

Vorlesung Netzbasierte Informationssysteme

Information Discovery: Text Mining

Prof. Dr. Adrian Paschke

Arbeitsgruppe Corporate Semantic Web (AG-CSW)
Institut für Informatik, Freie Universität Berlin
paschke@inf.fu-berlin.de
<http://www.inf.fu-berlin.de/groups/ag-csw/>



- Text Mining
 - Text Pre-processing
 - Features Generation
 - Features Selection
 - Text Mining
 - Classification- Supervised learning
 - Clustering- Unsupervised learning
 - Association Rule Mining

- **Spatial Data:** z.B. geographische Daten oder medizinische & Satellitenbilder
- **Multimedia Data:** Bilder, Audio, Video
- **Time-series Data:** z.B. Bankdaten und Aktien Daten
- **Text Data:** Wortbeschreibungen von Objekten, XML
- **World-Wide-Web:** Hoch unstrukturierte Text und Multimediadaten
- **Graph Data:** (Soziale) Netzwerke

- Viele Textdatenbanken existieren in der Praxis
- Nachrichtenartikel
- Forschungspublikationen
- Bücher
- Digitale Bibliotheken
- E-mail Nachrichten
- Webseiten

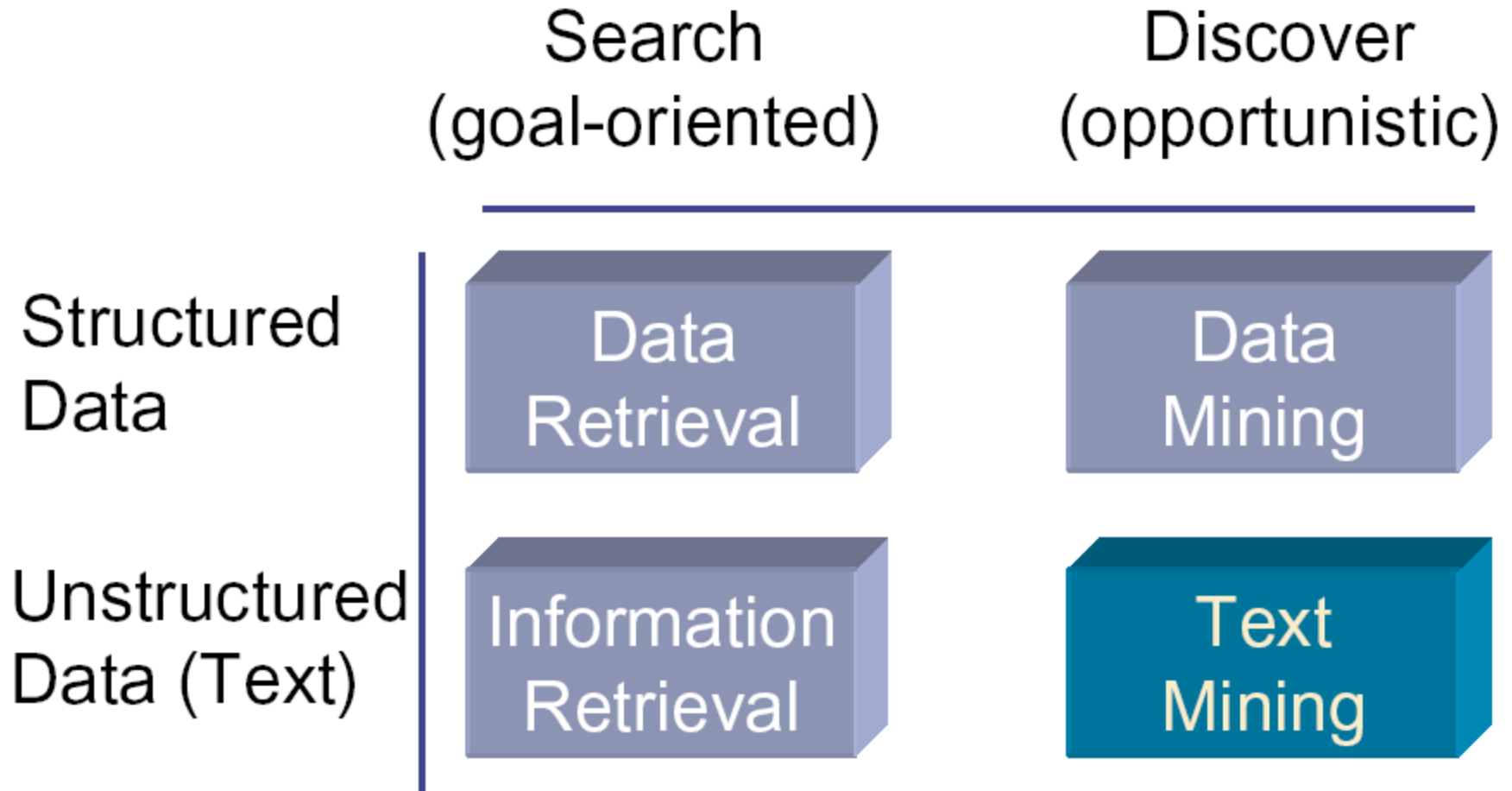
- Wachsen rapide in Größe und Wichtigkeit

- Textdatenbanken sind oft *semi-strukturiert*
- Beispiele:
 - Title
 - Author
 - Publication_Date
 - Length
 - Category
 - Abstract
 - Content

Strukturierte
Attribut/Werte Paare

Unstrukturiert

Search vs. Discovery





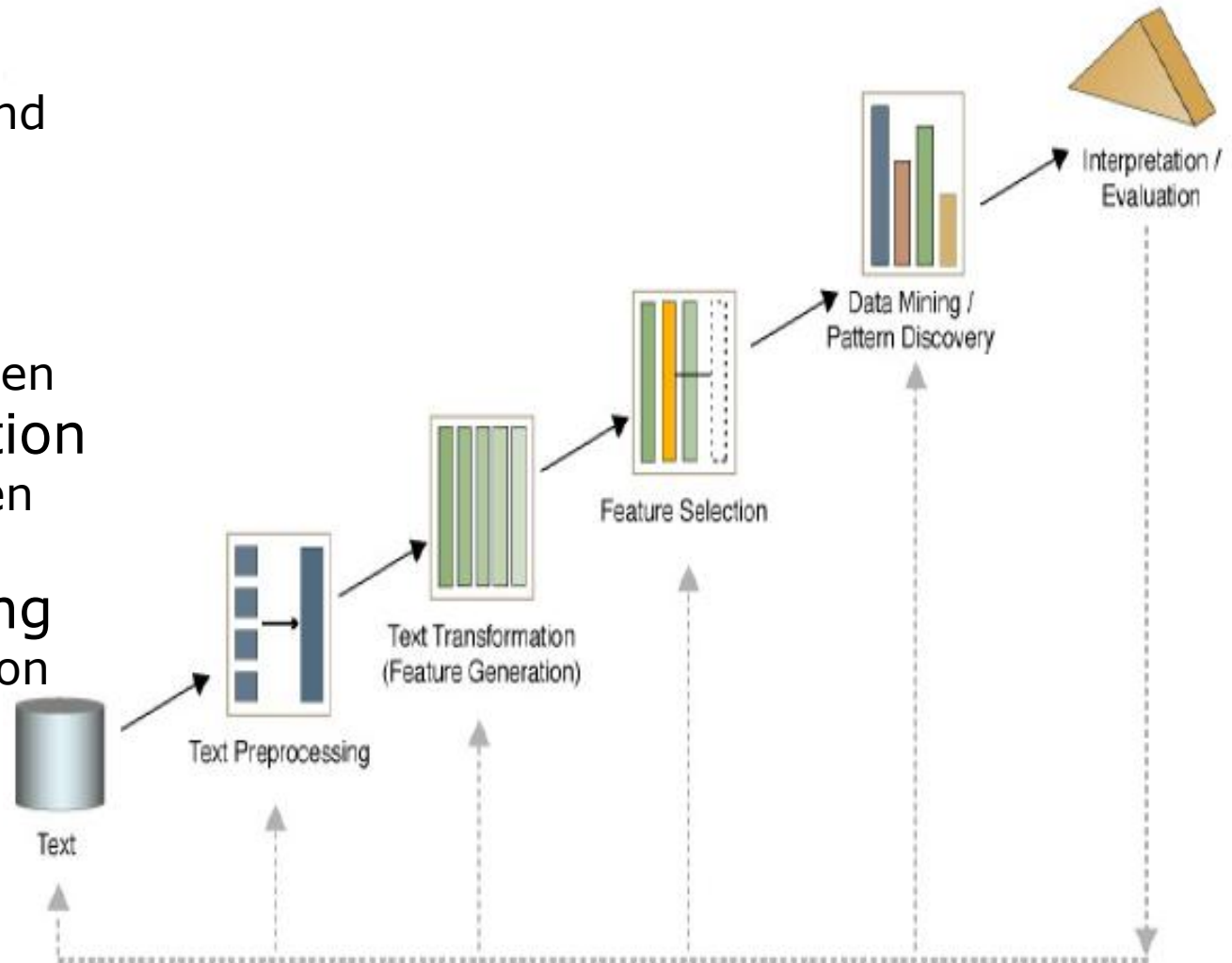
Text Mining

Machine Learning, by T. Mitchell

Data Mining – Concepts and Techniques, by Jiawei Han and
Micheline Kamber

Principle of Data Mining, by David J. Hand *et al*

- Text Preprocessing
 - Syntaktische und Semantische Analyse
- Features Generation
 - Menge an Worten
- Features Selection
 - Einfaches zählen
 - Statistik
- Text/Data Mining
 - *Classification* von Dokumenten
 - *Clustering* von Dokumenten
- Analyse der Ergebnisse



- Part Of Speech (pos) Tagging
 - Finde die jeweiligen POS für jedes Wort
z.B.: *John (Nomen) gave (Verb) the (Art.) ball (Nomen)*
- Word Sense Disambiguation
 - Context-basiert oder Nachbarschafts-basiert (proximity)
 - Sehr akkurat
- Parsing
 - Generiert einen **Parse Tree** (Graphen) für jeden Satz
 - Jeder Satz ist ein einzelner Graph

- Textdokument wird durch die darin enthaltenen Worte repräsentiert (und ihre Vorkommen)
 - z.B. "Lord of the rings" → {"the", "Lord", "rings", "of"}
 - Sehr effizient
 - Einfacheres Lernen
 - Ordnung der Worte ist nicht so wichtig für bestimmte Anwendungen
- Stemming: Identifiziert ein Wort durch seine Wurzel
 - z.B., flying, flew → fly
 - Reduziert Dimensionalität
- Stop words: Die meisten allgemeinen Worte helfen nicht im Text Mining und können entfernt werden
 - z.B., "the", "a", "an", "you" ...

- Aktuelle Schlüsselwort-orientierten Suchmaschinen können keine reichen Anfragen behandeln, wie
 - Find all books authored by “Adrian Paschke”.
- XML: Extensible Markup Language
 - XML Tags sowie Inhalte können als Features genutzt werden

```
<book> <title> NBI </title>  
      <author> <name> Adrian Paschke </name>  
      <affiliation> FUB </affiliation> </author>  
</book>
```

- Reduzierung der Dimensionalität
 - Lerner haben Schwierigkeiten mit Mehrdimensionalität
- Nicht relevante Features
 - Nicht alle Features helfen!
 - z.B. die Existenz eines Artikel in einem Nachrichtenartikel hilft wahrscheinlich nicht es als "Politik" oder "Sport" zu klassifizieren

- Clustering

- Dokumente, die gleiche Terme enthalten, werden als zusammengehörig angesehen

- Classification

- z.B. Identifikation von SPAM eMail
- Factor Analysis kann zur Reduzierung von Dimensionalität nützlich sein

- Association Rule Mining

- Sammle oft zusammen benutzte Schlüsselwörter und bilde Assoziationsregeln daraus

Text Mining: Klassifikation

- **Gegeben:** Eine Sammlung an gekennzeichneten Datensätzen (*training set*)
 - Jeder Datensatz enthält ein Satz an Features (*attributes*) und ein wahr/falsch Kennzeichen (*label*)
- **Finde:** Ein **Model** für eine Klasse als eine Funktion der Werte der enthaltenen Features
- **Ziel:** Vorher nicht gesehene Datensätze sollen eine Klasse so genau wie möglich zugeordnet werden
 - Ein **Testset** wird benutzt um die Genauigkeit des Models zu bestimmen. Normalerweise ist das Datenset in ein Trainings- und ein Testset unterteilt, wobei das Trainingset benutzt wird um das Model zu bilden und das Testset um das Model zu validieren.

Text Mining: Clustering

- **Gegeben:** Ein Satz and Dokumenten und ein Ähnlichkeitsmaß (*similarity measure*) zwischen den Dokumenten
- **Finde:** Cluster, so dass:
 - Dokumente in einem Cluster ähnlicher sind als die anderen Dokumente
 - Dokumenten in unterschiedlichen Clustern sind sich weniger ähnlich
- **Ziel:**
 - Finde ein korrektes Set and Dokumenten

Similarity Measures:

- *Euclidean Distance* wenn Attribute fortlaufend sind
- Andere Problem-spezifische Maße
 - z.B., “how many words are common in these documents”

- Überwachtes Lernen (*classification*)
 - Überwachung: Die Trainingsdaten (Beobachtungen, Messungen, etc.) werden gekennzeichnet mit einem *Label*, welches die Klasse der Beobachtung angibt
 - Neue Daten werden anhand des Trainingssets klassifiziert
- Unüberwachtes Lernen (*clustering*)
 - Die Klassenlabels der Trainingsdaten sind unbekannt
 - Gegeben ein Satz an Messungen, Beobachtungen, etc. mit dem Ziel Klassen oder Cluster in Daten zu bilden

- Korrekte Klassifizierung: Die bekannten Label der Testdaten sind identisch mit den Klassenergebnissen (*class result*) des Klassifikationsmodells
- Accuracy Ratio: Der Prozentsatz der korrekt durch das Model klassifizierten Testdaten
- *Abstandsmessung* zwischen Klassen kann genutzt werden
 - z.B., Klassifizierung von "football" Dokumenten als "basketball" Dokumente ist besser als sie als "crime" zu klassifizieren.

- **Gute Clustering Methode:** erzeugt hoch qualitative Cluster mit . . .
 - Hoher *intra-class* Ähnlichkeit
 - Niedriger *inter-class* Ähnlichkeit
- Die *Qualität* einer Clustering Methode wird auch gemessen durch seine Fähigkeit einige oder alle *versteckten* Muster zu finden

- Clustering

- Dokumente, die gleiche Terme enthalten, werden als zusammengehörig angesehen

- Classification

- z.B. Identifikation von SPAM eMail
- Factor Analysis kann zur Reduzierung von Dimensionalität nützlich sein

- Association Rule Mining

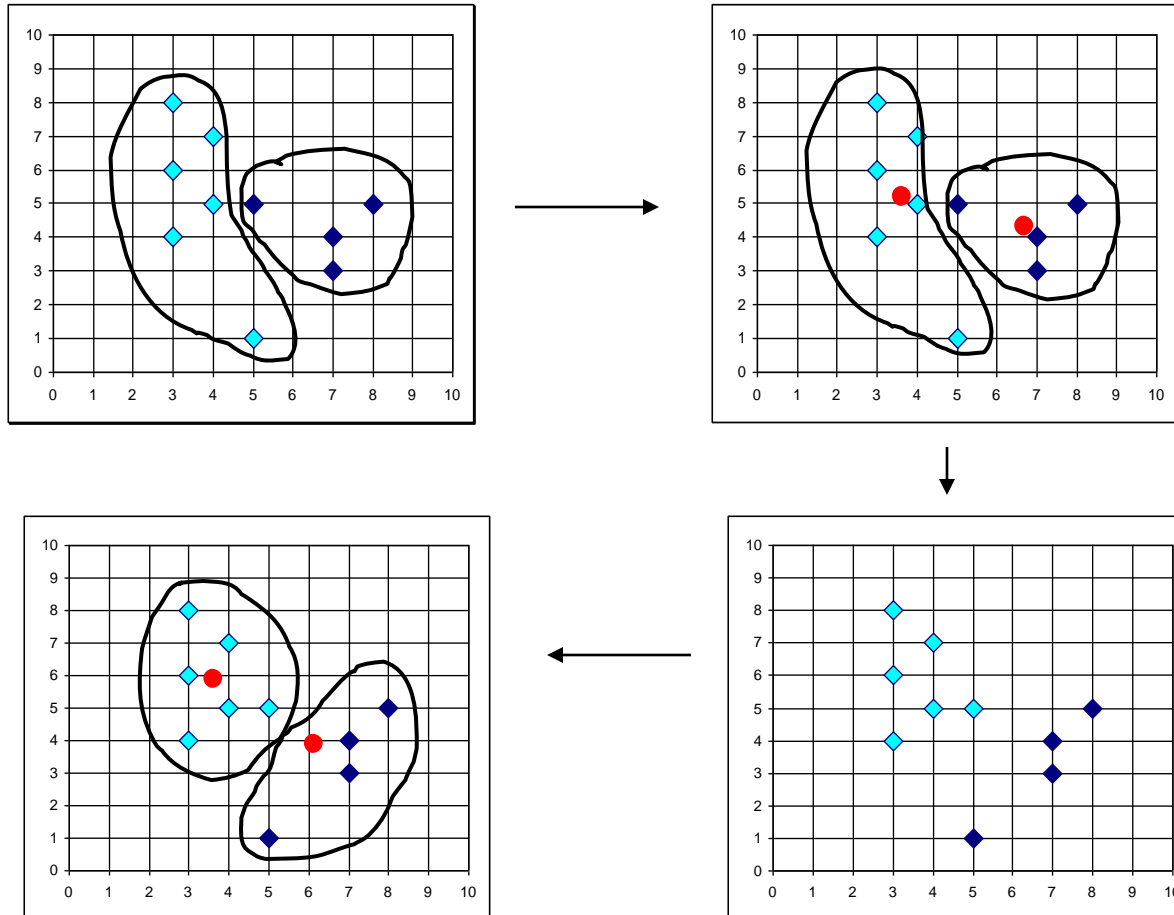
- Sammle oft zusammen benutzte Schlüsselwörter und bilde Assoziationsregeln daraus

- Partitionierungsmethoden
- Hierarchische Methoden

- **Partitionierungsmethode:** Konstruiere eine Partition von n Dokumenten in eine Menge von k Clustern
- **Gegeben:** Eine Menge von Dokumenten und die Anzahl k
- **Finde:** Eine Partition von k Clustern welche das gewählte Partitionskriterium optimiert
 - **Global optimal:** Enumeriere alle Partitionen
 - Heuristische Methoden: *k-means* und *k-medoids* Algorithmen
 - *k-means:* Jeder Cluster wird durch das Zentrum des Clusters repräsentiert

- *k-means* Algorithmus:
 1. Partitioniere Objekte in k nicht leere Subsets.
 2. Berechne Kernpunkte (**seed points**) als Zentren (**centroids**) der Cluster der aktuellen Partition. Ein "centroid" ist das Zentrum (mean point) des Clusters.
 3. Füge jedes Objekt zu dem Cluster mit dem nächsten Kernpunkt hinzu.
 4. Gehe zurück zu Schritt 2; beende wenn keine weitere neue Zuordnung.

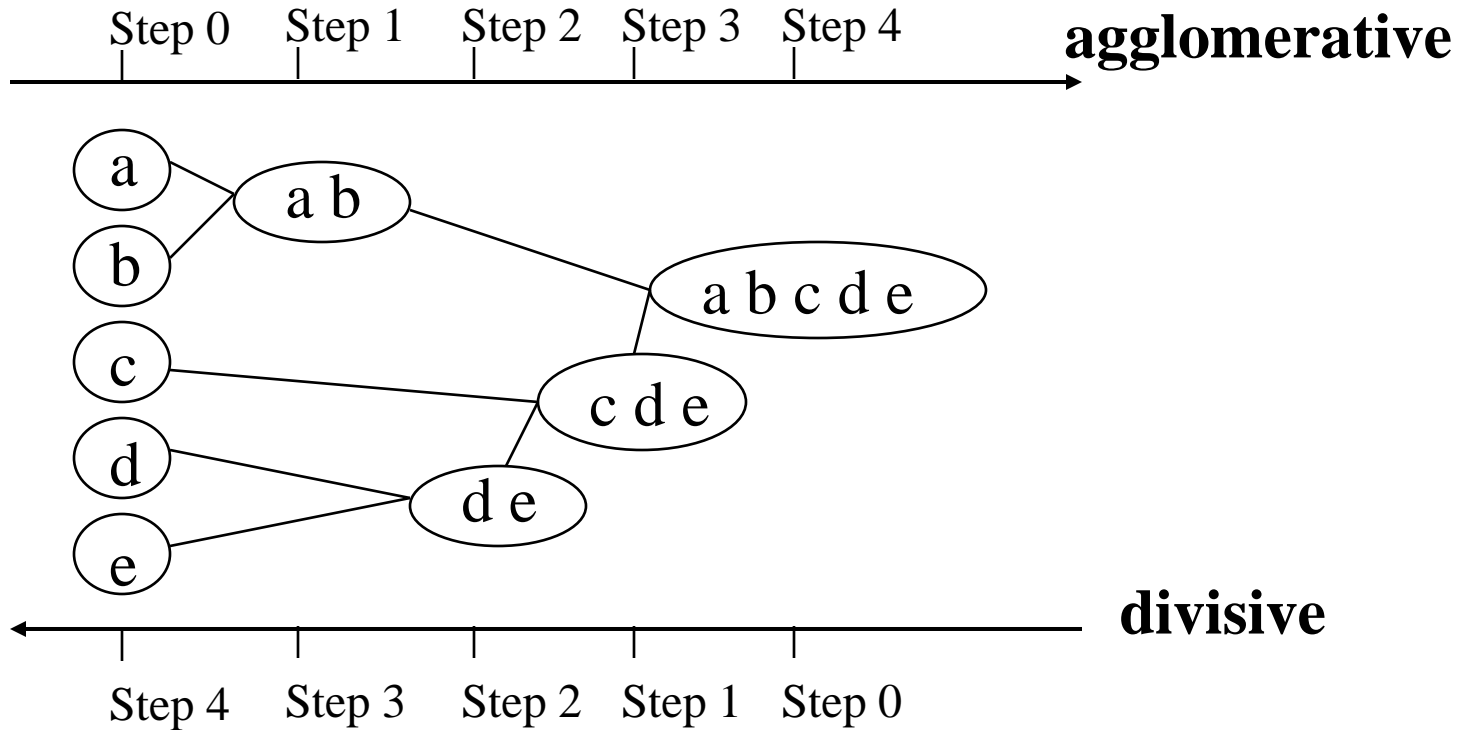
K-means Clustering: Beispiel



- Partitionierungsmethoden
- Hierarchische Methoden

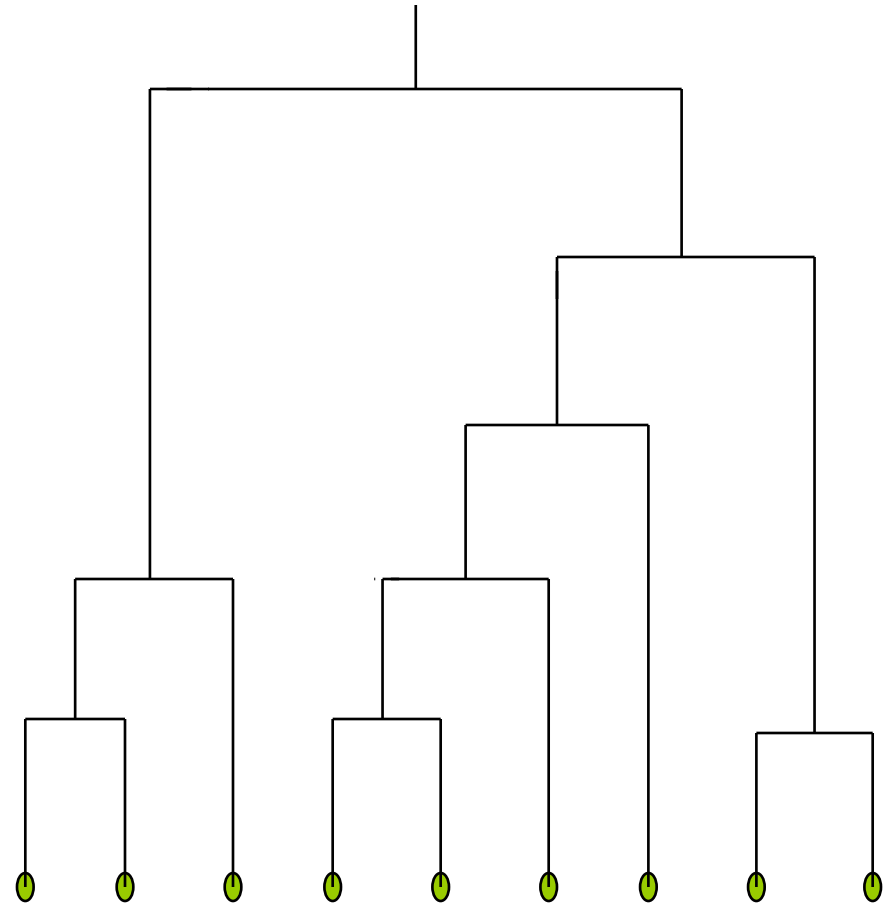
- **Agglomerative:**
 - Starte mit jedem Dokument als einfaches Cluster.
 - Eventuell gehören alle Dokumente zum gleichen Cluster.
- **Divisive:**
 - Starte mit allen Dokumenten in einem Cluster.
 - Eventuell bildet jeder Knoten einen eigenen Cluster.
- Die Anzahl an Clustern k wird nicht im Vorfeld benötigt
- Benötigt eine Terminierungsbedingung

Hierarchisches Clustering: Beispiel



Ein Dendrogramm: Hierarchisches Clustering

- Dendrogramm: Teile die Datenobjekte in mehreren Ebenen verschachtelter Partitionen auf (Baum an Clustern).
- Clustering der Datenobjekte wird durch Zerschneidung des Dendrogramms auf der gewünschten Ebene erreicht, dann formt jede **verbundene** Komponente ein Cluster.

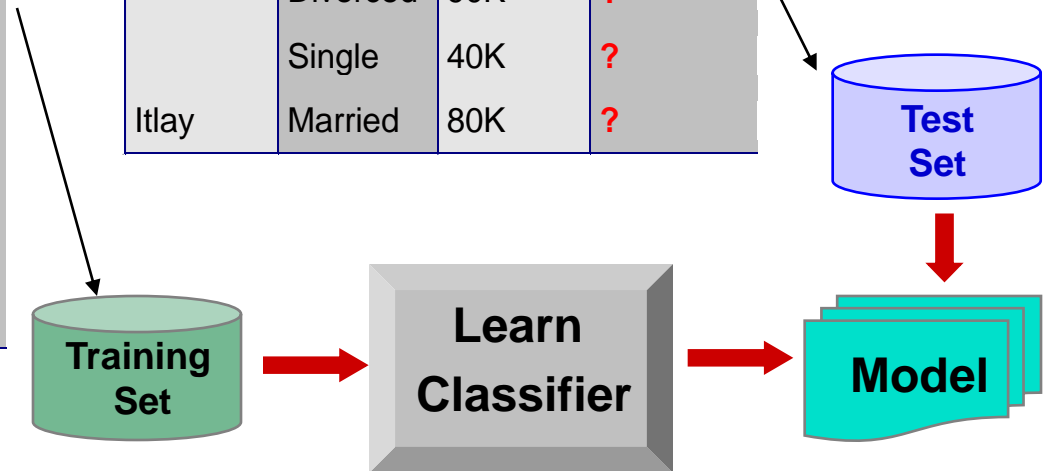


Klassifizierung: Ein Beispiel

categorical
categorical
continuous
class

Ex#	Country	Marital Status	Income	Hooligan
1	England	Single	125K	Yes
2	England	Married		Yes
3	England	Single	70K	Yes
4	Italy	Married	40K	No
5	USA	Divorced	95K	No
6	England	Married	60K	Yes
7	England		20K	Yes
8	Italy	Single	85K	Yes
9	France	Married	75K	No
10	Denmark	Single	50K	No

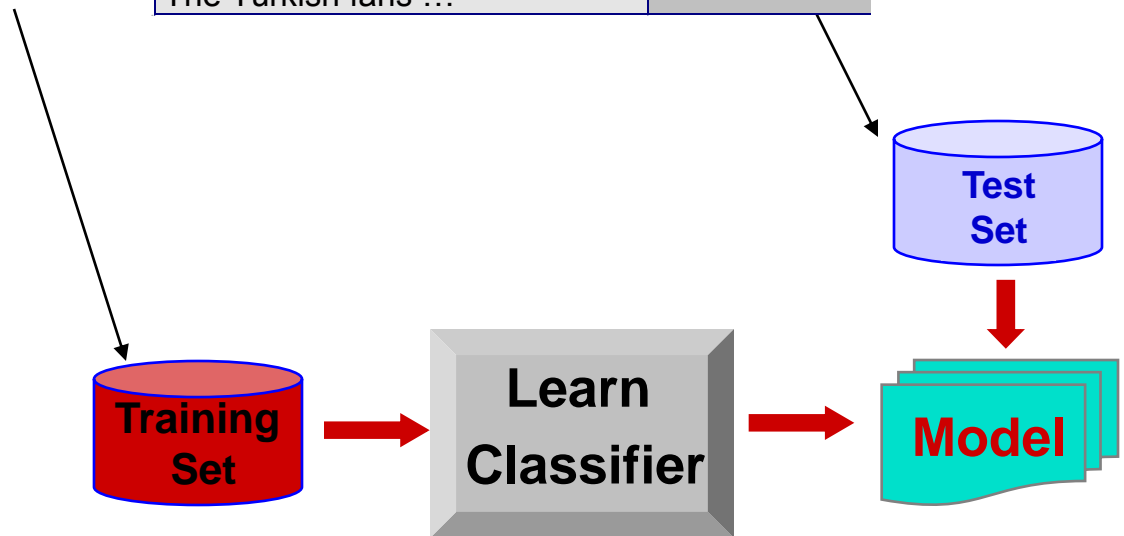
Country	Marital Status	Income	Hooligan
England	Single	75K	?
Turkey	Married	50K	?
England	Married	150K	?
	Divorced	90K	?
	Single	40K	?
Italy	Married	80K	?



Textklassifizierung: Ein Beispiel

Ex#	text	class
1	An English football fan ...	Yes
2	During a game in Italy ...	Yes
3	England has been beating France ...	Yes
4	Italian football fans were cheering ...	No
5	An average USA salesman earns 75K	No
6	The game in London was horrific	Yes
7	Manchester city is likely to win the championship	Yes
8	Rome is taking the lead in the football league	Yes

Hooligan	
A Danish football fan	?
Turkey is playing vs. France. The Turkish fans ...	?



- Clustering

- Dokumente, die gleiche Terme enthalten, werden als zusammengehörig angesehen

- Classification

- z.B. Identifikation von SPAM eMail
- Factor Analysis kann zur Reduzierung von Dimensionalität nützlich sein

- Association Rule Mining

- Sammle oft zusammen benutzte Schlüsselwörter und bilde Assoziationsregeln daraus

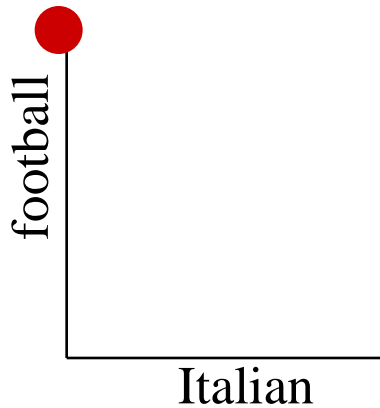
- Instanzbasierte Methoden
- Entscheidungsbäume
- Neuronale Netzwerke
- Bayesian Klassifizierung

- Instanz-basierte (Speicher-basiertes) Lernen
 - Speichere die Trainingsbeispiele und verzögere die Auswertung ("lazy evaluation") bis eine neue Instanz klassifiziert werden muss
- k -nearest Neighbor Ansatz
 - **Instanzen** (Beispiele) werden als **Punkte in einem Euklidischen Raum** repräsentiert

Textbeispiele im Euklidischen Raum

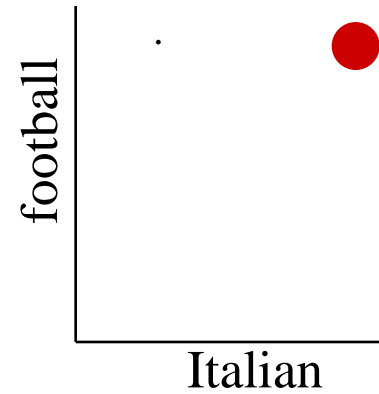
The English
football fan
is a hooligan.

·
·

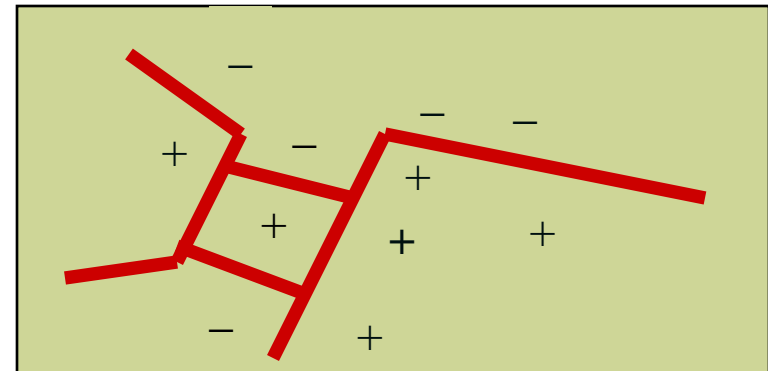
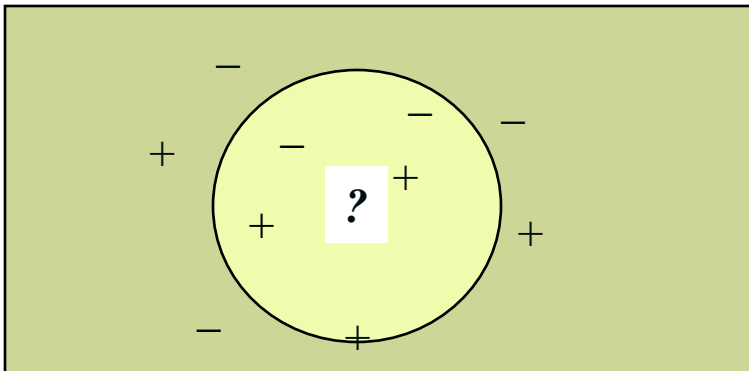


Similar to his
English equivalent,
the **Italian**
football fan
is a hooligan.

·
·



- Alle Instanzen korrespondieren mit Punkten im n -D Raum
- Der nächste Nachbar ist als Euklidische Entfernung definiert
- Der k -NN gibt den allgemeinsten Wert unter den k nächsten Trainingsbeispielen zurück
- Voronoi Diagramm: Die Entscheidungsfläche gebildet durch 1-NN für ein typisches Set an Trainingsbeispielen

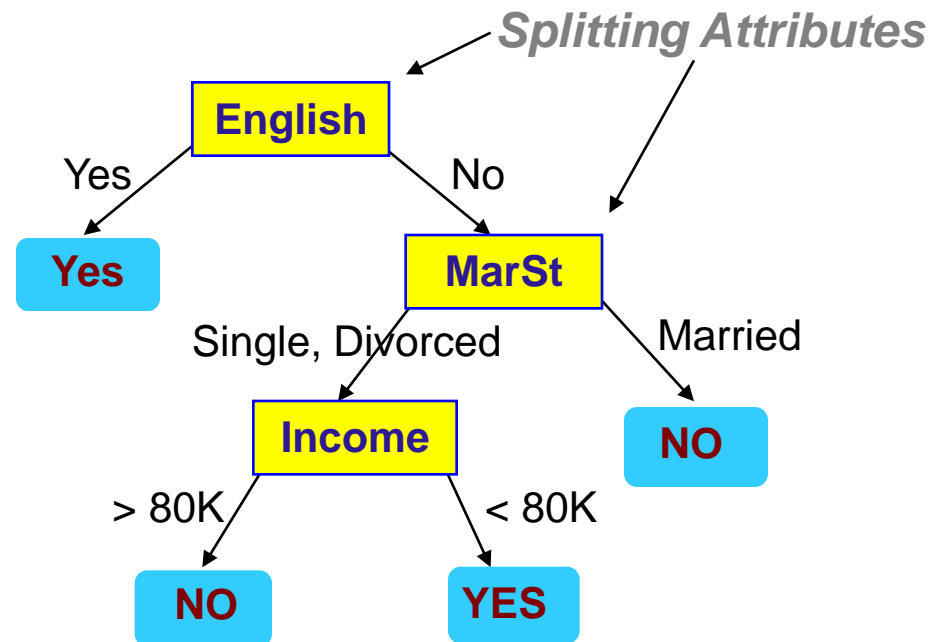


- Instanzbasierte Methoden
- Entscheidungsbäume
- Neuronale Netzwerke
- Bayessche Klassifizierung

Entscheidungsbaum: Ein Beispiel

categorical
categorical
continuous
class

Ex#	Country	Marital Status	Income	Hooligan
1	England	Single	125K	Yes
2	England	Married	100K	Yes
3	England	Single	70K	Yes
4	Italy	Married	40K	No
5	USA	Divorced	95K	No
6	England	Married	60K	Yes
7	England	Divorced	20K	Yes
8	Italy	Single	85K	Yes
9	France	Married	75K	No
10	Denmark	Single	50K	No

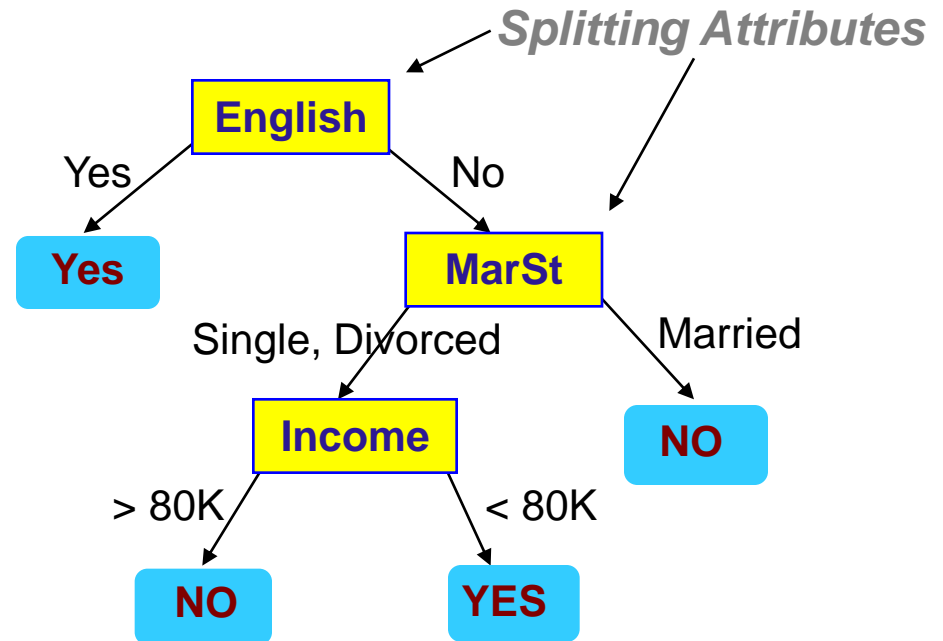


Das Aufteilungsattribut eines Knotens wird durch einen spezifischen Attributselektionsalgorithmus bestimmt

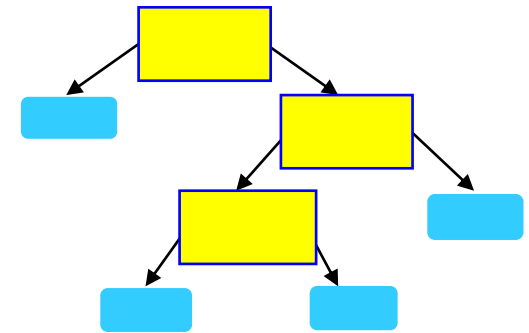
Entscheidungsbaum: Ein Textbispiel

text *class*

Ex#		Hooligan
1	An English football fan ...	Yes
2	During a game in Italy ...	Yes
3	England has been beating France ...	Yes
4	Italian football fans were cheering ...	No
5	An average USA salesman earns 75K	No
6	The game in London was horrific	Yes
7	Manchester city is likely to win the championship	Yes
8	Rome is taking the lead in the football league	Yes

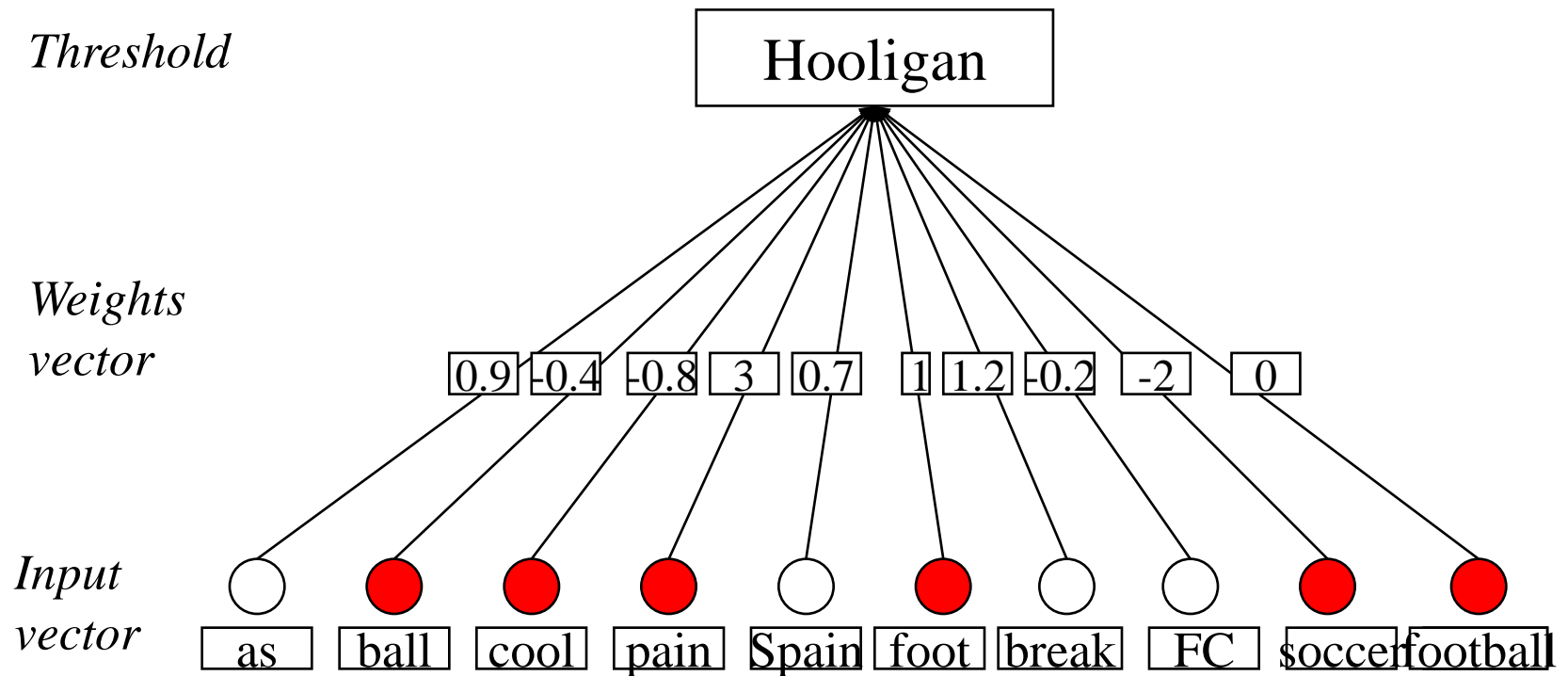


- Entscheidungsbaum (EB)
 - Ein Flow-Chart ähnliche Baumstruktur
 - Interne Knoten sind Tests auf Attribute
 - Abzweigungen sind Ergebnis des Tests
 - Blattknoten repräsentieren Klassenkennungen oder Klassenverteilung
- EB Erzeugung besteht aus zwei Phasen:
 - Baumkonstrurierung
 - Baumkürzung (Tree pruning)
 - Identifiziere und entferne Abzweigungen welche Lärm (**noise**) oder Ausreißer (**outliers**)
- Benutzung von Entscheidungsbäumen: Klassifizierung von unbekanntem Beispielen
 - Test der Attribute der Beispiele anhand des Entscheidungsbaums



- Instanzbasierte Methoden
- Entscheidungsbäume
- Neuronale Netzwerke
- Bayessche Klassifizierung

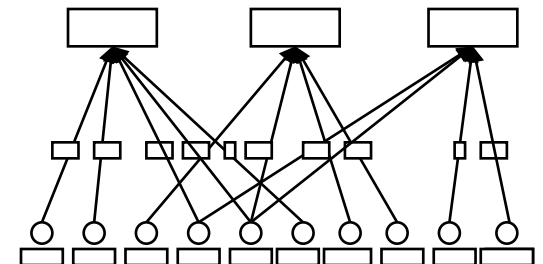
Ein einfaches Schichten "Perceptron"



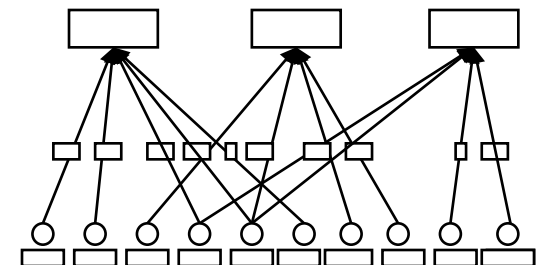
- Der *n*-dimensionale Eingabevektor wird zur Klassifizierung durch Multiplikation und Funktions Mapping genutzt

Einfach- vs. All-Klassifizierer

- Netzwerk von Schwellwertschranken
- Zielknoten repräsentieren Klassenkennungen
- Eingabeknoten repräsentieren die Relationen (features) im Beispiel
- Ein Beispiel ist **positive** für ein Netzwerk und **negative** für andere (abhängig vom Algorithmus)
- Allokation der Knoten (features) und Links sind Daten-getrieben (ein Link zwischen Feature i und Ziel j wird nur dann erzeugt, wenn i mit Ziel j aktiv ist).



- Ziel:
 - Ein Vektor mit Gewichten der fast alle Beispiele korrekt klassifiziert (unter Benutzung der Trainingsdaten)
- Schritte
 - Initialisiere die Gewichte mit zufälligen (konstanten) Werten
 - Gebe die Eingabebeispiele eins nach dem anderen in das Netzwerk ein
 - Für jeden Einheit
 - Berechne die Netzeingabe zu der Einheit als lineare Kombination aller Eingaben zu der Einheit
 - Berechne den Ausgabewert durch Nutzung der Aktivierungsfunktion (threshold)
 - Berechne den Fehler
 - Aktualisiere die Gewichte



- Vorteile
 - Vorhersagegenauigkeit ist generell sehr hoch
 - Robust, funktioniert auch, wenn Trainingsbeispiele Fehler enthalten
 - Schnelle Evaluierung der gelernten Zielfunktion
 - Leicht zu berechnen durch parallele Abarbeitung
- Nachteile
 - Lange Trainingszeiten
 - Schwierig die gelernte Funktion (Gewichte) zu verstehen
 - Schwierig **Domänenwissen** zu integrieren

- Instanzbasierte Methoden
- Entscheidungsbäume
- Neuronale Netzwerke
- Bayessche Klassifizierung

- Das Klassifizierungsproblem kann mit **Wahrscheinlichkeiten** formalisiert werden:
 $P(C|X)$ = Wahrscheinlichkeit, dass das Beispiel von Klasse C ist
z.B. $P(\text{Hooligan} \mid \text{English, fan, married...})$
- Idee: bestimme zu Beispiel X die Klassenkennung C so dass $P(C|X)$ maximal ist

- **Probabilistisches Lernen:** Berechnen expliziter Wahrscheinlichkeiten für Hypothesen ist unter den praktischsten Ansätzen für bestimmte Arten von
- **Inkrementell:** Jedes Trainingsbeispiel kann inkrementell die Wahrscheinlichkeit erhöhen/erniedrigen, dass einen Hypothese korrekt ist.
- **Vorwissen:** kann mit den beobachteten Daten kombiniert werden
- **Standard:**
 - Stellt einen Standard zur **optimalen Entscheidungsfindung** zur Verfügung, mit dem andere Methoden gemessen werden können.
 - In einer einfache Form, eine **Grundlinie** anhand der andere Methoden gemessen werden können

- **Bayessches Theorem:**

$$P(C|X) = P(X|C) \cdot P(C) / P(X)$$

- $P(X)$ ist konstant für alle Klassen

- Daher schätze $P(C|X)$ so dass:

$$P(C|X) \approx P(X|C) \cdot P(C)$$

- $P(C)$ = relative Frequenz von Klasse C Mustern

- Problem: Berechnung von $P(X|C)$ ist nicht durchführbar!

- X ist höchstwahrscheinlich ein Beispiel, dass vorher noch nicht gesehen wurde

- Naïve Annahme:
Feature Unabhängigkeit

$$P(x_1, \dots, x_k | C) = P(x_1 | C) \cdot \dots \cdot P(x_k | C)$$

- $P(x_i | C)$ wird als die relative Frequenz von Beispielen mit Wert x_i als Feature in Klasse C geschätzt
- Berechnung einfach!!!

- ... macht Berechnung möglich
- ... führt zu optimalen Klassifizieren wenn erfüllt
- ... aber ist nur selten in der Praxis erfüllt, da Attribute (Variablen) oft korreliert sind
- Ansätze diese Limitierungen zu überwinden:
 - **Bayessche Netzwerke**, welche Bayessche Reasoning mit kausalen Abhängigkeiten zwischen Features verbinden

- Clustering

- Dokumente, die gleiche Terme enthalten, werden als zusammengehörig angesehen

- Classification

- z.B. Identifikation von SPAM eMail
- Factor Analysis kann zur Reduzierung von Dimensionalität nützlich sein

- Association Rule Mining

- Sammle oft zusammen benutzte Schlüsselwörter und bilde Assoziationsregeln daraus

- Zielt auf die Erkennung interessanter Korrelationen oder anderer Beziehungen in Daten
- Finde Regel der Form

if A and B then C and D

- Welche Attribute in die Relation aufgenommen werden ist unbekannt

Term/document	d_1	\dots	d_n
t_1	$v_1^{(1)}$	\dots	$v_1^{(n)}$
\vdots	\vdots	\ddots	\vdots
t_m	$v_m^{(1)}$	\dots	$v_m^{(n)}$

- c1 Human machine *interface* for Lab ABC *computer* applications
- c2 A *survey* of *user* opinion of computer *system* *response* *time*
- c3 The EPS *user interface* management *system*
- c4 *System* and *human system* engineering testing of EPS
- c5 Relation of *user-perceived response time* to error measurement
- m1 The generation of random, binary, unordered *trees*
- m2 The intersection *graph* of paths in *trees*
- m3 *Graph minors* IV: Widths of *trees* and well-quasi-ordering
- m4 *Graph minors: A survey*

Beispiel: Terme und Dokumente

Terms	Documents								
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

- **Item:** Einfacher Term, **Itemset:** Set von Termen
- **Support/coverage** eines Itemset I : # von Dokumenten welche I enthalten
- **Minimum Support** σ : Grenzwert
- **Frequent Itemset** : mit support $> \sigma$.
- **Frequent Itemsets** repräsentieren Itemsets welche positiv korreliert sind

- *Frage: Mögliche Assoziationsregeln?*
- $A \Rightarrow B, E$
- $A, B \Rightarrow E$
- $A, E \Rightarrow B$
- $B \Rightarrow A, E$
- $B, E \Rightarrow A$
- $E \Rightarrow A, B$
- $_ \Rightarrow A, B, E$ (leere Regel), or $\text{true} \Rightarrow A, B, E$

- Annahme $R : I \Rightarrow J$ ist eine Assoziationsregel
 - $\text{sup}(R) = \text{sup}(I \text{ and } J)$ ist der *Support Count*
 - Unterstützung des Itemset „I and J“
 - $\text{conf}(R) = \text{sup}(R) / \text{sup}(I)$ ist die *Confidence* von R
 - Teildokumente mit I welche „I and J“ haben
- Assoziationsregeln mit minimalem *Support* und *Count* werde auch als “**starke**” Regeln bezeichnet.

- Q: Given frequent set $\{A, B, E\}$, what association rules have $\text{minsup} = 2$ and $\text{minconf} = 50\%$?

$$A, B \Rightarrow E : \text{conf} = 2/4 = 50\%$$

$$A, E \Rightarrow B : \text{conf} = 2/2 = 100\%$$

$$B, E \Rightarrow A : \text{conf} = 2/2 = 100\%$$

$$E \Rightarrow A, B : \text{conf} = 2/2 = 100\%$$

Don't qualify

$$A \Rightarrow B, E : \text{conf} = 2/6 = 33\% < 50\%$$

$$B \Rightarrow A, E : \text{conf} = 2/7 = 28\% < 50\%$$

$$_ \Rightarrow A, B, E : \text{conf} = 2/9 = 22\% < 50\%$$

TID	List of items
1	A, B, E
2	B, D
3	B, C
4	A, B, D
5	A, C
6	B, C
7	A, C
8	A, B, C, E
9	A, B, C

Beispiel:

Outlook	Temp.	Humidity	Windy	Play
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Rainy	Mild	High	FALSE	Yes
Rainy	Cool	Normal	FALSE	Yes
Rainy	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Sunny	Mild	High	FALSE	No
Sunny	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	FALSE	Yes
Sunny	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Mild	High	TRUE	No

1-Item	2-Item	3-Item	4-Item
Outlook=sunny (5)	Outlook=sunny temp=mild (2)	Outlook=sunny temp=hot humidity=high (2)	Outlook=sunny temp=hot humidity=high play=no (2)
Outlook= overcast (4)	Outlook=sunny temp=hot (2)	Outlook=sunny temp=hot play=no (2)	Outlook=sunny humidity=high windy=false play=no (2)
Outlook=rainy (5)	Outlook=sunny humidity=norm (2)	Outlook=sunny humidity=norm play=yes (2)	Outlook=over temp=hot windy=false play=no (2)
Temp=cool (4)	Outlook=sunny windy=true (2)	Outlook=sunny humidity=high windy=false (2)	Outlook=rainy temp=mild windy=false play=yes (2)
Temp=mild (6)	Outlook=sunny windy=true (2) ...	Outlook=sunny humidity=high play=no (3)	Outlook=rainy humidity=norm windy=false play=yes (2)

3-Item Set w/support 4:

Humidity = normal, windy = false, play = yes

Kandidaten für Assoziationsregeln: confidence

If humidity = normal and windy = false then play = yes 4/4

If humidity = normal and play = yes then windy = false 4/6

If windy = false and play = yes then humidity = normal 4/6

If humidity = normal then windy = false and play = yes 4/7

If windy = false then humidity = normal and play = yes 4/8

If play = yes then humidity = normal and windy = false 4/9

If-then humidity=normal and windy=false and play=yes 4/12

Beispiel: Von Sets zu Regeln

4-Item Set (w/support 2):

Temperature = cool, humidity = normal,
windy = false, play = yes

Kandidaten für Assoziationsregeln: Confidence (100%)

If temperature = cool, windy = false → humidity = normal, play = yes 2/2

If temperature = cool, humidity = normal, windy = false → play = yes 2/2

If temperature = cool, windy = false, play = yes → humidity = normal 2/2

“Beste” Regeln (Support = 4, Confidence = 100%)

If humidity = normal and windy = false → play = yes

If temperature = cool → humidity = normal

If outlook = overcast → play = yes

- Schritt 1: Finde alle (häufigen) *Item Sets*, welche den minimalen *Support* erfüllen
- Schritt 2: Finde alle Regeln, welche die minimale *Confidence* erfüllen
- Schritt 3: Pruning

- Text ist kompliziert zu verarbeiten, aber relativ gute Ergebnisse können mit Text Mining einfach erreicht werden
- Zusätzliche **Intelligenz** kann in das Text Mining integriert werden
 - Zu jeder Phase des Text Mining Prozesses
- Es gibt viele weitere **wissenschaftliche und statistische Text Mining Methoden**

- *Data Mining – Concepts and Techniques*, by Jiawei Han and Micheline Kamber
- *Principle of Data Mining*, by David J. Hand *et al*
- *Text Classification from Labeled and Unlabeled Documents using EM*, Kamal Nigam *et al*
- *Fast and accurate text classification via multiple linear discriminant projections*, S. Chakrabarti *et al*
- *Frequent Term-Based Text Clustering*, Florian Beil *et al*
- *The PageRank Citation Ranking: Bringing Order to the Web*, Lawrence Page and Sergey Brin
- *Untangling Text Data Mining*, by Marti. A. Hearst, <http://www.sims.berkeley.edu/~hearst/papers/acl99/acl99-tdm.html>



Fragen?