Vorlesung Maschinelles Lernen

Cluster Ensembles, Subspace Clustering, Distributed Clustering

Katharina Morik

LS 8 Informatik
Fakultät für Informatik
Technische Universität Dortmund

28.1.2014

Cluster Ensembles

- Lernaufgabe
  - Gegeben verschiedene clusterings eines Datensatzes
  - finde ein kombiniertes clustering.
- Wie müssen die Eingabe-clusterings beschaffen sein?
  - möglichst unterschiedlich
  - möglichst leicht/schnell zu berechnen
- Wie soll die Konsensusfunktion aussehen, damit das kombinierte clustering gut ist?
  - Gütemaß: inter cluster dissimilarity, intra cluster similarity
  - Kombiniertes clustering ist besser als irgend ein einzelnes.

Cluster Ensembles

- Lernaufgabe
  - Gegeben verschiedene clusterings eines Datensatzes
  - finde ein kombiniertes clustering.
- Wie müssen die Eingabe-clusterings beschaffen sein?
  - möglichst unterschiedlich
  - möglichst leicht/schnell zu berechnen
- Wie soll die Konsensusfunktion aussehen, damit das kombinierte clustering gut ist?
  - Gütemaß: inter cluster dissimilarity, intra cluster similarity
  - Kombiniertes clustering ist besser als irgend ein einzelnes.

Cluster Ensembles

- Lernaufgabe
  - Gegeben verschiedene clusterings eines Datensatzes
  - finde ein kombiniertes clustering.
- Wie müssen die Eingabe-clusterings beschaffen sein?
  - möglichst unterschiedlich
  - möglichst leicht/schnell zu berechnen
- Wie soll die Konsensusfunktion aussehen, damit das kombinierte clustering gut ist?
  - Gütemaß: inter cluster dissimilarity, intra cluster similarity
  - Kombiniertes clustering ist besser als irgend ein einzelnes.

Cluster Ensembles

- Lernaufgabe
  - Gegeben verschiedene clusterings eines Datensatzes
  - finde ein kombiniertes clustering.
- Wie müssen die Eingabe-clusterings beschaffen sein?
  - möglichst unterschiedlich
  - möglichst leicht/schnell zu berechnen
- Wie soll die Konsensusfunktion aussehen, damit das kombinierte clustering gut ist?
  - Gütemaß: inter cluster dissimilarity, intra cluster similarity
  - Kombiniertes clustering ist besser als irgend ein einzelnes.

Cluster Ensembles

- Lernaufgabe
  - Gegeben verschiedene clusterings eines Datensatzes
  - finde ein kombiniertes clustering.
- Wie müssen die Eingabe-clusterings beschaffen sein?
  - möglichst unterschiedlich
  - möglichst leicht/schnell zu berechnen
- Wie soll die Konsensusfunktion aussehen, damit das kombinierte clustering gut ist?
  - Gütemaß: inter cluster dissimilarity, intra cluster similarity
  - Kombiniertes clustering ist besser als irgend ein einzelnes.

Cluster Ensembles

- Lernaufgabe
  - Gegeben verschiedene clusterings eines Datensatzes
  - finde ein kombiniertes clustering.
- Wie müssen die Eingabe-clusterings beschaffen sein?
  - möglichst unterschiedlich
  - möglichst leicht/schnell zu berechnen
- Wie soll die Konsensusfunktion aussehen, damit das kombinierte clustering gut ist?
  - Gütemaß: inter cluster dissimilarity, intra cluster similarity
  - Kombiniertes clustering ist besser als irgend ein einzelnes.

Cluster Ensembles

- Lernaufgabe
  - Gegeben verschiedene clusterings eines Datensatzes
  - finde ein kombiniertes clustering.
- Wie müssen die Eingabe-clusterings beschaffen sein?
  - möglichst unterschiedlich
  - möglichst leicht/schnell zu berechnen
- Wie soll die Konsensusfunktion aussehen, damit das kombinierte clustering gut ist?
  - Gütemaß: inter cluster dissimilarity, intra cluster similarity
  - Kombiniertes clustering ist besser als irgend ein einzelnes.

Cluster Ensembles

- Lernaufgabe
  - Gegeben verschiedene clusterings eines Datensatzes
  - finde ein kombiniertes clustering.
- Wie müssen die Eingabe-clusterings beschaffen sein?
  - möglichst unterschiedlich
  - möglichst leicht/schnell zu berechnen
- Wie soll die Konsensusfunktion aussehen, damit das kombinierte clustering gut ist?
  - Gütemaß: inter cluster dissimilarity, intra cluster similarity
  - Kombiniertes clustering ist besser als irgend ein einzelnes.

Cluster Ensembles

- Lernaufgabe
  - Gegeben verschiedene clusterings eines Datensatzes
  - finde ein kombiniertes clustering.
- Wie müssen die Eingabe-clusterings beschaffen sein?
  - möglichst unterschiedlich
  - möglichst leicht/schnell zu berechnen
- Wie soll die Konsensusfunktion aussehen, damit das kombinierte clustering gut ist?
  - Gütemaß: inter cluster dissimilarity, intra cluster similarity
  - Kombiniertes clustering ist besser als irgend ein einzelnes.

Cluster Ensembles: formale Aufgabenstellung

Gegeben: X: Beobachtungen im d-dimensionalen Raum
G: Menge von clusterings $\{\varphi_1, \ldots, \varphi_H\}$

- Dabei: $\varphi_i = \{g_1^i, \ldots, g_K^i\}$, wobei $X = g_1^i \cup g_2^i \cup \cdots \cup g_K^i$

Gesucht: $\sigma = \{g_1, \ldots, g_K\}$, wobei

- $X = g_1 \cup g_2 \cup \cdots \cup g_k$ und
- $sim(x_1, x_2) > sim(x_1, x_3)$ gdw. $x_1, x_2 \in g_i, x_3 \notin g_i$

Cluster Ensembles: formale Aufgabenstellung

Gegeben: X: Beobachtungen im d-dimensionalen Raum
G: Menge von clusterings $\{\varphi_1, \ldots, \varphi_H\}$

- Dabei: $\varphi_i = \{g_1^i, \ldots, g_K^i\}$, wobei $X = g_1^i \cup g_2^i \cup \cdots \cup g_K^i$

Gesucht: $\sigma = \{g_1, \ldots, g_K\}$, wobei

- $X = g_1 \cup g_2 \cup \cdots \cup g_k$ und
- $sim(x_1, x_2) > sim(x_1, x_3)$ gdw. $x_1, x_2 \in g_i, x_3 \notin g_i$

Cluster Ensembles: formale Aufgabenstellung

Gegeben: X: Beobachtungen im d-dimensionalen Raum
G: Menge von clusterings $\{\varphi_1, \ldots, \varphi_H\}$

- Dabei: $\varphi_i = \{g_1^i, \ldots, g_K^i\}$, wobei $X = g_1^i \cup g_2^i \cup \cdots \cup g_K^i$

Gesucht: $\sigma = \{g_1, \ldots, g_K\}$, wobei

- $X = g_1 \cup g_2 \cup \cdots \cup g_k$ und
- $sim(x_1, x_2) > sim(x_1, x_3)$ gdw. $x_1, x_2 \in g_i, x_3 \notin g_i$

Cluster Ensembles: formale Aufgabenstellung

Gegeben: X: Beobachtungen im d-dimensionalen Raum
G: Menge von clusterings $\{\varphi_1, \ldots, \varphi_H\}$

- Dabei: $\varphi_i = \{g_1^i, \ldots, g_K^i\}$, wobei $X = g_1^i \cup g_2^i \cup \cdots \cup g_K^i$

Gesucht: $\sigma = \{g_1, \ldots, g_K\}$, wobei

- $X = g_1 \cup g_2 \cup \cdots \cup g_k$ und
- $sim(x_1, x_2) > sim(x_1, x_3)$ gdw. $x_1, x_2 \in g_i, x_3 \notin g_i$

Cluster Ensembles: formale Aufgabenstellung

Gegeben: X: Beobachtungen im d-dimensionalen Raum
G: Menge von clusterings $\{\varphi_1, \ldots, \varphi_H\}$

- Dabei: $\varphi_i = \{g_1^i, \ldots, g_K^i\}$, wobei $X = g_1^i \cup g_2^i \cup \cdots \cup g_K^i$

Gesucht: $\sigma = \{g_1, \ldots, g_K\}$, wobei

- $X = g_1 \cup g_2 \cup \cdots \cup g_k$ und
- $sim(x_1, x_2) > sim(x_1, x_3)$ gdw. $x_1, x_2 \in g_i, x_3 \notin g_i$

Cluster-Zugehörigkeit als neues Merkmal [4]

Gegeben:
X: Beobachtungen im d-dimensionalen Raum
G: Menge von clusterings $\{\varphi_1, \ldots, \varphi_H\}$ mit

- $\varphi_i = \{g_1^i, \ldots, g_k^i\}$
- $X = g_1^i \cup \ldots \cup g_k^i$

Gesucht:
$\sigma = \{g_1, \ldots, g_K\}$, wobei

- $X = g_1 \cup \ldots \cup g_K$
- $sim(x_1, x_2) > sim(x_1, x_3)$ gdw. $x_1, x_2 \in g_i, x_3 \notin g_i$

- Mehrere clusterings über den Attributen $f_1, \ldots, f_d$ (Eingabe G)
- Ein clustering über den Attributen $\varphi_1, \ldots, \varphi_H$ (Konsensus σ)

Attribut	f_1	$\ldots$	f_d	φ_1	$\ldots$	φ_H
Attributwerte				g_1^1 $\ldots$ g_K^1		g_1^H $\ldots$ g_K^H

Cluster-Zugehörigkeit als neues Merkmal [4]

Gegeben:

X: Beobachtungen im d-dimensionalen Raum

G: Menge von clusterings $\{\varphi_1, \ldots, \varphi_H\}$ mit

- $\varphi_i = \{g_1^i, \ldots, g_K^i\}$
- $X = g_1^i \cup \cdots \cup g_K^i$

Gesucht:

$\sigma = \{g_1, \ldots, g_K\}$, wobei

- $X = g_1 \cup \cdots \cup g_K$
- $sim(x_1, x_2) > sim(x_1, x_3)$ gdw. $x_1, x_2 \in g_i, x_3 \notin g_i$

- Mehrere clusterings über den Attributen $f_1, \ldots, f_d$ (Eingabe G)
- Ein clustering über den Attributen $\varphi_1, \ldots, \varphi_H$ (Konsensus σ)

Attribut	f_1	$\ldots$	f_d	φ_1	$\ldots$	φ_H
Attributwerte				g_1^1 $\ldots$ g_K^1		g_1^H $\ldots$ g_K^H

Cluster-Zugehörigkeit als neues Merkmal [4]

Gegeben:

X: Beobachtungen im d-dimensionalen Raum

G: Menge von clusterings $\{\varphi_1, \ldots, \varphi_H\}$ mit

- $\varphi_i = \{g_1^i, \ldots, g_K^i\}$
- $X = g_1^i \cup \cdots \cup g_K^i$

Gesucht:

$\sigma = \{g_1, \ldots, g_K\}$, wobei

- $X = g_1 \cup \cdots \cup g_K$
- $sim(x_1, x_2) > sim(x_1, x_3)$ gdw. $x_1, x_2 \in g_i, x_3 \notin g_i$

- Mehrere clusterings über den Attributen $f_1, \ldots, f_d$ (Eingabe G)
- Ein clustering über den Attributen $\varphi_1, \ldots, \varphi_H$ (Konsensus σ)

Attribut	f_1	$\ldots$	f_d	φ_1	$\ldots$	φ_H
Attributwerte				g_1^1 $\ldots$ g_K^1		g_1^H $\ldots$ g_K^H

Cluster-Zugehörigkeit als neues Merkmal [4]

Gegeben:

X: Beobachtungen im d-dimensionalen Raum

G: Menge von clusterings $\{\varphi_1, \ldots, \varphi_H\}$ mit

- $\varphi_i = \{g_1^i, \ldots, g_K^i\}$
- $X = g_1^i \cup \cdots \cup g_K^i$

Gesucht:

$\sigma = \{g_1, \ldots, g_K\}$, wobei

- $X = g_1 \cup \cdots \cup g_K$
- $sim(x_1, x_2) > sim(x_1, x_3)$ gdw. $x_1, x_2 \in g_i, x_3 \notin g_i$

- Mehrere clusterings über den Attributen $f_1, \ldots, f_d$ (Eingabe G)
- Ein clustering über den Attributen $\varphi_1, \ldots, \varphi_H$ (Konsensus σ)

Attribut	f_1	$\ldots$	f_d	φ_1	$\ldots$	φ_H
Attributwerte				g_1^1 $\ldots$ g_K^1		g_1^H $\ldots$ g_K^H

Cluster-Zugehörigkeit als neues Merkmal [4]

Gegeben:

X: Beobachtungen im d-dimensionalen Raum

G: Menge von clusterings $\{\varphi_1, \ldots, \varphi_H\}$ mit

- $\varphi_i = \{g_1^i, \ldots, g_K^i\}$
- $X = g_1^i \cup \cdots \cup g_K^i$

Gesucht:

$\sigma = \{g_1, \ldots, g_K\}$, wobei

- $X = g_1 \cup \cdots \cup g_K$
- $sim(x_1, x_2) > sim(x_1, x_3)$ gdw. $x_1, x_2 \in g_i, x_3 \notin g_i$

- Mehrere clusterings über den Attributen $f_1, \ldots, f_d$ (Eingabe G)
- Ein clustering über den Attributen $\varphi_1, \ldots, \varphi_H$ (Konsensus σ)

Attribut	f_1	$\ldots$	f_d	φ_1	$\ldots$	φ_H
Attributwerte				g_1^1 $\ldots$ g_K^1		g_1^H $\ldots$ g_K^H

Cluster-Zugehörigkeit als neues Merkmal [4]

Gegeben:

X: Beobachtungen im d-dimensionalen Raum

G: Menge von clusterings $\{\varphi_1, \ldots, \varphi_H\}$ mit

- $\varphi_i = \{g_1^i, \ldots, g_K^i\}$
- $X = g_1^i \cup \cdots \cup g_K^i$

Gesucht:

$\sigma = \{g_1, \ldots, g_K\}$, wobei

- $X = g_1 \cup \cdots \cup g_K$
- $sim(x_1, x_2) > sim(x_1, x_3)$ gdw. $x_1, x_2 \in g_i, x_3 \notin g_i$

- Mehrere clusterings über den Attributen $f_1, \ldots, f_d$ (Eingabe G)
- Ein clustering über den Attributen $\varphi_1, \ldots, \varphi_H$ (Konsensus σ)

Attribut	f_1	$\ldots$	f_d	φ_1	$\ldots$	φ_H
Attributwerte				g_1^1 $\ldots$ g_K^1		g_1^H $\ldots$ g_K^H

Cluster-Zugehörigkeit als neues Merkmal [4]

Gegeben:

X: Beobachtungen im d-dimensionalen Raum

G: Menge von clusterings $\{\varphi_1, \ldots, \varphi_H\}$ mit

- $\varphi_i = \{g_1^i, \ldots, g_K^i\}$
- $X = g_1^i \cup \cdots \cup g_K^i$

Gesucht:

$\sigma = \{g_1, \ldots, g_K\}$, wobei

- $X = g_1 \cup \cdots \cup g_K$
- $sim(x_1, x_2) > sim(x_1, x_3)$ gdw. $x_1, x_2 \in g_i, x_3 \notin g_i$

- Mehrere clusterings über den Attributen $f_1, \ldots, f_d$ (Eingabe G)
- Ein clustering über den Attributen $\varphi_1, \ldots, \varphi_H$ (Konsensus σ)

Attribut	f_1	$\ldots$	f_d	φ_1	$\ldots$	φ_H
Attributwerte				g_1^1 $\ldots$ g_K^1		g_1^H $\ldots$ g_K^H

Cluster-Zugehörigkeit als neues Merkmal [4]

Gegeben:

X: Beobachtungen im d-dimensionalen Raum

G: Menge von clusterings $\{\varphi_1, \ldots, \varphi_H\}$ mit

- $\varphi_i = \{g_1^i, \ldots, g_K^i\}$
- $X = g_1^i \cup \cdots \cup g_K^i$

Gesucht:

$\sigma = \{g_1, \ldots, g_K\}$, wobei

- $X = g_1 \cup \cdots \cup g_K$
- $sim(x_1, x_2) > sim(x_1, x_3)$ gdw. $x_1, x_2 \in g_i, x_3 \notin g_i$

- Mehrere clusterings über den Attributen $f_1, \ldots, f_d$ (Eingabe G)
- Ein clustering über den Attributen $\varphi_1, \ldots, \varphi_H$ (Konsensus σ)

Attribut	f_1	$\ldots$	f_d	φ_1	$\ldots$	φ_H
Attributwerte				g_1^1		g_1^H
				$\ldots$		$\ldots$
				g_K^1		g_K^H

Cluster-Zugehörigkeit als neues Merkmal [4]

Gegeben:

X: Beobachtungen im d-dimensionalen Raum

G: Menge von clusterings $\{\varphi_1, \ldots, \varphi_H\}$ mit

- $\varphi_i = \{g_1^i, \ldots, g_K^i\}$
- $X = g_1^i \cup \cdots \cup g_K^i$

Gesucht:

$\sigma = \{g_1, \ldots, g_K\}$, wobei

- $X = g_1 \cup \cdots \cup g_K$
- $sim(x_1, x_2) > sim(x_1, x_3)$ gdw. $x_1, x_2 \in g_i, x_3 \notin g_i$

- Mehrere clusterings über den Attributen $f_1, \ldots, f_d$ (Eingabe G)
- Ein clustering über den Attributen $\varphi_1, \ldots, \varphi_H$ (Konsensus σ)

Attribut	f_1	$\ldots$	f_d	φ_1	$\ldots$	φ_H
Attributwerte				g_1^1 $\ldots$ g_K^1		g_1^H $\ldots$ g_K^H

Konsensusfunktion Ko-Assoziation

Ko-Assotiationsmatrix

- neue Attribute für jede Beobachtung in co-Matrix eintragen
- für jedes Paar (x_i, x_j) und jedes Attribut Gleichheit der Attributwerte feststellen

$$d(a,b) = \begin{cases} 1, & \text{falls } a = b \\ 0, & \text{falls } a \neq b \end{cases}$$

- Zählen, wie oft (x_i, x_j) den gleichen Attributwert haben!

$$sim(x,y) = \sum_{i=1} d(\varphi_i(x), \varphi_i(y))$$

co	φ_1	...	φ_H
x_1	g_i		g_j
...			
x_N	g_i		g_j

$\delta(x_1, x_N) = 1$ für φ_1

$\delta(x_1, x_N) = 0$ für φ_H

sim	x_1	...	x_N
x_1	-		$sim(x_1, x_N)$
...			
x_N	$sim(x_1, x_N)$		-

Konsensusfunktion Ko-Assoziation

Ko-Assotiationsmatrix

- neue Attribute für jede Beobachtung in co-Matrix eintragen
- für jedes Paar (x_i, x_j) und jedes Attribut Gleichheit der Attributwerte feststellen

$$d(a,b) = \begin{cases} 1, & \text{falls } a = b \\ 0, & \text{falls } a \neq b \end{cases}$$

- Zählen, wie oft (x_i, x_j) den gleichen Attributwert haben!

$$sim(x,y) = \sum_{i=1} d(\varphi_i(x), \varphi_i(y))$$

co	φ_1	...	φ_H
x_1	g_i		g_j
...			
x_N	g_i		g_j

$\delta(x_1, x_N) = 1$ für φ_1

$\delta(x_1, x_N) = 0$ für φ_H

sim	x_1	...	x_N
x_1	-		$sim(x_1, x_N)$
...			
x_N	$sim(x_1, x_N)$		-

Konsensusfunktion Ko-Assoziation

Ko-Assotiationsmatrix

- neue Attribute für jede Beobachtung in co-Matrix eintragen
- für jedes Paar (x_i, x_j) und jedes Attribut Gleichheit der Attributwerte feststellen

$$d(a,b) = \begin{cases} 1, & \text{falls } a = b \\ 0, & \text{falls } a \neq b \end{cases}$$

- Zählen, wie oft (x_i, x_j) den gleichen Attributwert haben!

$$sim(x,y) = \sum_{i=1} d(\varphi_i(x), \varphi_i(y))$$

co	φ_1	...	φ_H
x_1	g_i		g_j
...			
x_N	g_i		g_j

$\delta(x_1, x_N) = 1$ für φ_1

$\delta(x_1, x_N) = 0$ für φ_H

sim	x_1	...	x_N
x_1	-		$sim(x_1, x_N)$
...			
x_N	$sim(x_1, x_N)$		-

Konsensusfunktion Ko-Assoziation

Ko-Assotiationsmatrix

- neue Attribute für jede Beobachtung in co-Matrix eintragen
- für jedes Paar (x_i, x_j) und jedes Attribut Gleichheit der Attributwerte feststellen

$$d(a,b) = \begin{cases} 1, & \text{falls } a = b \\ 0, & \text{falls } a \neq b \end{cases}$$

- Zählen, wie oft (x_i, x_j) den gleichen Attributwert haben!

$$sim(x,y) = \sum_{i=1} d(\varphi_i(x), \varphi_i(y))$$

co	φ_1	...	φ_H
x_1	g_i		g_j
...			
x_N	g_i		g_j

$\delta(x_1, x_N) = 1$ für φ_1

$\delta(x_1, x_N) = 0$ für φ_H

sim	x_1	...	x_N
x_1	-		$sim(x_1, x_N)$
...			
x_N	$sim(x_1, x_N)$		-

Konsensusfunktion Ko-Assoziation

Ko-Assotiationsmatrix

- neue Attribute für jede Beobachtung in co-Matrix eintragen
- für jedes Paar (x_i, x_j) und jedes Attribut Gleichheit der Attributwerte feststellen

$$d(a,b) = \begin{cases} 1, & \text{falls } a = b \\ 0, & \text{falls } a \neq b \end{cases}$$

- Zählen, wie oft (x_i, x_j) den gleichen Attributwert haben!

$$sim(x,y) = \sum_{i=1} d(\varphi_i(x), \varphi_i(y))$$

co	φ_1	...	φ_H
x_1	g_i		g_j
...			
x_N	g_i		g_j

$\delta(x_1, x_N) = 1$ für φ_1

$\delta(x_1, x_N) = 0$ für φ_H

sim	x_1	...	x_N
x_1	-		$sim(x_1, x_N)$
...			
x_N	$sim(x_1, x_N)$		-

Konsensusfunktion Ko-Assoziation

Ko-Assotiationsmatrix

- neue Attribute für jede Beobachtung in co-Matrix eintragen
- für jedes Paar (x_i, x_j) und jedes Attribut Gleichheit der Attributwerte feststellen

$$d(a,b) = \begin{cases} 1, & \text{falls } a = b \\ 0, & \text{falls } a \neq b \end{cases}$$

- Zählen, wie oft (x_i, x_j) den gleichen Attributwert haben!

$$sim(x,y) = \sum_{i=1} d(\varphi_i(x), \varphi_i(y))$$

co	φ_1	...	φ_H
x_1	g_i		g_j
...			
x_N	g_i		g_j

$\delta(x_1, x_N) = 1$ für φ_1

$\delta(x_1, x_N) = 0$ für φ_H

sim	x_1	...	x_N
x_1	-		$sim(x_1, x_N)$
...			
x_N	$sim(x_1, x_N)$		-

Algorithmus für den Konsens

Beliebiger clustering Algorithmus verwendet $sim(x, y)$ als Ähnlichkeitsmaß und

- maximiert Ähnlichkeit aller x_i im cluster und
- minimiert Ähnlichkeit aller x_i zwischen clusters.

Algorithmus für den Konsens

Beliebiger clustering Algorithmus verwendet $sim(x, y)$ als Ähnlichkeitsmaß und

- maximiert Ähnlichkeit aller x_i im cluster und
- minimiert Ähnlichkeit aller x_i zwischen clusters.

Konsensusfunktion Nutzen

Vergleich des Kandidaten-clusterings $\sigma = \{c_1, \ldots, c_K\}$ mit einem Eingabe-clustering $\varphi_i = \{g_1^i, \ldots, g_{K(i)}^i\}$: kann jedes cluster g_i der Eingabe (jedes Attribut) besser vorhergesagt werden, wenn man σ kennt?

$$U(\sigma, \varphi_i) = \sum_{r=1}^{K} p(c_r) \sum_{j=1}^{K(i)} p(g_j^i | c_r)^2 - \sum_{j=1}^{K} (i) p(g_j^i)^2$$

Nutzen von σ für alle φ:

$$U(\sigma, G) = \sum_{i=1}^{H} U(\sigma, \varphi_i)$$

$$p(c_r) = \frac{|c_r|}{N}$$

$$p(g_j^i | c_r) = \frac{|g_j^i \cap c_r|}{|c_r|}$$

$$p(g_j^i) = \frac{|g_j^i|}{N}$$

Wähle den Kandidaten mit dem größten Nutzen!

$$s_{\text{best}} = \arg\max_s U(s, G)$$

Konsensusfunktion Nutzen

Vergleich des Kandidaten-clusterings $\sigma = \{c_1, \ldots, c_K\}$ mit einem Eingabe-clustering $\varphi_i = \{g^i_1, \ldots, g^i_{K(i)}\}$: kann jedes cluster g_i der Eingabe (jedes Attribut) besser vorhergesagt werden, wenn man σ kennt?

$$U(\sigma, \varphi_i) = \sum_{r=1}^{K} p(c_r) \sum_{j=1}^{K(i)} p(g^i_j | c_r)^2 - \sum_{j=1}^{K} (i) p(g^i_j)^2$$

Nutzen von σ für alle φ:

$$U(\sigma, G) = \sum_{i=1}^{H} U(\sigma, \varphi_i)$$

Wähle den Kandidaten mit dem größten Nutzen!

$$s_{\text{best}} = \arg\max_s U(s, G)$$

Konsensusfunktion Nutzen

Vergleich des Kandidaten-clusterings $\sigma = \{c_1, \ldots, c_K\}$ mit einem Eingabe-clustering $\varphi_i = \{g_1^i, \ldots, g_{K(i)}^i\}$: kann jedes cluster g_i der Eingabe (jedes Attribut) besser vorhergesagt werden, wenn man σ kennt?

$$U(\sigma, \varphi_i) = \sum_{r=1}^{K} p(c_r) \sum_{j=1}^{K(i)} p(g_j^i | c_r)^2 - \sum_{j=1}^{K} (i) p(g_j^i)^2$$

Nutzen von σ für alle φ:

$$U(\sigma, G) = \sum_{i=1}^{H} U(\sigma, \varphi_i)$$

Wähle den Kandidaten mit dem größten Nutzen!

$$s_{\text{best}} = \arg\max_s U(s, G)$$

$$p(c_r) = \frac{|c_r|}{N}$$

$$p(g_j^i | c_r) = \frac{|g_j^i \cap c_r|}{|c_r|}$$

$$p(g_j^i) = \frac{|g_j^i|}{N}$$

Konsensusfunktion Nutzen

Vergleich des Kandidaten-clusterings $\sigma = \{c_1, \ldots, c_K\}$ mit einem Eingabe-clustering $\varphi_i = \{g_1^i, \ldots, g_{K(i)}^i\}$: kann jedes cluster g_i der Eingabe (jedes Attribut) besser vorhergesagt werden, wenn man σ kennt?

$$U(\sigma, \varphi_i) = \sum_{r=1}^{K} p(c_r) \sum_{j=1}^{K(i)} p(g_j^i | c_r)^2 - \sum_{j=1}^{K} (i) p(g_j^i)^2$$

Nutzen von σ für alle φ:

$$U(\sigma, G) = \sum_{i=1}^{H} U(\sigma, \varphi_i)$$

Wähle den Kandidaten mit dem größten Nutzen!

$$s_{\text{best}} = \arg\max_s U(s, G)$$

$$p(c_r) = \frac{|c_r|}{N}$$

$$p(g_j^i | c_r) = \frac{|g_j^i \cap c_r|}{|c_r|}$$

$$p(g_j^i) = \frac{|g_j^i|}{N}$$

Preprocessing für k-Means

- Wir können k-Means anwenden, wenn wir neue (numerische) Attribute formen! Wir müssen die Anzahl der Cluster vorgeben.
- Eigentlich haben wir die schon...
  für jede Eingabe φ: für jedes cluster g ein binäres Attribut
  $\delta(g_1, \varphi_i(x)) = 1$, falls $\varphi_i(x) = g_1, \delta(g_1, \varphi_i(x)) = 0$ sonst
- Einträge sind $[-1, 0[$, falls $\varphi_i(x) \neq g_j$, sonst $]0, 1]$

	g_1	...	g_K
x_1	$\delta(g_1, \varphi_i(x_i)) - p(g_1)$		$\delta(g_K, \varphi_i(x_1)) - p(g_K)$
...			
x_N	$\delta(g_1, \varphi_i(x_N)) - p(g_1)$		$\delta(g_K, \varphi_i(x_N)) - p(g_K)$

Preprocessing für k-Means

- Wir können k-Means anwenden, wenn wir neue (numerische) Attribute formen! Wir müssen die Anzahl der Cluster vorgeben.
- Eigentlich haben wir die schon...
  für jede Eingabe φ: für jedes cluster g ein binäres Attribut
  $\delta(g_1, \varphi_i(x)) = 1$, falls $\varphi_i(x) = g_1, \delta(g_1, \varphi_i(x)) = 0$ sonst
- Einträge sind $[-1, 0[$, falls $\varphi_i(x) \neq g_j$, sonst $]0, 1]$

	g_1	...	g_K
x_1	$\delta(g_1, \varphi_i(x_i)) - p(g_1)$		$\delta(g_K, \varphi_i(x_1)) - p(g_K)$
...			
x_N	$\delta(g_1, \varphi_i(x_N)) - p(g_1)$		$\delta(g_K, \varphi_i(x_N)) - p(g_K)$

Preprocessing für k-Means

- Wir können k-Means anwenden, wenn wir neue (numerische) Attribute formen! Wir müssen die Anzahl der Cluster vorgeben.
- Eigentlich haben wir die schon...
  für jede Eingabe φ: für jedes cluster g ein binäres Attribut
  $\delta(g_1, \varphi_i(x)) = 1$, falls $\varphi_i(x) = g_1, \delta(g_1, \varphi_i(x)) = 0$ sonst
- Einträge sind $[-1, 0[$, falls $\varphi_i(x) \neq g_j$, sonst $]0, 1]$

	g_1	...	g_K
x_1	$\delta(g_1, \varphi_i(x_i)) - p(g_1)$		$\delta(g_K, \varphi_i(x_1)) - p(g_K)$
...			
x_N	$\delta(g_1, \varphi_i(x_N)) - p(g_1)$		$\delta(g_K, \varphi_i(x_N)) - p(g_K)$

Fragen

Kann aus mehreren sehr schwachen clusterings per Konsensusfunktion ein sehr gutes clustering gemacht werden?

- Einfachste Eingabe-clusterings – wieviele?
- Konsensusfunktionen – wann ist welche besser?
  - Ko-Assoziation
  - Nutzen

Fehlermaß bei klassifizierten Daten: wieviele falsch eingeordnet?

- Iris, 3 Klassen, 150 Beobachtungen
- 2 Spiralen, 2 Klassen, 200 Beobachtungen
- 2 Halbringe, 2 Klassen, 400 Beobachtungen (ungleich verteilt)
- Galaxie, 2 Klassen, 4190 Beobachtungen

Fragen

Kann aus mehreren sehr schwachen clusterings per Konsensusfunktion ein sehr gutes clustering gemacht werden?

- Einfachste Eingabe-clusterings – wieviele?
- Konsensusfunktionen – wann ist welche besser?
  - Ko-Assoziation
  - Nutzen

Fehlermaß bei klassifizierten Daten: wieviele falsch eingeordnet?

- Iris, 3 Klassen, 150 Beobachtungen
- 2 Spiralen, 2 Klassen, 200 Beobachtungen
- 2 Halbringe, 2 Klassen, 400 Beobachtungen (ungleich verteilt)
- Galaxie, 2 Klassen, 4190 Beobachtungen

Fragen

Kann aus mehreren sehr schwachen clusterings per Konsensusfunktion ein sehr gutes clustering gemacht werden?

- Einfachste Eingabe-clusterings – wieviele?
- Konsensusfunktionen – wann ist welche besser?
  - Ko-Assoziation
  - Nutzen

Fehlermaß bei klassifizierten Daten: wieviele falsch eingeordnet?

- Iris, 3 Klassen, 150 Beobachtungen
- 2 Spiralen, 2 Klassen, 200 Beobachtungen
- 2 Halbringe, 2 Klassen, 400 Beobachtungen (ungleich verteilt)
- Galaxie, 2 Klassen, 4190 Beobachtungen

Fragen

Kann aus mehreren sehr schwachen clusterings per Konsensusfunktion ein sehr gutes clustering gemacht werden?

- Einfachste Eingabe-clusterings – wieviele?
- Konsensusfunktionen – wann ist welche besser?
  - Ko-Assoziation
  - Nutzen

Fehlermaß bei klassifizierten Daten: wieviele falsch eingeordnet?

- Iris, 3 Klassen, 150 Beobachtungen
- 2 Spiralen, 2 Klassen, 200 Beobachtungen
- 2 Halbringe, 2 Klassen, 400 Beobachtungen (ungleich verteilt)
- Galaxie, 2 Klassen, 4190 Beobachtungen

Fragen

Kann aus mehreren sehr schwachen clusterings per Konsensusfunktion ein sehr gutes clustering gemacht werden?

- Einfachste Eingabe-clusterings – wieviele?
- Konsensusfunktionen – wann ist welche besser?
  - Ko-Assoziation
  - Nutzen

Fehlermaß bei klassifizierten Daten: wieviele falsch eingeordnet?

- Iris, 3 Klassen, 150 Beobachtungen
- 2 Spiralen, 2 Klassen, 200 Beobachtungen
- 2 Halbringe, 2 Klassen, 400 Beobachtungen (ungleich verteilt)
- Galaxie, 2 Klassen, 4190 Beobachtungen

Fragen

Kann aus mehreren sehr schwachen clusterings per Konsensusfunktion ein sehr gutes clustering gemacht werden?

- Einfachste Eingabe-clusterings – wieviele?
- Konsensusfunktionen – wann ist welche besser?
  - Ko-Assoziation
  - Nutzen

Fehlermaß bei klassifizierten Daten: wieviele falsch eingeordnet?

- Iris, 3 Klassen, 150 Beobachtungen
- 2 Spiralen, 2 Klassen, 200 Beobachtungen
- 2 Halbringe, 2 Klassen, 400 Beobachtungen (ungleich verteilt)
- Galaxie, 2 Klassen, 4190 Beobachtungen

Fragen

Kann aus mehreren sehr schwachen clusterings per Konsensusfunktion ein sehr gutes clustering gemacht werden?

- Einfachste Eingabe-clusterings – wieviele?
- Konsensusfunktionen – wann ist welche besser?
  - Ko-Assoziation
  - Nutzen

Fehlermaß bei klassifizierten Daten: wieviele falsch eingeordnet?

- Iris, 3 Klassen, 150 Beobachtungen
- 2 Spiralen, 2 Klassen, 200 Beobachtungen
- 2 Halbringe, 2 Klassen, 400 Beobachtungen (ungleich verteilt)
- Galaxie, 2 Klassen, 4190 Beobachtungen

Fragen

Kann aus mehreren sehr schwachen clusterings per Konsensusfunktion ein sehr gutes clustering gemacht werden?

- Einfachste Eingabe-clusterings – wieviele?
- Konsensusfunktionen – wann ist welche besser?
  - Ko-Assoziation
  - Nutzen

Fehlermaß bei klassifizierten Daten: wieviele falsch eingeordnet?

- Iris, 3 Klassen, 150 Beobachtungen
- 2 Spiralen, 2 Klassen, 200 Beobachtungen
- 2 Halbringe, 2 Klassen, 400 Beobachtungen (ungleich verteilt)
- Galaxie, 2 Klassen, 4190 Beobachtungen

Experimente Ko-Assoziation

Einfachste Eingabe-clusterings:

- Projektion der Beobachtungen auf 1 Dimension, zufällig ausgewählt
- k-Means auf den projizierten Daten (schnell)

Konsensusfunktion Ko-Assoziation mit

- single-link,
  i. e. $\min\{dist(a,b)\}, a \in c_2, b \in c_2$
  bester Alg. bei Spiralen, Halbringen, schlecht bei Iris
- complete link
  i. e. $\max\{dist(a,b)\}$
  gut bei Iris, schlecht ansonsten
- average link
  i. e. $\text{avg}\{dist(a,b)\}$
  gut bei Iris, schlecht ansonsten

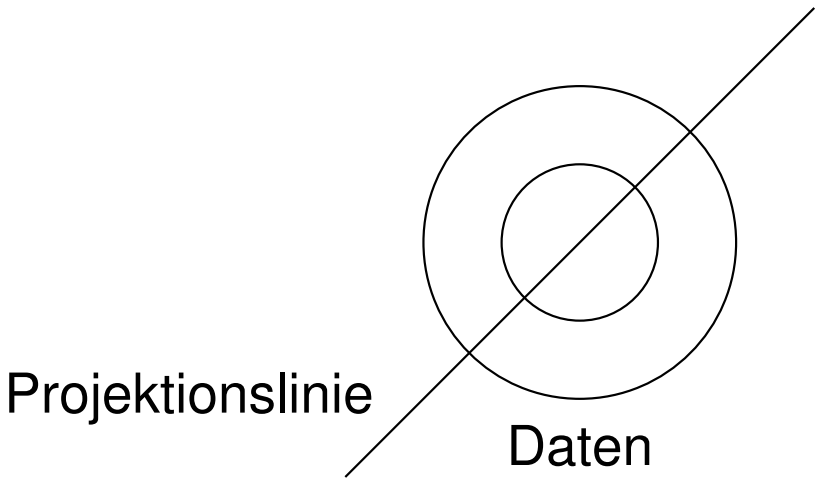
Bei Anzahl der Eingabe-clusterings > 50 sind die Ergebnisse besser als bestes einmal clustern der Originaldaten:

- z.B. Halbringe
  bestStandard 5,25% Fehler,
  EnsembleClustering 0 Fehler

Experimente Ko-Assoziation

Einfachste Eingabe-clusterings:

- Projektion der Beobachtungen auf 1 Dimension, zufällig ausgewählt
- k-Means auf den projizierten Daten (schnell)



Konsensusfunktion Ko-Assoziation mit

- single-link,
  i. e. $\min\{dist(a,b)\}, a \in c_2, b \in c_2$
  bester Alg. bei Spiralen, Halbringen,
  schlecht bei Iris
- complete link
  i. e. $\max\{dist(a,b)\}$
  gut bei Iris, schlecht ansonsten
- average link
  i. e. $\text{avg}\{dist(a,b)\}$
  gut bei Iris, schlecht ansonsten

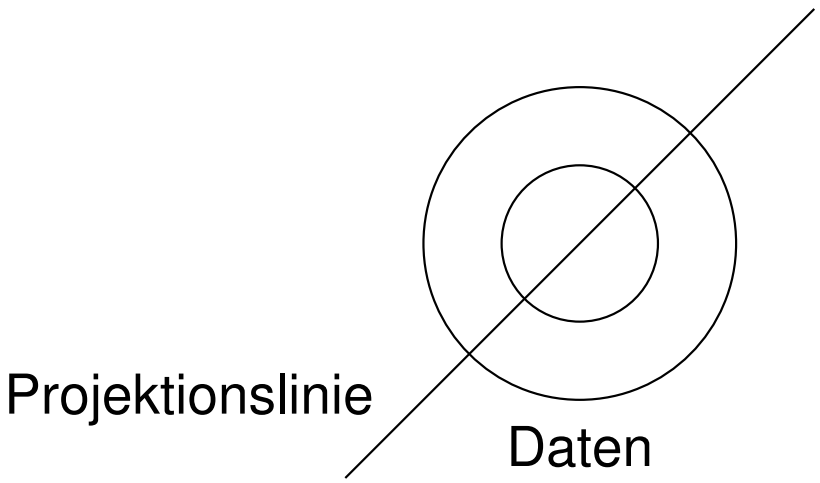
Bei Anzahl der Eingabe-clusterings > 50 sind die Ergebnisse besser als bestes einmal clustern der Originaldaten:

- z.B. Halbringe
  bestStandard 5,25% Fehler,
  EnsembleClustering 0 Fehler

Experimente Ko-Assoziation

Einfachste Eingabe-clusterings:

- Projektion der Beobachtungen auf 1 Dimension, zufällig ausgewählt
- k-Means auf den projizierten Daten (schnell)



Konsensusfunktion Ko-Assoziation mit

- single-link,
  i. e. $\min\{dist(a,b)\}, a \in c_2, b \in c_2$
  bester Alg. bei Spiralen, Halbringen,
  schlecht bei Iris
- complete link
  i. e. $\max\{dist(a,b)\}$
  gut bei Iris, schlecht ansonsten
- average link
  i. e. $\text{avg}\{dist(a,b)\}$
  gut bei Iris, schlecht ansonsten

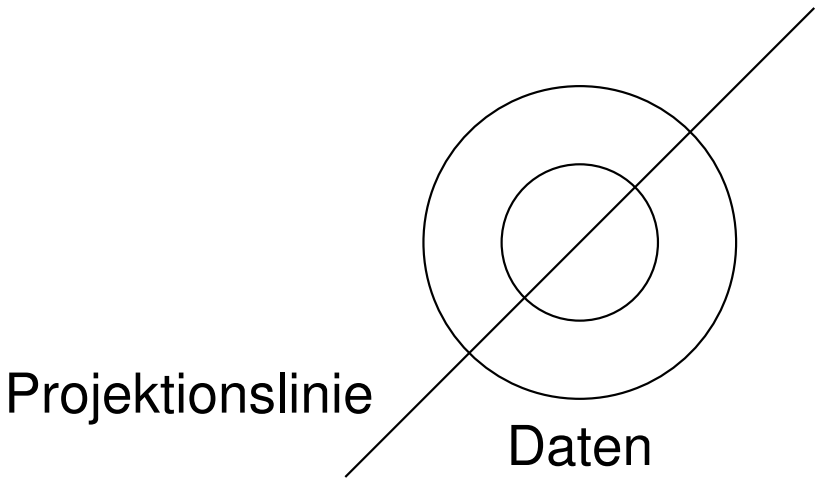
Bei Anzahl der Eingabe-clusterings > 50 sind die Ergebnisse besser als bestes einmal clustern der Originaldaten:

- z.B. Halbringe
  bestStandard 5,25% Fehler,
  EnsembleClustering 0 Fehler

Experimente Ko-Assoziation

Einfachste Eingabe-clusterings:

- Projektion der Beobachtungen auf 1 Dimension, zufällig ausgewählt
- k-Means auf den projizierten Daten (schnell)



Konsensusfunktion Ko-Assoziation mit

- single-link,
  i. e. $\min\{dist(a,b)\}, a \in c_2, b \in c_2$
  bester Alg. bei Spiralen, Halbringen,
  schlecht bei Iris
- complete link
  i. e. $\max\{dist(a,b)\}$
  gut bei Iris, schlecht ansonsten
- average link
  i. e. $\text{avg}\{dist(a,b)\}$
  gut bei Iris, schlecht ansonsten

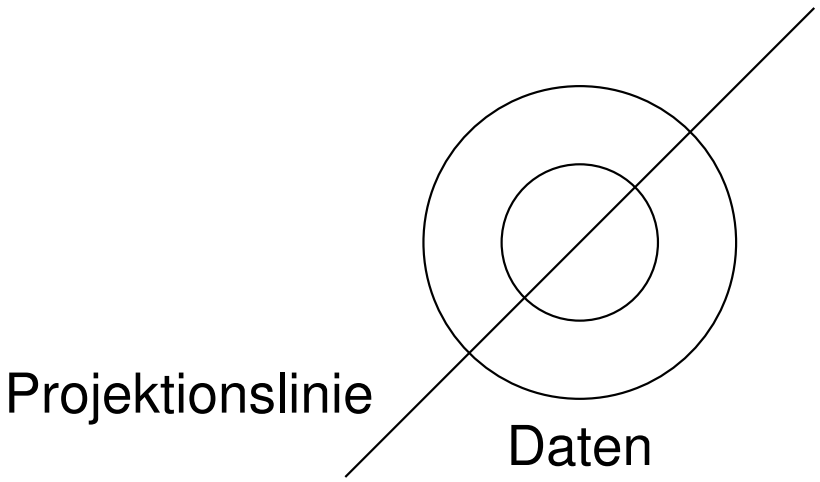
Bei Anzahl der Eingabe-clusterings > 50 sind die Ergebnisse besser als bestes einmal clustern der Originaldaten:

- z.B. Halbringe
  bestStandard 5,25% Fehler,
  EnsembleClustering 0 Fehler

Experimente Ko-Assoziation

Einfachste Eingabe-clusterings:

- Projektion der Beobachtungen auf 1 Dimension, zufällig ausgewählt
- k-Means auf den projizierten Daten (schnell)



Konsensusfunktion Ko-Assoziation mit

- single-link,
  i. e. $\min\{dist(a,b)\}, a \in c_2, b \in c_2$
  bester Alg. bei Spiralen, Halbringen, schlecht bei Iris
- complete link
  i. e. $\max\{dist(a,b)\}$
  gut bei Iris, schlecht ansonsten
- average link
  i. e. $\text{avg}\{dist(a,b)\}$
  gut bei Iris, schlecht ansonsten

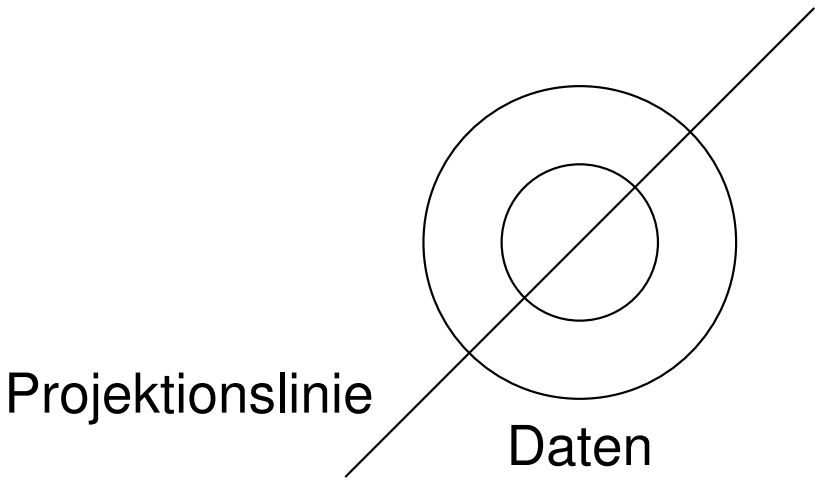
Bei Anzahl der Eingabe-clusterings > 50 sind die Ergebnisse besser als bestes einmal clustern der Originaldaten:

- z.B. Halbringe
  bestStandard 5,25% Fehler,
  EnsembleClustering 0 Fehler

Experimente Ko-Assoziation

Einfachste Eingabe-clusterings:

- Projektion der Beobachtungen auf 1 Dimension, zufällig ausgewählt
- k-Means auf den projizierten Daten (schnell)



Konsensusfunktion Ko-Assoziation mit

- single-link,
  i. e. $\min\{dist(a,b)\}, a \in c_2, b \in c_2$
  bester Alg. bei Spiralen, Halbringen, schlecht bei Iris
- complete link
  i. e. $\max\{dist(a,b)\}$
  gut bei Iris, schlecht ansonsten
- average link
  i. e. $\text{avg}\{dist(a,b)\}$
  gut bei Iris, schlecht ansonsten

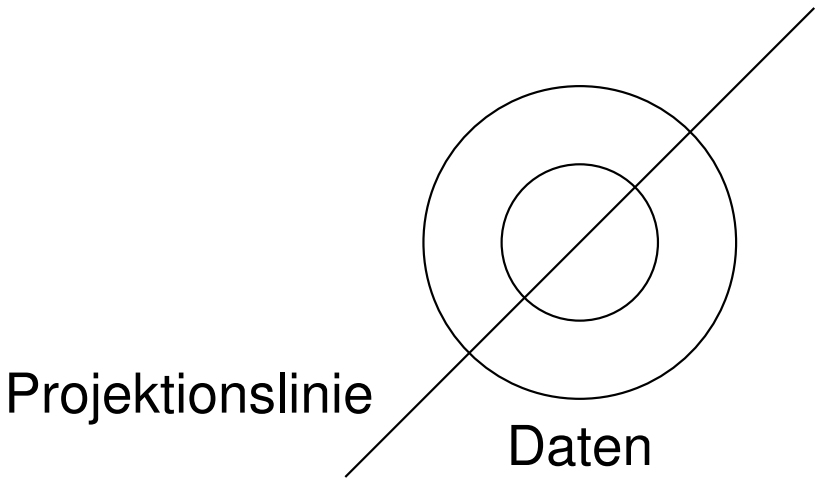
Bei Anzahl der Eingabe-clusterings > 50 sind die Ergebnisse besser als bestes einmal clustern der Originaldaten:

- z.B. Halbringe
  bestStandard 5,25% Fehler,
  EnsembleClustering 0 Fehler

Experimente Ko-Assoziation

Einfachste Eingabe-clusterings:

- Projektion der Beobachtungen auf 1 Dimension, zufällig ausgewählt
- k-Means auf den projizierten Daten (schnell)



Konsensusfunktion Ko-Assoziation mit

- single-link,
  i. e. $\min\{dist(a,b)\}, a \in c_2, b \in c_2$
  bester Alg. bei Spiralen, Halbringen, schlecht bei Iris
- complete link
  i. e. $\max\{dist(a,b)\}$
  gut bei Iris, schlecht ansonsten
- average link
  i. e. $\text{avg}\{dist(a,b)\}$
  gut bei Iris, schlecht ansonsten

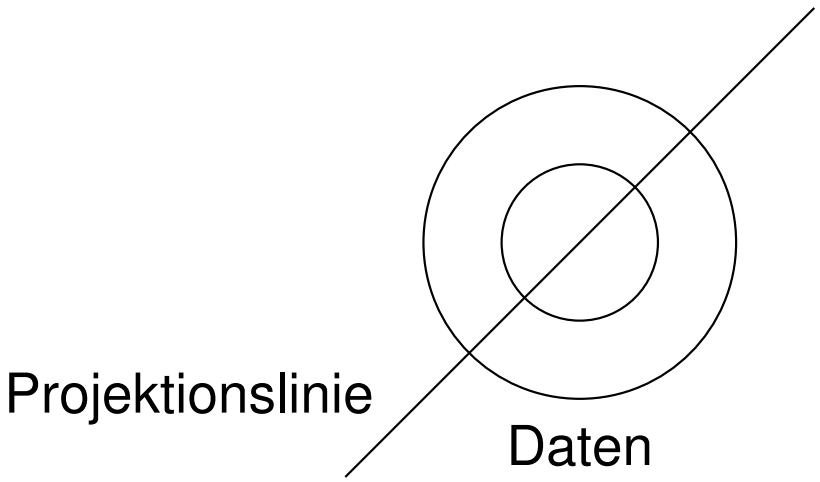
Bei Anzahl der Eingabe-clusterings > 50 sind die Ergebnisse besser als bestes einmal clustern der Originaldaten:

- z.B. Halbringe
  bestStandard 5,25% Fehler,
  EnsembleClustering 0 Fehler

Experimente Ko-Assoziation

Einfachste Eingabe-clusterings:

- Projektion der Beobachtungen auf 1 Dimension, zufällig ausgewählt
- k-Means auf den projizierten Daten (schnell)



Konsensusfunktion Ko-Assoziation mit

- single-link,
  i. e. $\min\{dist(a,b)\}, a \in c_2, b \in c_2$
  bester Alg. bei Spiralen, Halbringen, schlecht bei Iris
- complete link
  i. e. $\max\{dist(a,b)\}$
  gut bei Iris, schlecht ansonsten
- average link
  i. e. $\text{avg}\{dist(a,b)\}$
  gut bei Iris, schlecht ansonsten

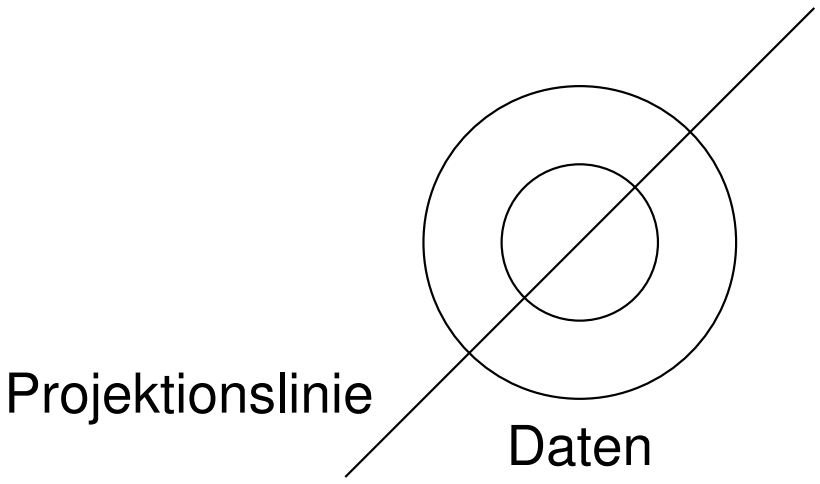
Bei Anzahl der Eingabe-clusterings > 50 sind die Ergebnisse besser als bestes einmal clustern der Originaldaten:

- z.B. Halbringe
  bestStandard 5,25% Fehler,
  EnsembleClustering 0 Fehler

Experimente Ko-Assoziation

Einfachste Eingabe-clusterings:

- Projektion der Beobachtungen auf 1 Dimension, zufällig ausgewählt
- k-Means auf den projizierten Daten (schnell)



Konsensusfunktion Ko-Assoziation mit

- single-link,
  i. e. $\min\{dist(a,b)\}, a \in c_2, b \in c_2$
  bester Alg. bei Spiralen, Halbringen, schlecht bei Iris
- complete link
  i. e. $\max\{dist(a,b)\}$
  gut bei Iris, schlecht ansonsten
- average link
  i. e. $\text{avg}\{dist(a,b)\}$
  gut bei Iris, schlecht ansonsten

Bei Anzahl der Eingabe-clusterings > 50 sind die Ergebnisse besser als bestes einmal clustern der Originaldaten:

- z.B. Halbringe
  bestStandard 5,25% Fehler,
  EnsembleClustering 0 Fehler

Experimente Nutzen

Einfachste Eingabe-clusterings:

- Projektion der Beobachtungen auf 1 Dimension, zufällig ausgewählt
- k-Means auf den projizierten Daten (schnell)

Konsensusfunktion Nutzen

- k-Means
- bester Alg. bei Galaxie

Günstig bei

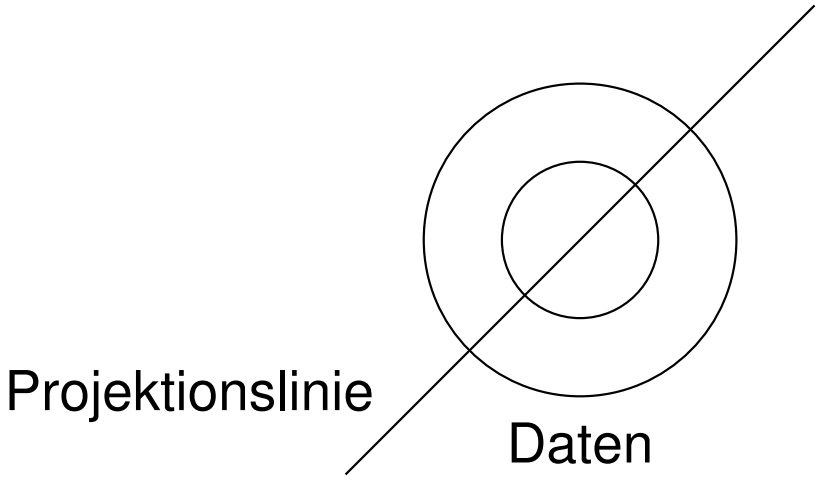
- vielen Beobachtungen (etwa $N > 150$)
- wenigen Eingabe-clusterings (etwa $H < 50$)

Laufzeit in $O(kNH)$

Experimente Nutzen

Einfachste Eingabe-clusterings:

- Projektion der Beobachtungen auf 1 Dimension, zufällig ausgewählt
- k-Means auf den projizierten Daten (schnell)



Konsensusfunktion Nutzen

- k-Means
- bester Alg. bei Galaxie

Günstig bei

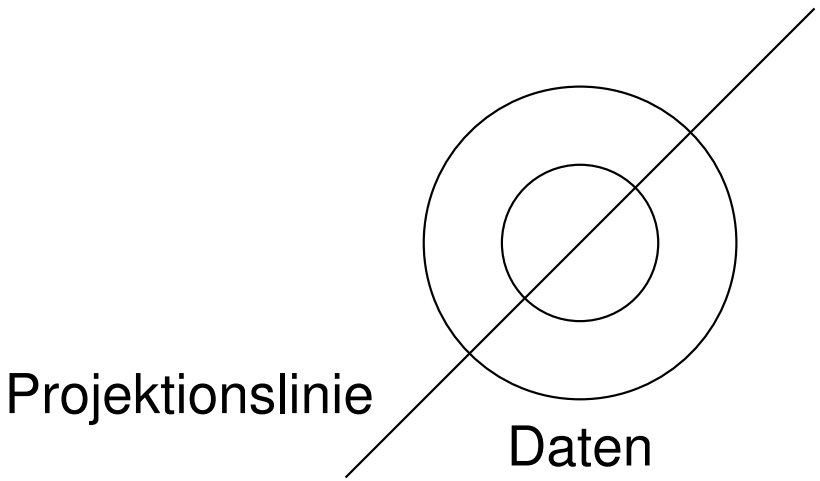
- vielen Beobachtungen (etwa $N > 150$)
- wenigen Eingabe-clusterings (etwa $H < 50$)

Laufzeit in $O(kNH)$

Experimente Nutzen

Einfachste Eingabe-clusterings:

- Projektion der Beobachtungen auf 1 Dimension, zufällig ausgewählt
- k-Means auf den projizierten Daten (schnell)



Konsensusfunktion Nutzen

- k-Means
- bester Alg. bei Galaxie

Günstig bei

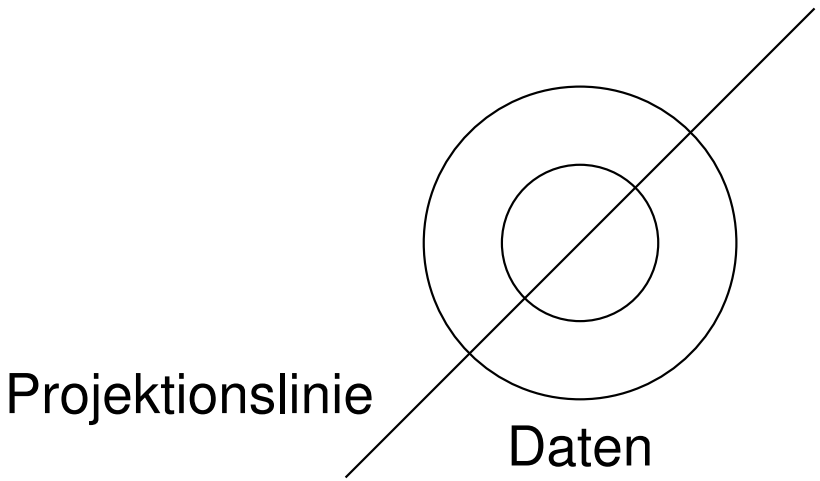
- vielen Beobachtungen (etwa $N > 150$)
- wenigen Eingabe-clusterings (etwa $H < 50$)

Laufzeit in $O(kNH)$

Experimente Nutzen

Einfachste Eingabe-clusterings:

- Projektion der Beobachtungen auf 1 Dimension, zufällig ausgewählt
- k-Means auf den projizierten Daten (schnell)



Konsensusfunktion Nutzen

- k-Means
- bester Alg. bei Galaxie

Günstig bei

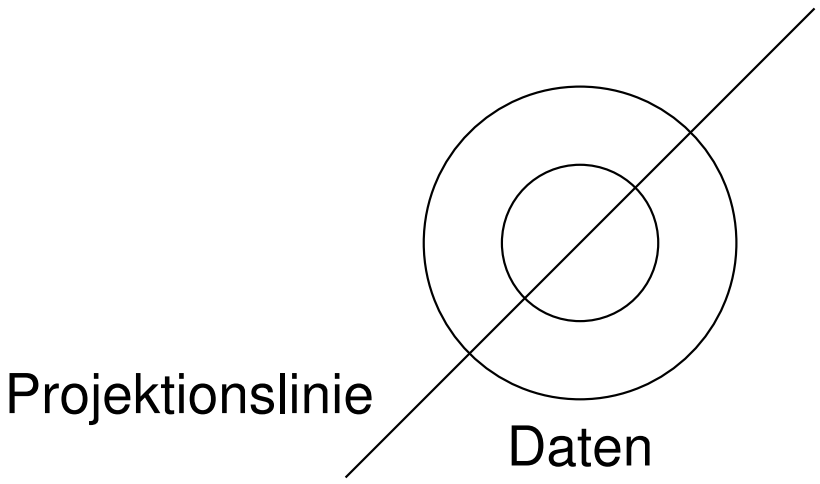
- vielen Beobachtungen (etwa $N > 150$)
- wenigen Eingabe-clusterings (etwa $H < 50$)

Laufzeit in $O(kNH)$

Experimente Nutzen

Einfachste Eingabe-clusterings:

- Projektion der Beobachtungen auf 1 Dimension, zufällig ausgewählt
- k-Means auf den projizierten Daten (schnell)



Konsensusfunktion Nutzen

- k-Means
- bester Alg. bei Galaxie

Günstig bei

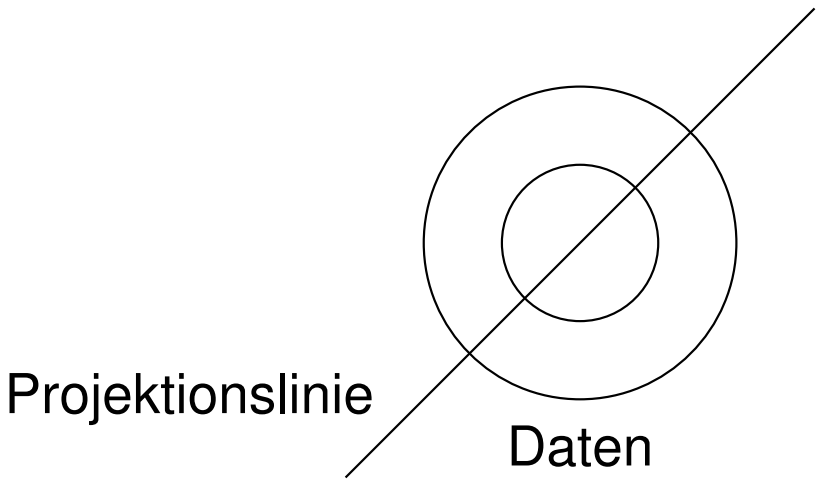
- vielen Beobachtungen (etwa $N > 150$)
- wenigen Eingabe-clusterings (etwa $H < 50$)

Laufzeit in $O(kNH)$

Experimente Nutzen

Einfachste Eingabe-clusterings:

- Projektion der Beobachtungen auf 1 Dimension, zufällig ausgewählt
- k-Means auf den projizierten Daten (schnell)



Konsensusfunktion Nutzen

- k-Means
- bester Alg. bei Galaxie

Günstig bei

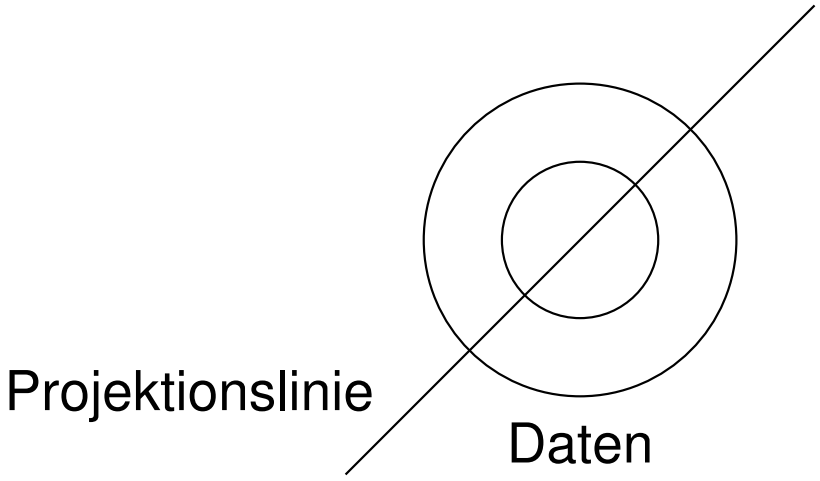
- vielen Beobachtungen (etwa $N > 150$)
- wenigen Eingabe-clusterings (etwa $H < 50$)

Laufzeit in $O(kNH)$

Experimente Nutzen

Einfachste Eingabe-clusterings:

- Projektion der Beobachtungen auf 1 Dimension, zufällig ausgewählt
- k-Means auf den projizierten Daten (schnell)



Konsensusfunktion Nutzen

- k-Means
- bester Alg. bei Galaxie

Günstig bei

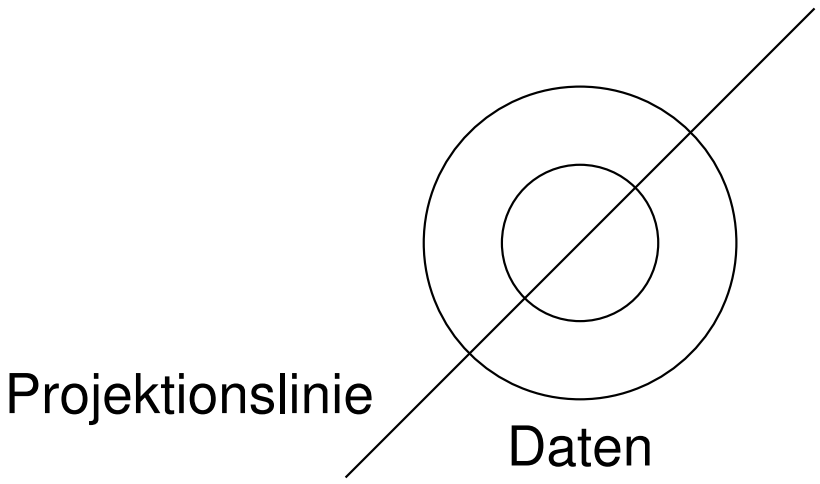
- vielen Beobachtungen (etwa $N > 150$)
- wenigen Eingabe-clusterings (etwa $H < 50$)

Laufzeit in $O(kNH)$

Experimente Nutzen

Einfachste Eingabe-clusterings:

- Projektion der Beobachtungen auf 1 Dimension, zufällig ausgewählt
- k-Means auf den projizierten Daten (schnell)



Konsensusfunktion Nutzen

- k-Means
- bester Alg. bei Galaxie

Günstig bei

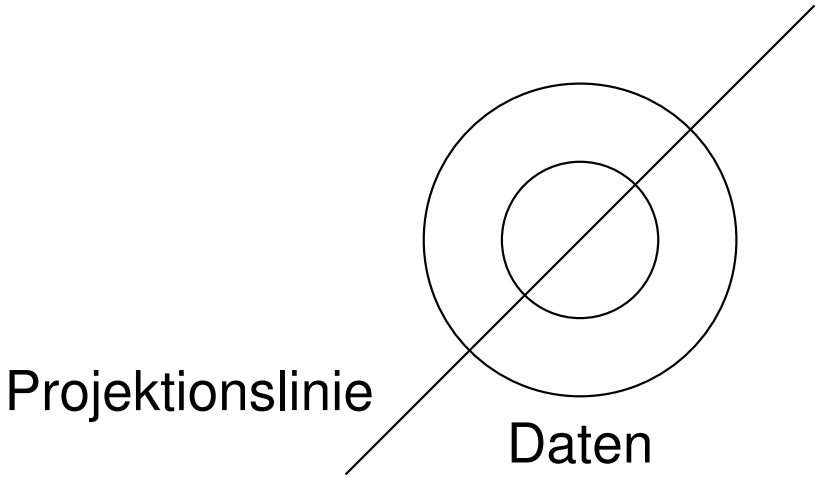
- vielen Beobachtungen (etwa $N > 150$)
- wenigen Eingabe-clusterings (etwa $H < 50$)

Laufzeit in $O(kNH)$

Experimente Nutzen

Einfachste Eingabe-clusterings:

- Projektion der Beobachtungen auf 1 Dimension, zufällig ausgewählt
- k-Means auf den projizierten Daten (schnell)



Konsensusfunktion Nutzen

- k-Means
- bester Alg. bei Galaxie

Günstig bei

- vielen Beobachtungen (etwa $N > 150$)
- wenigen Eingabe-clusterings (etwa $H < 50$)

Laufzeit in $O(kNH)$

Was wissen Sie jetzt?

- Cluster Ensembles formen aus mehreren Eingabe-clusterings ein neues clustering
- Der Ansatz von [4] zeigt, dass
  - Ensembles von vielen, sehr einfachen clusterings besser sein können als ein „anspruchsvolles“ clustering.
  - Konsensusfunktionen wiederum cluster-Algorithmen sind, wenn man die Eingabe-clusterings zu Merkmalen macht.
- Sie wissen, wie man aus Eingabe-clusterings
  - Ko-Assoziationsmatrix oder
  - Matrizen mit binären Merkmalen macht (s. [2])

Was wissen Sie jetzt?

- Cluster Ensembles formen aus mehreren Eingabe-clusterings ein neues clustering
- Der Ansatz von [4] zeigt, dass
  - Ensembles von vielen, sehr einfachen clusterings besser sein können als ein „anspruchsvolles“ clustering.
  - Konsensusfunktionen wiederum cluster-Algorithmen sind, wenn man die Eingabe-clusterings zu Merkmalen macht.
- Sie wissen, wie man aus Eingabe-clusterings
  - Ko-Assoziationsmatrix oder
  - Matrizen mit binären Merkmalen macht (s. [2])

Was wissen Sie jetzt?

- Cluster Ensembles formen aus mehreren Eingabe-clusterings ein neues clustering
- Der Ansatz von [4] zeigt, dass
  - Ensembles von vielen, sehr einfachen clusterings besser sein können als ein „anspruchsvolles“ clustering.
  - Konsensusfunktionen wiederum cluster-Algorithmen sind, wenn man die Eingabe-clusterings zu Merkmalen macht.
- Sie wissen, wie man aus Eingabe-clusterings
  - Ko-Assoziationsmatrix oder
  - Matrizen mit binären Merkmalen macht (s. [2])

Was wissen Sie jetzt?

- Cluster Ensembles formen aus mehreren Eingabe-clusterings ein neues clustering
- Der Ansatz von [4] zeigt, dass
  - Ensembles von vielen, sehr einfachen clusterings besser sein können als ein „anspruchsvolles“ clustering.
  - Konsensusfunktionen wiederum cluster-Algorithmen sind, wenn man die Eingabe-clusterings zu Merkmalen macht.
- Sie wissen, wie man aus Eingabe-clusterings
  - Ko-Assoziationsmatrix oder
  - Matrizen mit binären Merkmalen macht (s. [2])

Was wissen Sie jetzt?

- Cluster Ensembles formen aus mehreren Eingabe-clusterings ein neues clustering
- Der Ansatz von [4] zeigt, dass
  - Ensembles von vielen, sehr einfachen clusterings besser sein können als ein „anspruchsvolles“ clustering.
  - Konsensusfunktionen wiederum cluster-Algorithmen sind, wenn man die Eingabe-clusterings zu Merkmalen macht.
- Sie wissen, wie man aus Eingabe-clusterings
  - Ko-Assoziationsmatrix oder
  - Matrizen mit binären Merkmalen macht (s. [2])

Was wissen Sie jetzt?

- Cluster Ensembles formen aus mehreren Eingabe-clusterings ein neues clustering
- Der Ansatz von [4] zeigt, dass
  - Ensembles von vielen, sehr einfachen clusterings besser sein können als ein „anspruchsvolles“ clustering.
  - Konsensusfunktionen wiederum cluster-Algorithmen sind, wenn man die Eingabe-clusterings zu Merkmalen macht.
- Sie wissen, wie man aus Eingabe-clusterings
  - Ko-Assoziationsmatrix oder
  - Matrizen mit binären Merkmalen macht (s. [2])

Was wissen Sie jetzt?

- Cluster Ensembles formen aus mehreren Eingabe-clusterings ein neues clustering
- Der Ansatz von [4] zeigt, dass
  - Ensembles von vielen, sehr einfachen clusterings besser sein können als ein „anspruchsvolles“ clustering.
  - Konsensusfunktionen wiederum cluster-Algorithmen sind, wenn man die Eingabe-clusterings zu Merkmalen macht.
- Sie wissen, wie man aus Eingabe-clusterings
  - Ko-Assoziationsmatrix oder
  - Matrizen mit binären Merkmalen macht (s. [2])

Ansatz von [3]

Informationstheoretische Konsensusfunktion

- Wechselseitige Information zweier clusterings
  Normalized mutual information

$$NMI(\phi_a, \phi_b) = \frac{2}{N} \sum_{i=1}^{k(a)} \sum_{j=1}^{k(b)} |g_i^a \cap g_j^b| \log_{k(a)k(b)} \left(\frac{|g_i^a \cap g_j^b|}{|g_i^a||g_j^b|} \right)$$

- Der beste Konsens hat am meisten gemeinsam mit den meisten Eingabe-clusterings:

$$\sigma = \arg\max_{\sigma'} \sum_{i=1}^{r} NMI(\sigma', \phi_i)$$

- Alle möglichen clusterings σ' aufzählen und testen? Zu aufwändig!

Ansatz von [3]

Informationstheoretische Konsensusfunktion

- Wechselseitige Information zweier clusterings
  Normalized mutual information

$$NMI(\phi_a, \phi_b) = \frac{2}{N} \sum_{i=1}^{k(a)} \sum_{j=1}^{k(b)} |g_i^a \cap g_j^b| \log_{k(a)k(b)} \left(\frac{|g_i^a \cap g_j^b|}{|g_i^a||g_j^b|} \right)$$

- Der beste Konsens hat am meisten gemeinsam mit den meisten Eingabe-clusterings:

$$\sigma = \arg\max_{\sigma'} \sum_{i=1}^{r} NMI(\sigma', \phi_i)$$

- Alle möglichen clusterings σ' aufzählen und testen? Zu aufwändig!

Ansatz von [3]

Informationstheoretische Konsensusfunktion

- Wechselseitige Information zweier clusterings
  Normalized mutual information

$$NMI(\phi_a, \phi_b) = \frac{2}{N} \sum_{i=1}^{k(a)} \sum_{j=1}^{k(b)} |g_i^a \cap g_j^b| \log_{k(a)k(b)} \left(\frac{|g_i^a \cap g_j^b|}{|g_i^a||g_j^b|} \right)$$

- Der beste Konsens hat am meisten gemeinsam mit den meisten Eingabe-clusterings:

$$\sigma = \arg\max_{\sigma'} \sum_{i=1}^{r} NMI(\sigma', \phi_i)$$

- Alle möglichen clusterings σ' aufzählen und testen? Zu aufwändig!

Hypergraph als Repräsentation

Ein Clustering c ist ein Vektor
Jedes cluster von jedem clustering wird nun zu einer Hyperkante, die clusterings sind die Hyperknoten.

	c_1	c_2	c_3	$\Rightarrow$	g_1^1	g_2^1	g_3^1	g_1^2	g_2^2	g_3^2	g_1^3	g_2^3	g_3^3
x_1	1	2	1		1	0	0	0	1	0	1	0	0
x_2	1	2	1		1	0	0	0	1	0	1	0	0
x_3	1	2	2		1	0	0	0	1	0	0	1	0
x_4	2	3	2		0	1	0	0	0	1	0	1	0
x_5	2	3	3		0	1	0	0	0	1	0	0	1
x_6	3	1	3		0	0	1	1	0	0	0	0	1
x_7	3	1	3		0	0	1	1	0	0	0	0	1

Subspace Clustering

- Jedes clustering verwendet verschiedene Merkmale.
- Ensemble Clustering wird nun verwendet, um diese zusammenzuführen.

Daten	Dimensionalität Subspace	Anzahl clusterings	Qualität Ensemble NMI(σ, φ_I)	Beste NMI in einem Subspace
2D2K	1	3	0,68864	0,68864
8D5K	2	5	0,98913	0,76615
Pendig	4	10	0,59009	0,53197
Yahoo	128	20	0,38167	0,21403

Subspace Clustering

- Jedes clustering verwendet verschiedene Merkmale.
- Ensemble Clustering wird nun verwendet, um diese zusammenzuführen.

Daten	Dimensionalität Subspace	Anzahl clusterings	Qualität Ensemble NMI(σ, φ_I)	Beste NMI in einem Subspace
2D2K	1	3	0,68864	0,68864
8D5K	2	5	0,98913	0,76615
Pendig	4	10	0,59009	0,53197
Yahoo	128	20	0,38167	0,21403

Distributed Clustering

Eingabe-clusterings stammen von verschiedenen Quellen.

- Horizontale Aufteilung: alle Quellen verwenden die selben Attribute, z.B. Filialen einer Firma.
- Vertikale Aufteilung: die Quellen verwenden unterschiedliche Attribute, z.B. private Benutzer in einem peer-to-peer Forum zu Austausch von Filmen.

Man kann Ensemble clustering direkt anwenden.

Distributed Clustering

Eingabe-clusterings stammen von verschiedenen Quellen.

- Horizontale Aufteilung: alle Quellen verwenden die selben Attribute, z.B. Filialen einer Firma.
- Vertikale Aufteilung: die Quellen verwenden unterschiedliche Attribute, z.B. private Benutzer in einem peer-to-peer Forum zu Austausch von Filmen.

Man kann Ensemble clustering direkt anwenden.

Erstellen einer Ähnlichkeitsmatrix

$N \times N$ mit Ähnlichkeitseinträgen ist auf der H für r Eingabe-clusterings jetzt leicht als Matrizenmultiplikation durchführbar:

$$Sim = \frac{1}{r} HH^T$$

- Beliebigen cluster-algorithmus anwenden.
- [3] haben zwei weitere spezielle Algorithmen für clustering von Hypergraphen entwickelt.

Erstellen einer Ähnlichkeitsmatrix

$N \times N$ mit Ähnlichkeitseinträgen ist auf der H für r Eingabe-clusterings jetzt leicht als Matrizenmultiplikation durchführbar:

$$Sim = \frac{1}{r} HH^T$$

- Beliebigen cluster-algorithmus anwenden.
- [3] haben zwei weitere spezielle Algorithmen für clustering von Hypergraphen entwickelt.

Distributed Clustering

Verschiedene Quellen liefern Eingabe-Clusterings.

- Horizontale Aufteilung: alle Quellen verwenden die selben Attribute, aber andere Beobachtungen, z.B. Filialen einer Firma.
- Vertikale Aufteilung: jede Quelle verwendet andere Attribute, z.B. Benutzer in einem peer-to-peer Netz über Filme.

Wir können direkt Ensemble Clustering verwenden. Aber:

- Aufwändig, alle Beobachtungen zusammen zu führen!
- Datenschutz (privacy)!

Distributed Clustering

Verschiedene Quellen liefern Eingabe-Clusterings.

- Horizontale Aufteilung: alle Quellen verwenden die selben Attribute, aber andere Beobachtungen, z.B. Filialen einer Firma.
- Vertikale Aufteilung: jede Quelle verwendet andere Attribute, z.B. Benutzer in einem peer-to-peer Netz über Filme.

Wir können direkt Ensemble Clustering verwenden. Aber:

- Aufwändig, alle Beobachtungen zusammen zu führen!
- Datenschutz (privacy)!

Distributed Clustering

Verschiedene Quellen liefern Eingabe-Clusterings.

- Horizontale Aufteilung: alle Quellen verwenden die selben Attribute, aber andere Beobachtungen, z.B. Filialen einer Firma.
- Vertikale Aufteilung: jede Quelle verwendet andere Attribute, z.B. Benutzer in einem peer-to-peer Netz über Filme.

Wir können direkt Ensemble Clustering verwenden. Aber:

- Aufwändig, alle Beobachtungen zusammen zu führen!
- Datenschutz (privacy)!

Distributed Clustering

Verschiedene Quellen liefern Eingabe-Clusterings.

- Horizontale Aufteilung: alle Quellen verwenden die selben Attribute, aber andere Beobachtungen, z.B. Filialen einer Firma.
- Vertikale Aufteilung: jede Quelle verwendet andere Attribute, z.B. Benutzer in einem peer-to-peer Netz über Filme.

Wir können direkt Ensemble Clustering verwenden. Aber:

- Aufwändig, alle Beobachtungen zusammen zu führen!
- Datenschutz (privacy)!

Datenschutz

- Es sollen nicht (alle) Daten bekannt gemacht werden, sondern nur die Verteilung in den Daten.
- Die Daten werden also durch lokale Modelle generalisiert. Das globale Modell wird aus diesen Modellen, nicht aus den ursprünglichen Daten gebildet.
- Nach den lokalen Verteilungen werden künstlich Beispiele erzeugt.
- Globales Modell = Mischung von Gauss-Modellen!

Datenschutz

- Es sollen nicht (alle) Daten bekannt gemacht werden, sondern nur die Verteilung in den Daten.
- Die Daten werden also durch lokale Modelle generalisiert. Das globale Modell wird aus diesen Modellen, nicht aus den ursprünglichen Daten gebildet.
- Nach den lokalen Verteilungen werden künstlich Beispiele erzeugt.
- Globales Modell = Mischung von Gauss-Modellen!

Datenschutz

- Es sollen nicht (alle) Daten bekannt gemacht werden, sondern nur die Verteilung in den Daten.
- Die Daten werden also durch lokale Modelle generalisiert. Das globale Modell wird aus diesen Modellen, nicht aus den ursprünglichen Daten gebildet.
- Nach den lokalen Verteilungen werden künstlich Beispiele erzeugt.
- Globales Modell = Mischung von Gauss-Modellen!

Datenschutz

- Es sollen nicht (alle) Daten bekannt gemacht werden, sondern nur die Verteilung in den Daten.
- Die Daten werden also durch lokale Modelle generalisiert. Das globale Modell wird aus diesen Modellen, nicht aus den ursprünglichen Daten gebildet.
- Nach den lokalen Verteilungen werden künstlich Beispiele erzeugt.
- Globales Modell = Mischung von Gauss-Modellen!

Distributed Model-Based Clustering

Gegeben:

- $\{X_i\}, i = 1, \ldots, n$ Datenquellen
- $\{\varphi_i\}m, i = 1, \ldots, n$ unterliegende (lokale) Modelle
- $\{v_i\}, i = 1, \ldots, n$ nicht-negative Gewichtung der Modell

Finde das optimale globale Modell $\sigma = \arg\min Q(\sigma')$, wobei Q ein Fehlermaß (z.B. KL) und σ' aus einer Familie von Modellen ist.

Distributed Model-Based Clustering

Gegeben:

- $\{X_i\}, i = 1, \ldots, n$ Datenquellen
- $\{\varphi_i\} m, i = 1, \ldots, n$ unterliegende (lokale) Modelle
- $\{v_i\}, i = 1, \ldots, n$ nicht-negative Gewichtung der Modell

Finde das optimale globale Modell $\sigma = \arg\min Q(\sigma')$, wobei Q ein Fehlermaß (z.B. KL) und σ' aus einer Familie von Modellen ist.

Distributed Model-Based Clustering

Gegeben:

- $\{X_i\}, i = 1, \ldots, n$ Datenquellen
- $\{\varphi_i\}m, i = 1, \ldots, n$ unterliegende (lokale) Modelle
- $\{v_i\}, i = 1, \ldots, n$ nicht-negative Gewichtung der Modell

Finde das optimale globale Modell $\sigma = \arg\min Q(\sigma')$, wobei Q ein Fehlermaß (z.B. KL) und σ' aus einer Familie von Modellen ist.

Durchschnittsmodell finden

Das DMC-Problem ist äquivalent zu dem, ein Modell zu finden, das nahe am Durchschnitt der lokalen Modelle bezüglich der KL-Divergenz ist.

- KL-Divergenz: $KL(x, y) = \sum_{i=1}^{d} x_i \log\left(\frac{x_i}{y_i}\right)$
- Durchschnitt aller lokalen Modelle so dass $p_\sigma(x) = \sum_{i=1}^{n} v_i p_{\phi_i}(x)$

Finde das Modell, das dem Durchschnitt der lokalen Modelle bzgl. KL-Divergenz am nächsten ist!

Durchschnittsmodell finden

Das DMC-Problem ist äquivalent zu dem, ein Modell zu finden, das nahe am Durchschnitt der lokalen Modelle bezüglich der KL-Divergenz ist.

- KL-Divergenz: $KL(x, y) = \sum_{i=1}^{d} x_i \log\left(\frac{x_i}{y_i}\right)$
- Durchschnitt aller lokalen Modelle so dass $p_\sigma(x) = \sum_{i=1}^{n} v_i p_{\phi_i}(x)$

Finde das Modell, das dem Durchschnitt der lokalen Modelle bzgl. KL-Divergenz am nächsten ist!

Approximierender Algorithmus für DMC

Input $\{\varphi_i\}, i = 1, \ldots, n$ mit $\{v_i\}, i = 1, \ldots, n$
Familie F: Gaussian Mixture Models

Output $\sigma_a \approx \arg\min_{\sigma' \in F} \sum_{i=1}^{n} v_i D_{KL}(\phi_i, \sigma')$

1. Erzeuge ein Durchschnittsmodell, so dass $p_{\bar{\sigma}}(x) = \sum_{i=1}^{n} v_i p_{\phi_i}(x)$
2. Ziehe daraus eine Stichprobe X'
3. Wende EM an, um σ_a zu erhalten mit

$$\sigma_a = \arg\max_{\sigma' \in F} L(X', \sigma') = \arg\max \frac{1}{m} \sum_{j=1}^{m} \log\left(p_{\sigma'}(x_j)\right)$$

Approximierender Algorithmus für DMC

Input $\{\varphi_i\}, i = 1, \ldots, n$ mit $\{v_i\}, i = 1, \ldots, n$
Familie F: Gaussian Mixture Models

Output $\sigma_a \approx \arg\min_{\sigma' \in F} \sum_{i=1}^{n} v_i D_{KL}(\phi_i, \sigma')$

1. Erzeuge ein Durchschnittsmodell, so dass $p_{\bar{\sigma}}(x) = \sum_{i=1}^{n} v_i p_{\phi_i}(x)$
2. Ziehe daraus eine Stichprobe X'
3. Wende EM an, um σ_a zu erhalten mit

$$\sigma_a = \arg\max_{\sigma' \in F} L(X', \sigma') = \arg\max \frac{1}{m} \sum_{j=1}^{m} \log\left(p_{\sigma'}(x_j)\right)$$

Approximierender Algorithmus für DMC

Input $\{\varphi_i\}, i = 1, \ldots, n$ mit $\{v_i\}, i = 1, \ldots, n$
Familie F: Gaussian Mixture Models

Output $\sigma_a \approx \arg\min_{\sigma' \in F} \sum_{i=1}^{n} v_i D_{KL}(\phi_i, \sigma')$

1. Erzeuge ein Durchschnittsmodell, so dass $p_{\bar{\sigma}}(x) = \sum_{i=1}^{n} v_i p_{\phi_i}(x)$
2. Ziehe daraus eine Stichprobe X'
3. Wende EM an, um σ_a zu erhalten mit

$$\sigma_a = \arg\max_{\sigma' \in F} L(X', \sigma') = \arg\max \frac{1}{m} \sum_{j=1}^{m} \log\left(p_{\sigma'}(x_j)\right)$$

Approximierender Algorithmus für DMC

Input $\{\varphi_i\}, i = 1, \ldots, n$ mit $\{v_i\}, i = 1, \ldots, n$
Familie F: Gaussian Mixture Models

Output $\sigma_a \approx \arg\min_{\sigma' \in F} \sum_{i=1}^{n} v_i D_{KL}(\phi_i, \sigma')$

1. Erzeuge ein Durchschnittsmodell, so dass $p_{\bar{\sigma}}(x) = \sum_{i=1}^{n} v_i p_{\phi_i}(x)$
2. Ziehe daraus eine Stichprobe X'
3. Wende EM an, um σ_a zu erhalten mit

$$\sigma_a = \arg\max_{\sigma' \in F} L(X', \sigma') = \arg\max \frac{1}{m} \sum_{j=1}^{m} \log\left(p_{\sigma'}(x_j)\right)$$

Approximierender Algorithmus für DMC

Input $\{\varphi_i\}, i = 1, \ldots, n$ mit $\{v_i\}, i = 1, \ldots, n$
Familie F: Gaussian Mixture Models

Output $\sigma_a \approx \arg\min_{\sigma' \in F} \sum_{i=1}^{n} v_i D_{KL}(\phi_i, \sigma')$

1. Erzeuge ein Durchschnittsmodell, so dass $p_{\bar{\sigma}}(x) = \sum_{i=1}^{n} v_i p_{\phi_i}(x)$
2. Ziehe daraus eine Stichprobe X'
3. Wende EM an, um σ_a zu erhalten mit

$$\sigma_a = \arg\max_{\sigma' \in F} L(X', \sigma') = \arg\max \frac{1}{m} \sum_{j=1}^{m} \log\left(p_{\sigma'}(x_j)\right)$$

Mixture Models

Gewichtete Linearkombinationen von K einzelnen Verteilungen (Dichten) mit Parametern θ,

$$p(x) = \sum_{k=1}^{K} p_k\left(x \mid \theta_k\right) \pi_k$$

π_k: Die Wahrscheinlichkeit, dass eine zufällig gezogene Beobachtung aus der k-ten Verteilung kommt.

Hintergrung

Zur Erinnerung:

- Mixture Models
- log-likelihood schätzen für Mixture Models
- EM

Hintergrung

Zur Erinnerung:

- Mixture Models
- log-likelihood schätzen für Mixture Models
- EM

Hintergrung

Zur Erinnerung:

- Mixture Models
- log-likelihood schätzen für Mixture Models
- EM

log-likelihood für Mixture Models

Wahrscheinlichkeit für Daten D gegeben Parameter θ, wobei wir die Werte H eines Attributs für die Beobachtungen nicht kennen.

$$\log p(D \mid q) = \log \sum_H p(G, H \mid \theta)$$

Wenn wir immerhin die Verteilung $Q(H)$ kennen:

$$\log \sum_H Q(H) \frac{p(D, H \mid \theta)}{(H)} \geq \sum_H Q(H) \log \frac{p(D, H \mid \theta)}{Q(H)} = BL(Q, \theta)$$

Iteratives Vorgehen

- likelihood(θ) bei unbekannten Werten H schwer zu finden!
- $BL(Q, \theta)$ ist untere Schranke von likelihood(θ).
- Abwechselnd:
  - Maximieren von $BL(Q, \theta)$ bei festen Parametern θ

$$Q^{k+1} = \arg\max_{Q} BL(Q^k, \theta^k)$$

  - Maximieren $BL(Q, \theta)$ bei fester Verteilung $Q = p(H)$

$$\theta^{k+1} = \arg\max_{\theta} BL(Q^k, \theta^k)$$

Das macht EM-Vorgehen.

Iteratives Vorgehen

- likelihood(θ) bei unbekannten Werten H schwer zu finden!
- $BL(Q, \theta)$ ist untere Schranke von likelihood(θ).
- Abwechselnd:
  - Maximieren von $BL(Q, \theta)$ bei festen Parametern θ

$$Q^{k+1} = \arg\max_{Q} BL(Q^k, \theta^k)$$

  - Maximieren $BL(Q, \theta)$ bei fester Verteilung $Q = p(H)$

$$\theta^{k+1} = \arg\max_{\theta} BL(Q^k, \theta^k)$$

Das macht EM-Vorgehen.

Iteratives Vorgehen

- likelihood(θ) bei unbekannten Werten H schwer zu finden!
- $BL(Q, \theta)$ ist untere Schranke von likelihood(θ).
- Abwechselnd:
  - Maximieren von $BL(Q, \theta)$ bei festen Parametern θ

$$Q^{k+1} = \arg\max_{Q} BL(Q^k, \theta^k)$$

  - Maximieren $BL(Q, \theta)$ bei fester Verteilung $Q = p(H)$

$$\theta^{k+1} = \arg\max_{\theta} BL(Q^k, \theta^k)$$

Das macht EM-Vorgehen.

Iteratives Vorgehen

- likelihood(θ) bei unbekannten Werten H schwer zu finden!
- $BL(Q, \theta)$ ist untere Schranke von likelihood(θ).
- Abwechselnd:
  - Maximieren von $BL(Q, \theta)$ bei festen Parametern θ

$$Q^{k+1} = \arg\max_{Q} BL(Q^k, \theta^k)$$

  - Maximieren $BL(Q, \theta)$ bei fester Verteilung $Q = p(H)$

$$\theta^{k+1} = \arg\max_{\theta} BL(Q^k, \theta^k)$$

Das macht EM-Vorgehen.

Iteratives Vorgehen

- likelihood(θ) bei unbekannten Werten H schwer zu finden!
- $BL(Q, \theta)$ ist untere Schranke von likelihood(θ).
- Abwechselnd:
  - Maximieren von $BL(Q, \theta)$ bei festen Parametern θ

$$Q^{k+1} = \arg\max_{Q} BL(Q^k, \theta^k)$$

  - Maximieren $BL(Q, \theta)$ bei fester Verteilung $Q = p(H)$

$$\theta^{k+1} = \arg\max_{\theta} BL(Q^k, \theta^k)$$

Das macht EM-Vorgehen.

EM-Vorgehen [1]

Estimation:

- Schätze Wahrscheinlichkeit, dass eine Beobachtung zu einem cluster gehört
- Speichere Matrix: Beobachtungen X cluster

Maximization

- Passe clustering den Wahrscheinlichkeiten an.
- Bis es keine Verbesserung mehr ergibt: gehe zu E-Schritt.

EM-Vorgehen [1]

Estimation:

- Schätze Wahrscheinlichkeit, dass eine Beobachtung zu einem cluster gehört
- Speichere Matrix: Beobachtungen X cluster

Maximization

- Passe clustering den Wahrscheinlichkeiten an.
- Bis es keine Verbesserung mehr ergibt: gehe zu E-Schritt.

EM-Vorgehen [1]

Estimation:

- Schätze Wahrscheinlichkeit, dass eine Beobachtung zu einem cluster gehört
- Speichere Matrix: Beobachtungen X cluster

Maximization

- Passe clustering den Wahrscheinlichkeiten an.
- Bis es keine Verbesserung mehr ergibt: gehe zu E-Schritt.

EM-Vorgehen [1]

Estimation:

- Schätze Wahrscheinlichkeit, dass eine Beobachtung zu einem cluster gehört
- Speichere Matrix: Beobachtungen X cluster

Maximization

- Passe clustering den Wahrscheinlichkeiten an.
- Bis es keine Verbesserung mehr ergibt: gehe zu E-Schritt.

Approximierender Algorithmus für DMC

Input $\{\varphi_i\}, i = 1, \ldots, n$ mit $\{v_i\}, i = 1, \ldots, n$
Familie F: Gaussian Mixture Models

Output $\sigma_a \approx \arg\min_{\sigma' \in F} \sum_{i=1}^{n} v_i D_{KL}(\phi_i, \sigma')$

1. Erzeuge ein Durchschnittsmodell, so dass $p_{\bar{\sigma}}(x) = \sum_{i=1}^{n} v_i p_{\phi_i}(x)$
2. Ziehe daraus eine Stichprobe X'
3. Wende EM an, um σ_a zu erhalten mit

$$\sigma_a = \arg\max_{\sigma' \in F} L(X', \sigma') = \arg\max \frac{1}{m} \sum_{j=1}^{m} \log\left(p_{\sigma'}(x_j)\right)$$

Approximierender Algorithmus für DMC

Input $\{\varphi_i\}, i = 1, \ldots, n$ mit $\{v_i\}, i = 1, \ldots, n$
Familie F: Gaussian Mixture Models

Output $\sigma_a \approx \arg\min_{\sigma' \in F} \sum_{i=1}^{n} v_i D_{KL}(\phi_i, \sigma')$

1. Erzeuge ein Durchschnittsmodell, so dass $p_{\bar{\sigma}}(x) = \sum_{i=1}^{n} v_i p_{\phi_i}(x)$
2. Ziehe daraus eine Stichprobe X'
3. Wende EM an, um σ_a zu erhalten mit

$$\sigma_a = \arg\max_{\sigma' \in F} L(X', \sigma') = \arg\max \frac{1}{m} \sum_{j=1}^{m} \log\left(p_{\sigma'}(x_j)\right)$$

Approximierender Algorithmus für DMC

Input $\{\varphi_i\}, i = 1, \ldots, n$ mit $\{v_i\}, i = 1, \ldots, n$
Familie F: Gaussian Mixture Models

Output $\sigma_a \approx \arg\min_{\sigma' \in F} \sum_{i=1}^{n} v_i D_{KL}(\phi_i, \sigma')$

1. Erzeuge ein Durchschnittsmodell, so dass $p_{\bar{\sigma}}(x) = \sum_{i=1}^{n} v_i p_{\phi_i}(x)$
2. Ziehe daraus eine Stichprobe X'
3. Wende EM an, um σ_a zu erhalten mit

$$\sigma_a = \arg\max_{\sigma' \in F} L(X', \sigma') = \arg\max \frac{1}{m} \sum_{j=1}^{m} \log\left(p_{\sigma'}(x_j)\right)$$

Approximierender Algorithmus für DMC

Input $\{\varphi_i\}, i = 1, \ldots, n$ mit $\{v_i\}, i = 1, \ldots, n$
Familie F: Gaussian Mixture Models

Output $\sigma_a \approx \arg\min_{\sigma' \in F} \sum_{i=1}^{n} v_i D_{KL}(\phi_i, \sigma')$

1. Erzeuge ein Durchschnittsmodell, so dass $p_{\bar{\sigma}}(x) = \sum_{i=1}^{n} v_i p_{\phi_i}(x)$
2. Ziehe daraus eine Stichprobe X'
3. Wende EM an, um σ_a zu erhalten mit

$$\sigma_a = \arg\max_{\sigma' \in F} L(X', \sigma') = \arg\max \frac{1}{m} \sum_{j=1}^{m} \log\left(p_{\sigma'}(x_j)\right)$$

Approximierender Algorithmus für DMC

Input $\{\varphi_i\}, i = 1, \ldots, n$ mit $\{v_i\}, i = 1, \ldots, n$
Familie F: Gaussian Mixture Models

Output $\sigma_a \approx \arg\min_{\sigma' \in F} \sum_{i=1}^{n} v_i D_{KL}(\phi_i, \sigma')$

1. Erzeuge ein Durchschnittsmodell, so dass $p_{\bar{\sigma}}(x) = \sum_{i=1}^{n} v_i p_{\phi_i}(x)$
2. Ziehe daraus eine Stichprobe X'
3. Wende EM an, um σ_a zu erhalten mit

$$\sigma_a = \arg\max_{\sigma' \in F} L(X', \sigma') = \arg\max \frac{1}{m} \sum_{j=1}^{m} \log\left(p_{\sigma'}(x_j)\right)$$

Privacy

- Ein feines Gauss-Modell mit wenig Varianz, womöglich zentriert an jedem Datenpunkt, gibt dann doch genau die Daten wieder.
  - Auch wenn nur das Modell weitergegeben wird – keine Anonymität!
- Ein grobes Modell mit nur einem Gauss-Kegel und hoher Varianz hat hohe Anonymität – niedrige Wahrscheinlichkeit, die Daten aus dem Modell zu generieren.
- Immerhin wird das globale Modell auch aus mittelfeinen lokalen Gauss-Modellen noch ordentlich. . .

Privacy

- Ein feines Gauss-Modell mit wenig Varianz, womöglich zentriert an jedem Datenpunkt, gibt dann doch genau die Daten wieder.
  - Auch wenn nur das Modell weitergegeben wird – keine Anonymität!
- Ein grobes Modell mit nur einem Gauss-Kegel und hoher Varianz hat hohe Anonymität – niedrige Wahrscheinlichkeit, die Daten aus dem Modell zu generieren.
- Immerhin wird das globale Modell auch aus mittelfeinen lokalen Gauss-Modellen noch ordentlich. . .

Privacy

- Ein feines Gauss-Modell mit wenig Varianz, womöglich zentriert an jedem Datenpunkt, gibt dann doch genau die Daten wieder.
  - Auch wenn nur das Modell weitergegeben wird – keine Anonymität!
- Ein grobes Modell mit nur einem Gauss-Kegel und hoher Varianz hat hohe Anonymität – niedrige Wahrscheinlichkeit, die Daten aus dem Modell zu generieren.
- Immerhin wird das globale Modell auch aus mittelfeinen lokalen Gauss-Modellen noch ordentlich. . .

Privacy

- Ein feines Gauss-Modell mit wenig Varianz, womöglich zentriert an jedem Datenpunkt, gibt dann doch genau die Daten wieder.
  - Auch wenn nur das Modell weitergegeben wird – keine Anonymität!
- Ein grobes Modell mit nur einem Gauss-Kegel und hoher Varianz hat hohe Anonymität – niedrige Wahrscheinlichkeit, die Daten aus dem Modell zu generieren.
- Immerhin wird das globale Modell auch aus mittelfeinen lokalen Gauss-Modellen noch ordentlich. . .

Was wissen Sie jetzt?

- Sie kennen das Kriterium der wechselseitigen Information.

$$NMI(\phi_a, \phi_b) = \frac{2}{n} \sum_{i=1}^{k(a)} \sum_{j=1}^{k(b)} |g_i^a \cap g_j^b| \log_{k(a)k(b)} \left(\frac{|g_j^a \cap g_j^b|}{|g_i^a||g_j^b|} \right)$$

- Das kann man zur Konsensusfunktion machen.
- Sie kennen die Hypergraph-Representation der Eingabe-clusterings.
- Ensemble clustering ist eine Art, clusterings mit unterschiedlichen Attributen zusammen zu führen. Es gibt noch andere Subspace clustering Algorithmen!
- Ensemble clustering ist eine Art, verteilte Datenbestände zu clustern. Es gibt noch andere!
- Wenn man nicht die Daten austauscht, sondern Modelle der Daten, schützt man die Daten und wahrt Anonymität.

Was wissen Sie jetzt?

- Sie kennen das Kriterium der wechselseitigen Information.

$$NMI(\phi_a, \phi_b) = \frac{2}{n} \sum_{i=1}^{k(a)} \sum_{j=1}^{k(b)} |g_i^a \cap g_j^b| \log_{k(a)k(b)} \left(\frac{|g_j^a \cap g_j^b|}{|g_i^a||g_j^b|} \right)$$

- Das kann man zur Konsensusfunktion machen.
- Sie kennen die Hypergraph-Representation der Eingabe-clusterings.
- Ensemble clustering ist eine Art, clusterings mit unterschiedlichen Attributen zusammen zu führen. Es gibt noch andere Subspace clustering Algorithmen!
- Ensemble clustering ist eine Art, verteilte Datenbestände zu clustern. Es gibt noch andere!
- Wenn man nicht die Daten austauscht, sondern Modelle der Daten, schützt man die Daten und wahrt Anonymität.

Was wissen Sie jetzt?

- Sie kennen das Kriterium der wechselseitigen Information.

$$NMI(\phi_a, \phi_b) = \frac{2}{n} \sum_{i=1}^{k(a)} \sum_{j=1}^{k(b)} |g_i^a \cap g_j^b| \log_{k(a)k(b)} \left(\frac{|g_j^a \cap g_j^b|}{|g_i^a||g_j^b|} \right)$$

- Das kann man zur Konsensusfunktion machen.
- Sie kennen die Hypergraph-Representation der Eingabe-clusterings.
- Ensemble clustering ist eine Art, clusterings mit unterschiedlichen Attributen zusammen zu führen. Es gibt noch andere Subspace clustering Algorithmen!
- Ensemble clustering ist eine Art, verteilte Datenbestände zu clustern. Es gibt noch andere!
- Wenn man nicht die Daten austauscht, sondern Modelle der Daten, schützt man die Daten und wahrt Anonymität.

Was wissen Sie jetzt?

- Sie kennen das Kriterium der wechselseitigen Information.

$$NMI(\phi_a, \phi_b) = \frac{2}{n} \sum_{i=1}^{k(a)} \sum_{j=1}^{k(b)} |g_i^a \cap g_j^b| \log_{k(a)k(b)} \left(\frac{|g_j^a \cap g_j^b|}{|g_i^a||g_j^b|} \right)$$

- Das kann man zur Konsensusfunktion machen.
- Sie kennen die Hypergraph-Representation der Eingabe-clusterings.
- Ensemble clustering ist eine Art, clusterings mit unterschiedlichen Attributen zusammen zu führen. Es gibt noch andere Subspace clustering Algorithmen!
- Ensemble clustering ist eine Art, verteilte Datenbestände zu clustern. Es gibt noch andere!
- Wenn man nicht die Daten austauscht, sondern Modelle der Daten, schützt man die Daten und wahrt Anonymität.

Was wissen Sie jetzt?

- Sie kennen das Kriterium der wechselseitigen Information.

$$NMI(\phi_a, \phi_b) = \frac{2}{n} \sum_{i=1}^{k(a)} \sum_{j=1}^{k(b)} |g_i^a \cap g_j^b| \log_{k(a)k(b)} \left(\frac{|g_j^a \cap g_j^b|}{|g_i^a||g_j^b|} \right)$$

- Das kann man zur Konsensusfunktion machen.
- Sie kennen die Hypergraph-Representation der Eingabe-clusterings.
- Ensemble clustering ist eine Art, clusterings mit unterschiedlichen Attributen zusammen zu führen. Es gibt noch andere Subspace clustering Algorithmen!
- Ensemble clustering ist eine Art, verteilte Datenbestände zu clustern. Es gibt noch andere!
- Wenn man nicht die Daten austauscht, sondern Modelle der Daten, schützt man die Daten und wahrt Anonymität.

Was wissen Sie jetzt?

- Sie kennen das Kriterium der wechselseitigen Information.

$$NMI(\phi_a, \phi_b) = \frac{2}{n} \sum_{i=1}^{k(a)} \sum_{j=1}^{k(b)} |g_i^a \cap g_j^b| \log_{k(a)k(b)} \left(\frac{|g_j^a \cap g_j^b|}{|g_i^a||g_j^b|} \right)$$

- Das kann man zur Konsensusfunktion machen.
- Sie kennen die Hypergraph-Representation der Eingabe-clusterings.
- Ensemble clustering ist eine Art, clusterings mit unterschiedlichen Attributen zusammen zu führen. Es gibt noch andere Subspace clustering Algorithmen!
- Ensemble clustering ist eine Art, verteilte Datenbestände zu clustern. Es gibt noch andere!
- Wenn man nicht die Daten austauscht, sondern Modelle der Daten, schützt man die Daten und wahrt Anonymität.

Literatur I

[1] Arthur P Dempster, Nan M Laird, and Donald B Rubin.
Maximum likelihood from incomplete data via the em algorithm.
Journal of the Royal Statistical Society. Series B (Methodological), pages 1–38, 1977.

[2] Boris Mirkin.
Reinterpreting the category utility function.
Machine Learning, 45(2):219–228, 2001.

[3] Alexander Strehl and Joydeep Ghosh.
Cluster ensembles—a knowledge reuse framework for combining multiple partitions.
The Journal of Machine Learning Research, 3:583–617, 2003.

[4] Alexander Topchy, Anil K Jain, and William Punch.
Combining multiple weak clusterings.
In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 331–338. IEEE, 2003.