
KDD 2 : TEIL 2

Data Mining in strukturierten Objekten

Skript zur Vorlesung
Knowledge Discovery in Databases II
im Sommersemester 2008

Skript © 2007 Matthias Schubert

318

Data Mining in Strukturierten Objekten

Bis jetzt:

Datenobjekte werden durch Feature-Vektoren repräsentiert:

Aber:

- reelle Objekte können durch viele unterschiedliche Informationen charakterisiert werden.
=> Nicht alle Objekte lassen sich gut als Vektor von Grunddatentypen darstellen
- unterschiedliche Bedeutung der Features
- Strukturierung der Objekte durch Teilobjekte integriert zusätzlich Information.
- Integration von Domain-Knowledge durch Ausnutzen der gegebenen Modellierung

319

Arten von Strukturierten Objekten

1. **Multirepräsentierte Objekte:**
Tupel aus Objekten unterschiedlicher Objekträume.
Bsp.: Farbverteilung und Texturbeschreibung eines Pixelbilds
2. **Multi-Instanz Objekte:**
Mengen aus Objekten. Alle Objekte sind Element des gleichen Objektraums
Bsp.: Konfigurationen eines Moleküls, Warenangebot eines Händlers,..
3. **Sequenzen**
Abfolge von Objekten i.d.R. des gleichen Objektraums
Bsp.: Videos, Audio-Daten, Aminosäureketten, Zeitreihen...
4. **Bäume**
Bäume aus Objekten, wobei Knoten und Kanten durch andere Objekte beschrieben sein können.
Bsp.: Stammbäume, XML-Dateien...
5. **Graphen**
gerichtete/ungerichtete Graphen aus Objekten, wobei Knoten und Kanten durch andere Objekte beschrieben sein können.
Bsp.: Proteine, Bildsegmente, ...

320

Auswahlkriterium für Grad der Strukturierung

Jede Struktur kann als Spezialfall von Graphen aufgefasst werden.

Aber:

- häufig haben Teilobjekte keine bekannte Beziehung zueinander
- Beziehung zwischen Objekten ist oft vernachlässigbar
- Verwendung von graphstrukturierten Daten sehr aufwendig
 - Graph-Isomorphie ist nicht polynomiell berechenbar
 - Subgraph-Isomorphie ist NP-hart

Fazit: Die verwendete Strukturierung der Daten sollte so einfach wie möglich sein, aber die wesentlichen Merkmale erhalten.

Bsp: Zum Vergleich 2er Videos ist die zeitliche Abfolge von Szenen häufig nicht relevant. 2 Videos können schon als ähnlich betrachtet werden, wenn sie ähnliche Szenen in unterschiedlicher Reihenfolge enthalten.
(Sequenz von Szenen => Menge von Szenen)

321

Skript zur Vorlesung
Knowledge Discovery in Databases II
im Sommersemester 2009

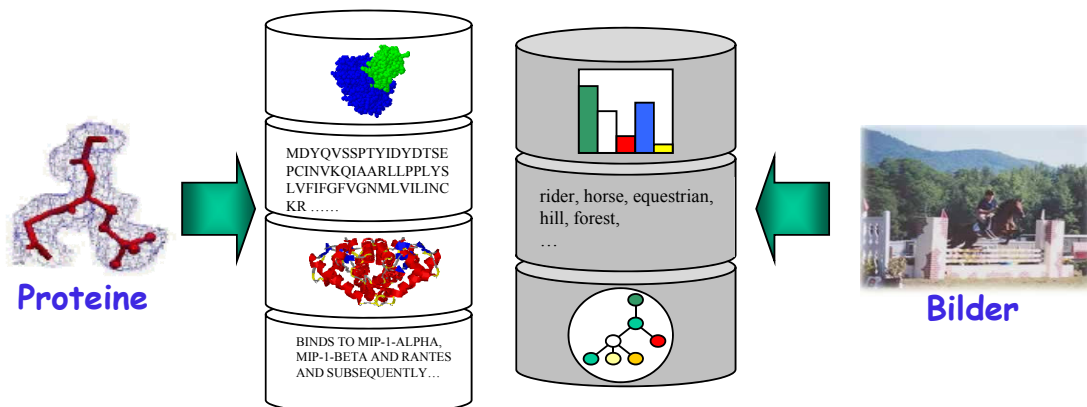
Kapitel 6: Multirepräsentiertes Data Mining

Skript © 2009 Matthias Schubert

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

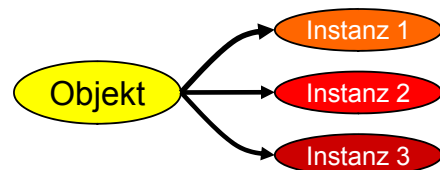
322

Grundsituation



Gründe für Multirepräsentierte Objekte:

- unterschiedliche Featuretransformationen
- unterschiedliche Messtechniken
- unterschiedliche Aspekte desselben Objekts



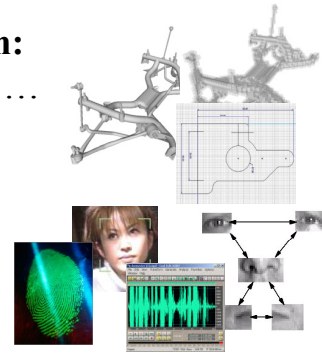
➔ Multirepräsentierte Objekte

323

Was sind Multirepräsentierte Objekte ?

Weitere Anwendungen mit multirepräsentierten Daten:

- CAD-Bauteile: Voxel, Polygonzüge, Formhistogramme ...
- Biometrische Daten: Sprachmuster, Gesichtszüge, Fingerabdrücke...



Formal:

- Objektrepräsentation $o = (r_1, \dots, r_n) \in R_1 \times \dots \times R_n$,
wobei R_i ein Darstellungsraum für die i -te Komponente mit $1 \leq i \leq n$.
- $R_i = O_i \cup \{-\}$, wobei O is a Featureraum
und “-” ein Symbol für fehlende Instanzen.

324

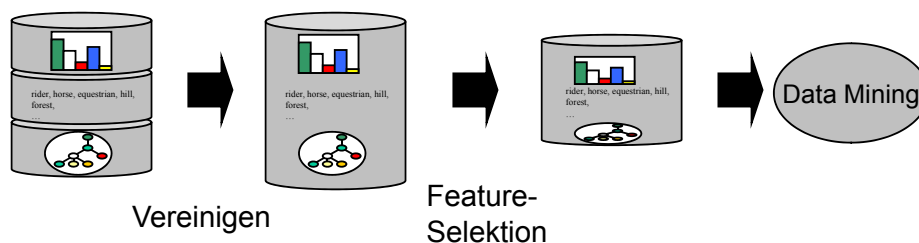
Probleme mehreren Repräsentationen

Grundproblem:

- alle notwendigen Informationen sollen dem Algorithmus zur Verfügung stehen => Verwende alle verfügbaren Informationen
- zu viele unnötige Features können das Ergebnis negativ beeinflussen => Verwende nur notwendige Features

Standard Lösungsansatz:

1. Bilde einen gemeinsamen Feature-Space aus allen Features jeder Repräsentation.
2. Benutze Feature-Reduktion oder Feature-Selektion.
3. Wende Data Mining auf reduzierten Feature-Raum an.



325

Probleme bei mehreren Repräsentationen

Probleme des Standard-Ansatzes :

- nicht alle Features sind vergleichbar.
 - Worthäufigkeit und Farbhäufigkeit bei Bildern
 - Aminosäurehäufigkeit und Zellmilieu bei Proteinen
- bei Nebeneinanderstellen unterschiedlicher Arten von Features, entsteht Informationsverlust bzgl. Vergleichbarkeit der Features

Lösungsansatz:

- ⇒ Data Mining Algorithmus erhält Tupel aus Feature-Vektoren oder anderen Objektdarstellungen.
- ⇒ Relevanz für Problem ist häufig auf Ebene der Repräsentationen zu entscheiden.
Beispiel: Spielen Farben ein Rolle bei Bildähnlichkeit?
- ⇒ hochdimensionale Daten können besser verarbeitet werden, indem Features nach Bedeutung gruppiert betrachtet werden.
=> Wissen über Zusammengehörigkeit der Features bleibt erhalten.
- ⇒ Verwendung von Ensemble Techniken

326

Multirepräsentierte Algorithmen

Möglichkeit zur Kombination mehrerer Repräsentationen:

1. Kombination auf Feature-Ebene:

- unterschiedliche Merkmale werden aus verschiedenen Repräsentationen in einen Feature-Vektor vereint.
- Feature-Selektion oder Selektion der Repräsentation sollen irrelevante Information ausschließen.

Bereits behandelt in Kap.2

2. Kombination der Distanzen und Ähnlichkeiten:

Bestimme Objektähnlichkeit in jeder Repräsentation und kombiniere Ähnlichkeitsaussagen.

3. Kombination auf Muster-Ebene:

Bestimme Muster in jeder Repräsentation und kombiniere die Muster zu allgemeinen Mustern.

Bsp: Kombination der Klassenwahrscheinlichkeiten aus mehreren Repräsentationen.

327

Kapitelübersicht

6.1 Einleitung

Grundproblematik und Motivation

6.2 Multirepräsentierte Ähnlichkeits- und Distanzfunktionen

Lernen von Kombinationsregeln, Normalisierungen

6.3 Klassifikation mit Multirepräsentierten Objekten

Kombination von Klassifikatoren

6.4 Co-Training

Verwendung mehrerer Repräsentation zum Labeln neuer Trainingsobjekte

6.5 Multirepräsentiertes Clustering

dichtebasiertes Clustering, partitionierendes Clustering

328

6.2 Multirepräsentierte Ähnlichkeits- und Distanzfunktionen

Integration der verschiedenen Repräsentationen über Kombination von Ähnlichkeitsmaßen oder Distanzen.

Idee: Erhalte die Trennung der einzelnen Repräsentationen bei und kombiniere auf Ebene der Ähnlichkeitsaussagen.

Beispiel: gewichtete Linear-Kombination

$d_i(o_1, o_2)$: lokale Metrik oder lokaler Kernel in R_i .

$$D_{kombi}(o_1, o_2) = \sum_{R_i \in R} w_i \cdot d_i(o_1, o_2)$$

329

Normalisierung

Der Wertebereich von Distanzen in unterschiedlichen Repräsentationen kann sich stark unterscheiden.

⇒ Normalisierung der Ähnlichkeits- und Distanzwerte ist essentiell

Normalisierung mit Erwartungswert: Sei μ_i^{orig} der Erwartungswert aller Distanzen, die in Repräsentation R_i beobachtet wurden.

$$d_i(o, q) = \frac{d_i^{orig}(o, q)}{\mu_i^{orig}}$$

Bereichs-Normalisierung:

$$d_i(o, q) = \frac{d_i^{orig}(o, q) - \min_{r, s \in D} \{d_i^{orig}(r, s)\}}{\max_{r, s \in D} \{d_i^{orig}(r, s)\} - \min_{r, s \in D} \{d_i^{orig}(r, s)\}}$$

330

Gewichtung der Repräsentationen

- Normalisierung erzeugt Vergleichbarkeit bzgl. der Wertebereiche.
- Semantische Aussage der Repräsentationen ist nicht so einfach bestimmbar. Z.B.: Ist ein sehr ähnliches Farbhistogramm weniger aussagekräftig als eine sehr ähnliche Textbeschreibung?

⇒ Gewichtung der Repräsentationen ist essentiell für Nutzen

Annahme gewichtete Linearkombination: $D_{kombi}(o_1, o_2) = \sum_{R_i \in R} w_i \cdot d_i(o_1, o_2)$

- Auswahl durch Domain-Experten
- Integration in das Optimierungsproblem des Klassifikators (Hyper-Kernels und SVMs)
- explizites Lernen von Gewichten

331

Lernen von Gewichten

Formuliere Ähnlichkeit als lineares Klassifikationsproblem:

Normalenvektor der trennenden Hyperebene setzt sich aus Gewichten zusammen:

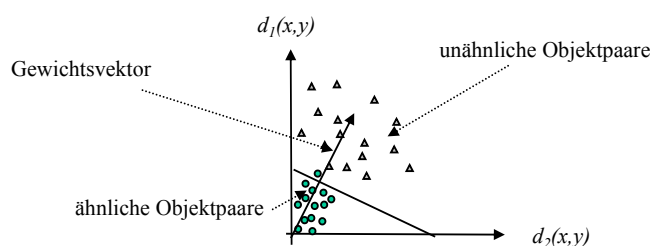
Trainingsobjekte: Paare von ähnlichen und unähnlichen Objekten

Klassen: {„ähnlich“, „unähnlich“}

Feature-Space: Abstandsvektor, $v_i = d_i(x,y)$ für alle Representation R_i $1 \leq i \leq n$

Vorgehen:

- Bestimme Abstandsvektoren auf DB-Sample
(Vorsicht: Es gibt quadratisch viele Abstandsvektoren! => Sample)
- Trainiere linearen Klassifikator
- Bestimme Gewichtungsvektor aus Normalenvektor der Trennebene (MMH).



332

Kombination von Distanzen/Ähnlichkeiten

Bemerkungen:

- Vorsicht: lineare Klassifikatoren garantieren keine positiven Gewichte für alle Repräsentationen!
- Alternativ kann auch der gelernte Klassifikator direkt zur Kombination der Ähnlichkeiten bzw. Distanzen verwendet werden. In diesem Fall ersetzt die Wahrscheinlichkeit für die Klasse „unähnlich“ die Distanz.
- Bei komplexeren Kombinationsregeln müssen die Metrik- bzw. Kernel-Eigenschaften erneut geprüft werden, falls das anschließende Data-Mining-Verfahren diese Eigenschaften benötigt.

333

Multi-Repräsentierte Ähnlichkeitsschätzer

Bisher:

- Distanzmetrik/Skalarprodukt = (Un-)ähnlichkeitsmaß
- Ähnlichkeit ist linear und kann beliebig groß werden (Kernel) oder Unähnlichkeit ist linear und kann unendlich ansteigen.
- Kombinationen von Gewichtungsfunktionen brauchen Trainingsbeispiele

Aber:

- Ähnlichkeit und Unähnlichkeit sind beschränkt:
 - ab einer gewissen Ähnlichkeit werden Objekte als gleich wahrgenommen
 - man unterscheidet ab einer gewissen Unähnlichkeit nicht weiter
- Trainingsbeispiele, die die Ähnlichkeit 2er Objekte beschreiben sind schwer zu erzeugen. (Selbst Menschen labeln häufig inkonsistent!)
- Ein und derselbe Distanzwert kann bei unterschiedlicher Objektähnlichkeit beobachtet werden.

334

Multi-Repräsentierte Ähnlichkeitsschätzer

Lösungsansatz:

Beschreibe Ähnlichkeit, als Wahrscheinlichkeit, dass ein Benutzer beide Objekte als ähnlich betrachtet.

⇒ Abstand in einer Repräsentation wird zu einem Feature das statistisch mit der Ähnlichkeitsaussage korreliert ist.

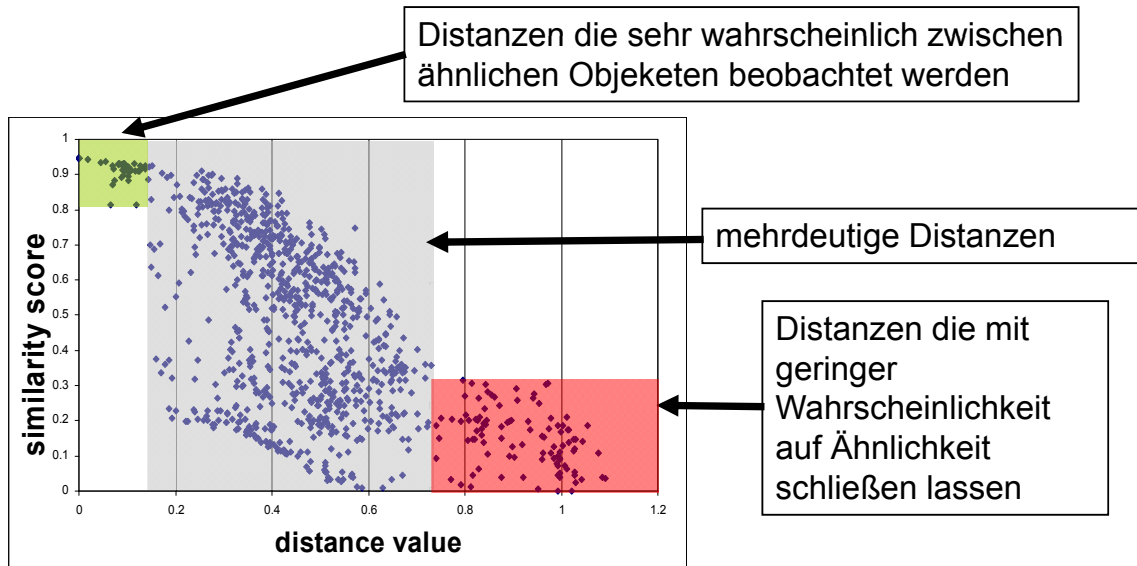
⇒ Ähnlichkeit wird also als bedingte Wahrscheinlichkeit angesehen.

⇒ Die Unähnlichkeitswahrscheinlichkeit kann dann als Ranking-Kriterium für Ähnlichkeitsanfragen, kNN-Klassifikatoren und distanzbasierte Algorithmen verwendet werden.

335

Multi-Repräsentierte Ähnlichkeitsschätzer

Beobachtung: Betrachte Distanzen und Ähnlichkeitsaussagen.



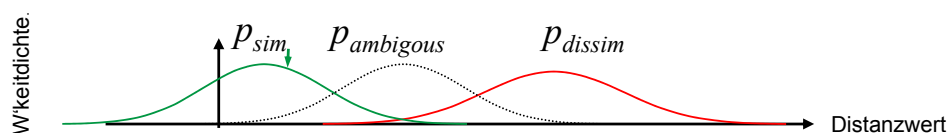
- ⇒ Distanzen sind nur für große und kleine Werte mit Ähnlichkeit korreliert.
- ⇒ mittlere Distanzen können alles bedeuten.

Multi-Repräsentierte Ähnlichkeitsschätzer

Idee:

- Modelliere die Unsicherheit der Aussage als Wahrscheinlichkeitsdichte.
- zur Bestimmung der Ähnlichkeit werden zunächst 2 Verteilungen über die Distanzen von ähnlichen und unähnlichen Objekten betrachtet.
- die Verteilung der Unsicherheit kann dann als kombinierte Wahrscheinlichkeit betrachtet werden, dass beide Verteilungen den gleichen Distanzwert liefern.
- Die Unsicherheit in einem Intervall von Distanzen wird dann wie folgt berechnet:

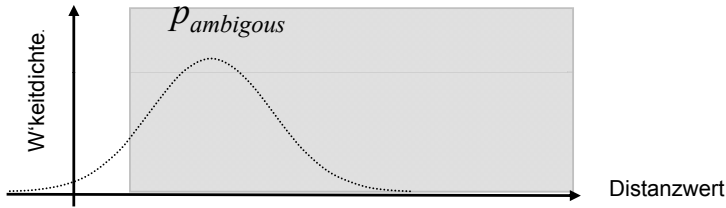
$$P_{\text{ambiguous}}(a, b) = \frac{\int_a^b p_s(x)p_d(x)dx}{\int_{-\infty}^{\infty} p_s(x)p_d(x)dx}$$



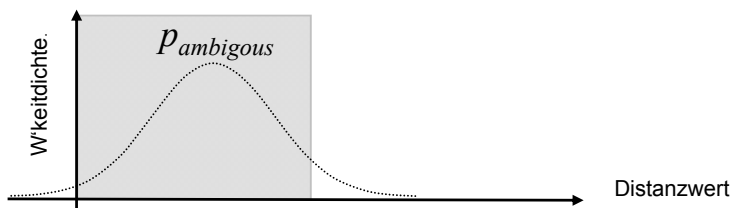
Multi-Repräsentierte Ähnlichkeitsschätzer

Ein Distanz hat eine sichere Aussage, wenn sie nicht zweideutig ist:

- eine sichere Aussage bei der die Distanz kleiner ist als die Mehrzahl der unsicheren Aussagen deutet auf Ähnlichkeit hin.



- eine sichere Aussage bei der die Distanz größer ist als die Mehrzahl der unsicheren Aussagen deutet auf Unähnlichkeit hin.



338

Multi-Repräsentierte Ähnlichkeitsschätzer

Die Wahrscheinlichkeit, dass 2 Objekte bzgl. einer Repräsentation ähnlich sind kann man also folgendermaßen bestimmen:

$$L_{sim}^i(o_1, o_2) = P_{definite}^i(d_i(o_1, o_2) < \delta) = P_{ambiguous}^i(d_i(o_1, o_2) \geq \delta)$$

Kombiniert man diese Wahrscheinlichkeiten für alle Repräsentationen ergibt sich folgendes Ähnlichkeitsmaß:

$$P_{SIM}(o_1, o_2) = \prod_{i=1}^R L_{sim}^i(o_1, o_2) \cong \sum_{i=1}^R \ln(L_{sim}^i(o_1, o_2))$$

Wird eine Distanz benötigt wird das Komplement verwenden:

$$P_{DISSIM}(o_1, o_2) = \prod_{i=1}^R L_{dissim}^i(o_1, o_2) \cong \sum_{i=1}^R \ln(L_{dissim}^i(o_1, o_2))$$

339

Multi-Repräsentierte Ähnlichkeitsschätzer

Wie bestimmt man die Dichtefunktionen für $p_{sim}(x)$ und $p_{dissim}(x)$?

Training mit Beispielen:

- benutze ein ausreichendes Sample von Objektpaaren
- Label können als graduelle Ähnlichkeit oder binär vorliegen (Bei Einteilung in Klassen sind alle Objekte einer Klasse ähnlich.)
- bestimme eine Normalverteilung bei der Ähnlichkeits- bzw. Unähnlichkeitswert eines Objektvergleichs das Gewicht in der Berechnung darstellt.

$$\mu_r^{sim} = \frac{\sum_{o_1, o_2 \in S} sim(o_1, o_2) \cdot d_r(o_1, o_2)}{\sum_{o_1, o_2 \in S} sim(o_1, o_2)}$$

$$Var_r^{sim} = \frac{\sum_{o_1, o_2 \in S} sim(o_1, o_2) \cdot (d_r(o_1, o_2) - \mu_r^{sim})^2}{\sum_{o_1, o_2 \in S} sim(o_1, o_2)}$$

$$\mu_r^{dissim} = \frac{\sum_{o_1, o_2 \in S} dissim(o_1, o_2) \cdot d_r(o_1, o_2)}{\sum_{o_1, o_2 \in S} dissim(o_1, o_2)}$$

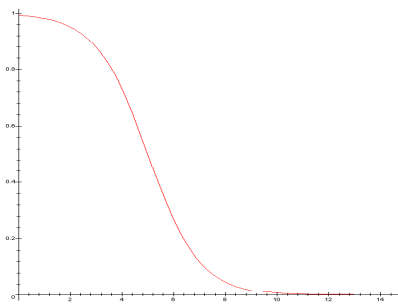
$$Var_r^{dissim} = \frac{\sum_{o_1, o_2 \in S} dissim(o_1, o_2) \cdot (d_r(o_1, o_2) - \mu_r^{dissim})^2}{\sum_{o_1, o_2 \in S} dissim(o_1, o_2)}$$

340

Multi-Repräsentierte Ähnlichkeitsschätzer

- Nach der Bestimmung der Verteilung muss noch das Integral über den Dichtefunktionen berechnet werden.
- Da die Stammfunktion der Normalverteilung unbekannt ist, muß die kumulierte Dichtefunktion approximiert werden.
- Dies kann mit Hilfe einer Sigmoidfunktion erreicht werden:
 - a: regelt Verschiebung
 - b: regelt Steigung

$$sigmoid_{a,b}(x) = \frac{1}{1 + e^{a+b \cdot x}}$$



- Anpassen des Sigmoiden über numerische Methoden (Regression)

341

Multi-Repräsentierte Ähnlichkeitsschätzer

Training ohne Beispiellabel:

Idee: Objekte die in allen Repräsentationen sehr kleine bzw. sehr große Distanzen haben, sind wahrscheinlich auch semantisch ähnlich bzw. unähnlich.

Vorgehen: Co-Learning

- Bilde ein Qualitätskriterium, das die Konsistenz der Wahrscheinlichkeitsaussagen über alle Repräsentationen hinweg mißt
- Verwende EM-Ansatz der dieses Qualitätsmaß minimiert

Ziel: Die Ähnlichkeitsaussagen für dasselbe Objektpaar sollen in jeder Repräsentation möglichst gleich sein.

342

Multi-Repräsentierte Ähnlichkeitsschätzer

EM-Algorithmus zum iterativen Anpassen der Ähnlichkeit:

E-Step: Qualitätsmaß für die Übereinstimmung:

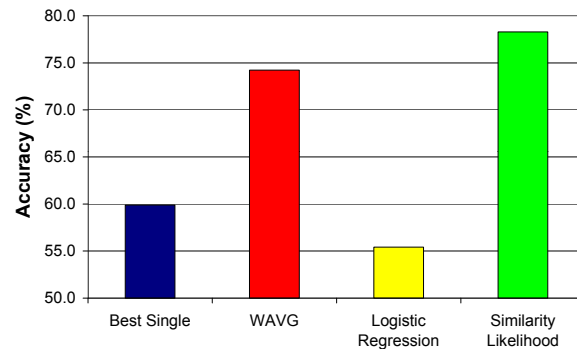
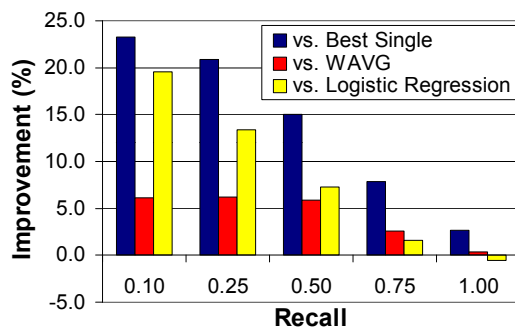
$$L_s^{AVG}(o_1, o_2) = \frac{\sum_{i=1}^R L_S^i(o_1, o_2)}{R}$$
$$Var(S) = \sum_{o_1, o_2 \in S} \sum_{i=1}^R [L_S^i(o_1, o_2) - L_s^{AVG}(o_1, o_2)]^2$$

M-Step: Benutze denselben Ablauf wie bei der Methode mit Trainingsdaten. Anstatt der beobachteten Ähnlichkeitswerte wird allerdings $L_s^{AVG}(o_1, o_2)$ verwendet.

343

Multi-Repräsentierte Ähnlichkeitsschätzer

Ergebnisse: 500 Audiotitel in 6 Repräsentationen.



- Verbesserung der Precision für verschiedene Recall-Werte
- 1 NN Klassifikation
- WAVG entspricht dem normalisiertem Durchschnitt
- LogRegression ist die Anwendung logistischer Regression zur bestimmung von Gewichten (vgl. Folie 273)

344

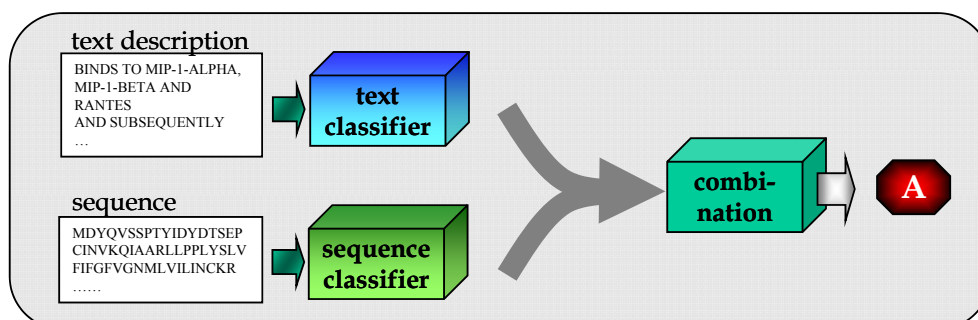
6.3 Multirepräsentierte Klassifikation

Eingabe : $o \in R_1 \times \dots \times R_n$,

wobei R_i der Darstellungsraum für die i -te Repräsentation ist.

Kombination mehrerer Klassifikatoren (Classifier Combination):

1. Trainiere Klassifikator für jede Repräsentation getrennt.
2. Klassifiziere neues Objekte mit jedem Klassifikator
3. Kombiniere die Resultate der Klassifikatoren zur einer globalen Klassenvorhersage.



345

Kombination der Klassifikatoren

Beispiel: Man betrachte 2 Doktoren A und B , die entscheiden sollen, ob ein Patient krank ist oder gesund.

Untersuche den Nutzen der zweiten Meinung in den Fällen in denen nur 1 Dr. richtig liegt, beide Dr. richtig liegen, keiner richtig liegt.

Fall A: kein zusätzlicher Nutzen durch zweiten Experten.

Fall D: kein Nutzen durch irgendeinen Experten.

Fall B und C: Nutzen durch 2 Experten, wenn sich der richtig liegende durchsetzt.

Fazit: Nutzen kann nur bei widersprüchlichen Diagnosen auftreten, sofern sich der Richtige durchsetzt.

		Dr. A	
		richtig	falsch
Dr. B	richtig	A	B
	falsch	C	D

346

Kombination der Klassifikatoren

Der Nutzen der Multirepräsentierten Klassifikation hängt von 2 Faktoren ab:

1. Unterscheidet sich die Vorhersage der einzelnen Klassifikatoren hinreichend genug, um genug potentiellen Nutzen zuzulassen.
=> Wie kann man entscheiden, ob Repräsentationen widersprüchliche Aussagen erzeugen ?
2. In Falle widersprüchlicher Vorhersagen, sollte die Aussage gewählt werden die richtig ist.
=> Wie kann man die Konfidenz der Aussagen bestimmen ?

347

Potenzieller Nutzen der Kombination

Unabhängig von der richtigen Aussage kann Nutzen nur dann entstehen, wenn widersprüchlich Klassenvorhersagen auftreten.

- Klassenvorhersage sollte nicht positiv korreliert sein
Grund: Kein Nutzen, da überwiegend gleiche Aussagen.
- Klassenvorhersage sollte auch nicht negativ korreliert sein.
Grund:
 - Bei vollständiger negativer Korrelation, kann immer nur ein Klassifikator richtig liegen.
 - Die durchschnittliche Vorhersagequalität der Klassifikatoren muss schlecht sein.
=> Experten raten nur !!

Fazit: *Die Klassenvorhersagen und damit die Repräsentationen sollten möglichst unabhängig voneinander sein!*

348

Kombination mehrerer Klassifikatoren

Wie kombiniert man Klassenvorhersagen so, dass die richtige Vorhersage bevorzugt wird?

1. Jeder Klassifikator gibt für jede Klasse A und ein Objekt x eine Vorhersagewahrscheinlichkeit c_A zurück.

Für Konfidenzvektor $c^f(x)$ gilt:
$$\sum_{A \in C} c_A^r(x) = 1$$

2. Klassifikation durch Kombination der Konfidenzevektoren $c^f(x)$:

$$pred(X) = \underset{A \in C}{\mathbf{argmax}} \left(\underset{r \in R}{\Theta} (c_A^r) \right) \text{ mit } \Theta \in \{ \min, \max, \sum, \prod \}$$

349

Kombination mehrerer Klassifikatoren

Beispiel:

Gegeben: 2 Repräsentation für Bildobjekte: Farbhistogramme(R1) und Texturvektoren(R2).

Klassen = { „enthält Wasseroberfläche“=A, „keine Wasseroberfläche“=B }

Bayes Klassifikatoren K1 (für R1) und K2 (für R2)

Kombination mit Summe.

Klassifikation von Bild b:

$$K1(b)=c1=(0.45, 0.55); K2(b) = c2=(0.6, 0.4)$$

Kombination mit Durchschnitt (Summe):

$$c_{\text{global}} = (1.05, 0.95) * \frac{1}{2} = (0.525, 0.475) \text{ und } \text{argmax}(c_{\text{global}}) = A$$

Kombination mit Produkt:

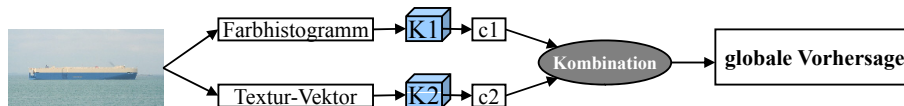
$$c_{\text{global}} = (0.27, 0.22) \text{ und } \text{argmax}(c_{\text{global}}) = A$$

Kombination mit Maximum:

$$c_{\text{global}} = (0.6, 0.55) \text{ und } \text{argmax}(c_{\text{global}}) = A$$

Kombination mit Minimum:

$$c_{\text{global}} = (0.45, 0.4) \text{ und } \text{argmax}(c_{\text{global}}) = A$$



350

Kombination mehrerer Klassifikatoren

Probleme:

- Performanz der kombinierten Klassenvorhersage ist stark von der Aussagekraft des Konfidenzvektor abhängig.

Ist die Konfidenz der Aussage nicht stark mit der Richtigkeit korreliert, wird häufig die falsche Aussage bevorzugt.

- Bestimmte Klassifikatoren erzeugen keinen Konfidenzvektor sondern nur eine Klassenvorhersage.

=> Verfahren zum Abschätzen der Klassenkonfidenz für allgemeine Klassifikatoren

=> Ableitung von zuverlässigen Konfidenzwerten für unterschiedliche Arten von Klassifikatoren.

351

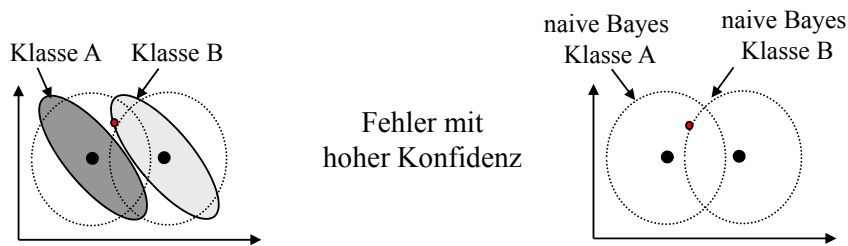
Ableiten von Konfidenzvektoren

Bayes Klassifikatoren

Bayes-Klassifikatoren berechnen ohnehin eine Wahrscheinlichkeit pro Klasse.

Vorsicht: Konfidenzwerte häufig unzuverlässig, weil sie ausdrücken inwieweit ein Objekt zum gelernten Prozess passt.

Aber: Wenn Daten schlecht durch zugrunde liegenden statistischen Prozess beschrieben werden, entstehen leicht falsche Klassifikationsergebnisse mit hoher Konfidenz.



352

Multirepräsentierte k NN-Klassifikation

Idee: Um die Konfidenz für einen k NN-Klassifikator zu bestimmen betrachte die Eindeutigkeit (\equiv Entropie) in der Entscheidungsmenge (k NN-Sphäre).



Intuition: In R_1 scheint der Bereich um das Objekt eindeutig zu Klasse \bullet zu gehören. In R_2 ist der Bereich um das Objekt durch beide Klassen \blacktriangle und \bullet gegeben.

\Rightarrow Vorhersage in R_1 scheint zuverlässiger zu sein als die in R_2

353

Multirepräsentierte k NN-Klassifikation

[Kriegel, Pryakhin, Schubert 2005]

Klassifikator $K:O \rightarrow C$ bildet $o \in O$ auf eine Klasse $c \in C$

O besteht aus den Repräsentationen $R_1 \times \dots \times R_n$

Klassifikation des Objekts $o = (r_1, \dots, r_n)$:

Bestimme Entscheidungsmenge $sphere_i(o, k)$ in jeder Repräsentation R_i

wobei $r_i \neq "-"$ (= die Repräsentation ist vorhanden)

$$sphere_i(o, k) = \{o_1, \dots, o_k \mid o_1, \dots, o_k \in DB_i \wedge \neg \exists o' \in DB_i \setminus \{o_1, \dots, o_k\}$$

$$\wedge \neg \exists \lambda, 1 \leq \lambda \leq k : dist_i(o', r_i) \leq dist_i(o_\lambda, r_i)\}$$

Bestimme Konfidenzvektor $cv_i(o)$ auf folgende Art und Weise:

$$I \quad d_i^{norm}(o, u) = \frac{dist_i(o, u)}{\max_{v \in sphere_i(o, k)} dist_i(o, v)}$$

$$II \quad \hat{c}(o)_{i,j} = \sum_{u \in sphere_i(o, k) \wedge c(u) = c_j} \frac{1}{d_i^{norm}(o, u)^2}$$

$$III \quad \forall j, 1 \leq j \leq |C| : cv_{i,j}(o) = \frac{\hat{c}(o)_{i,j}}{\sum_{k=1}^{|C|} \hat{c}(o)_{i,k}}$$

$$IV \quad cv_i(o) = (cv_{i,1}(o), \dots, cv_{i,|C|}(o))$$

354

Multirepräsentierte k NN-Klassifikation

Kombination der cv_i in jeder Repräsentation mit Gewichten:

$$Cl_{mr}(o) = \max_{j=1, \dots, |C|} \sum_{i=1}^m w_i \cdot cv_{i,j}(o)$$

Gewicht $w_{o,i}$ der Objekts o in Repräsentation i :

$$w_{o,i} = \begin{cases} 0 & , \text{ falls } r_i = "-" \\ \frac{1 + \sum_{j=1}^{|C|} (cv_{i,j}(o) * \log_{|C|} cv_{i,j}(o))}{\sum_{k=1}^m (1 + \sum_{j=1}^{|C|} (cv_{k,j}(o) * \log_{|C|} cv_{k,j}(o)))} & , \text{ sonst} \end{cases}$$

Idee: Je „reiner“ die Nachbarschaft eines Objekts o ist,

desto zuverlässiger ist die Aussage des k NN Klassifikators in Rep. R_i .

355

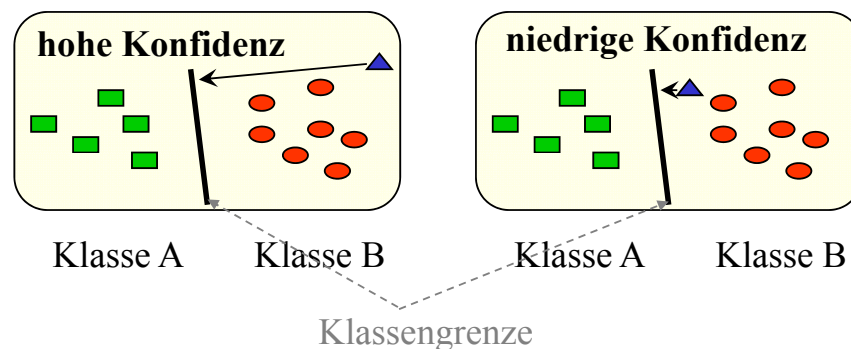
Konfidenzabschätzung mit Confidence-Ranges

Wie kann man unabhängig vom Typ des Klassifikators Aussagen über die Zuverlässigkeit einer Klassenvorhersage treffen?

Beobachtung: Die Unsicherheit nimmt an den Klassengrenzen zu. Objekte die weit von der Klassengrenze liegen, sind weit von anderen Klassen entfernt und damit verhältnismäßig sicher klassifiziert.

Idee: Bestimme inwieweit ein Objekt o verschoben werden kann, ohne dass sich die Klassenvorhersage ändert.

=> Je größer die Verschiebung desto stabiler die Klassenvorhersage.



356

Konfidenzabschätzung mit Confidence-Ranges

Klassifikator Kombination mit Confidence Ranges:

- Confidence Range $CRange(o)$ ist der Betrag der minimalen Verschiebung, die die Klasse von o verändern würde.

$$CRange(o) = \min \left\{ \|v\| \mid v \in F \wedge CL(o) \neq CL(o+v) \right\}$$

mit F als zugrunde liegenden Feature-Raum und $CL(o)$ als Klassifikator.

- **Interpretation:** Wie groß muss $CRange(o)$ sein um mit $x\%$ Konfidenz sagen zu können, dass o richtig klassifiziert wurde?

=> Lerne eine Funktion die $CRange(o)$ auf Konfidenz $CE_i(o)$ mappt.

$CE_i(o)$ wird als Confidence Estimate von o in R_i bezeichnet.

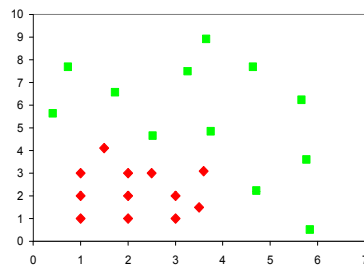
- Kombinierte Klassifikation: Wähle die Klasse, die in der Repräsentation R_i mit dem höchsten Wert für $CE_i(o)$ vorhergesagt wurde.

$$CL_{global}(o) = CL_{\arg \max_{0 \leq j \leq n} \{CE_j(o)\}}(o)$$

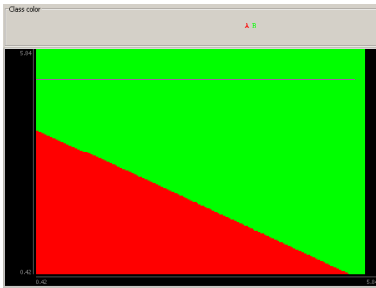
357

Bestimmung der Confidence-Range

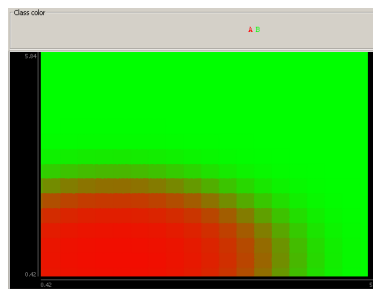
originale Datemenge



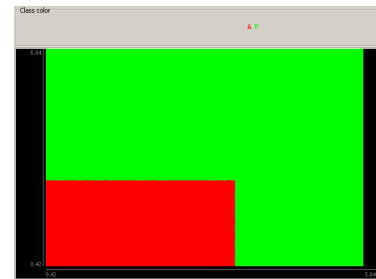
lineare SVM



naive Bayes

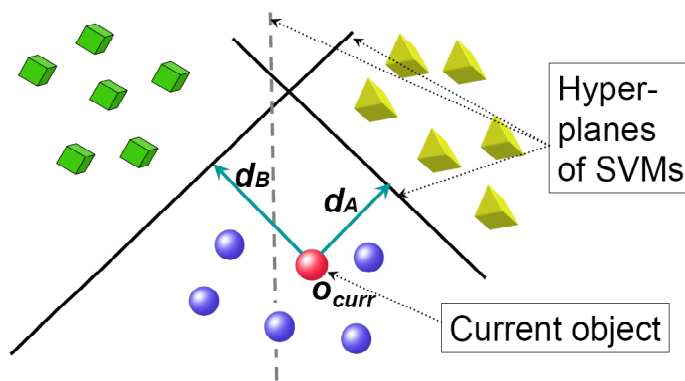


Entscheidungsbaum



Bestimmung der Confidence-Range für SVM

Support Vector Machine



Minimal

Class Set $C = \{ \text{cube}, \text{sphere}, \text{pyramid} \}$

d_A, d_B : distances between o_{curr} and borders of its class

Bestimmung der Confidence-Range für Bayes-Kl.

Bayes Classification

$$p(c_{pred}) \cdot p(\mathbf{x}|c_{pred}) = p(c_{pred}) \cdot p(\mathbf{x}|c_{other})$$
$$\Rightarrow p(c_{pred}) \cdot p(\mathbf{x}|c_{pred}) - p(c_{pred}) \cdot p(\mathbf{x}|c_{other}) = 0$$

$$\min_{x \in \mathbb{R}^d} d(o, x)$$

$$s.t. p(c_{pred}) \cdot p(\mathbf{x}|c_{pred}) - p(c_{pred}) \cdot p(\mathbf{x}|c_{other}) = 0$$

nicht lineares Optimierungsproblem,
Lösung z.B. mit Gradient Descent Verfahren

$$\min_{x \in \mathbb{R}^d} d(o, x)$$

$$s.t. (x - \mu_1)^T \times (\Sigma_1)^{-1} \times (x - \mu_1)$$

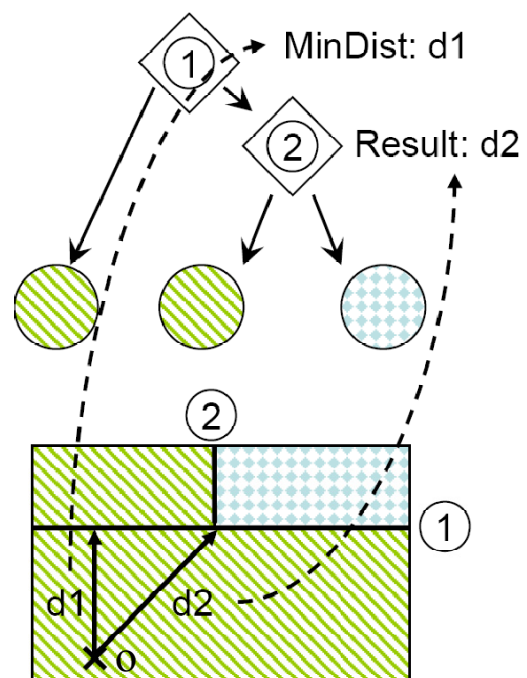
$$-(x - \mu_2)^T \times (\Sigma_2)^{-1} \times (x - \mu_2) - \ln \frac{p(c_1) \cdot \Sigma_2}{p(c_2) \cdot \Sigma_1} = 0$$

360

Bestimmung der Confidence-Range für Entscheidungsab.

Entscheidungsbaum

- normaler Durchlauf des Entscheidungsbaums um Objekte o zu klassifizieren
- Während des Durchlaufs werden die nicht verfolgten Unterbäume und Blätter in einer Priority-Queue(PQ) gespeichert
- die Einträge in PQ sind nach dem minimalen Abstand zu o sortiert.
- Nach finden des Blattknotens in dem o enthalten ist, wird PQ solange abgearbeitet bis Blatt mit anderer Klassenzuordnung auftaucht



361

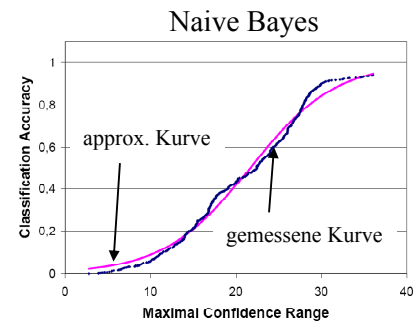
Von Confidence Ranges zu Confidence Estimates

Ziel: Ableiten eines Konfidenzwertes aus $\text{CRange}(o)$.

Beobachtung: Betrachte Vorhersagenauigkeit für alle Objekte mit $\text{CRange}(o) < x$.

Die beobachtete Kurve hat die Form einer Sigmoid-Funktion.

$$\text{sig}(x) = \frac{1}{1 + e^{-\alpha x + \beta}}$$



⇒ Lerne Sigmoid-Funktion, die die Genauigkeit für einen bestimmten Wert von $\text{CRange}(o)$ vorhersagt.

$$CE_i(o) = \frac{1}{1 + e^{-\alpha_i \text{CRange}_i(o) + \beta_i}} \quad \text{Confidence Estimate } CE_i(o)$$

362

Bestimmen von $CE_i(o)$

für jede Repräsentation i :

- bestimme für jedes Objekt o , $\text{CRange}(o)$ und ob die Klassenvorhersage richtig ist mit 3-fold Cross-Validation.

- Bilde eine Kurve aus folgende Werten: $x = \text{CRange}(o)$ (max. CRange)
 $y = |\text{R}(x)|/|\text{DB}(x)|$ (Accuracy)

$$\text{DB}(x) = \{ o \in \text{DB} \wedge \text{CRange}(o) < x \}$$

$$\text{R}(x) = \{ o \in \text{DB}(x) \wedge \text{CL}(o) = \text{K}(o) \}$$

- Approximiere Graphen mit Regression => Parameter für CE_i
(Verwendung der Methode von Levenberg und Marquadt)

363

Klassifikation mit Confidence Ranges

Geg.: CL_i und CE_i sind bereits trainiert für jede Repräsentation i .

Klassifikation nach folgenden Schema:

1. Bestimme $CL_i(p)$ und $CRange_i(o)$ für alle R_i .
2. Bestimme $CE_i(o)$ für alle $CRange_i(o)$.
3. Bestimme R_p bei dem $CE_p(o)$ am höchsten.
4. Klassifiziere o durch $CL_p(o)$

364

6.4 Co-Training

Multiple Repräsentationen können auch dazu verwendet werden eine Trainingsmenge zu erweitern.

Gegeben: 2 Repräsentationen für die sowohl gelabelte als auch nicht gelabelte Objekte vorhanden sind.

Idee:

Benutze Klassifikator um neue Trainingsobjekte aus ungelabelten Datenobjekten zu erzeugen.

Aber: Wieso braucht man dazu mehrere Repräsentationen ?

365

Generieren von Trainingsobjekte mit nur 1 Repräsentation

Versuch:

- Trainiere Klassifikator **CL** auf allen gelabelten Objekten
- klassifiziere k ungelabelte Objekte und füge sie in die Trainingsmenge ein.
- Trainiere nächsten Klassifikator auf der neuen Trainingsmenge

Problem:

- neue Daten werden mit dem Modell von **CL** gelabelt
- damit neue Trainingsobjekte CL verändern können, müssten sie aber Widersprüche zum bisherigen Modell enthalten

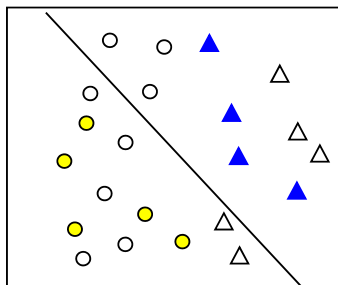
=> Generieren von Trainingsobjekten mit einer Repräsentation verstärkt nur die Schwächen des Klassifikators

366

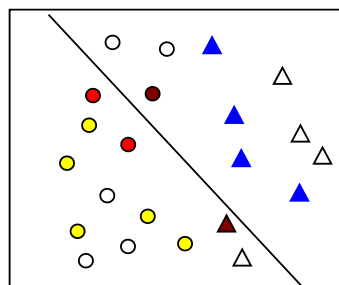
Generieren von Trainingsobjekte mit nur 1 Repräsentation

Beispiel:

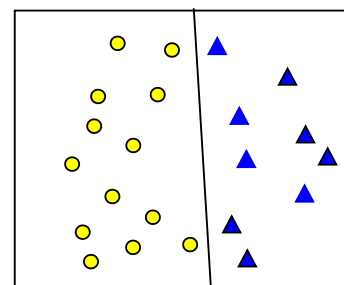
- blau = gelabelte Objekte Dreieck-Klasse
- gelb = gelabelte Objekte Kreis Klasse
- rot = relabelte Objekte mit CL_1



Training auf originalen Daten



Training mit relabelten Daten



optimale Lösung

Fazit:

- Die roten Objekte bestätigen nur die Annahmen des Klassifikators, können diese aber nicht verbessern.
- Zur Verbesserung wären von CL unabhängig Informationen notwendig.

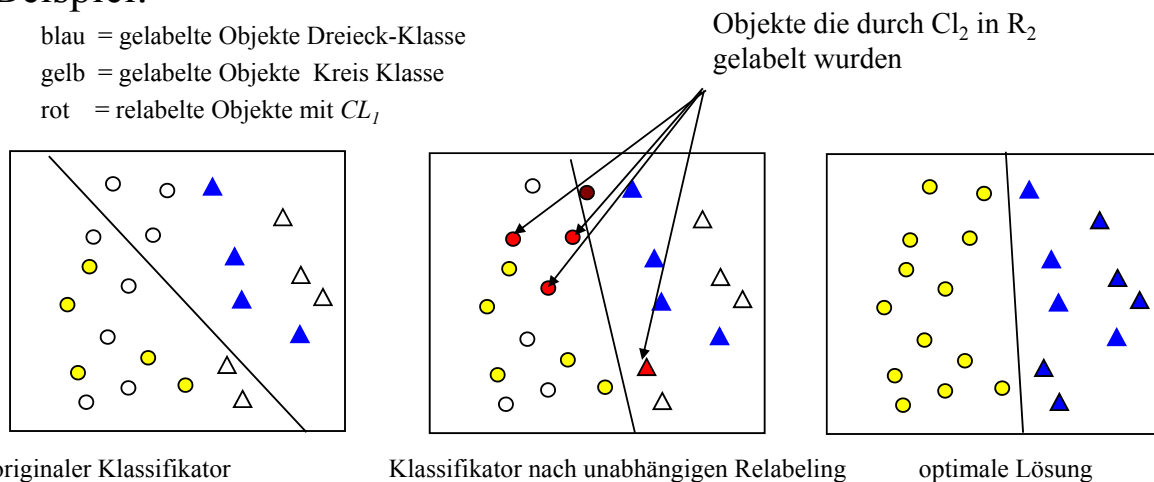
367

Co-Training

Idee: Klassifikatoren aus anderen Repräsentationen labeln Objekte, mit für diese Repräsentation neuen Informationen.

Beispiel:

- blau = gelabelte Objekte Dreieck-Klasse
- gelb = gelabelte Objekte Kreis Klasse
- rot = relabelte Objekte mit CL_1



=> Durch neue unabhängig gelabelte Objekte kann sich ein Klassifikator verbessern.

368

Der Co-Training Algorithmus

Gegeben: 2 Mengen aus multirepräsentierten Objekten

TR = gelabelte Trainingsmenge, U = Menge ungelabelter Objekte.

Co-Training Algorithmus

For k times do

For each R_i Do

Trainiere CL_i für Repräsentation i .

Ziehe Sample aus U .

generiere neue Label mit CL_i .

füge neu gelabelte Objekte zu TR hinzu

369

Ansatz ist von 2 Aspekten abhängig:

1. Jeder beteiligte Basisklassifikator muss für sich selbst ausreichend genau sein:
Bei schlechter Vorhersagequalität einzelner Klassifikatoren sind neue Label nicht zuverlässig genug.
2. Basisklassifikatoren müssen hinreichend unabhängig voneinander sein.
Sind sich die Klassifikatoren einig entsteht kein Nutzenpotential .

6.5 Clustering Multirepräsentierter Objekte

Anforderungen an Clustering-Algorithmen für Multirepräsentierte Objekte:

- Integration aller Informationsquellen.
- Eigenschaften in unterschiedlichen Repräsentationen müssen unterschiedlich behandelt werden.
- spezialisierte Techniken für unterschiedliche Arten von Repräsentationen sollten verwendet werden.
(Zugriffsmethoden, Indexstrukturen, Distanzmaße ...).
- Der Aufwand sollte möglichst nur linear mit jeder Repräsentation ansteigen.

Clustering Multirepräsentierter Objekte

Naive Lösungen zum Clustering Multirepräsentierter Objekte

Clustere nur die vielversprechendste Repräsentation:

- + bereits bekannte Clusteringalgorithmen sind verwendbar
- Verwendet nicht alle Repräsentationen.
- Welche Repräsentation ist die beste.

Verwende zusammengesetzten Featureraum:

- + verwendet alle Repräsentationen.
- Die Anzahl der Feature in jeder Repräsentation ergibt eine ungewollte Gewichtung (z.B. Text-Vektoren und Farbhistogramme)
- Keine Verwendung spezialisierter Distanzmaße (z.B. Edit-Distanz, Cosinus-Distanz, Jaccard...)
- Unterstützung für Ähnlichkeitsanfragen schwierig.
Keine spezialisierten Index und Speicherstrukturen.

372

Dichtebasiertes Multirepräsentiertes Clustering

Anforderung: Kombiniere Repräsentationen

- so früh wie möglich (während des Clusterings)
 - nur ein Durchlauf durch alle Repräsentationen
- so spät wie möglich (nach den ε -Bereichs-Anfragen)
 - ermöglicht Verwendung spezieller Indexstrukturen und Distanzmaße



Idee: Definiere die Kernobjekt Eigenschaft neu

373

Vereinigungs-Methode

Idee: Ein Objekt ist in einem dichten Bereich, wenn k Nachbarn in allen Repräsentationen in der ε -Umgebung liegen.

Geeignet für : "sparse" Daten mit viel Rauschen.

Vereinigungs-Kernobjekt:

Sei $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_m \in \mathcal{R}^+, \text{MinPts} \in \mathbb{N}, o \in O$ ist ein **Vereinigungs-Kernobjekt**, falls

$$\left| \bigcup_{R_i(o) \in O} N_{\varepsilon_i}^{R_i}(o) \right| \geq \text{MinPts}, \text{ wobei } N_{\varepsilon_i}^{R_i}(o) \text{ die lokale } \varepsilon\text{-Nachbarschaft in Repr. } i \text{ ist.}$$

Direkte Vereinigungserreichbarkeit:

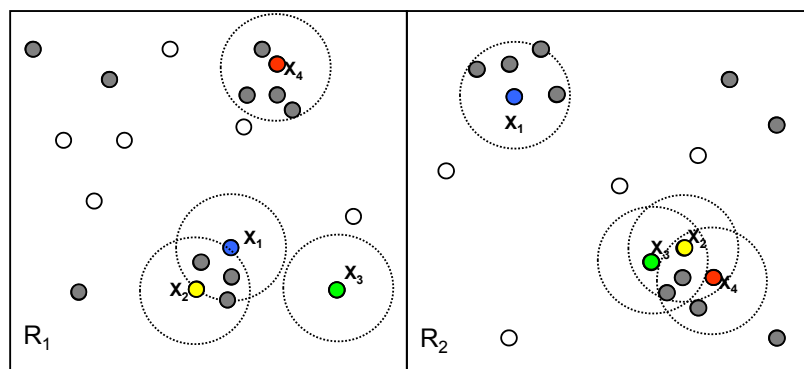
Objekt $p \in O$ ist **direkt vereinigungserreichbar** von $q \in O$ bzgl. $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_m$ und MinPts , falls q ein Vereinigungs-Kernobjekt in O ist und es gilt:

$$\exists i \in \{1, \dots, m\} : R_i(p) \in N_{\varepsilon_i}^{R_i}(q)$$

374

Vereinigungs-Methode

Clusterexpansion bei der Vereinigungsmethode



MinPts = 3

375

Schnitt-Methode

Idee: Ein Objekt ist in einem dichten Bereich, falls es k Objekte in den ε -Nachbarschaften aller Repräsentationen gibt.

Geeignet für: dichte Repräsentationen and unzuverlässige lokale Feature-Vektoren.

Schnitt-Kernobjekt:

Sei $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_m \in \mathbb{R}^+, \text{MinPts} \in \mathbb{N}$. $o \in O$ ist ein **Schnitt-Kernobjekt**, falls

$$\left| \bigcap_{R_i(o) \in O} N_{\varepsilon_i}^{R_i}(o) \right| \geq \text{MinPts} \quad , \text{ wobei } N_{\varepsilon_i}^{R_i}(o) \text{ die lokale } \varepsilon\text{-Nachbarschaft in Repr. } i \text{ ist.}$$

Direkt schnitterreichbar:

Objekt $p \in O$ ist **direkt schnitterreichbar** von $q \in O$ bzgl.

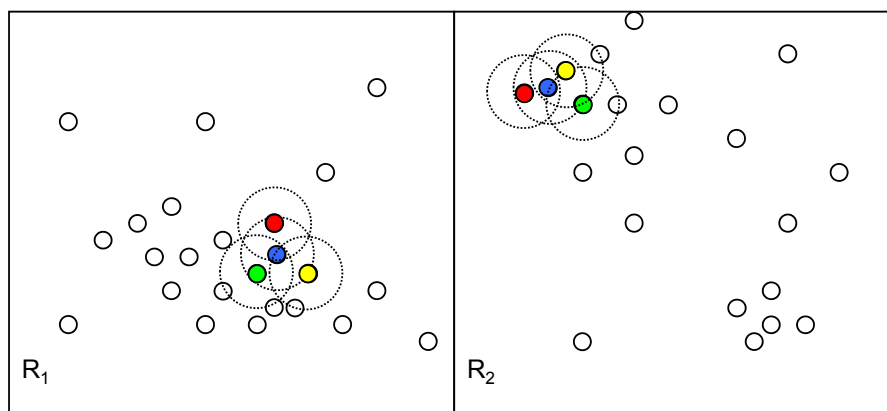
$\varepsilon_1, \varepsilon_2, \dots, \varepsilon_m$ und MinPts , falls q ein Schnitt-Kernobjekt in O ist und es gilt:

$$\forall i \in \{1, \dots, m\}: R_i(p) \in N_{\varepsilon_i}^{R_i}(q)$$

376

Schnitt-Methode

Clusterexpansion mit Schnitt-Methode



MinPts = 3

377

Qualitätsmaß für MR-Clustering

Definition: Sei O eine Datenmenge und $C = \{C_i | C_i \subset O\}$ ein Clustering. Weiterhin sei $K = \{K_i | K_i \subset O\}$ ein Referenz-Clustering (= tatsächliches Clustering).

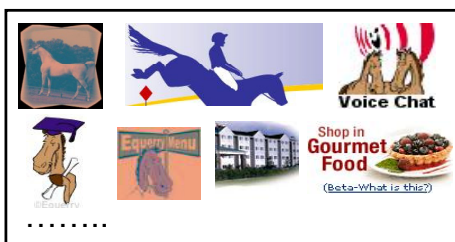
$$quality_K(C) = \sum_{C_i \in C} \frac{|C_i|}{|O|} \cdot (1 + entropy_K(C_i))$$

- Vergleicht Clustering mit Referenz-Clustering
- Bevorzugt Clusterings mit wenig Rauschen.
- Jeder Cluster sollte möglichst nur zu einem Cluster im Referenz-Clustering gehören. => Niedrige Entropie

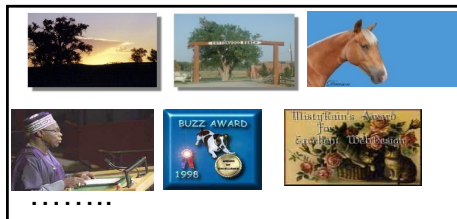
378

Beispiel-Ergebnisse auf Bilddaten

Cluster in den einzelnen Repr.



Beispiele für Bilder im Cluster IC 5
(nur Farbhistogramme)



Beispiele für Bilder im Cluster IC 5
(nur Segmentbäume)



Cluster IC5 der auf beiden Repräsentationen mit der Intersectionmethode gebildet wurde.

379

Distanzen in allen Repräsentationen müssen vergleichbar sein

=> Normalisierung der Distanzen

2 mögliche Vorteile durch mehrere Repräsentationen

- Verbesserung der Clusterqualität (Precision)
=> 2 Objekte sind dann ähnlich,
wenn sie in allen Repräsentationen ähnlich sind
- Vergrößere die Clusterabdeckung (Recall)
=> es reicht wenn ein Objekt in einer Repräsentation ähnlich ist

Integration dieser Idee in OPTICS durch Redefinition der
Kerndistanz und der Erreichbarkeitsdistanz

380

Intersection Clustering (Schnittmethode)

Idee: Ein Objekt ist in einem dichten Raum, wenn es k Objekte gibt, die bzgl. aller Repräsentationen ähnlich sind.

Intersection k -nächste-Nachbar-Distanz: $k \in \mathcal{N}$

$$NN - Dist_k^\cap = \max \left\{ \max_{i=1..m} \{d_i(o, q)\} \mid q \in NN_k^\cap(o) \right\}$$

Intersection Kerndistanz: $k \in \mathcal{N}$, $\varepsilon \in \mathcal{R}$

$$CORE_{\varepsilon, k}^\cap = \begin{cases} NN - Dist_k^\cap(o) & \text{if } |N_\varepsilon^\cap(o)| \geq k \\ \infty & \text{else} \end{cases}$$

Intersection Erreichbarkeitsdistanz:

$$REACH_{\varepsilon, k}^\cap(p, o) = \max \left\{ CORE_{\varepsilon, k}^\cap(p), \max_{i=1..m} \{d_i(o, p)\} \right\}$$

381

Union Clustering (Vereinigungsmethode)

Idee: Ein Objekt ist in einer dichten Region, wenn es k Objekte gibt, die in einer beliebigen Repräsentation ähnlich sind.

Union-k-Nächste-Nachbar Distanz: $k \in \mathcal{N}$

$$NN - Dist_k^\cup = \max \{ \min_{i=1..m} \{ d_i(o, q) \} \mid q \in NN_k^\cup(o) \}$$

Union-Kerndistanz: $k \in \mathcal{N}$, $\varepsilon \in \mathcal{R}$

$$CORE_{\varepsilon, k}^\cup = \begin{cases} NN - Dist_k^\cup(o) & \text{if } |N_\varepsilon^\cup(o)| \geq k \\ \infty & \text{else} \end{cases}$$

Union-Erreichbarkeitsdistanz:

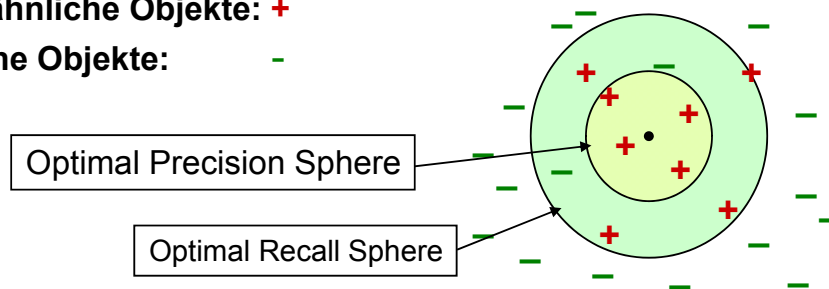
$$REACH_{\varepsilon, k}^\cup(p, o) = \max \{ CORE_{\varepsilon, k}^\cup(p), \min_{i=1..m} \{ d_i(o, p) \} \}$$

382

Bedeutung der Repräsentationen

wirklich ähnliche Objekte: +

unähnliche Objekte: -



Möglichen Interpretationen der ε -Nachbarschaft:

hohe Precision- und Recall-Werte

=> 1 Rep. lässt gutes Clustering zu

niedrige Precision- und Recall-Werte

=> alle Rep. lassen kein gutes Clustering zu

hohe Precision- aber niedrige Recall-Werte

=> benutze Vereinigungs-Methode

niedrige Precision- aber hohe Recall-Werte

=> verwende Schnitt-Methode

383

Kombinationsbäume

Klassifiziere Repräsentationen nach:

- Precision-Räume: alle Objekt weisen gute Precision-Sphere auf.
- Recall-Räume: Alle Objekte weisen gute Recall-Sphere auf.

Kombination der Precision-Spaces mit der Vereinigungsmethode

=> Ergebnis: besserer Recall-Space.

Kombination der Recall-Spaces mit der Schnitt-Methode

=> Ergebnis: besserer Precision-Space.

Für große Mengen an Repräsentationen kombiniere beide Methoden mit Kombinationsbäumen.

384

Kombinationsbäume

Gegeben Repräsentationen $R = \{ R_1, \dots, R_m \}$ ein Kombinationsbaum KB ist :

Baum mit beliebigen Grad $g > 0$

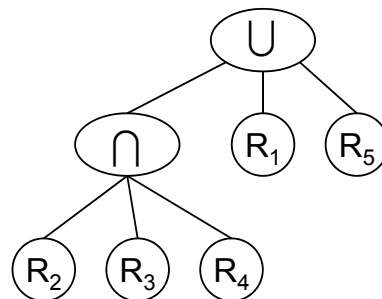
Blätter entsprechen den Repräsentationen

Innere Knoten entsprechen den Operatoren $\theta \in \{ \cup, \cap \}$

Semantik:

\cup verbessert Recall

\cap verbessert Precision



385

Verwendung von Kombinationsbäumen

Kombinationsbäume implizieren folgendes rekursives Distanzmaß:

$$d^n(o, p) = \begin{cases} \min_{c \in n.children} \{d^c(o, p)\} & \text{if } n.label = \cup \\ \max_{c \in n.children} \{d^c(o, p)\} & \text{if } n.label = \cap \\ d_i(o, p) & \text{if } n.label = R_i \end{cases}$$

für den vollständigen Baum: $d_{CT}(o, p) = d^{CT.root}(o, p)$

Bemerkung: Für die sinnvolle Interpretation der Distanzen ist eine vorherige Normalisierung notwendig.

386

Multirepräsentiertes Partitionierendes Clustering

[Bickel, Scheffer 2004]

Auch partitionierende Clustering-Verfahren wie, k-Means, k-medoid oder EM, können auf multiple Repräsentationen angepasst werden.

Grundannahme: Cluster entstehen durch einen statistischen Prozeß je Repräsentation.

- ⇒ jedes Objekt gehört zu genau einem Cluster in jeder Repräsentation
- ⇒ multirepräsentierte Cluster bestehen 1 Cluster in jeder Repräsentation der exakt die gleichen Objekte enthält.

Idee: Korrektes Clustering in einer Repräsentation impliziert ein korrektes multirepräsentiertes Clustering.

Aber: Partitionierendes Clustering Algorithmen terminieren in lokalen Minima

⇒ benutze mehrere Repräsentation, um nicht in lokalen Minima zu terminieren

387

Multirepräsentiertes Partitionierendes Clustering

Zur Anwendung von partitionierenden Clustering Algorithmen wird eine Zielfunktion und ein Modellbildungsschritt benötigt:

- Zielfunktion:
$$MRTD^2 = \sum_{R_i \in R} \sum_{C_{ik} \in C_i} \sum_{x \in C_{ik}} d(x, c_{ik})^2$$

$R = \{R_1, \dots, R_n\}$ Repräsentationen,

C_{ik} : k -ter Cluster in R_i , c_{ik} : Centroid des k -ten Cluster in R_i .

- Modelbildung: Gruppierung der Objekte bzgl. R_j
Berechnung neuer Centroide c_{ik} in R_i bzgl. Aufteilung aus R_j

- Consensus-Clustering bei Nicht-Terminieren:

MR-partitionierendes Clustering garantiert kein globales Maximum

=> Algorithmus kann $MRTD^2$ nicht mehr verbessern und es gibt kein einheitliches Clustering

=> Bilde globales Cluster-Modell aus allen Punkten, die in allen Repräsentationen richtig eingeordnet wurden.

$$c_k = \frac{\sum_{x \in \bigcap_{R_j \in R} C_{jk}} x}{\left| \bigcap_{R_j \in R} C_{jk} \right|}$$

388

Multirepräsentiertes k-Means

Gegeben: Suche k Cluster in DB aus multirep. Objekten aus n Repräsentation R_i

Algorithmus: *MR-k-Means*

aktR:=1

$MRTD^2_{old} = \infty$

Initialisiere Clustering in R_{aktR}

Berechne Objektaufteilung pro Cluster

Bestimme $MRTD^2_{neu}$

Wiederhole bis $(MRTD^2_{old} - MRTD^2_{neu}) \leq \varepsilon$

oldR:=aktR

aktR:=(aktR+1)MOD n

Bestimme Centroide in R_{aktR} mit der Aufteilung aus R_{oldR} ;

$MRTD^2_{old} := MRTD^2_{neu}$;

uosdate($MRTD^2_{neu}$)

Ende der Wiederholung

Berechne Consensus Clustering

389

Multirepräsentiertes Partitionierendes Clustering

Bemerkungen:

- Ansatz ist auch auf EM und k-Medoid Verfahren anwendbar.
- Algorithmus terminiert nicht zwangsläufig
- Verbesserung der lokalen Clusterings durch Verwendung der anderen Repräsentationen.
- Ansatz nimmt keine Wertung der Repräsentationen vor.
=> alle Repräsentationen beeinflussen das Ergebnis gleich stark.

390

Literatur

- Kriegel H.-P., Kunath P., Pryakhin A., Schubert M.: ***MUSE: Mult-Represented Similarity Estimates***, in proc. 24th International Conference on Data Engineering (ICDE 2008), Cancun, México, 2008
- Abfalg J., Kriegel H.-P., Pryakhin A., Schubert M.: ***Multi-Represented Classification based on Confidence Estimation*** in proc. 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2007), Nanjing, China
- Achtert E., Kriegel H.-P., Pryakhin A., Schubert M.: ***Clustering Multi-Represented Objects Using Combination Trees*** in proc. 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2006), Singapore
- Kriegel H.-P., Pryakhin A., Schubert M.: ***Multi-represented kNN-Classification for Large Class Sets*** 10th International Conference on Database Systems for Advanced Applications (DASFAA 2005), Beijing, China.
- Kailing K., Kriegel H.-P., Pryakhin A., Schubert M.: ***Clustering Multi-Represented Objects with Noise*** Proc. 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'04), Sydney, Australia, 2004.
- Bickel S., Scheffer T.: ***Multi-View Clustering***, 4th IEEE International Conference on Data Mining (ICDM 2004).
- Blum. A, Mitchell T.: ***Combining Labeled and Unlabeled Data with Co-Training***, Workshop on Computational Learning Theory (COLT 98)

391