Bildverarbeitung: Clusterung

D. Schlesinger – TUD/INF/KI/IS

Einfache Clusterung Aufgabe

Aufgabe: partitioniere eine Menge der Objekte auf sinnvolle Teile – Clusters.
Die Objekte eines Clusters sollen „ähnlich“ sein.

Clustermenge: K
Indexmenge: $I = \{1, 2, \ldots, |I|\}$
Merkmalsvektoren: x^i, $i \in I$.
Partitionierung: $C = (I_1, I_2, \ldots, I_{|K|})$, $I_k \cap I_{k'} = \emptyset$ für $k \neq k'$, $\bigcup_k I_k = I$

Jeder Cluster hat einen „Repräsentant“ y^k

Die Aufgabe:

$$\sum_k \sum_{i \in I_k} \|x^i - y^k\|^2 \to \min_{C,y}$$

Alternativ – gesucht wird eine Abbildung $C : I \to K$

$$\sum_i \|x^i - y^{C(i)}\|^2 \to \min_{y,C}$$

$$\sum_i \min_k \|x^i - y^k\|^2 \to \min_y$$

K-Means Algorithmus

Initialisiere Clusterzentren y^k zufällig.
Wiederhole bis sich die Clusterung C ändert:

1) Klassifikation:

$$C(i) = \arg\min_{k'} \|x^i - y^{k'}\|^2 \quad \rightarrow i \in I_k$$

2) Aktualisierung der Zentren:

$$y^k = \arg\min_y \sum_{i \in I_k} \|x^i - y\|^2 = \sum_{i \in I_k} x^i$$

– NP-vollständig.
– Bei $|K| \ll |I|$ bleibt kein Cluster frei (bei der global optimalen Lösung).
– K-Means Konvergiert zum lokalen Optimum $\rightarrow$ abhängig von der Initialisierung (Beispiel lokaler Konvergenz auf der Tafel).

Varianten/Generalisierungen

Ein anderer Abstandsmaß, zum Beispiel $\|x^i - y^k\|$ anstatt $\|x^i - y^k\|^2$:

beim K-Means ist die Klassifikation 1) dasselbe,

die Aktualisierung 2) – finde das Zentrum des Umkreises der Punkte x^i, $i \in I_k$

Problem: die Merkmale x lassen sich nicht immer mitteln.

Eine Verallgemeinerung basiert auf der Beobachtung

$$\sum_i \|x^i - \bar{x}\|^2 \sim \sum_{ij} \|x^i - x^j\|^2,$$

daraus folgt

$$\sum_k \sum_{ij \in I_k} \|x^i - x^j\|^2 = \sum_k \sum_{ij \in I_k} d(i,j) \to \min_C,$$

mit der Abstandsmatrix d (die auf unterschiedlichste Weisen definiert werden kann).

Noch eine Variante – Minimierung der Durchmesser:

$$\max_k \max_{ij \in I_k} d(i,j) \to \min_C$$

Diese aufgabe ist bei $|K| = 2$ polynomiell lösbar.

Abstandsmaße

Graph basierte Abstandsmaße.

Gegeben ist ein Graph, dessen Knoten die Elementen von I sind. Jede Kante ist mit $d(i, j)$ bewertet. Der Abstand zwischen i und j ist die (summarische) Länge des kürzesten Pfaden zwischen den entsprechenden Knoten im Graphen.
$\rightarrow$ So ein Abstandsmaß ist eine Metrik.

Pfad basierte Abstandsmaße (auch Graphen).

Die Idee – selbst wenn zwei Merkmale x^i und x^j von einander weit entfernt sind, gehören sie eher zum selben Cluster, wenn ein Pfad $(x^i, x^l, x^{l'}, \ldots, x^j)$ existiert so, dass die Abstände $d(x^l, x^{l'})$ klein sind.
$\rightarrow$ Der minimale aufspannende Baum wird benötigt.

Abstandsmaße für „andere“ Objekte (nicht $\in \mathbb{R}^n$).

Zum Beispiel:
Edit distance (Levenstein Abstand) zwischen zwei Folgen,
Graph Isomorphismus basierte Abstände zwischen Graphen

etc.

Die Elementen sind die Pixel des Bildes, die Merkmale sind Farbwerte,
man zerlege das Bild auf Teile, die jeweils „charakteristischen Farben“ entsprechen.

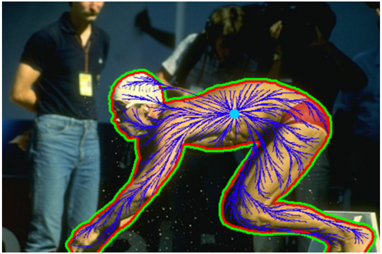
Beispiel der Farbreduktion auf 8 Farben:

Die Elementen sind die Pixel des Bildes.
Idee – die Matrix der Abstände d berücksichtigt

a) Unterschied der lokalen Merkmalswerte (z.B. der Farbe),

b) Die räumliche Entfernung der Pixel von einander.

Beispiel: „Geodesics“.



Graphen basiertes Abstandsmaß. Kanten verbinden nah liegende Pixel.
Die Kantenbewertung – Farbdifferenz

Die kürzesten Wege von allen Pixeln zu einem (vom Nutzer gegebenem) bilden einen Baum (effizient berechenbar).

Die Längen werden mit einem Schwellwert verglichen.

Der Vordergrund ist zusammenhängend.

Andere ähnliche Vorgehensweisen:
Semidefinite Programmierung, Normalized Cut, usw.