- k-means Clustering der obigen Darstellungen möglich
- **Aber**: Bei großem Vokabular sind alle Dokumente weit weg (“Fluch der hohen Dimensionen”)

Clustern von Text

Menge von Dokumenten = Korpus
Verschiedene Darstellungen denkbar:

- Vorhandensein von Worten (Binärvektoren, Mengen)
- Anzahl von Worten (Bag-of-Words)
- Term-Frequency-Inverse-Document-Frequency (TF-IDF)

$$tfidf_{w,d} = \frac{\text{Häufigkeit Wort } w \text{ in Dokument } d}{\max_{d'} \text{Häufigkeit Wort } w \text{ in Dokument } d'} \log \frac{N}{N_w}$$

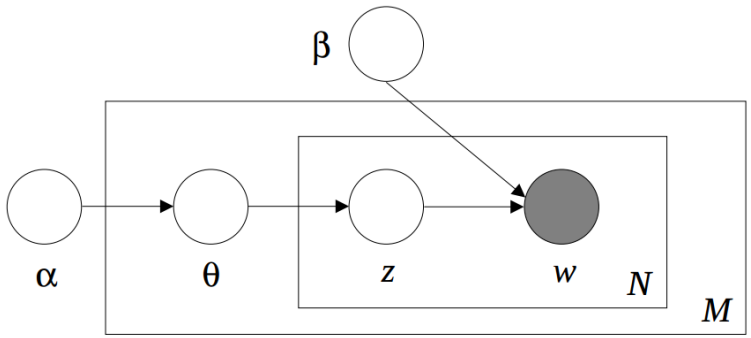
N_w ist Anzahl Dok. mit Wort w; N ist Anzahl aller Dok.

- k-means Clustering der obigen Darstellungen möglich
- **Aber**: Bei großem Vokabular sind alle Dokumente weit weg (“Fluch der hohen Dimensionen”)

Clustern von Text mit Graphischen Modellen (Topic Models)

Idee: Datengenerierender Prozess von Dokument d ist:

- Wähle Länge des Dokuments N_d aus $\mathbb{P}(N_d \mid \lambda)$ [Poisson]
- Wähle Themenverteilung θ_d aus $\mathbb{P}(\theta_d \mid \alpha)$ [Dirichlet]
- Erzeuge die Worte $w = 1 \ldots N_d$:
  - Wähle ein Thema z_w aus $\mathbb{P}(z_w \mid \theta)$ [Kategorisch]
  - Wähle ein Wort aus $\mathbb{P}(w \mid z_w, \beta)$ [Kategorisch]



Lernen: Bestimme λ, α, β, und θ via Expectation-Maximization (Maximum-Likelihood für unvollständige Daten)