

Skript zur Vorlesung
Knowledge Discovery in Databases II
im Sommersemester 2008

Kapitel 2: Feature-Selektion und Feature-Reduktion

Skript © 2008 Matthias Schubert
(Folien 21,22,60-63 aus KDD-Skript 2005)

<http://www.dbs.ifi.lmu.de/Lehre/KDD2>

2. Featurereduktion und Featureselektion

Inhalt dieses Kapitels

2.1 Einführung

Motivation, „*Curse of Dimensionality*“

2.2 Feature-Selektion

Methoden zur Auswahl einer geeigneten Unterräumen

2.3 Feature-Reduktion

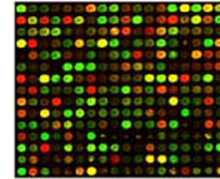
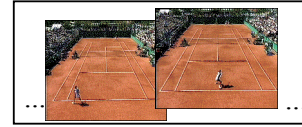
Generierung neuer Featureräume

2.1 Einführung

„Reale Daten sind meist sehr hochdimensional“

Beispiele:

- Fernseh Bildern
 - Fernsehbilder werden in Farbhistogramme zerlegt
 - Je nach Farbauflösung: 100 – 1.000 dimensionale Feature-Vektoren pro Bild
- Biologie: Microarray Daten
 - Ein Feature entspricht z.B. einem Gen im menschlichen Körper
 - Je nach Versuchsaufbau: 20.000 dimensionale Vektoren
- Biologie: Metabolomdaten
 - Ein Feature entspricht der Konzentration eines Stoffwechselprodukts im Blut
 - Je nach Messgenauigkeit: 50 – 2000 dimensionale Feature-Vektoren



24

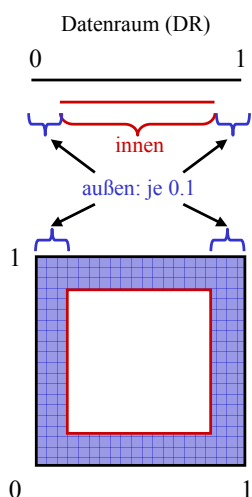
Grundprobleme

„Fluch der Dimensionalität“ (*Curse of Dimensionality*)

- Distanz zum nächsten Nachbarn unterscheidet sich im Hochdimensionalen kaum von der Distanz zu einem beliebigen Nachbarn

$$\frac{\text{nächsteNachbarnDist}}{\text{entferntesteNachbarnDist}} \approx 1$$

- Die Wahrscheinlichkeit, dass Datenobjekte am Rand des Datenraumes liegen, steigt mit der Anzahl der Dimensionen exponentiell



$$\begin{aligned} 1D: \quad \mathcal{P}_{DR} &= 1^1 = 1 \\ \mathcal{P}_{\text{innen}} &= 0.8^1 = 0.8 \\ \mathcal{P}_{\text{außen}} &= 1 - 0.8 = 0.2 \end{aligned}$$

$$\begin{aligned} 2D: \quad \mathcal{P}_{DR} &= 1^2 = 1 \\ \mathcal{P}_{\text{innen}} &= 0.8^2 = 0.64 \\ \mathcal{P}_{\text{außen}} &= 1 - 0.64 = 0.36 \end{aligned}$$

$$\begin{aligned} 3D: \quad \mathcal{P}_{DR} &= 1^3 = 1 \\ \mathcal{P}_{\text{innen}} &= 0.8^3 = 0.512 \\ \mathcal{P}_{\text{außen}} &= 1 - 0.512 = 0.488 \end{aligned}$$

$$\begin{aligned} 10D: \\ \mathcal{P}_{\text{außen}} &= 0.893 \end{aligned}$$

25

Grundprobleme

andere Effekte des *Curse of Dimensionality*:

- Die Werte in jeder Dimension sind verrauscht.
D.h. die Werte schwanken durch Störeinflüsse, die mit dem Objekt an sich nichts zu tun haben.
- bei zunehmender Dimensionalität wird die Summe der Störeinflüsse so groß, dass die beobachteten Unterschiede zwischen 2 Objekten von den Störeinflüssen dominiert werden.
=> Summe der Unterschiede hängt von der Ausprägung der Störeinflüsse ab und nicht von den Objekteigenschaften.
=> Abstand zwischen Objekt gleicht sich immer weiter an, da Störeinflüsse gleichverteilt über alle Objekte.

26

Grundprobleme

- Patterns und Modelle auf hochdimensionalen Daten sind oft schwer interpretierbar.
=> Lange Entscheidungsregeln
- Effizienz bei hochdimensionalen Daten häufig problematisch.
=> Indexstrukturen degenerieren auf hochdimensionalen Daten
=> Distanzberechnungen werden i.d.R. teurer
- Muster treten nur in Teilräumen auf, aber nicht im gesamten Feature Raum
- Cliques von korrelierten Features dominieren das Objektverhalten.

27

2.2. Feature-Selektion

Idee: Bei sehr vielen Features sind nicht alle notwendig, um die Daten zu beschreiben:

- Features sind nicht aussagekräftig für ein Problem
- Informationsgehalt stark korrelierter Features ist fast identisch

Die Einschränkung auf einen Teilraum des gesamten Datenraums kann Verfahren effizienter und effektiver machen.

Lösung:

Streiche alle überflüssigen Dimensionen aus dem Featureraum.

28

Feature-Selektion

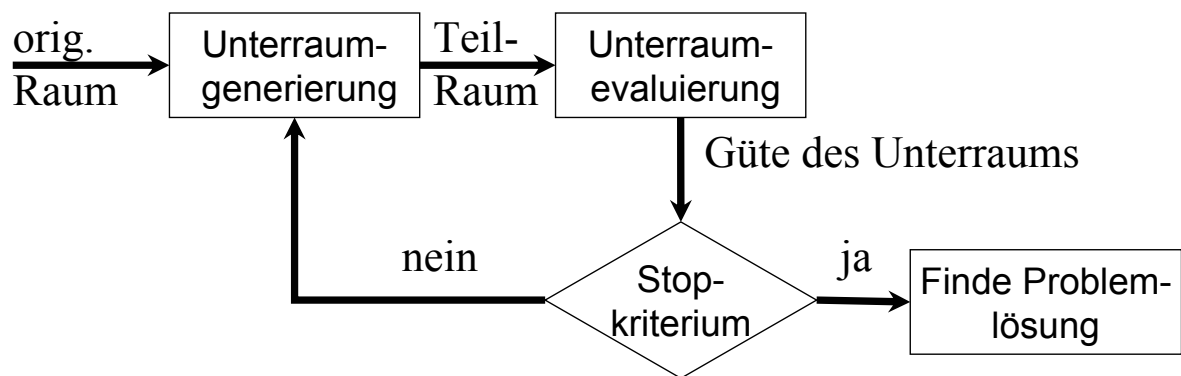
Gegeben: Vektorraum $F = D_1 \times \dots \times D_n$ mit $D = \{D_1, \dots, D_n\}$.

Gesucht: Minimaler Unterraum M über $D' \subseteq D$, der für ein gegebenes Data Mining Problem eine optimale Lösung erlaubt.

- Minimalität erhöht Effizienz und verringert den Curse-of-Dimensionality
- Optimale Lösung ist ein sehr breiter Begriff, da Featurereduktion Vorverarbeitung zu sehr unterschiedlichen Problemstellungen sein kann.
- Problem ist sehr komplex, da es 2^n mögliche Unterräume gibt.
=> vollständige Suche wäre nur auf kleinen Dimensionalitäten möglich, bei denen Featureselektion nicht notwendig ist

29

Genereller Ablauf der Feature-Selektion



- obiges Schema deckt einen Großteil der Algorithmen ab (weitere Methoden denkbar: z.B. Clustering der Features im Objektraum)
- Unterscheidung anhand Unterraumgenerierung, Unterraumevaluierung und Stopkriterium.

30

Überblick über Teillösungen

1. Welcher der 2^n Unterräume müssen oder sollen bei der Suche untersucht werden? (Unterraumgenerierung und Suche)
 - Greedy-Ansätze
 - Heuristische Ansätze
 - optimale Ansätze mit Monotonie
 - Suchrichtung (Hinzufügen oder Löschen von Dimensionen)
2. Wann erlaubt ein Unterraum eine optimale Problemlösung? (Unterraumevaluierung)
 - monotone oder nicht monotone Kriterien
 - problemspezifische oder unspezifische Kriterien
 - supervised oder unsupervised Kriterien

31

Überblick über Teillösungen

3. Wann kann der Algorithmus aufhören ? (Stop-Kriterium)

- optimales Ergebnis wurde gefunden
- alle Unterräume wurden untersucht.
- vom Benutzer wird höchste Anzahl an Iterationen vorgegeben
- Güte erreicht einen Mindestwert
- eine angegebene Anzahl an Features wurde selektiert

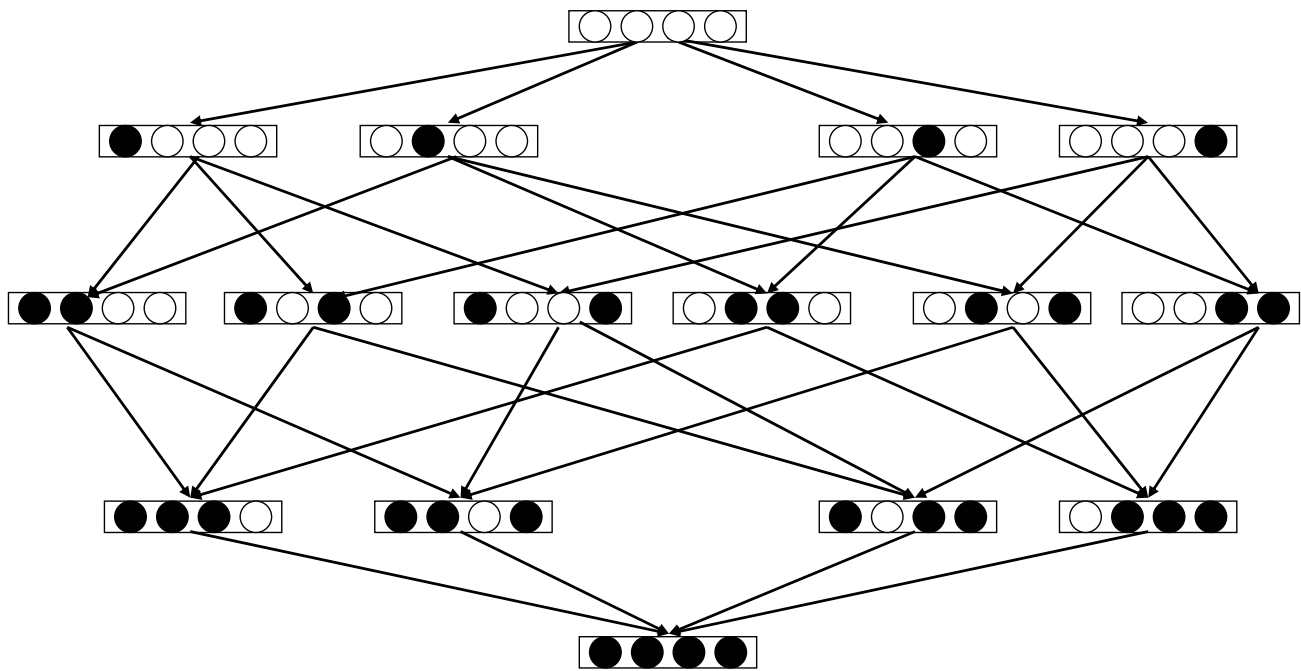
Prinzipiell sind verschieden Lösungen der 3 Teilaufgaben miteinander kombinierbar.

32

Unterraumgenerierung und Suche

- *Backward-Elimination*: Beginne mit dem gesamten Featurespace F und entferne überflüssige Dimensionen.
- *Forward-Selection*: Untersuche einzelne Dimensionen $D^i \in \{D_1, \dots, D_n\}$. und bilde neue Unterräume durch Kombination.
- *Greedy*: Kombiniere immer die besten Unterräume oder lösche immer die schlechteste Dimension.
- *Random*: Bilde zufällige Unterräume und evaluiere diese.
- *Suche*: Untersuche alle möglichen Unterräume, die durch Löschen oder Kombination bereits untersuchter Unterräume gebildet werden können.
- *Branch and Bound-Verfahren*: Schließe Lösungen, die nicht mehr maximale Qualität erreichen können, aus.

33



Qualitätskriterien

- Modellabhängige Beurteilung (Wrapper-Methoden):
Direkte Anwendung des Data Mining Algorithmus und Beurteilung des Ergebnisses.
 - Klassifikation: Trainiere Klassifikator auf Unterraum und beurteile Genauigkeit
 - Clustering: Clustere Unterraum und beurteile Clusterqualität (Silhouetten-Koeffizient, Max. Likelihood, ..)
- Modellunabhängige Beurteilung (Filter-Methoden):
 - Supervised und Unsupervised Trainingdaten
 - Diskrete oder Reelle Domänen
 - Beurteilung einer Dimension oder eines ganzen Teilraums
- Monotone Qualitätsmaße:
erlaubt die Verwendung von Branch and Bound Algorithmen

Kriterien zur Evaluation von Teilräumen

Zentrale Frage: *Wie Aussagegleich sind 2 Repräsentationen derselben Datenmenge ?*

Repräsentationen können hier einzelne Features, Unterräume oder auch Klassenlabels sein.

Einteilung der Kriterien:

1. Vorwissen

- mit Klassen gelabelt (Supervised Feature Selection)
- ohne Einteilung in Klassen (Unsupervised Feature Selection)

2. Art der gemessenen Information

1. Statistische und Informationstheoretische Ansätze
2. Distanzbasierter Ansatz
3. Konsistenzbasierter Ansatz

36

Einteilung nach Vorwissen

Supervised Feature-Selection:

Bewertung der Unterräume nach Korrelation zu den Klassen und Trennung der Klassen.

- Irrelevante Features sind nicht mit Klasse korreliert
- Redundante Features werden über Minimalität ausgeschlossen

Unsupervised Feature-Selektion:

Problem: Relevanz der Features kann nicht beurteilt werden, da kein Wissen über Ziel und Zweck des Datenraums

- ⇒ Elimination von redundanten Features
- ⇒ Beurteilung nach der Trennung potentieller Cluster (Wrapper-Ansatz)

**Maße zur Bestimmung der Korrelation sind notwendig für beide Ansätze !
(Korrelation zwischen Klassen und Features oder zwischen Features)**

37

1. Statistische und informationstheoretische Maße

- Basieren auf Verteilungen bzgl. der Klassen und der Features/Unterräume
Achtung bei reellen Features ist Schätzung einer Zufallsvariable nicht direkt durchführbar. Daher:
 - Split zur Umwandlung von reellen Features in diskrete Features
 - Annahme über Verteilungsfunktion (z.B. Normalverteilung,..)
- Wie stark korrelieren die Verteilungen von Klassen und Features/Unterräumen?
- Wie gut kann Klassenlabel im Unterraum vorhergesagt werden?
- Wie stark unterscheidet sich Aufteilung bzgl. dieses Features/Unterraums von zufälliger Aufteilung ?

38

1. Statistische Verfahren

Idee: Bewerte wie gut jede Dimension die Klassen „unterscheidet“.

Kriterien: Information Gain (vgl. Entscheidungsbäume)

Zerlege Trainingsmenge anhand Feature/Unterraum in Teilmengen (Unterteilung: nach Werten oder Splitkriterien).

Die *Entropie* für eine Menge T von Trainingsobjekten ist definiert als

$$\text{entropie}(T) = -\sum_{i=1}^k p_i \cdot \log p_i \quad (p_i \text{ steht für Häufigkeit der Klasse } i \text{ in } T)$$

$$\text{entropie}(T) = 0, \text{ falls } p_i = 1 \text{ für ein } i$$

$$\text{entropie}(T) = 1 \text{ für } k = 2 \text{ Klassen mit } p_i = 1/2$$

$$\text{informationsgewinn}(T, a) = \text{entropie}(T) - \sum_{j=1}^m \frac{|T(a)_j|}{|T|} \cdot \text{entropie}(T(a)_j)$$

mit $T(a)_j$ Teilmenge von T , für die Attribut a den Wert j -ten Wert annimmt.

Bei reellwertigen Attributen muss ein Split gefunden werden.

39

1. Statistische Verfahren

χ^2 -Statistik

Bewertet Unabhängigkeit einer Dimension von einer Klasse.

(Aufteilung der Daten anhand Splitwert s oder anhand diskreten Attribut werten)

$A = |\{o \mid x \leq s \wedge \text{Class}(o) = C_j\}|$ Objekte in C_j mit Wert $x \leq$ Splitwert

$B = \left| \bigcup_{l \neq j} \{o \mid x \leq s \wedge \text{Class}(o) = C_l\} \right|$ Objekte anderer Klassen mit $x \leq$ Splitwert.

$C = |\{o \mid x > s \wedge \text{Class}(o) = C_j\}|$ Objekte in C_j mit $x >$ Splitwert.

$D = \left| \bigcup_{l \neq j} \{o \mid x > s \wedge \text{Class}(o) = C_l\} \right|$ Objekte anderer Klassen, mit $x >$ Splitwert

χ^2 -Statistik ist definiert durch: $\chi^2(t, C_j) = \frac{|DB|(AD-CB)^2}{(A+C)(B+D)(A+B)(C+D)}$

Je höher Maximum oder Durchschnitt über alle Klassen desto besser das Feature a :

$$\chi_{\max}^2(a) = \max_{i=1}^m \{\chi^2(a, C_i)\} \quad \text{oder} \quad \chi_{\text{avg}}^2(a) = \sum_{i=1}^m \Pr(C_i) \chi^2(a, C_i)$$

40

1. Statistische Verfahren

Mutual Information (MI)

Maß für gegenseitige Abhängigkeit zweier Zufallsvariablen.

Hier: Vergleich der Abhängigkeit von der allgemeinen

Klassenverteilung mit Verteilung in Dimension/Unterraum.

1. diskreter Fall.

$$I(X, Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

2. Kontinuierlicher Fall

$$I(X, Y) = \int_Y \int_X p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dx dy$$

41

2. Distanzbasierte Kriterien

Idee: Unterraum ist gut, wenn Abstand von Objekten innerhalb einer Klasse durchschnittlich kleiner ist, als zwischen Objekten unterschiedlicher Klassen.

Qualitätsmaß:

Für alle Objekte $o \in DB$ berechne den nächsten Nachbarn in Klasse $C = \text{Class}(o)$ $NN_C(o)$ und den kleinsten nächsten Nachbarn $NN_{K \neq C}(o)$ in einer der anderen Klassen.

Güte des Unterraums U :

$$Q(U) = \frac{1}{|DB|} \cdot \sum_{o \in DB} \frac{NN_{K \neq C}^U(o)}{NN_C^U(o)}$$

42

3. Konsistenzbasierte Kriterien

Idee: Existieren im Unterraum U identische Vektoren u, v mit $v_i = u_i$ $1 \leq i \leq d$ aber unterschiedlichen Klassen labels $C(u) \neq C(v)$.

=> Unterraum ist inkonsistent

Maß für Konsistenz von U :

$X_U(A)$: Anzahl aller zu A identischen Vektoren

$IC_U(A)$: Inkonsistenz bzgl A in U

$$IC_U(A) = X_u(A) - \max_k X_u^k(A)$$

$$\text{Für ganz } U: IC(U) = \frac{\sum_{A \in S} IC_U(A)}{|S|}$$

Monotonie: $U_1 \subset U_2 \Rightarrow IC(U_1) \geq IC(U_2)$

43

3. Konsistenzbasierte Kriterien (2)

Vorteile:

- kann auf Unterräume angewendet werden und nicht nur auf Dimensionen
- Monotonie ermöglicht die effiziente Suche nach optimalen Unterräumen mit Branch and Bound

Nachteil:

- Anwendung beschränkt auf nominale Attribute (keine ordinalen oder reell-wertigen Attribute)

44

Beispiel 1: Greedy-Ansatz mit Information Gain

Gegeben: Klassifikationsproblem über F .

Ziel: Selektiere k Dimensionen

- Berechne IG für jede Dimension $D' \in \{D_1, \dots, D_n\}$ (bei reell-wertigen Attributen müssen alle möglichen Splitpunkte betrachtet werden)
- Sortiere Dimensionen $\{D_1, \dots, D_n\}$ nach IG
- Wähle die k besten Dimensionen

Nachteil:

- Dimensionen werden einzeln betrachtet: Klasse und Dimensionswert müssen direkt korreliert sein.
- Korrelierte Dimensionen: Auswahl von bedeutungsgleichen Dimensionen falls diese am stärksten mit der Klasse korreliert

45

Beispiel 2: Random-Ansatz mit Modellabhängiger Beurteilung

Gegeben: Klassifikationsproblem über F .

Ziel: Selektiere k Dimensionen

- Berechne die Klassifikationsgenauigkeit mit Überkreuzvalidierung für n zufällige Teilräume der Dimensionalität k .
- Wähle den Unterraum, in dem die beste Klassifikationsgüte gefunden wurde.

Nachteil:

- kann je nach Anzahl der zu testenden Unterräume sehr lange dauern.
- bester Unterraum mit k Dimensionen muss nicht im Sample enthalten sein

46

Beispiel 3: Branch and Bound mit Inkonsistenzkriterium

Gegeben: Klassifikationsproblem über F .

Ziel: Selektiere k Dimensionen

Backward-Elimination mit Branch and Bound:

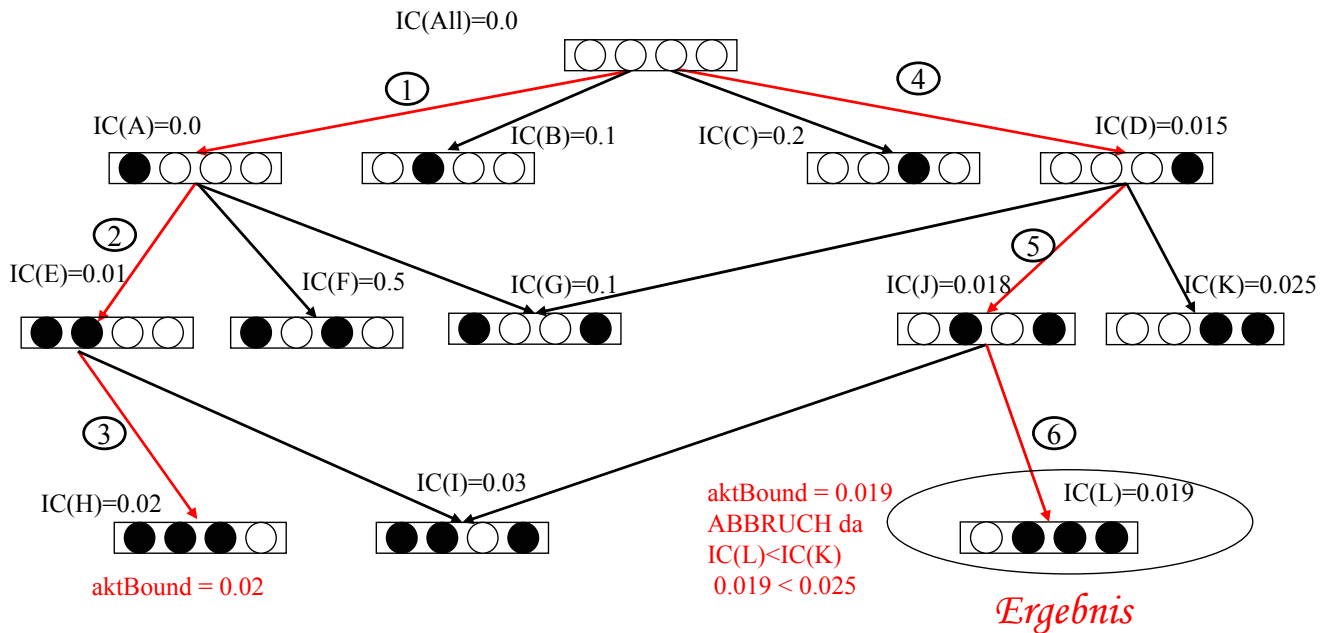
```
FUNCTION BBwithInkonsistency(Featurespace F, int k)
    queue.init(ASCENDING);
    queue.add(F, IC(F))
    aktBound:= INFINITY;
    WHILE queue.NotEmpty() oder aktBound < queue.top() DO
        aktURaum := queue.top();
        FOR ALL Unterräume U von aktURaum DO
            IF U.dimensionality() = k THEN
                IF IC(U) < aktBound THEN
                    aktBound := IC(U);
                    BestURaum := U;
            ELSE
                queue.add( U, IC(U));
    RETURN BestURaum
```

47

Beispiel 3: Branch and Bound mit Inkonsistenzkriterium

Beispiel: $k = 1$.

○ Feature selektiert ● Feature ausgeschlossen



48

Beispiel 3: Branch and Bound mit Inkonsistenzkriterium

Vorteile:

- liefert optimale Lösung bzgl. Inkonsistenzkriterium meist relativ effizient

Nachteile:

- Komplexität immer noch exponentiell da Suchraum immer noch exponentiell
- Anwendbarkeit des Inkonsistenzkriteriums ist auf nominale Attribute eingeschränkt.

49

Beispiel 4: Genetischer Algorithmus

Gegeben: Klassifikationsproblem über F .

Ziel: Selektiere k Dimensionen

Vorgehen: Genetischer Algorithmus

Gegeben:

- Population von Lösungen := Menge k -dimensionaler Unterräume
- Fitnesskriterium: Korrelationsmaß zwischen Klassen und Unterräumen
- Mutationregel und Mutationwahrscheinlichkeit:
mit Wahrscheinlichkeit $x\%$ wird Dimension D_i in U durch D_j ersetzt
- Fortpflanzung: Kombinationsregel für 2 Unterräume U_1 und U_2 :
Wähle 50% der Dimensionen aus U_1 und 50% aus U_2
- Selektionsregel: Alle Kandidationräume die $z\%$ schlechtere Fitness haben als der beste der bisherigen Generation sind nicht lebensfähig.
- Freilos: Zusätzlich zur Selektion kann jeder Unterraum mit Wahrscheinlichkeit $u\%$ in die nächste Generation übernommen.

50

Beispiel 4: Genetischer Algorithmus

Ablauf:

Initialisiere Population

WHILE Max_Fitness > Old_Fitness DO

 Mutiere Population gemäß Mutationsrate

 WHILE nextGeneration < PopulationSize DO

 Generiere neuen Kandidaten K durch Fortpflanzung

 IF K hat Freilos oder K ist fit enough THEN

K darf in die nächste Generation

 RETURN fittester Unterraum

51

Beispiel 4: Genetischer Algorithmus

Bemerkung:

- hier nur Skizze des Grundalgorithmus (viele Erweiterungen)
- Konvergenz meist nur unter „Simulated Annealing“
(Freiloswahrscheinlichkeit sinkt mit Anzahl der Generationen)

Vorteil:

- Vermeidung von lokalen Maxima
- häufig gute Approximation des optimalen Unterraums

Nachteile:

- kann lange Laufzeiten aufweisen
- viele Parameter müssen richtig gewählt werden, um einen guten Trade-Off zwischen Qualität und Laufzeit zu erzielen

52

Beispiel 5: Feature-Clustering mit Korrelation

Gegeben: Clusteringproblem über F .

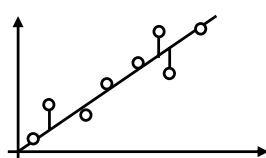
Ziel: Reduziere Featurespace auf k Dimensionen.

Vorgehen: Da man irrelevante Attribute nicht erkennen kann, beschränkt man sich auf die Elimination von redundanter Information.

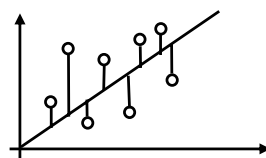
Idee: Clustere die Feature im Raum der Objekte und selektiere 1 Repräsentanten pro Cluster.

Maß für die Abhängigkeit Ähnlichkeit der Features:

- Korrelation zwischen 2 Features:
$$COR(X, Y) = \frac{COV(X, Y)}{\sqrt{VAR(X) \cdot VAR(Y)}}$$
- Regression: Bilde Regressionsgerade aus X für Y und messe quadratischen Fehler. Fehler klein \Rightarrow starke Korrelation.

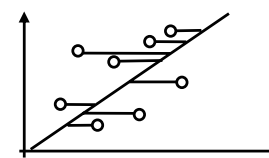


stark abhängige Dimensionen



schwach abhängige Dimensionen

\neq



(Achtung: Asymmetrie)

53

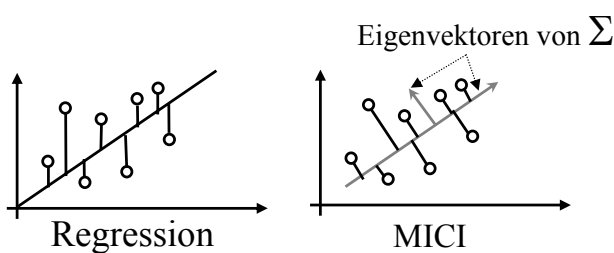
Beispiel 5: Feature-Clustering mit Korrelation

Maximaler Information Compression Index (MICI):

Idee: Messe den kleinsten Eigenwert der Kovarianzmatrix Σ zwischen beiden verglichenen Unterräumen.

$$\begin{aligned}\det(\Sigma - \lambda E) &= \det \begin{pmatrix} \text{VAR}(X) - \lambda & \text{COV}(X, Y) \\ \text{COV}(X, Y) & \text{VAR}(Y) - \lambda \end{pmatrix} \\ &= (\text{VAR}(X) - \lambda) \cdot (\text{VAR}(Y) - \lambda) - \text{COV}(X, Y)^2 \\ \Rightarrow 0 &= \lambda^2 - \lambda \cdot \text{VAR}(Y) - \lambda \cdot \text{VAR}(X) + \text{VAR}(X) \cdot \text{VAR}(Y) - \text{COV}(X, Y)^2 \\ \Rightarrow \lambda &= \frac{(\text{VAR}(Y) + \text{VAR}(X)) \pm \sqrt{(\text{VAR}(Y) + \text{VAR}(X))^2 + 4 \cdot 1 \cdot \text{VAR}(X) \cdot \text{VAR}(Y) - \text{COV}(X, Y)^2}}{2 \cdot 1}\end{aligned}$$

$$\text{MICI}(X, Y) = \text{VAR}(Y) + \text{VAR}(X) - \sqrt{(\text{VAR}(Y) + \text{VAR}(X))^2 + 4 \cdot \text{VAR}(X) \cdot \text{VAR}(Y) - \text{COV}(X, Y)^2}$$



MICI:

- Symmetrisch
- Kann aus beiden Dimensionen 1 gebildet werden, die die gesamte Information beider widerspiegelt.

54

Beispiel 5: Feature-Clustering mit Korrelation

Ablauf:

- Cluster Feature mit k-medoid Clustering für k Cluster und Abstandsmaß MICI.
- Selektiere die Clusterrepräsentanten als Featuredimensionen

Bemerkung:

- Versucht für jede Gruppe abhängiger Dimensionen eine representative Dimension
- Anwendung anderer Clustering Algorithmen denkbar.
(K-Means: Wähle Dimension die am nächsten am Cluster-Centroid liegt)
- Häufig werden Cluster-Algorithmen für Streams verwendet, wegen Ihrer Laufzeit $O(d)$

55

Beispiel 5: Feature-Clustering mit Korrelation

Vorteile:

- Verhältnismäßig schnelle Selektionsmethode
- Kommt ohne Klasseneinteilung aus

Nachteile:

- Meist kein eindeutiges Ergebnis, da Clustering von Parametern und Reihenfolge abhängen kann.
- Repräsentative Dimensionen wechseln bei unterschiedlichen Cluster Algorithmen
- basiert auf paarweiser Korrelation
=>höherwertige Dimensionen werden nicht untersucht.

56

Feature-Selektion

Diskussion:

- Viele Algorithmen basierend auf unterschiedlichen Heuristiken
- Feature können aus 2 Gründen eliminiert werden:
 - es existieren andere bedeutungsgleiche Feature (Redundanz)
 - Features sind nicht mit der Aufgabe korreliert
- häufig können auch schon nicht optimale Ergebnisse sowohl Effizienz als auch Effektivität verbessern
- Vorsicht: Selektierte Features müssen keinen direkten Einfluß auf Zielvariable haben, sondern können auch nur von den gleichen versteckten Einflüssen abhängen.

57

2.3 Featurereduktion

Idee: Anstatt Features einfach wegzulassen, generiere einen neuen niedrigdimensionalen Featureraum aus allen Features:

- Redundante Features können zusammengefasst werden
- Irrelevantere Features haben einen entsprechend kleineres Gewicht in den neuen Feature

Lösungsansätze:

- Referenzpunktansatz
- Hauptkomponentenanalyse (PCA)
- Single-Value-Decomposition (SVD)

58

1. Referenzpunkt Transformation

Idee:

Position eines Objekts kann häufig recht gut über den Abstand zu anderen Objekten beschrieben werden.

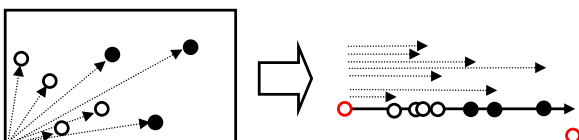
Wähle k Referenzpunkte und beschreibe Objekte über den k -dimensionalen Vektor der Abstände zu den Referenzpunkten.

Gegeben: Vektorraum $F = D_1 \times \dots \times D_n$ mit $D = \{D_1, \dots, D_n\}$.

Gesucht: k -dimensionaler Raum R , der für ein gegebenes Data Mining Problem eine optimale Lösung erlaubt.

Methode: Für die Menge der Referenzpunkte $R = \{r_1, \dots, r_k\}$ und Distanzmaß $d()$:

Transformation von Vektor $x \in F$:
$$r = \begin{pmatrix} d(r_1, x) \\ \vdots \\ d(r_k, x) \end{pmatrix}$$



59

1. Referenzpunkt Transformation

- Abstandsmaß ist meist durch Applikation gegeben.
- Auswahl der Referenzpunkte:
 - Wähle Centroide der Klassen oder Cluster-Centroide als Referenzpunkte
 - Häufig Wahl der Referenzpunkte am Rand und möglichst weit weg von allen Datenobjekten.

Vorteile:

- leicht umzusetzender Ansatz

Nachteil:

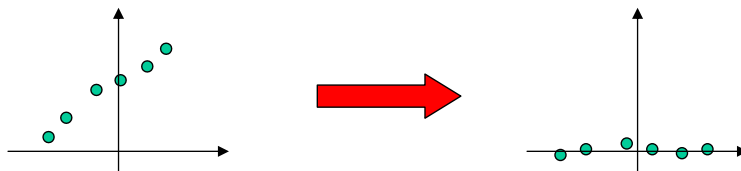
- selbst bei gleicher Featureanzahl ist die Abbildung nicht eindeutig
- Performanz stark von der Wahl der Referenzpunkte abhängig.

60

Hauptachsentransformation (PCA)

Ziel: Rotiere den Datenraum so, dass

- die Abhängigkeiten zwischen den Merkmalen verschwinden
- Abstände und Winkel der Vektoren erhalten bleiben



Gesucht ist also...

- eine orthonormale Abbildung,
- die die Richtung stärkster Varianz auf die erste Achse abbildet
- die Richtung zweitstärkster Varianz auf die zweite usw.

61

PCA

- Wir beginnen mit der Kovarianz-Matrix: $\Sigma = 1/n \sum_{x \in D} (x-\mu)(x-\mu)^T$
- Die Matrix wird zerlegt in
 - eine Orthonormalmatrix $V = [e_1, \dots, e_d]$ (Eigenvektoren)
 - und eine Diagonalmatrix $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_d)$ (Eigenwerte)
 - so dass gilt: $\Sigma = V \Lambda V^T$
- Bei Weglassen von k Basisvektoren e_j entsteht ein neuer Unterraum. Die Transformation der Vektoren aus X in diesen neuen Unterraum hat den quadratischen Fehler:

$$\chi^2 = \sum_{j=1}^k \lambda_j$$

⇒ Wähle die k Eigenvek leinsten Eigenwerten

62

PCA

Dimensionsreduktion via PCA

1. Berechne Kovarianzmatrix Σ
2. Berechne Eigenwerte und Eigenvektoren von Σ
3. Bestimme die k kleinsten Eigenwerte und lösche deren Eigenvektoren (V^c)
4. Die resultierenden Eigenvektoren bilden die Basis für den neuen Unterraum
5. Entwickle die Vektoren der Daten $X = \{x_1, \dots, x_n\}$ nach dieser neuen Unterraumbasis:

$$y_i = V^c x_i$$

⇒ Resultierende Daten $Y = \{y_1, \dots, y_n\}$ sind $(d-k)$ -dimensional

63

Single Value Decomposition (SVD)

Verallgemeinerung der PCA: Auch anwendbar wenn Kovarianz-Matrix singulär.
Im Textumfeld häufig als Latent Semantic Indexing (LSI) bezeichnet.

Grundidee:

Bestimme Zerlegung der Objekt-Feature-Matrix.

$$\begin{array}{c}
 n \text{ Objekte} \\
 \left[\begin{array}{c} \mathbf{M} \\ \mathbf{d} \times \mathbf{n} \end{array} \right] = \left[\begin{array}{c} \mathbf{T} \\ \mathbf{d} \times \mathbf{k} \end{array} \right] \cdot \left[\begin{array}{c} \mathbf{S} \\ \mathbf{k} \times \mathbf{k} \end{array} \right] \cdot \left[\begin{array}{c} \mathbf{D}^T \\ \mathbf{k} \times \mathbf{n} \end{array} \right]
 \end{array}$$

T : links-singuläre Vektoren, orthogonal

S : singuläre Werte, Diagonalmatrix

D : rechts-singuläre Vektoren, orthogonal

Zerlegung mittels numerischer Algorithmen zur Matrix-Faktorisierung.

(nicht Thema der Vorlesung)

SVD (2)

Feature-Reduktion auf $j < k$ Features:

- Sortiere TSD^T nach der Größe der k Diagonaleinträge in S
- Streiche die k-j Zeilen mit den niedrigsten singulären Werten

$$\begin{array}{c}
 n \text{ Objekte} \\
 \left[\begin{array}{c} \mathbf{M}' \\ \mathbf{d} \times \mathbf{n} \end{array} \right] = \left[\begin{array}{c} \mathbf{T}' \\ \mathbf{d} \times \mathbf{j} \end{array} \right] \cdot \left[\begin{array}{c} \mathbf{S}' \\ \mathbf{j} \times \mathbf{j} \end{array} \right] \cdot \left[\begin{array}{c} \mathbf{D}'^T \\ \mathbf{j} \times \mathbf{n} \end{array} \right]
 \end{array}$$

- M' ist Näherung von M
- Anstatt der A Attribute werden die Objekte im Raum j-dimensionalen Raum der Singular Values betrachtet => die Matrix D' beschreibt die Objekte.

SVD (3)

Problem: Momentan sind nur Trainingsdaten umgewandelt.

Wie werden neue Objekte in den neuen Feature-Raum transformiert ?

Lösungsansatz: Folding-In

Neues Objekte o.

Durch Umformung erhält man aus $M' = T' \cdot S' \cdot D'^T$ folgende Umrechnung:

$$M' = T' \cdot S' \cdot D'^T \Rightarrow S'^T \cdot T'^T \cdot M' = D'^T$$

$$\boxed{S'^T \cdot T'^T \cdot o = r} \quad \text{Umrechnungsformel}$$

Vorteile:

- Anwendbar auf alle möglichen Objekt-Attribut Matrizen
- Reduktion auf die wichtigsten Konzepte

Nachteile:

- Matrixfaktorisierung ist ein verhältnismäßig teures Verfahren
- basiert auf linearen Abbildungen zwischen Attributen und Objekten

Literatur

- A. Blum and P. Langley: *Selection of Relevant Features and Examples in Machine Learning*, Artificial Intelligence (97), 1997
- H. Liu and L. Yu: *Feature Selection for Data Mining* (WWW), 2002
- L.C. Molina, L. Belanche, Â. Nebot: *Feature Selection Algorithms: A Survey and Experimental Evaluations*, ICDM 2002, Maebashi City, Japan
- P. Mitra, C.A. Murthy and S.K. Pal: *Unsupervised Feature Selection using Feature Similarity*, IEEE Transactions on pattern analysis and Machine intelligence, Vol. 24, No. 3, 2004
- S. Deerwester, S. Dumais, R. Harshman: *Indexing by Latent Semantic Analysis*, Journal of the American Society of Information Science, Vol. 41, 1990
- J. Dy, C. Brodley: *Feature Selection for Unsupervised Learning*, Journal of Machine Learning Research 5, 2004
- I. Guyon, A. Elisseeff: *An Introduction to Variable and Feature Selection*, Journal of Machine Learning Research 3, 2003
- M. Dash, H. Liu, H. Motoda: *Consistency Based Feature Selection*, 4th Pacific-Asia Conference, PADKK 2000, Kyoto, Japan, 2000