

Skript zur Vorlesung  
Knowledge Discovery in Databases II  
im Sommersemester 2007

# Kapitel 6: Multi-Instanz Data Mining

Skript © 2007 Matthias Schubert

<http://www.dbs.ifi.lmu.de/Lehre/KDD>

259

---

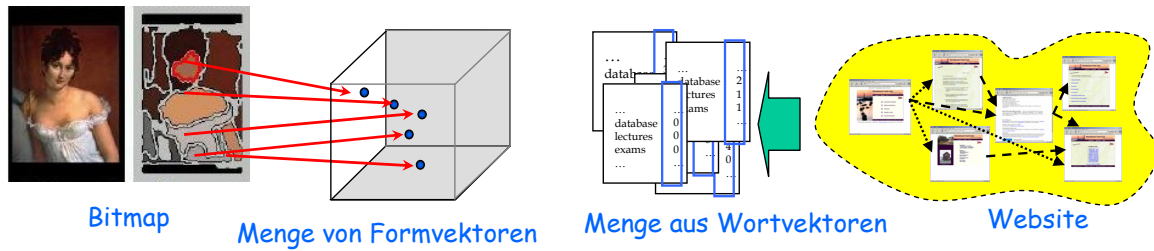
## Kapitelübersicht

---

- 6.1 Einleitung und Motivation  
Applikationen, Spezifikation, Bedeutungen
- 6.2. Aggregationsbasierte Ansätze
- 6.3 Distanzmaße und Kernel für Multi-Instanz Objekte  
SMD, Hausdorff, Matching Distance, Convolution Kernel
- 6.4 Multi-Instanz-Klassifikation und Multi-Instanz Lernen  
allgemeine Klassifikatoren, Lernen mit APRs, EM-DD
- 6.5 Multi-Instanz Clustering  
MI-EM, COSMIC

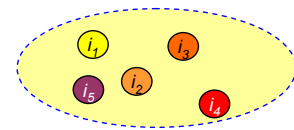
260

# Was sind Multi-Instanz Objekte ?



**Multi-Instanz Objekte treten auf bei:**  
mehrere Komponenten (z.B. CAD-Daten)  
unterschiedlichen Konfigurationen (z.B. Proteine)  
mengenartigen Objekten (z.B. Warenkorb)

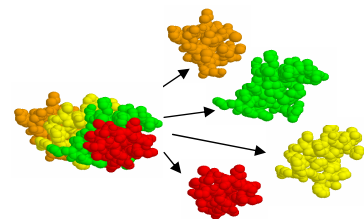
**Wichtig:** 1. Alle Instanzen im gleichen Featureraum.  
2. Reihenfolge wird nicht berücksichtigt.



# Was sind Multi-Instanz Objekte ?

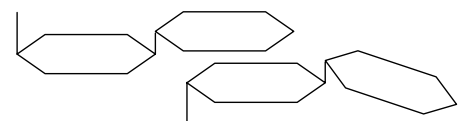
## Proteine

- Proteine bestehen i.d.R. aus mehreren Aminosäureketten (AS-Ketten)
- betrachte jede Kette als Instanz
- ein Protein ist eine Menge von AS-Sequenzen



## Macro-Moleküle

- verschieden räumliche Ausrichtungen
- jede räumliche Ausrichtung ist eine Instanz
- Menge aller Ausrichtung stellt das Moleküle dar

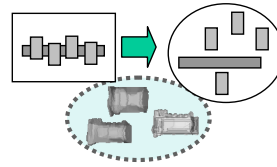


# Was sind Multi-Instanz Objekte ?

---

## Weitere Anwendungen :

- CAD-Bauteile: Mengen von Raumprimitiven /Überdeckungen.



- HTML-Dokumente: Menge aus thematischen Blöcken.



- Video-Daten: Video entspricht Menge von Shots (= Kameraeinstellungen)



## Formal:

Objektrepräsentation  $o = \{r_1, \dots, r_n\} \in 2^R$

wobei  $R$  der Darstellungsraum der Komponenten ist.



263

---

## Möglichkeiten zur Behandlung von Multi-Instanz Objekten

---

### 1. Aggregation

Rückführen der Instanz-Mengen auf einen Repräsentanten

### 2. Ähnlichkeitsmaße

Definiere Abstandsmaße und Kernel für mengenwertige Objekte (Multi-Instanz Objekte). Verarbeitung über Abstands-basiertes Data Mining oder Kernel Verfahren.

### 3. Algorithmen auf Multi-Instanz Objekten

Direkte Integration der Multi-Instanz Objekte in die Verfahren.

Klassisches Multi-Instanz Lernen

264

## 6.1. Aggregationsbasierte Ansätze

---

**Idee:** Beschreibe Menge durch einen einzigen Feature-Vektor, der die Eigenschaften des Multi-Instanz(MI) Objekts zusammenfasst.

z.B. Aggregation durch Centroid

=> einfaches Verfahren, das eine Menge über Durchschnittswerte beschreibt

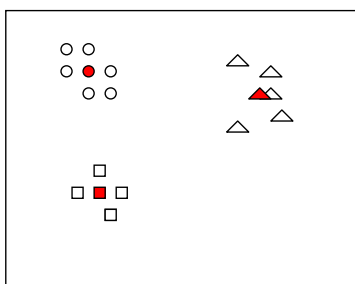
**Probleme:**

- Charakteristika einzelner Objekte gehen in der Menge unter
- Kardinalität der MI-Objekte wird nicht verwendet
- weit entfernte Instanzen werden von Centroiden schlecht beschrieben

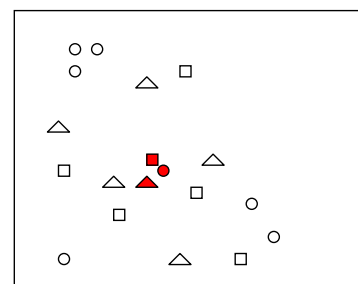
265

## Aggregationsbasierte Ansätze

---



1. Fall: Daten für Aggregation geeignet



2. Fall: Daten werden durch Centroide schlecht approximiert.

**Fazit:** Je nach Charakteristika der Daten können Centroide oder andere Aggregationen MI-Objekte gut darstellen.

=> Wenn Instanzen eines Objekts alle ähnlich zu einander sind, dann ist Aggregation meist eine ausreichende Darstellung des Objekts.

266

## 6.2 Distanzmaße und Kernel für Multi-Instanz Objekte

**Idee:** Zur Anwendung vieler Data Mining Algorithmen ist nur die Verwendung bestimmter Abstands- oder Ähnlichkeitsfunktionen notwendig.

⇒ Definiere Distanz- und Kernel-Funktionen auf MI-Objekten

Distanzen und Ähnlichkeiten zwischen Mengen können auf unterschiedliche Art und Weise definiert werden:

- Wie viele Instanzen müssen bei 2 MI-Objekten ähnlich sein?
- Darf zu einer Instanz aus Objekt  $O_1$  mehr als eine ähnliche Instanz im Objekt  $O_2$  vorhanden sein, damit  $O_1$  und  $O_2$  als ähnlich betrachtet werden?

⇒ mehrere Ähnlichkeitsmaße die je nach Datentyp und Applikation Verwendung finden.

267

## Hausdorff Distanz

**Idee:** Die Ähnlichkeit zweier MI-Objekte kann über die maximale minimale Distanz zweier Instanzen festgelegt werden.

⇒ minimale Distanz = ähnlichste Instanz im anderen MI-Objekt

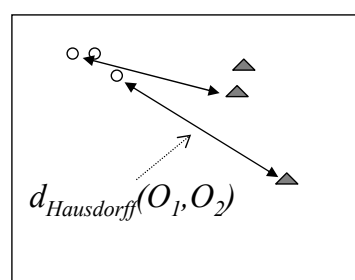
⇒ maximale, minimale Distanz = Ähnlichkeit der Instanz, die am wenigsten durch eine Distanz im anderen MI-Objekt dargestellt wird.

⇒ Jede Instanz in  $O_1$  hat mindestens eine Instanzen aus  $O_2$  im Umkreis von  $d_{Hausdorff}(O_1, O_2)$ .

**Definition:** Hausdorff Distanz

Seien  $O_1, O_2$  zwei MI-Objekte und  $d(x, y)$  ein Distanzmaß im zugehörigen Repräsentationsraum  $R$ . Dann ist die Hausdorff-Distanz wie folgt definiert:

$$d_{Hausdorff}(O_1, O_2) = \max\left(\max_{o_i \in O_1} \left(\min_{o_j \in O_2} (d(o_i, o_j))\right), \max_{o_i \in O_2} \left(\min_{o_j \in O_1} (d(o_i, o_j))\right)\right)$$



268

# Minimal-Hausdorff-Distanz

**Idee:** Zwei MI-Objekte sind gleich, wenn sie mindestens 1 ähnliche Instanz enthalten.

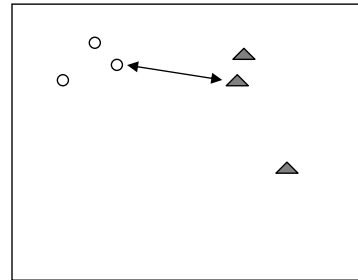
**Definition:** Minimal-Hausdorff-Distanz

Seien  $O_1, O_2$  zwei MI-Objekte und  $d(x, y)$  ein Distanzmaß im zugehörigen Repräsentationsraum  $R$ . Dann ist die Minimal-Hausdorff-Distanz wie folgt definiert:

$$d_{\text{Hausdorff}}(O_1, O_2) = \min_{o_i \in O_1} \left( \min_{o_j \in O_2} (d(o_i, o_j)) \right)$$

**Bemerkungen:**

Min. Hausdorff Dist. stellt die geringste Anforderung an die Ähnlichkeit 2er MI-Objekte.



269

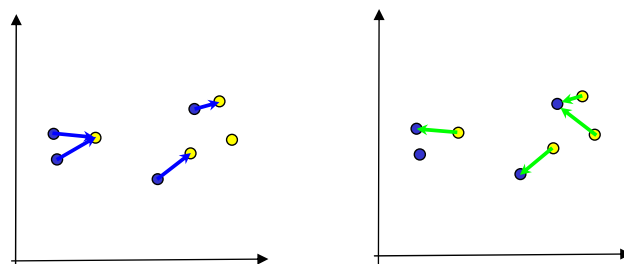
# Summe Minimaler Distanzen (SMD)

**Idee:** Zwei Objekte sind ähnlich, wenn es für jede Instanz in beiden Objekten mindestens eine ähnliche Instanz im jeweils anderen Objekt gibt.

**Definition:**

Seien  $O_1, O_2$  zwei MI-Objekte und  $d(x, y)$  ein Distanzmaß im zugehörigen Repräsentationsraum  $R$ . Dann ist die SMD wie folgt definiert:

$$d_{\text{SMD}}(O_1, O_2) = \frac{1}{2} \left( \frac{1}{|O_1|} \sum_{o_i \in O_1} \left( \min_{o_j \in O_2} (d(o_i, o_j)) \right) + \frac{1}{|O_2|} \sum_{o_j \in O_2} \left( \min_{o_i \in O_1} (d(o_i, o_j)) \right) \right)$$



270

# Berechnung von Hausdorff und SMD

Berechnung von Hausdorff und SMD Distanzen in  $O(|O_1| \cdot |O_2| \cdot d)$

- unter der Annahme:  $d(x,y)$  in  $O(d)$
- Begründung für jede Instanz in  $O_1$  muss die Distanz zu allen Distanzen in  $O_2$  bestimmt werden.

Metrikeigenschaften:

- Hausdorff-Distanz ist metrisch
- Minimal-Hausdorff und SMD sind nur symmetrisch und reflexiv. Die Dreiecksungleichung gilt aber für beide Distanzmaße nicht.

271

# Minimal Matching Distanz (MMD)

**Idee:** 2 MI-Objekte sind ähnlich, wenn es zu jeder Instanz **genau** eine ähnliche Instanz im anderen Objekt gibt.

**Definition:**

Seien  $O_1, O_2$  zwei MI-Objekte und  $d(x,y)$  ein Distanzmaß im zugehörigen Repräsentationsraum  $R$ . Dann ist die Minimal-Matching-Distanz wie folgt definiert:

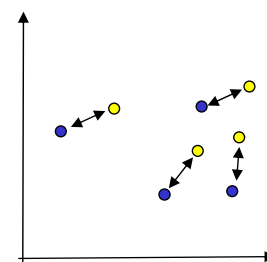
$$d_{MM}(O_1, O_2) = \min_{\pi \in \Pi(O_1)} \left( \sum_{k=1}^{|O_1|} d(o_{1,\pi(k)}, o_{2,k}) + w(o_{1,\pi(k)}) \right)$$

o.B.d.A sei  $|O_1| > |O_2|$ . Weiter sei  $\Pi(O_1)$  die Menge aller Permutationen der Instanzen von  $O_1$  und  $w(o_{i,j})$  ein Straf-Faktor für nicht zugeordnete Instanzen ist.

**Bemerkung:**

MMD ist eine Metrik, wenn  $w(o_{i,j})$  groß genug ist. D.h. größer als alle auftretenden Distanzen zwischen 2 Instanzen.

=> ein Objekt zu Matchen ist im ungünstigsten Fall immer noch günstiger als es einfach nicht zu matchen.



272

# Minimal Matching Distanz

**Berechnung:** Lösung des Zuordnungsproblems mit Ungarischer Methode (Zeitkomplexität  $O(n^3)$ ).

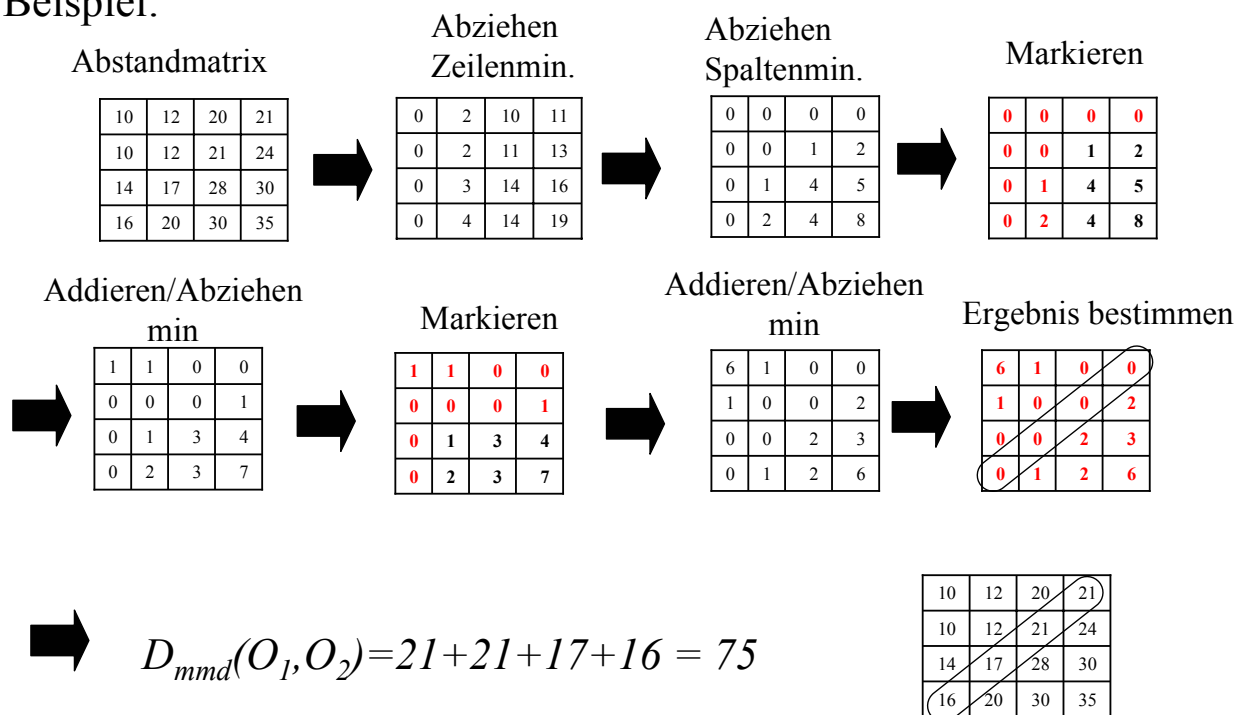
**Algorithmus:**

1. Abstandsmatrix zwischen Instanzen beider Objekte aufstellen
2. Quadratisieren der Matrix (Einträge durch  $w(o_{i,j})$  auffüllen)
3. Abziehen des Minimums in jeder Zeile
4. Abziehen des Minimums in jeder Spalte
5. Markieren aller Zeilen und Spalten, so dass alle Nuller markiert sind und einer minimale Anzahl markiert ist.
6. Falls minimale Anzahl an markierten Spalten und Zeilen = n, dann permurtiere Matrix so, dass 0 auf der Hauptdiagonalen liegen und gib Ergebnis zurück
7. Falls Anzahl markierter Zeilen und Spalten < n
  - a. Suche Minimum aller nicht markierten Elemente
  - b. Subtrahiere Minimum von aller unmarkierten Elementen
  - c. Addiere Minimum auf die Schnittlelemente der markierten Zeilen und Spalten
  - d. gehe zu Schritt 5

273

## Berechnung Minimal Matching Distanz

Beispiel:



274



# Kernelfunktionen für Multi-Instanz Objekte

---

**Idee:** 2 MI-Objekte werden über die aufsummierte Ähnlichkeit der Instanzen in einem Objekt mit den Instanzen im anderen Objekt verglichen. Ähnlichkeit der Instanzen kann mit einem beliebigen anderen Kernel für Objekte bestimmt werden.

**Definition:** Convolution Kernel

Seien  $O_1, O_2$  zwei MI-Objekte und  $K(x,y)$  eine Kernelfunktion im zugehörigen Repräsentationsraum  $R$ . Dann ist der Convolution Kernel wie folgt definiert:

$$K_{Convolution}(O_1, O_2) = \sum_{o_{1,i} \in O_1, o_{2,j} \in O_2} K(o_{1,i}, o_{2,j})$$

**Bemerkung:**

- Grundidee ist ähnlich zu Average-Link Distance (Durschnitt der paarweisen Distanzen)
- Convolution Kernel sind Mercer-Kernel und können daher in SVM, Kernel-PCA, usw. korrekt verwendet werden.

275

---

## 6.4 Multi-Instanz Klassifikation

---

**Gegeben:**  $DB = 2^F$  mit  $F$  ein Featureraum

Trainingobjekte Objekte  $(O,c)$  mit  $O \in DB$  und  $c \in C$

*Welche Instanzen  $o_i \in O$  sind für die Zugehörigkeit von  $O$  zu Klasse  $c$  ausschlaggebend ?*

**Klassisches Multi-Instanz Lernen:**

- Unterscheide relevant und nicht-relevant
- Objekt ist relevant wenn mind. 1 Instanz relevant ist.

**Generelle Multi-Instanz Klassifikation:**

- Mehr als 2 Klassen
- Instanzen sind mit einer oder mehreren Klassen assoziiert.

276

# Generelle Multi-Instanz Klassifikation

---

## Problem:

MI-Objekte der gleichen Klasse müssen nicht vollständig ähnlich sein. Daher sind die Möglichkeiten eine Klasse zu beschreiben recht umfangreich.

## Allgemeiner Ansatz zum beschreiben von MI-Klassen:

- Klassen lassen sich über bestimmte Instanz-Konzepte definieren (Fußballmannschaft hat 1 **Torwart** und 10 **Feldspieler**)
- Jedes Konzept beschreibt einen Typ von Instanzen
- Konzepte können in Klassen vorhanden sein oder nicht
- Eventuell spielt Kardinalität der Instanzen pro Konzept eine Rolle (5 Torwarte und 1 Feldspieler sind keine Fußballmannschaft)

277

# Generelle Multi-Instanz Klassifikation

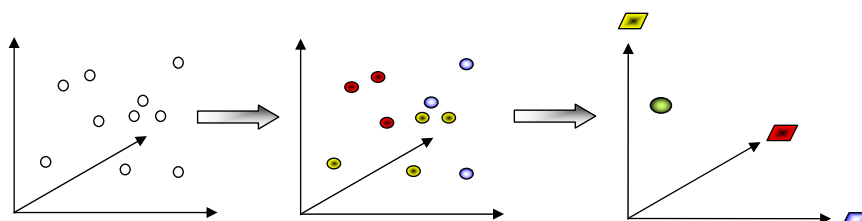
---

## Klassifikation von Multi-Instanz Objekten mit bekannten Konzepten

**Gegeben:** Eine Menge  $C$  von MI-Klassen, eine Menge  $K$  von Instanz-Konzepten.  $DB$  eine Datenbank mit MI-Objekten. Funktion  $CL(O) = C_i \in C$  beschreibt tatsächliche Klasse von MI Objekt  $O$ . Funktion  $KL(o_j) = K_l \in K$  beschreibt tatsächliches Konzept von Instanz  $o_j$ .

**Idee:** Zweistufige Klassifikation.

- Lerne 2 Klassifikatoren. Einen der Instanzen auf Konzepte abbildet und einen der MI-Objekte basierend der vorkommenden Konzepte auf Klassen abbildet.
- 1. Klassifikator ist dabei Klassifikationsproblem im Feature-Raum.
- 2. Klassifikator wird im Feature der Konzepthäufigkeiten trainiert



278

# Generelle Multi-Instanz Klassifikation

Klassifikation von Multi-Instanz Objekten mit unbekannten Konzepten

**Gegeben:** Eine Menge  $C$  von MI-Klassen.  $DB$  eine Datenbank mit MI-Objekten. Funktion  $CL(O) = C_i \in C$  beschreibt tatsächliche Klasse von MI Objekt  $O$ .

**Problem:** Die Konzepte, die eine Klasse definieren existieren, sind aber unbekannt.

=> explizites Trainieren eines Konzept-Klassifikators ist nicht möglich

**Lösung:**

- Trainiere einen Instanz-Klassifikator der folgendes vorhersagt:  
„Wie wahrscheinlich ist es, dass eine Instanz  $o_j$  in einem MI-Objekt der Klasse  $C$  auftritt?“
- Kombiniere die Wahrscheinlichkeiten für alle Instanzen im MI-Objekte und gib die Klasse mit der max. Wahrscheinlichkeit zurück. (Modelieren der Menge über einen Multinomial-Prozess = „Ziehen aus Urne mit Zurücklegen“)

**Bemerkung:**

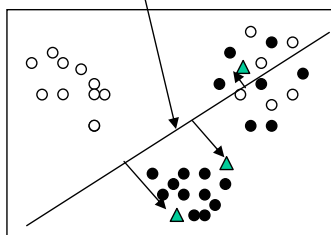
- Verfahren setzt wieder zuverlässige Konfidenz-Werte voraus.
- Multinomial-Prozess basiert nimmt Unabhängigkeit zwischen den Instanzen an.

279

# Generelle Multi-Instanz Klassifikation

Beispiel: 2 Klassen mit 3 unbekannten Konzepten

linearer Instanz-Klassifikator

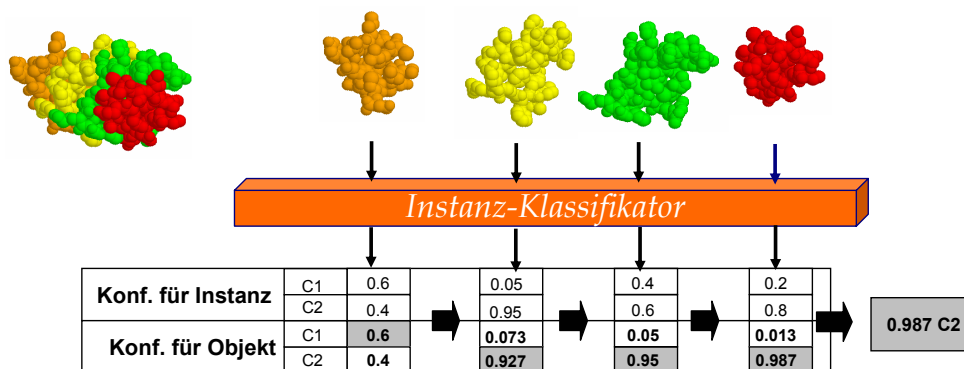


- Trainingsmenge für Instanz-Klassifikator für die Klasse A
$$TR_A = \bigcup_{O_i \in DB} \{o_j \in O_i \wedge CL(O_i) = A\}$$
- Klassifikator sollte implizit die Konzepte trennen
- Konzepte, die in mehreren Klassen vorkommen sollten an den Grenzen liegen.
- der Verwendete Klassifikator sollte daher in der Lage sein „*multi-modale*“ Klassen zu trennen.  
(multimodal = Klassen werden durch Kombination mehrerer Prozesse gebildet)
- Konfidenz für die Zugehörigkeit zu einer Klasse sollte von Zugehörigkeitswahrscheinlichkeit zu einem Konzept und der Klassenspezifität für eine Klasse abhängen.

280

# Generelle Multi-Instanz Klassifikation

Beispiel: Kombination der Wahrscheinlichkeiten



Konfidenz von  $O$  für Klasse  $C_k$ :  
 (Satz von Bayes)  $\Pr[C_k | O] = \frac{\Pr[C_k] \cdot P[O | C_k]}{\sum_{i \in C} \Pr[C_i] \cdot P[O | C_i]}$

mit  $\Pr[W | O_k] = \prod_{p_i \in W} \Pr[I_i | C_k]$

281

## Klassisches Multi-Instanz Lernen

Problemstellung: Es existiert genau 1 unbekanntes Konzept  $K_{rel}$ . Alle MI-Objekte, die mindestens 1 Instanz  $o_{rel}$  mit  $K(o_{rel}) = K_{rel}$  beinhalten werden zur „relevant“-Klasse gezählt. Alle anderen werden als „irrelevant“ eingestuft.

Beispiel:

1- Riecht eine Molekül nach Moshus oder nicht ? [Dietterich et al. 1998]

Moleküle werden als MI Objekte betrachtet. Instanz = räumliche Konformation. 1 Konformation kann hier ausschlaggebend dafür sein, dass Molekül nach Moshus riecht. Umgekehrt riecht ein Molekül nicht, wenn keine deartige Koformation vorhanden ist.

2- Suche Lungenembolien

Patient soll auf Blutgerinsel in der Lunge untersucht werden. CT liefert pro Patient eine Menge von verdächtigen Punkten in der Lunge. Patient ist nur dann gesund wenn kein verdächtiger Punkt eine Embolie ist.

=> relevant = krank und existiert mind eine Lungenembolie muss der Patient behandelt werden

282

# Klassisches Multi-Instanz Lernen

## Lösungsansätze:

Klassifiziere alle einzelnen Instanzen

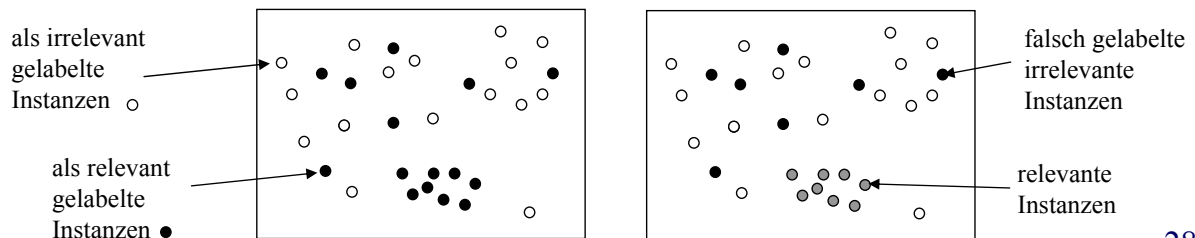
=> Falls eine Instanz relevant ist, ist auch das Objekt relevant.

## Problem:

Gegeben sind nur gelabelte MI-Objekte und nicht gelabelte relevante Instanzen.

**Bemerkung:** *Klassisches Multi-Instanz Lernen kann als spezielles Klassifikationsszenario der Single-Instanz-Klassifikation betrachtet werden*

- Instanzen aus irrelevanten MI-Objekten sind zwangsläufig alle irrelevant  
=> Beispiele für irrelevante Instanzen
- Instanzen aus relevanten MI-Objekten können relevant oder irrelevant sein  
=> Trainings-Instanzen für „relevant“-Klasse ist stark verrauscht.



283

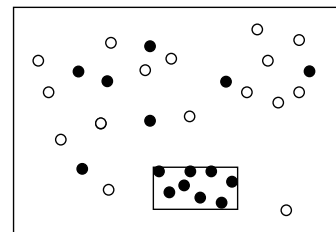
# Klassisches Multi-Instanz Lernen

## MI-Lernen mit achsenparallelen Rechtecken

Finde das kleinste Rechteck, das keine Instanz aus einem irrelevanten MI-Objekt enthält und mindestens 1 Instanz aus jedem relevanten MI-Objekt.

## Bemerkungen:

- Annahme der Unabhängigkeit der Dimensionen.
- Es gibt mehrere Algorithmen
- Rechtecke müssen nicht in allen Dimensionen beschränkt sein



## Erfolgreichster Algorithmus: Iterated Discrimination

- Beginne mit einer positiven Instanz und expandiere ein achsenparalleles Rechteck (Methode: Grow)
- Selektiere relevante Features. Expansion nur entlang der relevanten Dimensionen (Methode: Discrim)
- Nachdem ein minimales umgebendes Rechteck gefunden wurde, wird Rechteck noch weiter expandiert um Generalisierung zu verbessern. (Methode: Expand)

284

# Klassisches Multi-Instanz Lernen

---

## Iterated Discrim: Grow

wähle positive Start-Instanz  $o^*$ .

WHILE NOT alle  $O_i$  mit  $CL(O_i) = „relevant“$  enthalten Instanz in  $R$  DO

    Wähle positive Instanz  $s$

        mit Objekt von  $s$  noch nicht in  $R$  repräsentiert

        und  $R$  wächst minimal nach hinzufügen von  $s$

Vergrößere  $R$  um  $s$

//Backfitting (optionale Verbesserung)

    betrachte alle vorherigen Entscheidungen und vertausche

    gegebenenfalls Instanzen aus bereits abgedeckten Objekten

    um  $R$  zu verkleinern

285

# Klassisches Multi-Instanz Lernen

---

## Iterated Discrim:

### Discrim

Dimension  $d$  ist „stark diskriminierend“ bzgl. der negativen Instanz  $o_i$  falls:

1.  $o_i$  liegt weiter außerhalb  $R$  als 1  $\hat{A}$
2.  $o_i$  liegt weiter außerhalb  $R$  als bei allen anderen Features

### Algorithmus

1. Ranke Dimensionen absteigende nach Anzahl „stark diskriminierter“ Instanzen
2. Wähle beste Dimension  $d$
3. Streiche alle Instanzen, die in  $d$  stark diskriminiert werden
4. Falls genug Features selektiert wurden => Abbruch  
    sonst springe zu Schritt 1.

286

# Klassisches Multi-Instanz Lernen

---

## Iterated Discrim: Expand

Verbessere Generalisierung durch weitere Expansion entlang relevanter Dimensionen.

**Idee:** Schätze Wahrscheinlichkeit, dass neue positive Instanz innerhalb der Grenzen liegen.

Verschiebe Grenzen bis W'keit, dass neue positive Instanz über Grenzwert  $\epsilon$  liegt.

Wahrscheinlichkeit mit Kernel-Density-Estimation.

## gesamtes Ablaufschema:

Hauptschleife

Grow (Finde Rechteck bzgl. aktueller Feature-Menge)

Discrim (Selektiere aktuelle Features)

Expand (Generalisiere die gefundenen Features)

287

# Klassisches Multi-Instanz Lernen

---

## Expectation Maximization Diverse Density Klassifikation (EM-DD)

**Idee:** Beschreibe das „relevant“-Konzept durch einen Punkt  $h$  und

Gewichtungsfaktoren  $s_d$  für die Dimensionen  $D=\{d_1, \dots, d_m\}$  des Feature-Raums.

Bestimmen der Klassenkonfidenz für „relevant“-Klasse:

$$Label(O_i | h, \vec{s}) = \max_j \left\{ \exp \left[ - \sum_{i=1}^m (s_i (o_{j,i} - h_i))^2 \right] \right\}$$

Mit  $l=0$  für „relevant“ und  $l=1$  für „irrelevant“ kann man die Qualität des Modells mit der negativ logarithmischen Diverse Density (NLDD) beschreiben:

$$NLDD(h, \vec{s}, DB) = \sum_{i=1}^{|DB|} \left( - \log \left( l_i - Label(O_i | h, \vec{s}) \right) \right)$$

288

# Klassisches Multi-Instanz Lernen

---

## Der EM-DD Algorithmus:

Initialisiere  $h$  //z.B. Centroid von Sample Instanzen aus rel. Obj,  $s_i = 0.1$

While(  $NLDD_{\text{new}} < NLDD_{\text{old}}$  )

FOR ALL  $O_i$  in DB mit  $CL(O_i) = \text{„relevant“}$  DO

$$o_i^* = \arg \max_{o_{ij} \in O_i} (\text{Label}(O_i | h, \vec{s}))$$

$$h' = \arg \max_{h \in H} \prod_{i=1}^n \Pr(l_i | h, \vec{s}, o_i^*) \quad // \text{Suche mit Gradient Descent}$$

$NLDD_{\text{old}} = NLDD_{\text{new}}$

$NLDD_{\text{new}} = NLDD(h', D)$

$h = h'$

return  $h$

Anmerkung:  $\Pr(l_i | h, \vec{s}, o_i^*) = \exp \left[ - \sum_{i=1}^m (s_i (o_i^* - h_i))^2 \right]$

289

---

## Multi-Instanz Klassifikation

Abschließende Bemerkungen:

### *allgemeines MI-Lernen*

- Wenige Arbeiten für allgemeine Multi-Instanz Klassifikation
- Viele Lösungsansätze über Distanzen oder Kernel

### *klassisches MI-Lernen*

- sehr viele Arbeiten im Bereich klassisches Multi-Instanz Lernen
    - Citation-kNN und Bayes-kNN (MI nächste Nachbar Lerner)
    - Multi-Instanz Entscheidungsbäume und Regeln
    - Neuronale Netze für MI-Probleme
- ⇒ EM-DD momentan der Leistungsfähigste  
Algorithmus auf Standard Benchmark (MUSK 1 & 2)

290



## 6.5 Multi-Instanz Clustering

---

- MI-Objekte werden meist mit distanzbasierten Verfahren wie k-Medoid oder DBSCAN, OPTICS geclustert.
  - Auswahl der Distanzfunktion beeinflusst das Ergebnis meist sehr stark
  - nur reine distanzbasierte Verfahren anwendbar (z.B. k-Means nicht anwendbar da keine Centroide auf Mengen von MI-Objekten gebildet werden können)
- konzeptbasiertes MI-Clustering  
Beim Clustering von Multi-Instanz Objekten gelten wieder ähnliche Annahmen wie bei der Klassifikation:
  1. Instanzen sind Ausprägungen bestimmter Konzepte.
  2. Multi-Instanz Objekte werden über die Art und Anzahl der in Ihnen auftretenden Konzepte beschrieben.=> Ähnliche MI-Objekte enthalten Ausprägungen derselben Konzepte

### Ausblick:

- statistischen konzeptionelles MI-Clustering
- dichtebasierte konzeptionelles MI-Clustering

291

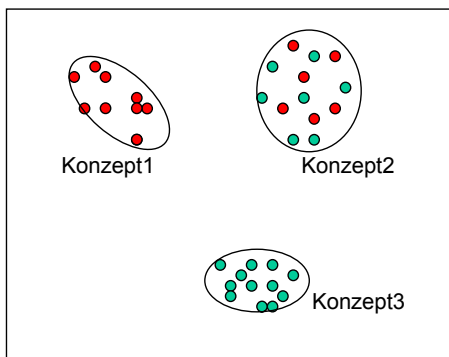
## Konzeptionelles Multi-Instance Clustering

---

### Idee:

Jede Instanz  $o_i \in O$  repräsentiert ein Konzept.

Multi-Instanz (MI-) Cluster sind dann Verteilungen über den Konzepten.



**MI-Cluster1** enthält Instanzen aus Konzept1 und Konzept 2.

**MI-Cluster2** enthält Instanzen aus Konzept2 und Konzept 3.

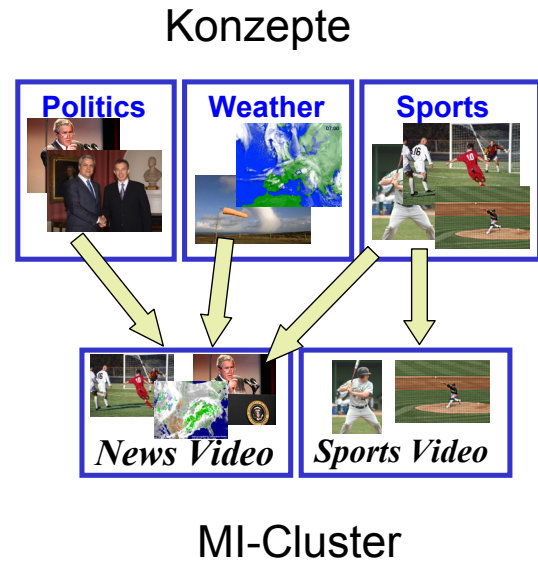
Beschreibung eines MI-Clusters = Clusterbeschreibung der beteiligten Konzepte

292

# Konzeptionelles Multi-Instance Clustering

## Beispiel: Video Daten

- Videos werden als Mengen von Shots/Szenen betrachtet
  - Shots haben einen bestimmten Typ oder ein Konzept (*Weather*)
  - Vollständige Videos sind Mengen von Shots (=MI-Objekte)
- ⇒ MI-Cluster sind MI-Objekte, die Instanzen der gleichen Konzepte enthalten.  
(Sport-Videos enthalten nur Sport-Shots)



293

## Ein statistisches Multi-Instance Modell

### Definition 1: Instance Set

- **DB** ist eine Menge von MI-Objekten  $o = \{i_1, \dots, i_k\}$
- Das Instance Set  $I_{DB}$  von DB ist: 
$$I_{DB} = \bigcup_{DB} o$$

### Definition 2: Instance Model

Ein Instance Model  $IM$  für  $I_{DB}$  ist definiert durch:

- $k$  statistische Prozesse, die die Konzepte im Instanz-Feature-Raum beschreiben z.B.. Gauß-Prozesse mit Erwartungswert  $\mu_j$  und Kovarianzmatrix  $\Sigma_j$ .
- eine Apriori-Verteilung über diesen Prozessen  $\Pr[ k_j ]$ .

294

# Ein statistisches Multi-Instance Modell

## Definition 3: Multi-Instanz Cluster Model

- Ein Menge  $C$  von MI-Clustern über den Instance Model  $IM$ .
- Alle MI-Cluster  $c \in C$  werden beschrieben durch:
  - Apriori-Wahrscheinlichkeit  $Pr[c]$ ,
  - eine Verteilung  $Pr[Card(o) | c]$ , die die Kardinalität beschreibt.
  - Eine bedingte Wahrscheinlichkeit  $Pr[i \in k | i \in o \in c]$  (shorter:  $Pr[k | c]$ ) für jedes Konzept  $k$  in  $IM$ .

Die totale W'keit eines Objekts  $o$  wird wie folgt berechnet:

$$Pr[o] = \sum_{c \in C} Pr[c] \cdot Pr[Card(o) | c] \cdot \prod_{i \in o} \prod_{k \in IM} Pr[k | c]^{Pr[k|i]}$$

Die a-posteriori-Wahrscheinlichkeit für  $o$  und Cluster  $c$  wie folgt:

$$Pr[c | o] = \frac{1}{Pr[o]} Pr[c] \cdot Pr[Card(o) | c] \cdot \prod_{i \in o} \prod_{k \in IM} Pr[k | c]^{Pr[k|i]}$$

# Ein statistisches Multi-Instance Modell

**Beispiel:** 2 MI-Cluster

Cluster 1: ▲

50 % Apriori-W'keit

erwartete Anzahl von Instanzen: 2 3

Konzept1	Konzept2	Konzept3
0.2 <span style="border: 1px solid black; padding: 2px;">1</span>	0.01	0.79 <span style="border: 1px solid black; padding: 2px;">2</span>

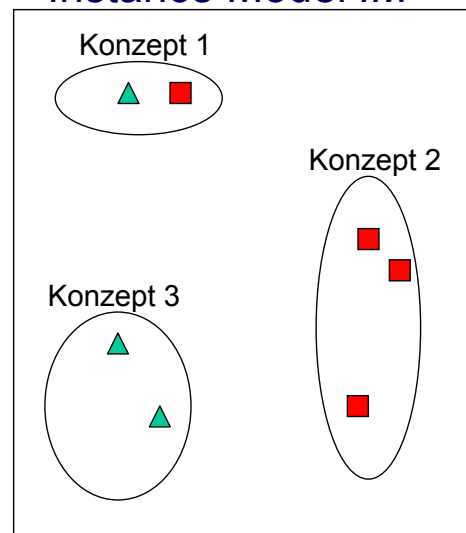
Cluster 2: ■

50 % Apriori-W'keit

erwartete Anzahl von Instanzen: 5 4

Konzept1	Konzept2	Konzept3
0.1 <span style="border: 1px solid black; padding: 2px;">1</span>	0.89 <span style="border: 1px solid black; padding: 2px;">3</span>	0.01

## Instance Model $IM$



## Überblick über den Algorithmus:

- 1- Leite ein Mixture-Model ( $IM$ ) für das Instance Set  $I$  ab.
- 2- Berechne eine gute Start-Verteilung der MI Objekte
- 3- Anpassen der Verteilungen über den Konzepten mit EM

---

## Schritt (1) und Schritt (2)

---

Schritt(1):

Bilde  $I_{DB}$  und verwende EM-Clustering um  $IM$  abzuleiten.

Schritt(2):

- Für alle MI-Objekte  $O$  bilde einen “Confidence Summary Vector”  $CSV(o)$ .
  - Jede Dimension entspricht einem Konzept.
  - Die  $i$ -te Komponente des  $CSV(o)$  ist definiert durch:

$$CSV_j(o) = \sum_{i \in O} \Pr[k_j] \cdot \Pr[i | k_j]$$

- Cluster  $CSV_d$  mit k-Means clustering um eine initiale Zuordnung zu bestimmen.

## Schritt 3

---

### Schritt 3:

**E-Step:** Bestimme Log-Likelihood des aktuellen Modells  $M$ .

$$E(M) = \sum_{o \in DB} \log \sum_{c_i \in C} \Pr[c_i | o]$$

**M-Step:** Anwendung der folgenden Parameterverbesserungen:

update Apriori-W'keit:  $W_{c_i} = \Pr[c_i] = \frac{1}{\text{Card}(DB)} \sum_{o \in DB} \Pr[c_i | o]$

update Vert. Kardinalität:  $l_{c_i} = \frac{\sum_{o \in DB} \Pr[c_i | o] \cdot \text{Card}(o)}{\text{Card}(DB)} \cdot \frac{1}{\text{MAXLENGTH}}$

update Konzept Verteilung:  $P_{k_j, c_i} = \Pr[k_j, c_i] = \frac{\sum_{o \in DB} \left( \Pr[c_i | o] \cdot \sum_{u \in o} \Pr[u | k_j] \right)}{\sum_{o \in DB} \Pr[c_i | o]}$

299

---

## EM-Clustering auf MI-Objekten

---

### Fazit:

- Konzeptionelles MI-Clustering, das ohne explizite Annahme eines bestimmte Ähnlichkeitsbegriff auskommt
- 3. Schritt basiert auf einem Multinomial-Prozess über den Konzeptbeschreibung.
- Komplexität wie bei EM auf den Instanzen
- Nachteil: Verfahren funktioniert schlecht, wenn nicht bekannt ist wie viele Konzepte und wie viele Cluste relevant sind

=> dichtebasierter Ansatz

300