Ludwig Maximilians Universität München
Institut für Informatik
Lehr- und Forschungseinheit für Datenbanksysteme

Skript zur Vorlesung

Knowledge Discovery in Databases II

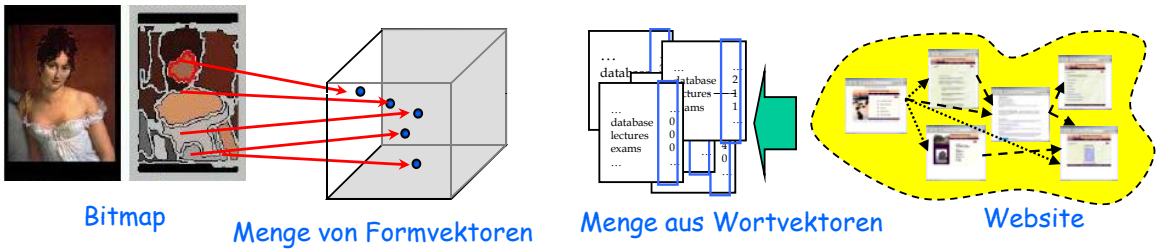
im Sommersemester 2010

Kapitel 7:Multi-Instanz Data Mining

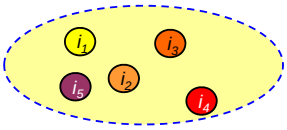
http://www.dbs.ifi.lmu.de/Lehre/KDD

Kapitelübersicht

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)

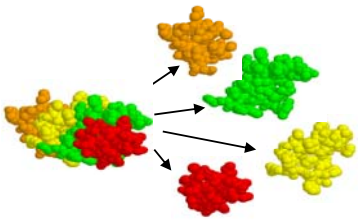


Wichtig: 1. Alle Instanzen im gleichen Featureraum.
2. Reihenfolge wird nicht berücksichtigt.

Was sind Multi-Instanz Objekte ?

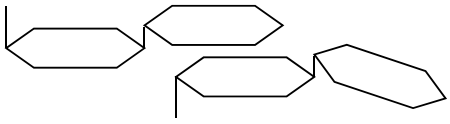
Proteine

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



Macro-Moleküle

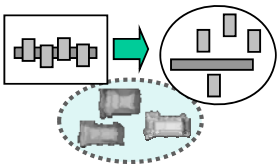
- verschiedene räumliche Ausrichtungen
- jede räumliche Ausrichtung ist eine Instanz
- Menge aller Ausrichtungen stellt das Moleküle dar



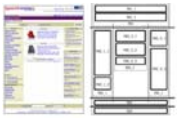
Was sind Multi-Instanz Objekte ?

Weitere Anwendungen :

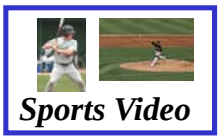
• CAD-Bauteile: Mengen von Raumprimitiven / Überdeckungen.



• HTML-Dokumente: Menge aus thematischen Blöcken.



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



Formal:

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

wobei R der Darstellungsraum der Komponenten ist.

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 Abstandsbasiertes Data Mining oder Kernel Verfahren.

3. **Algorithmen auf Multi-Instanz Objekten**
   Direkte Integration der Multi-Instanz Objekte in die Verfahren.
   Klassisches Multi-Instanz Lernen

7.1. Aggregationsbasierte Ansätze

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

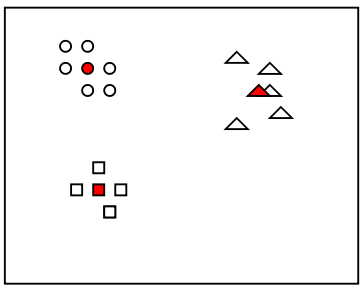
z.B. Aggregation durch Centroid

=> einfaches Verfahren, das eine Menge über Durchschnittswerte beschreibt

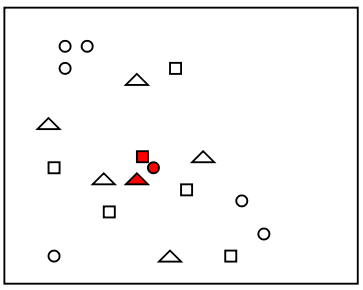
Probleme:

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

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 die Instanzen eines Objekts alle ähnlich zueinander sind, dann ist eine Aggregation meist eine ausreichende Darstellung des Objekts.

7.2 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.

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

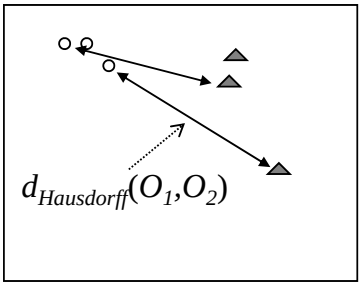
⇒ maximale, minimale Distanz = Ähnlichkeit der Instanz, die am wenigsten durch eine Distanz im anderen MI-Objekt dargestellt wird.

⇒ Jede Instanz in O_1 hat mindestens eine Instanz 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:

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



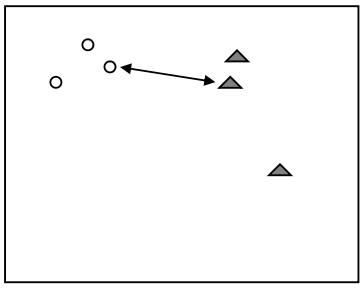
Minimal-Hausdorff-Distanz

Idee: Zwei MI-Objekte sind gleich, wenn sie mindestens eine 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:

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



Bemerkungen:

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

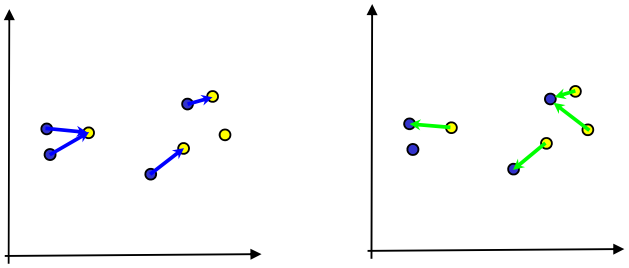
Summe Minimaler Distanzen (SMD)

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:

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



Berechnung von Hausdorff und SMD

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

- 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.

Minimal Matching Distanz (MMD)

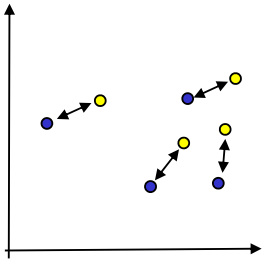
Idee: 2 MI-Objekte sind ähnlich, wenn es zu jeder Instanz **genau** eine ähnliche Instanz im anderen Objekt gibt.

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

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

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

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.



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 eine minimale Anzahl markiert ist.
6. Falls minimale Anzahl an markierten Spalten und Zeilen = n, dann permutiere 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 allen unmarkierten Elementen
   c. Addiere Minimum auf die Schnittelemente der markierten Zeilen und Spalten
   d. gehe zu Schritt 5

Berechnung Minimal Matching Distanz

Beispiel:

Abstandmatrix

10	12	20	21
10	12	21	24
14	17	28	30
16	20	30	35

Abziehen Zeilenmin.

0	2	10	11
0	2	11	13
0	3	14	16
0	4	14	19

Abziehen Spaltenmin.

0	0	0	0
0	0	1	2
0	1	4	5
0	2	4	8

Markieren

0	**0**	**0**	**0**
0	**0**	**1**	**2**
0	**1**	**4**	**5**
0	**2**	**4**	**8**

Addieren/Abziehen min

1	1	0	0
0	0	0	1
0	1	3	4
0	2	3	7

Markieren

1	**1**	**0**	**0**
0	**0**	**0**	**1**
0	**1**	**3**	**4**
0	**2**	**3**	**7**

Addieren/Abziehen min

2	1	0	0
1	0	0	1
0	0	2	3
0	1	2	6

Ergebnis bestimmen

2	**1**	**0**	**0**
1	**0**	**0**	**1**
0	**0**	**2**	**3**
0	**1**	**2**	**6**

$$D_{mmd}(O_1,O_2)=21+21+17+16 = 75$$

10	12	20	21
10	12	21	24
14	17	28	30
16	20	30	35

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.

7.4 Multi-Instanz Klassifikation

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

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 mindestens eine Instanz relevant ist.

Generelle Multi-Instanz Klassifikation:

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

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 zur Beschreibung 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 die Kardinalität der Instanzen pro Konzept eine Rolle (5 Torwärte und 1 Feldspieler sind keine Fußballmannschaft)

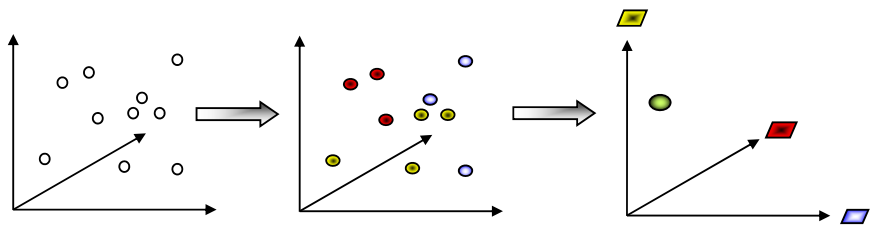
Generelle Multi-Instanz Klassifikation

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 auf den vorkommenden Konzepten auf Klassen abbildet.
- 1. Klassifikator ist dabei Klassifikationsproblem im Feature-Raum.
- 2. Klassifikator wird im Feature der Konzepthäufigkeiten trainiert



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_1 in einem MI-Objekt der Klasse C auftritt?“

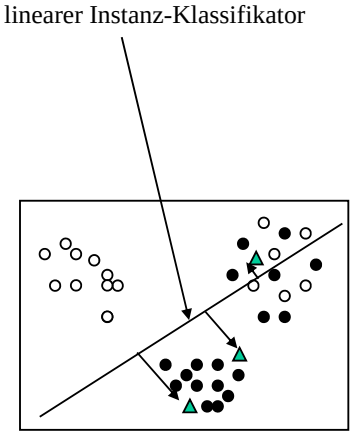
- Kombiniere die Wahrscheinlichkeiten für alle Instanzen im MI-Objekte und gebe die Klasse mit der maximalen Wahrscheinlichkeit zurück. (Modellieren der Menge über einen Multinomial-Prozess = „Ziehen aus Urne mit Zurücklegen“)

Bemerkung:

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

Generelle Multi-Instanz Klassifikation

Beispiel: 2 Klassen mit 3 unbekannten Konzepten



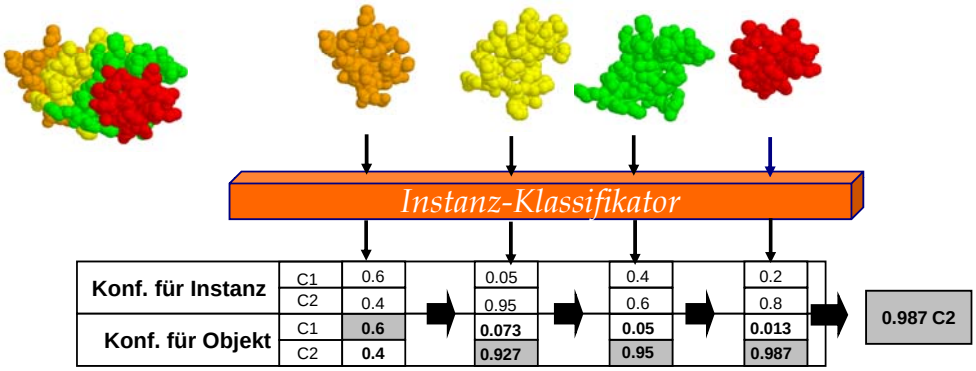
- Trainingsmenge für Instanz-Klassifikator für die Klasse A

$$TR_A = \bigcup_{O_i \in DB} \{o_j \in O_i \wedge CL(O_i) = A\}$$

- 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.

Generelle Multi-Instanz Klassifikation

Beispiel: Kombination der Wahrscheinlichkeiten



Konfidenz von O für Klasse C_k: (Satz von Bayes)

$$\Pr[C_k \mid O] = \frac{\Pr[C_k] \cdot P[O \mid C_k]}{\sum_{i \in C} \Pr[C_i] \cdot P[O \mid C_i]}$$

mit

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

Klassisches Multi-Instanz Lernen

Problemstellung: Es existiert genau 1 unbekanntes Konzept K_{rel}. Alle MI-Objekte, die mindestens eine 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 Moschus oder nicht ? [Dietterich et al. 1998]

Moleküle werden als MI Objekte betrachtet. Instanz = räumliche Konformation. Eine Konformation kann hier ausschlaggebend dafür sein, dass Molekul nach Moschus riecht. Umgekehrt riecht ein Molekül nicht, wenn keine derartige Konformation vorhanden ist.

2- Suche Lungenembolien

Patient soll auf Blutgerinnsel 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 es existiert mindestens eine Lungenembolie muss der Patient behandelt werden

Klassisches Multi-Instanz Lernen

Lösungsansätze:

Klassifiziere alle einzelnen Instanzen

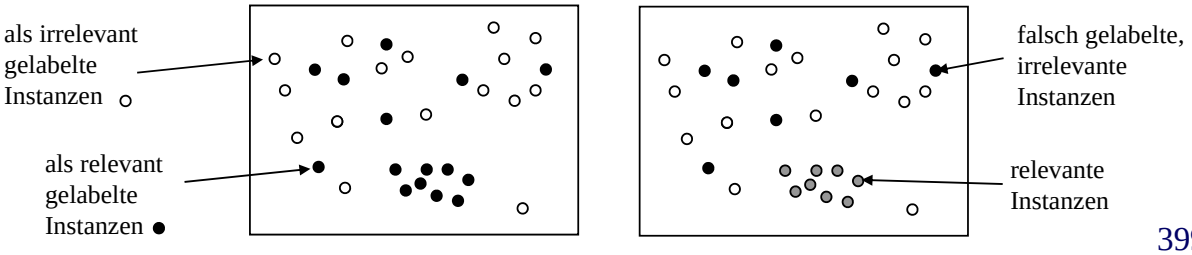
=> Falls eine Instanz relevant ist, ist auch das Objekt relevant.

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



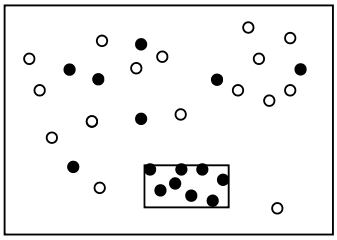
Klassisches Multi-Instanz Lernen

MI-Lernen mit achsenparallelen Rechtecken

Finde das kleinste Rechteck, das keine Instanz aus einem irrelevanten MI-Objekt enthält und mindestens eine 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)

Klassisches Multi-Instanz Lernen

Iterated Discrim: Grow

wähle positive Start-Instanz *o**.
WHILE NOT alle O_i mit $CL(O)$ = „*relevant*“ enthalten Instanz in *R* DO
Wähle positive Instanz *s*
deren Objekt noch nicht in *R* repräsentiert ist
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

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 Â
2. o_i liegt weiter außerhalb *R* als bei allen anderen Features

Algorithmus

1. Ranke Dimensionen absteigend 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.

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 ε 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)

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 $D=\{d_1,..,d_m\}$ des Feature-Raums.

Bestimmen der Klassenkonfidenz für „relevant"-Klasse:

$$Label(O_1 \mid h, \vec{s}) = \max_j \left\{ \exp\left[-\sum_{i=1}^{m} \left(s_i \left(o_{j,i} - h_i \right) \right)^2 \right] \right\}$$

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

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

Klassisches Multi-Instanz Lernen

Der EM-DD Algorithmus:

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

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

FOR ALL O_i in DB mit CL(O_i) = „relevant“ DO

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

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

$NLDD_{old} = NLDD_{new}$

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

$h = h'$

return h

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

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 Nachbarn Lerner)
  - Multi-Instanz Entscheidungsbäume und Regeln
  - Neuronale Netze für MI-Probleme

  $\Rightarrow$ EM-DD momentan der Leistungsfähigste Algorithmus auf Standard Benchmark (MUSK 1 & 2)

7.5 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

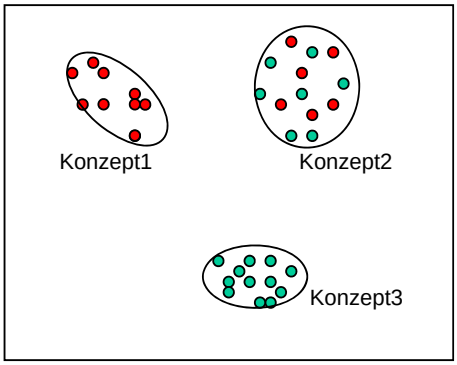
Ausblick:

- statistisches konzeptionelles MI-Clustering
- dichtebasiertes konzeptionelles MI-Clustering

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

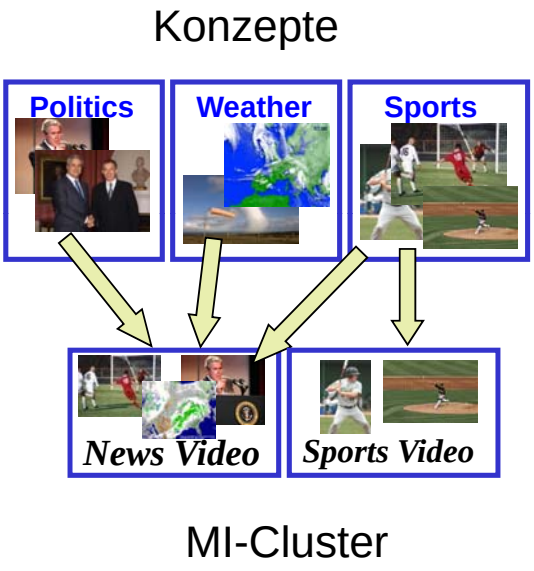
Konzeptionelles Multi-Instance Clustering

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)

$\Rightarrow$ MI-Cluster sind MI-Objekte, die Instanzen der gleichen Konzepte enthalten.

(Sport-Videos enthalten nur Sport-Shots)



Ein statistisches Multi-Instance Modell

Definition 1: Instance Set

- ***DB*** ist eine Menge von MI-Objekten $o = \{i_1,\ldots,i_k\}$
- 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 μ_j und Kovarianzmatrix Σ_j.
- eine Apriori-Verteilung über diesen Prozessen Pr[k_j].

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* [$i \in k \mid i \in o \in c$] (kürzer: *Pr* [*k* | *c*]) für jedes Konzept *k* in *IM.*

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

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

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

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

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 **1**	0.01	0.79 **2**

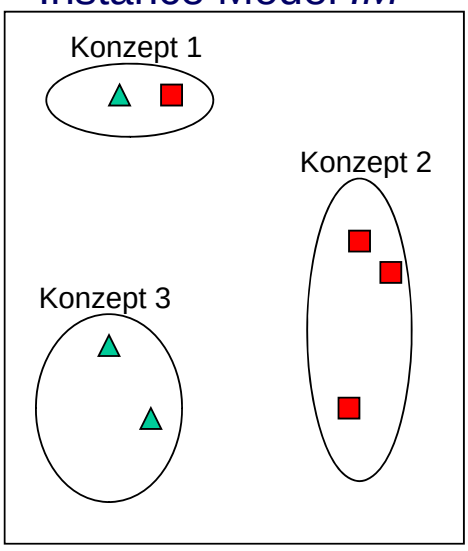
Cluster 2: ■

50 % Apriori-W'keit

erwartete Anzahl von Instanzen: 5 **4**

Konzept1	Konzept2	Konzept3
0.1 **1**	0.89 **3**	0.01

Instance Model *IM*



EM-Clustering auf MI-Objekten

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:

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

- Clustere 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.

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

M-Step: Anwendung der folgenden Parameterverbesserungen:

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

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

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

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 Cluster relevant sind

=> dichtebasierter Ansatz

COSMIC: Conceptually Specified MI-Clusters

Grundidee:

- dichtebasiertes hierarchisches Clustering für MI-Objekte
- Konzepte werden durch dichte Bereiche im Instanzraum beschrieben. (robust gegenüber Parameter-Wahl)

  => Clustering der Instanzen mit OPTICS-ähnlichem Verfahren
- MI-Objekte bestehen dann wieder aus Realisierungen dieser Konzepte.
- MI-Cluster werden über Menge von Objekten beschrieben die Instanzen in den gleichen Konzepten haben.

  => Cluster können überlappen.

  => Cluster können Ober- und Teilcluster von einander sein, wenn Beschreibung nur eine Teilmenge der Konzepte enthält.

Konzept Gitter

- Ein Object *O* wird wird durch die Menge aller enthaltenen Konzepte $Desc(O) \subseteq A$ beschrieben.(*A* = Menge aller Konzepte)
- Die binäre Relation *I* bildet dann Konzepte auf Objekte ab: $I \subseteq DB \times A$

Cluster: 1. $C = \{o \in DB \mid \forall a \in Desc(C) : (o,a) \in I\}$

2. $Desc(C) = \{a \in A \mid \forall o \in C : (o,a) \in I\}$

Beispiel: Gegeben eine Bilddatensatz *DB* und die Konzepte *A:* $\{Haus, Dach\} \subseteq A$

- $C = \{$*alle Bilder die Häuser mit Dächern darstellen*$\}$
- $Desc(C) = \{Haus, Dach\}$

$\Rightarrow$ Es existiert kein Bild $i \in DB$, das ein Haus mit einem Dach darstellt, das nicht Element von *C* ist.

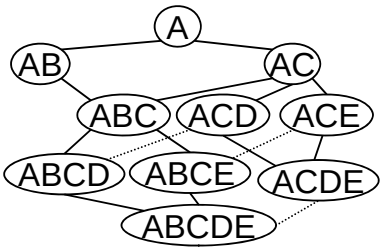
$\Rightarrow$ Es existiert kein weiteres Konzept *k'*, das ebenfalls alle Bilder des Clusters *C* beschreiben würde.

Konzeptionelles Dichtebasiertes MI-Clustering

1. Gegeben $O = \{o_1,..,o_n\}$: bilde $o_k \in IR^d$ auf das diskrete Attribut A ab
   => Clustering der Instanzen

2. Bilde alle möglichen MI-Cluster
   => Konzept Gitter

3. Jeder Cluster beschreibt eine Menge von MI-Objekten mit einer Menge von ähnlichen Instanzen
   - ⇒ die Cluster überlappen
   - ⇒ die Größe von $Desc(C)$ beschreibt die Stärke der Verbindung
   - ⇒ Das Konzept Gitter beschreibt also alle Möglichkeiten um Cluster zu bilden



Beispiel für ein Konzept Gitter

Übersicht über COSMIC

1. Bilde Instanz-Menge und verwende angepassten OPTICS-Algorithmus um Reachability Plot abzuleiten.
2. Durchlaufe Reachability Plot mit einem Top-Down-Sweepline Algorithmus, der Konzepte und Konzeptgitter expandiert.

Herausforderungen:

Sehr viele dichtebasierte Instanz-Cluster

=> Auswahl der Instanz-Cluster zur Einschränkung der Konzepte

Kriterien:
1. Ist Instanzcluster allgemein genug für Konzept?
2. Ist Konzept notwendig um MI-Cluster zu unterscheiden?

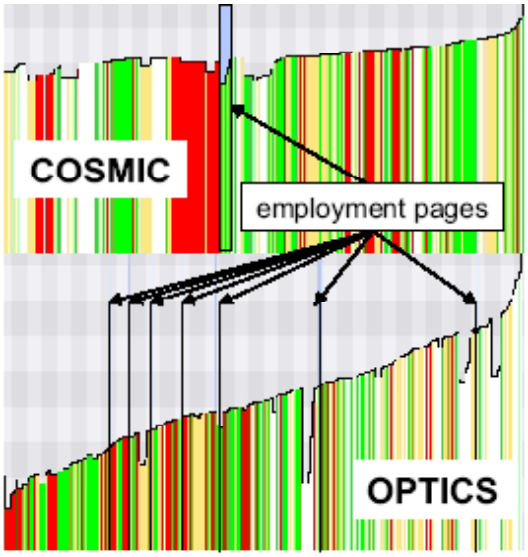
Clustering der Instanzen

Problem: Ein Instanz-Cluster stellt kein Konzept dar, falls nur Instanzen einer oder zu weniger Objekte enthalten sind.
(z.B. Konzept beschreibt nur 1 MI-Objekt)

Lösung: *Erweitere Prädikat*
Für jedes Objekt *O* zählt bei der Bestimmung der Kern-Distanz nur noch die nächst gelegene Instanz anderer Objekte.

⇒ Kerndistanz hängt von mindestens MinPts Objekten ab, anstatt von MinPts Instanzen.

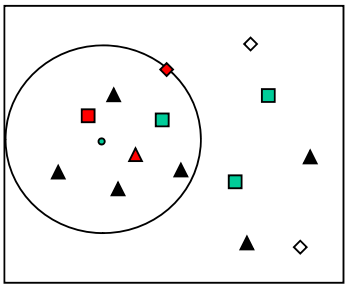
⇒ Bei der Erreichbarkeit werden wie bisher alle Objekte gezählt.



Beispiel Websites: Nur die Berücksichtigung der Objektzugehörigkeit erzeugt nützliche Konzepte.

Clustering der Instanzen

Definition: *Concept-Core-Distance*
Sei $MinObjs \in IN$, $\varepsilon \in IR^+$ und DB eine Menge von MI-Objekten. $I_{DB} = \bigcup_{DB} o$. Als *MinObjs*-nächste-Nachbarn einer Instanz i bezeichnet man die kleinste Menge $N^{MI}_{MinObjs}(i) \subseteq I_{DB}$, so dass folgende Bedingungen gelten:



$$(1) \forall p \in N^{MI}_{MinObjs}(i), \forall q \in DB \setminus N^{MI}_{MinObjs}(i) : d(p,i) < d(q,i)$$

$$(2) \left| \left\{ MiObj(x) \middle| x \in N^{MI}_{MinObjs}(i) \right\} \right| \geq MinObjs$$

Dann ist $d_{MinObs}(i) = \max\left\{ d(i,q) \middle| q \in N^{MI}_{MinObs}(i) \right\}$ und die Concept-Core-Distance ist definiert durch:

$$ConceptCoreDist^{\varepsilon}_{MinObs}(i) = \begin{cases} d_{MinObs}(i) & : d_{MinObs}(i) \leq \varepsilon \\ \infty & : d_{MinObs}(i) > \varepsilon \end{cases}$$

Definition: Concept-Reachability-Dinstance

$$\text{ConceptReachDist}^{\varepsilon}_{\text{MinObs}}(i, j) = \max\left\{ ConceptCoreDist^{\varepsilon}_{MinObs}(i), d(i, j) \right\}$$

Ableiten von Konzepten

Gegeben: Reachability-Plot mit ConceptReachDist erstellt.

Probleme:

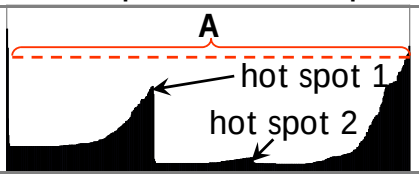
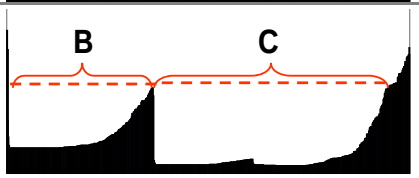
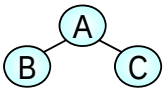
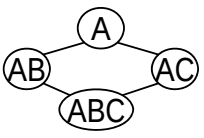
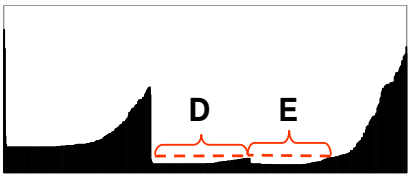
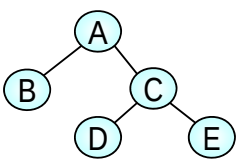
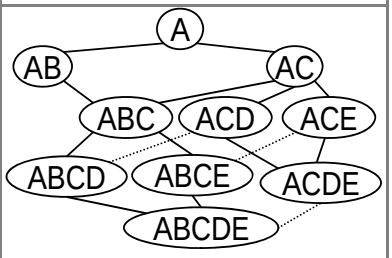
- Konzepte sind nur implizit in Tälern vorhanden.
  => Ableiten der Konzepte.
- Nicht alle Konzepte beschreiben einen ausreichend großen Cluster => Ableiten von überflüssigen Konzepten vermeiden.

Ideen:

- allgemeine Konzepte werden eher benötigt um MI-Cluster zu beschreiben => top-down
- Leite keine Konzepte ab deren Vaterkonzept nicht Teil einer Clusterbeschreibung ist.

Bestimmung von Konzepten und Clustern

Beispielablauf für das Ableiten von Konzepten und Konzept-Gittern

	Reach. plot and hot spots	Konzepte	Konzept Gitter
Step 1			
Step 2			
Step 3			

Hot Spots im Reachability Plot

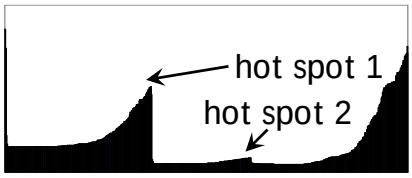
Extraktion der Konzepte aus dem Plot:

- Hot-Spot: Punkt an dem 2 Cluster im Plot unterschieden werden

  (1) $reach(i) > reach(i+1)$

  (2) $\exists l \in IN : reach(i-l) < reach(i) \wedge \forall k : (i-l) < k < i : reach(i) = reach(k)$

  $reach(i)$:= Erreichbarkeitsdistanz auf Position i im Plot
- Hot-Spots werden während des Clusterns der Instanzen erkannt und in einer Priority-Liste bezüglich der maximalen Erreichbarkeitsdistanz abgelegt.
- Hot-Spots trennen Cluster voneinander.



Bestimmung von Konzepten und Clustern

- COSMIC durchläuft die Liste der Hot Spots
- Jeder Hot Spot erzeugt mind. einen neuen Konzept-Kandidaten
- Ist das Vater-Konzept des Kandidaten nicht in der Konzept-hierarchie, dann braucht der Kandidat nicht weiter betrachtet werden.
- Falls Vater in Hierarchie, teste ob mit dem neuen Konzept neue MI-Cluster gebildet werden können.
- Falls ja wird das Konzept zur Konzepthierarchie hinzugenommen und das Konzept Gitter um die neuen Cluster erweitert.

Fazit COSMIC

- COSMIC leitet alle möglichen MI-Cluster bzgl. eines Reachability Plots ab.
- Kann man die Konzepte (Instanz Cluster) mit einem Cluster-Modell gut approximieren (z.B. Centroid, Gauß-Kurve), dann ergibt die Menge der Konzept Beschreibungen eine Beschreibung der MI-Cluster.
- Obwohl COSMIC die Kardinalität der einzelnen Konzept nicht betrachtet, können die resultierenden Cluster diesbezüglich noch untersucht werden
- Overhead für die Analyse des Instanz-Plots ist verschwindend gering im Vergleich zum Clustering der Instanzen

Literatur

- Kriegel H.-P, Pryakhin A., Schubert M. : ***An EM-Approach for Clustering Multi-Instance Objects,*** Proc. 10th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD 2006), Singapore, 2006.
- Kriegel H.-P., Pryakhin A., Schubert M., Zimek A. : ***COSMIC: Conceptually Specified Multi-Instance Clusters*** in proc. 6th int. Conference on Data Mining **(ICDM 2006)**, Hong Kong, China
- Dietterich T.G., Lathrop R.H., Lozano-Perez T. : ***Solving the Multiple Instance Problem with Axis-Parallel Rectangles,*** Artificial Intelligence, vol. 89, num.1-2, Seiten 31-71, 1997
- Weidmann N., Frank E., Pfahringer B. : ***A Two-Level Learning Method for Generalized Multi-instance Problems***. ECML 2003: S. 468-479
- Gärtner T., Flach P.A., Kowalczyk A., Smola A.j. : ***Multi-Instance Kernels***, Proceedings of the 19th International Conference on Machine Learning, p. 179-186, 2002
- Zhang Q., Goldman S. : ***EM-DD: An improved multiple-instance learning technique***. Neural Information Processing Systems 14, 2001.
- Eiter T., Mannila H. : ***Distance Measures for Point Sets and Their Computation***. Acta Informatica, 34(2):103-133, 1997.
- Brecheisen S, Kriegel H.-P., Kröger P., Pfeifle M., Schubert M. : ***Using Sets of Feature Vectors for Similarity Search on Voxelized CAD Objects***, Proc. ACM SIGMOD Int. Conf. on Management of Data (**SIGMOD'2003**), San Diego, CA, 2003