- Vortragsthema

Klassifikation

Agenda

I Begriff Klassifikation
- Abgrenzung
- Anforderungen
- Anwendungsgebiete

II Klassifikatoren
- Entscheidungsbaum
- naive Bayes – Algorithmus
- Künstlich neuronale Netze
- k-nearest neighbour – Methode
- Support Vector Machine

III Dimensionsreduktion

Umsetzung in Software

Vergleich der Algorithmen

Quellenangaben

Fragen

- eine Data-Mining-Methode, die der Zusammenfassung von Datenobjekten zu disjunkten Klassen dient
- Klasse = Menge von zusammengehörigen Datenobjekten
- im Gegensatz zum Clustering müssen die Klassen bekannt sein
- zwei Phasen:
  1. Lernphase
  2. Klassifizierung von Datenobjekten
- Attribute (nominal, numerisch) der Datenobjekte

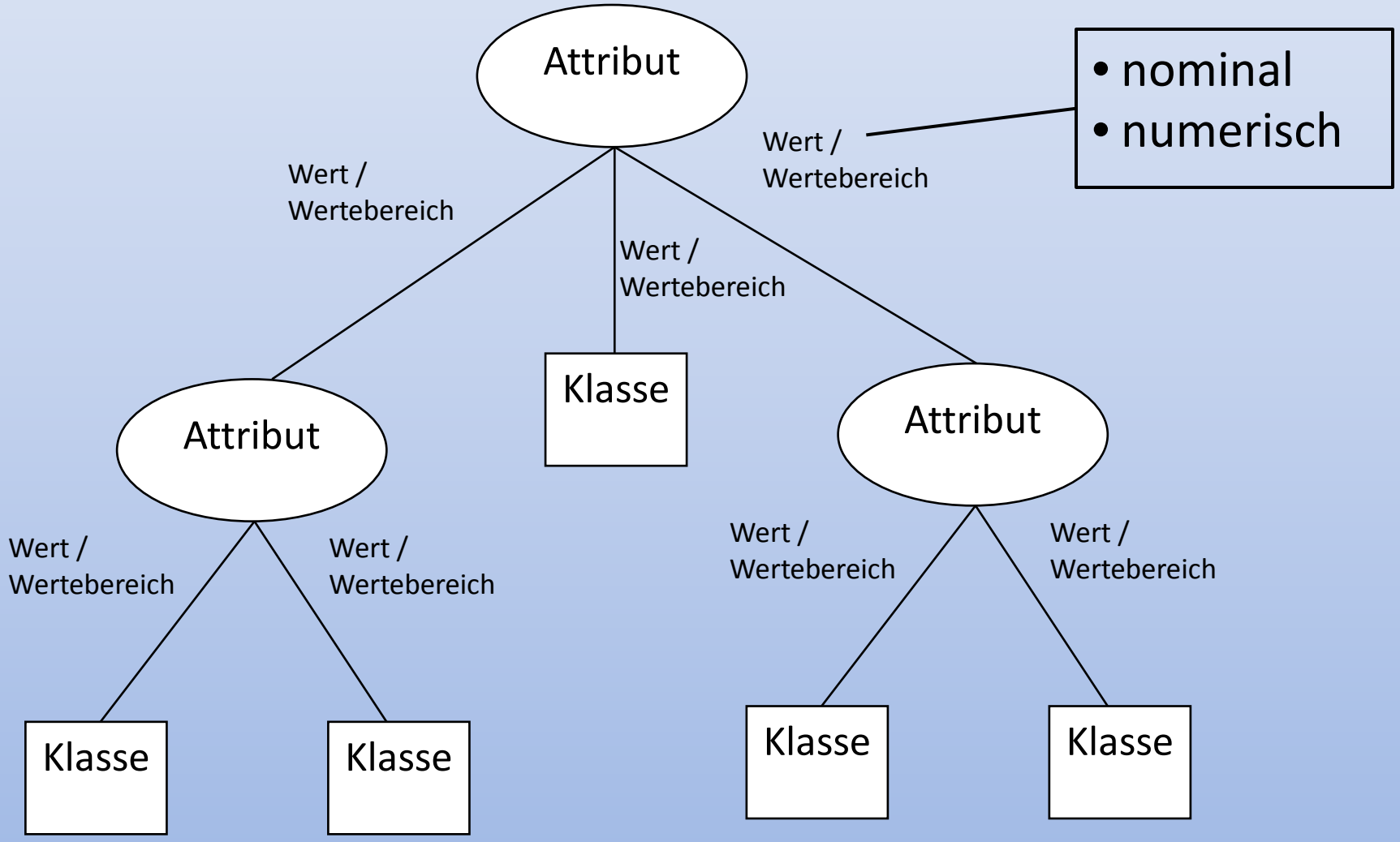
Anforderungen an einen Klassifikator und Daten

- möglichst korrekte Klassifizierung der Daten
- eine akzeptable Performance und Skalierbarkeit
- Ausreißern, fehlende Attribute
- Overfitting vermeiden
- Trainingsdaten sind notwendig

Anwendungsgebiete

- Medizin
- Marketing & Marktanalysen
- Versicherung
- Mustererkennung

Entscheidungsbaum


Attribut
• nominal
• numerisch
Wert / Wertebereich
Wert / Wertebereich
Wert / Wertebereich
Attribut
Klasse
Attribut
Wert / Wertebereich
Wert / Wertebereich
Wert / Wertebereich
Wert / Wertebereich
Klasse
Klasse
Klasse
Klasse

Konstruktion eines Entscheidungsbaumes

Informationsmenge:

- Maß für den Informationsgehalt, der nach dem Wissen über den Attributwert vorhanden ist
- auch als Entropie bezeichnet
- Einheit: bits
- $\text{info}([p_1, p_2, \ldots, p_n]) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 \ldots - p_n \log_2 p_n$

  n – Anzahl der Klassen
- $p_i = \frac{\#\, des\ Attributwerts}{\#\, aller\ Werte\ dieses\ Attributs}$
- n = 1 -> info(p_1) = 0

Beispiel - Wetterdaten

Informationsgewinn Wurzel: $\text{info}([9,5]) = -\frac{5}{14}\log_2\frac{5}{14} - \frac{9}{14}\log_2\frac{9}{14} = 0.940$

Informationsgewinn aller Attribute:

outlook

$\text{info}_{overcast}([4,0]) = \text{info}_{overcast}([4]) = 0$

$\text{info}_{rainy}([2,3]) = 0.971$

$\text{info}_{sunny}([3,2]) = 0.971$

$\text{Mittelwert}: \frac{4}{14}\cdot 0 + \frac{5}{14}\cdot 0.971 + \frac{5}{14}\cdot 0.971 = 0.693$

$\text{Informationsgewinn} = 0.940 - 0.693 = 0.247$

temperature

$\text{info}_{cool}([3,1]) = 0.811$

$\text{info}_{mild}([4,2]) = 0.919$

$\text{info}_{warm}([2,2]) = 1$

$\text{Mittelwert} = 0.911$

$\text{Informationsgewinn} = 0.029$

humidity

$\text{info}_{high}([3,4]) = 0.985$

$\text{info}_{normal}([6,1]) = 0.592$

$\text{Mittelwert} = 0.789$

$\text{Informationsgewinn} = 0.152$

windy

$\text{info}_{true}([2,6]) = 0.811$

$\text{info}_{false}([3,3]) = 1$

$\text{Mittelwert} = 0.892$

$\text{Informationsgewinn} = 0.048$

outlook = sunny

temperature

$\text{info}_{cool}([1]) = 0$

$\text{info}_{mild}([1,1]) = 1$

$\text{info}_{hot}([2]) = 0$

$\text{Mittelwert} = 0.400$

$\text{Informationsgewinn} = 0.971 - 0.400 = 0.571$

humidity

$\text{info}_{high}([2]) = 0$

$\text{info}_{normal}([3]) = 0$

$\text{Mittelwert} = 0$

$\text{Informationsgewinn} = 0.971$

windy

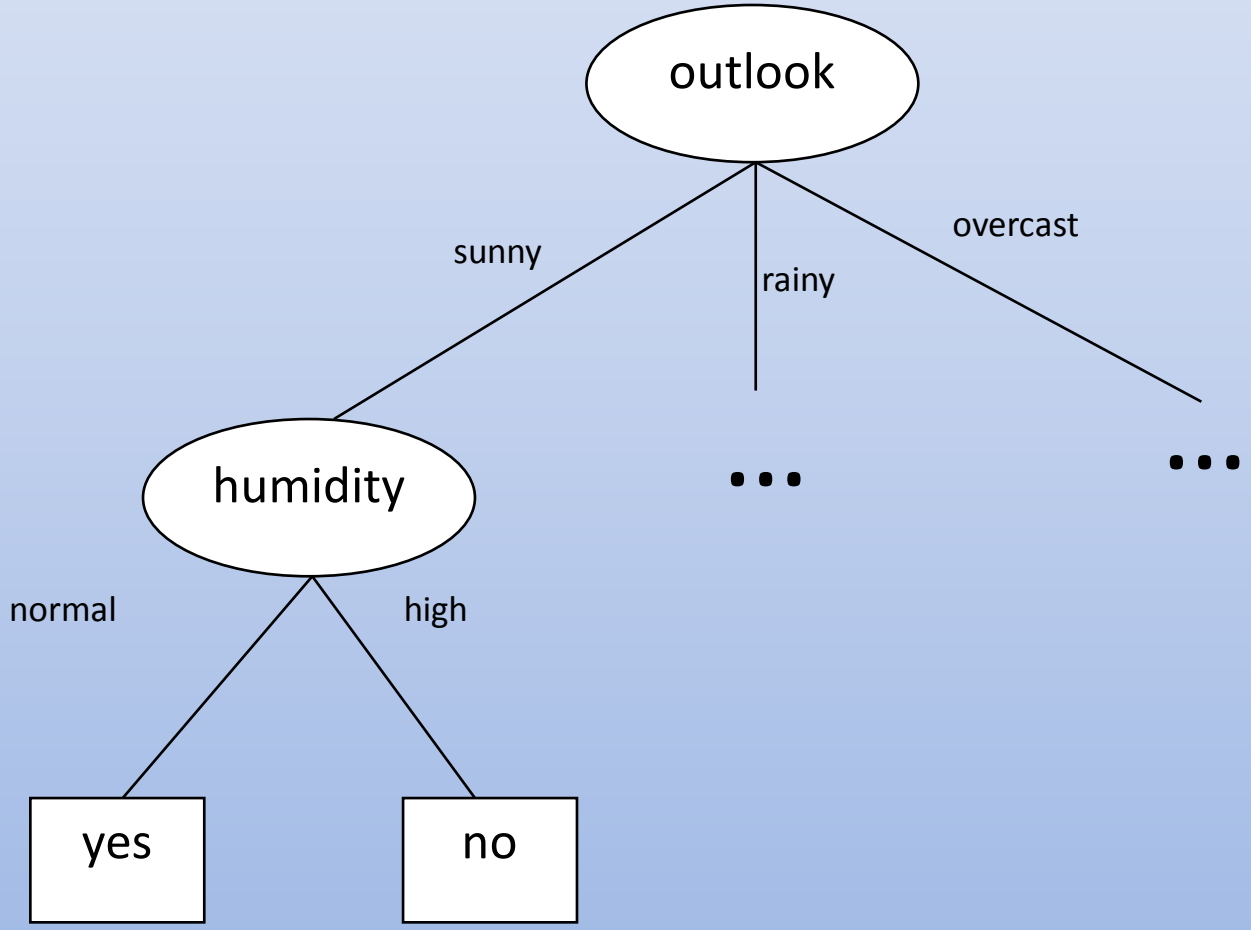
$\text{info}_{true}([1,1]) = 1$

$\text{info}_{false}([1,2]) = 0.918$

$\text{Mittelwert} = 0.951$

$\text{Informationsgewinn} = 0.020$

Beispiel - Wetterdaten



fehlende Attributwerte

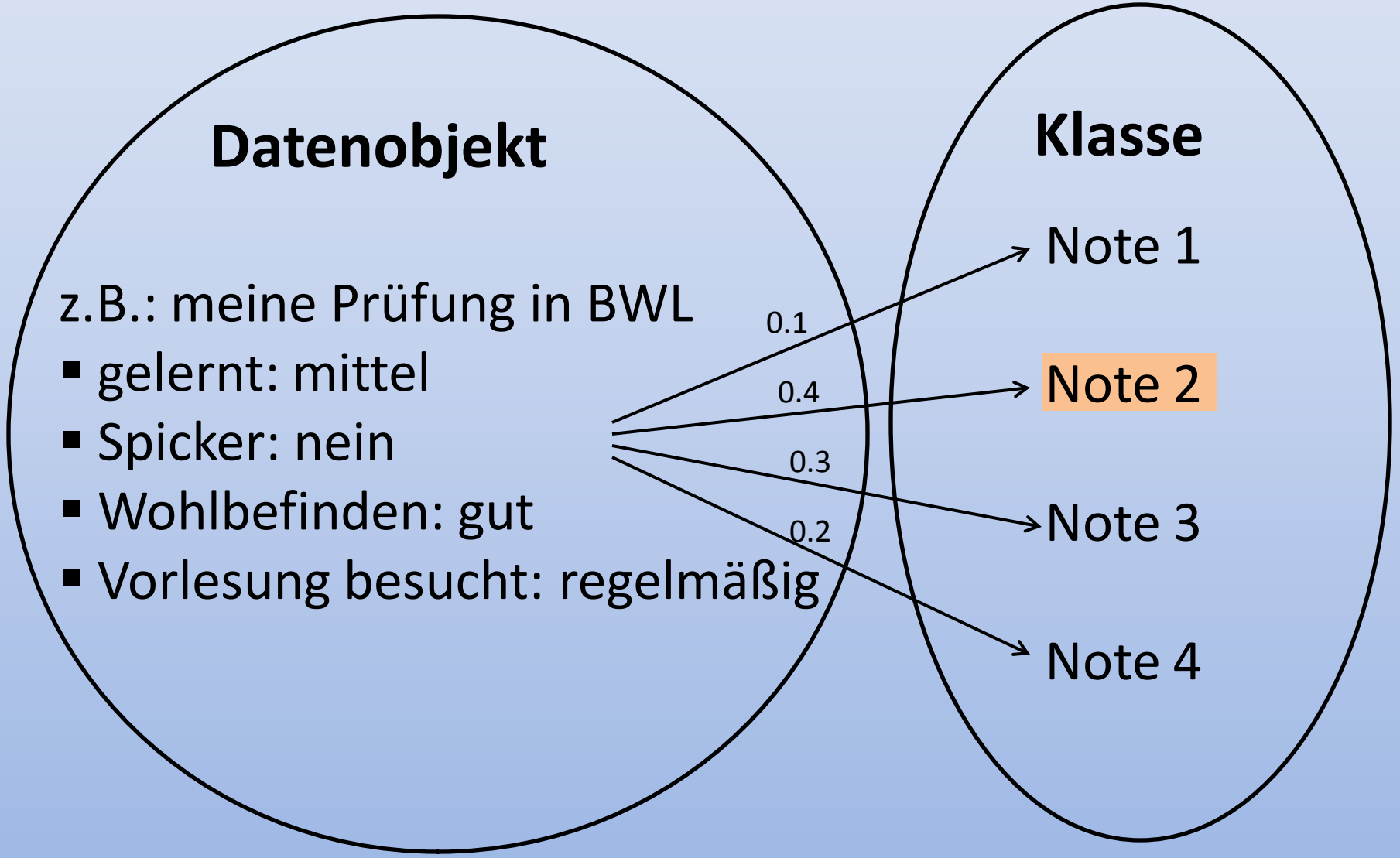
- fehlende Attributwerte stellen einen eigenen Zweig dar
- die in der Trainingsmenge gebräuchlichste Verzweigung wird gewählt
- jede Verzweigung des fehlenden Attributwertes wird durchlaufen, die Blätter werden anhand der Anzahl der Elemente der Trainingsinstanz numerisch gewichtet (sehr komplex)

Attribute mit vielen Verzweigungen

- Mehrfachverzweigung mit sehr vielen untergeordneten Knoten
- z.B. numerische Attribute: Wertebereiche für numerische Attribute
- IDs: höchster Informationsgewinn -> Verzweigung nach diesem Attribut ?
- statt Informationsgewinn wird das Gewinnverhältnis verwendet (siehe Literatur)

naive Bayes - Verfahren

- stochastisches Klassifizierungsmodell



formal:

n – #Klassen

m – #Merkmale/Attribute

$$Klassenzuordnung = index\{\max_{1 \le i \le n}(P(C_i \mid X))\}$$

C_i – Klasse i

X - Datenobjekt

[Los02]

$$P(C_i) - a - priori - Wahrscheinlichkeit$$

$$P(C_i \mid X) - a - posteriori - Wahrscheinlichkeit$$

$$P(C_i \mid X) = \frac{P(X \mid C_i) \cdot P(C_i)}{P(X)}$$

$$P(X \mid C_i) = \prod_{j=1}^{m} P(x_j \mid C_i)$$

- für kontinuierliche Merkmale ist eine Verteilungsfunktion nötig

Beispiel Wetter

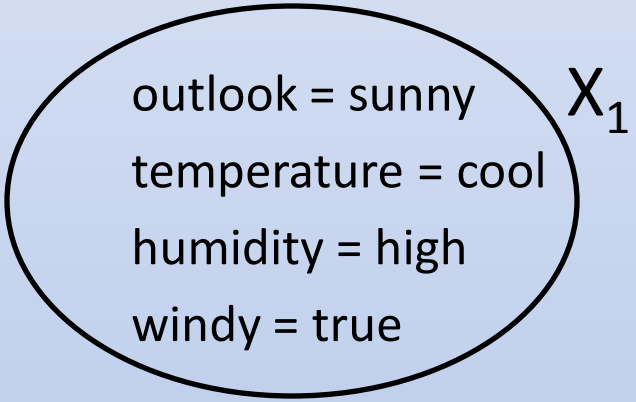
Beispiel Wetter

[Wit01]

Grundlage der Klassifikation

	Anzahl		relative Häufigkeit	
	P(yes)	P(no)	$P(x_j \mid yes)$	$P(x_j \mid no)$
	yes	no	yes	no
play	9	5	9/14	5/14
outlook				
sunny	2	3	2/9	3/5
overcast	4	0	4/9	0/5
rainy	3	2	3/9	2/5
temperature				
hot	2	2	2/9	2/5
mild	4	2	4/9	2/5
cool	3	1	3/9	1/5
humidity				
high	3	4	3/9	4/5
normal	6	1	6/9	1/5
windy				
false	6	2	6/9	2/5
true	3	3	3/9	3/5

Klassifikation



$$index\{\max_{1 \le i \le n}(P(C_i \mid X))\}$$

$$P(C_i \mid X) = \frac{P(X \mid C_i) \cdot P(C_i)}{P(X)}$$

[Wit01]

Klassifikation

$$P(yes \mid X_1) = \frac{\frac{2}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{9}{14}}{P(X)} = \frac{0.0053}{0.0053 + 0.0206} = 0.205$$

$$P(no \mid X_1) = \frac{\frac{3}{5} \cdot \frac{1}{5} \cdot \frac{4}{5} \cdot \frac{3}{5} \cdot \frac{5}{14}}{P(X)} = \frac{0.0206}{0.0053 + 0.0206} = 0.795$$

-> Zuordnung zu Klasse ‚no‘

[Wit01]

Problem

Attributwert in den Trainingsdaten gibt es nicht in Kombination mit einer Klasse (siehe outlook=overcast) -> Laplace-Schätzung

	Anzahl		relative Häufigkeit		Laplace Schätzung	
	P(yes)	P(no)	$P(x_j \mid yes)$	$P(x_j \mid no)$		
	yes	no	yes	no	yes	no
outlook						
sunny	2	3	2/9	3/5	3/12	4/8
overcast	4	0	4/9	0/5	5/12	1/8
rainy	3	2	3/9	2/5	4/12	3/8

[Wit01]

fehlende Werte

fehlende Wert stellen keine Probleme dar

-> Trainingsdaten: Instanz wird bei der Häufigkeitszählung für die Attributwert-Klassenkombination nicht berücksichtig

-> Klassifikation: Attribut wird ausgelassen

[Wit01]

numerische Attribute

nominale Werte

temperature	P(yes)	P(no)	$P(x_j \mid yes)$	$P(x_j \mid no)$
hot	2	2	2/9	2/5
mild	4	2	4/9	2/5
cool	3	1	3/9	1/5

numerische Werte

temperature	yes	no
nummerische Attributwerte	83	85
	70	80
	68	65
	64	72
	69	71
	75	
	75	
	72	
	81	
Mittelwert µ	73	74,6
Standardab-weichung σ	6,2	7,9

$$\sigma = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2$$

[Wit01]

numerische Attribute

temperature	yes	no
nummerische Attributwerte	83	85
	70	80
	68	65
	64	72
	69	71
	75	
	75	
	72	
	81	
Mittelwert µ	73	74,6
Standardab-weichung σ	6,2	7,9

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

rein nominale Attribute

P(temperature = mild | yes)

$$P(yes \mid X_1) = \frac{\frac{2}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{9}{14}}{P(X)}$$

f(temperature = 66| yes) = 0.034

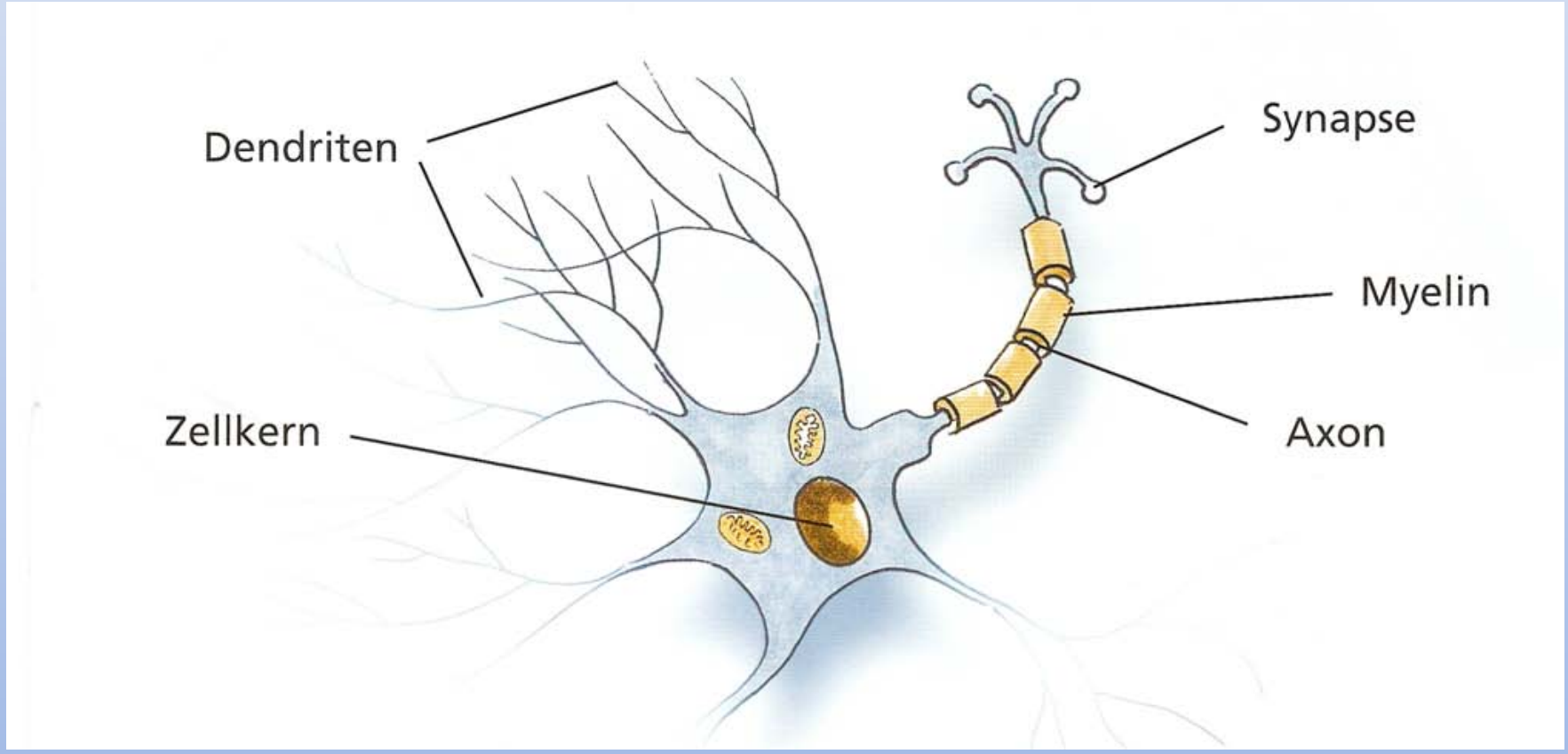
$$P(yes \mid X_1) = \frac{\frac{2}{9} \cdot 0.034 \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{9}{14}}{P(X)}$$

Künstliche Neuronale Netze

- KNN

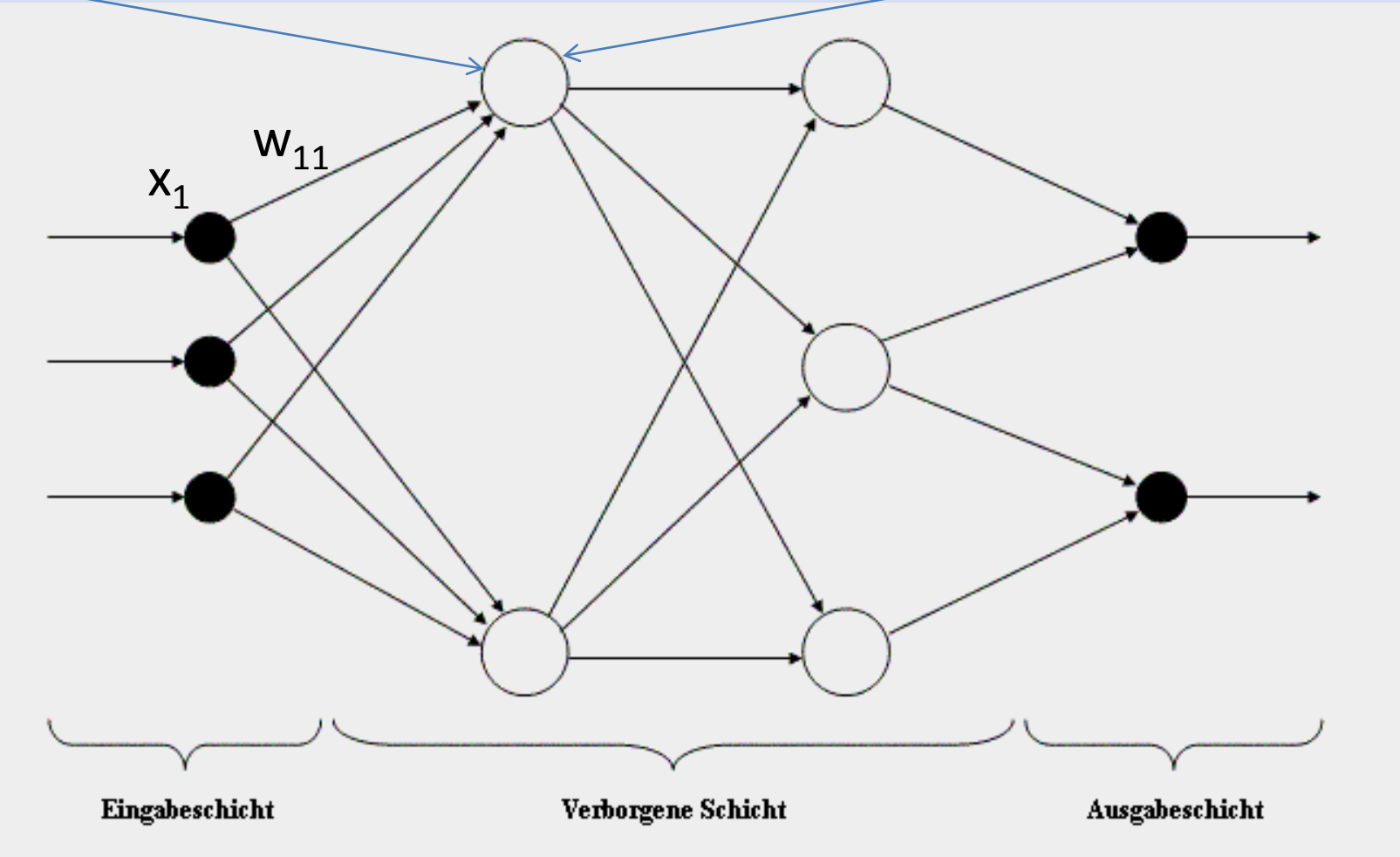
[google – Bildersuche]

Neuron (Nervenzelle)



Aufbau eines KNN

$f_{Eingabe} = \sum_{i=1}^{n} x_i w_{ij}$ n-# Neuronen der Vorgängerschicht, j - Neuron der Schicht $f_{Aktivierung}$



Aktivierungsfunktion

- Schwellwert bzw. Sprungfunktion
- lineare Funktion
- Sigmoidfunktion

Backpropagation-Algorithmus

- Trainingsdaten werden in das Netz gespeist
- die Differenz zwischen Netzausgabe und gewünschten Ausgabe wird als Fehler berechnet
- liegt der Fehler über der Toleranzgrenze wird er zurück propagiert -> die Gewichtungen der Neuronenverknüpfungen werden abhängig vom Einfluss auf den Fehler angepasst
- Vorgang wiederholt sich bis der Fehler innerhalb der Toleranzgrenze liegt
- Backpropagation stellte ein Näherungsverfahren dar

k – nearest neighbour

Idee

- Einordnung der Datenobjekt in gleiche Klasse wie ähnliche Trainingsobjekte
- Beschreibung einer Klasse durch ein Trainingsobjekt meist unzureichend
- Klassenzentrum bezeichnet ein Trainingsobjekt
- Verwendung einer Diskriminanzfunktion
- Berechnung des Abstands zu jedem Klassenzentrum zum Zeitpunkt der Klassifikation

- nearest neighbour

[Sch09]

Diskriminanzfunktion

$J(i)$ – Menge der Klassenzentren für Klasse i

$z_i(j)$ – j-te Zentrum der Klasse i

x – *Datenobjekt*

m – *# Klassen*

$$\mathrm{d_i}(x) = \min_{1 \le j \le J(i)} \{ z_i^T(j) \cdot z_i(j) - 2 \cdot z_i^T(j) \cdot x \}$$

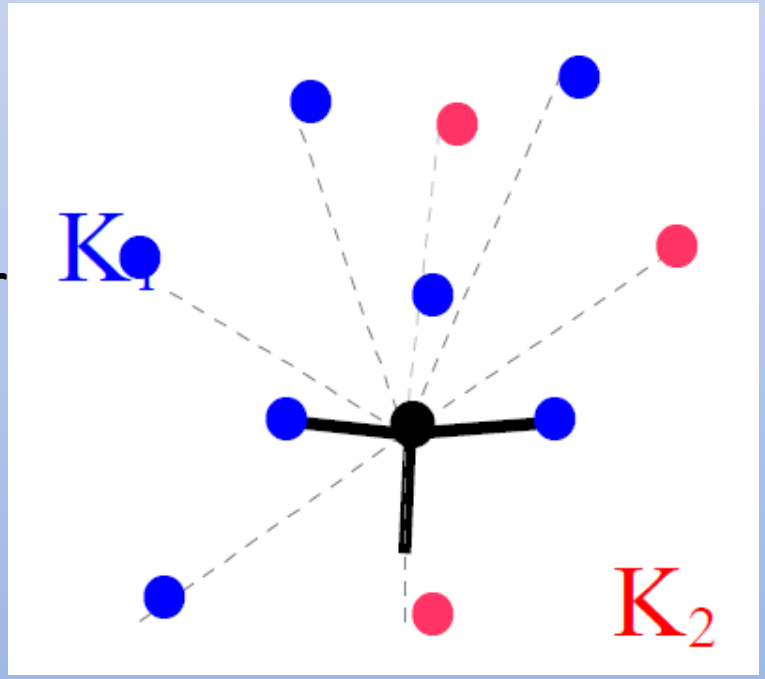
$$Klassenindex = index\{\min_{1 \le i \le m} d_i(x)\}$$

NN -> kNN

- schwächere Berücksichtigung von ‚Ausreißern‘ wünschenswert -> Berechnung aller Werte

$$d_{ij}(x) = z_i^T(j) \cdot z_i(j) - 2 \cdot z_i^T(j) \cdot x$$

- Klassenzuordnung erfolgt nach Bestimmung der k-kleinsten $d_{ij}(x)$
- um so größer k desto schwächer der Einfluss von Ausreißern
- Bsp: k=3



Support-Vector-Machine

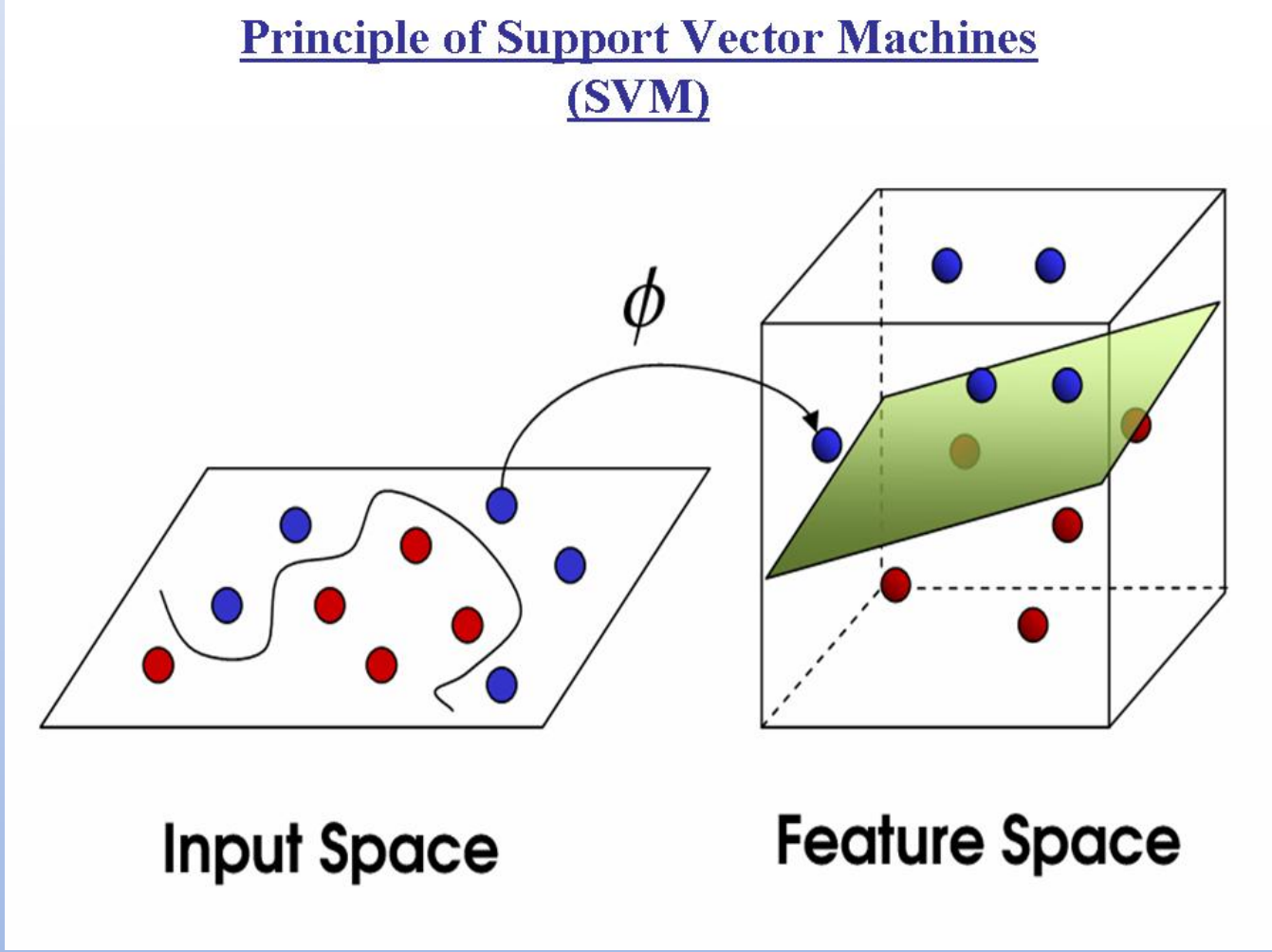
- Erweiterung der linearen Klassifikation
- benutzt Hyperebene als Trennfunktion
- es ist möglich auch nicht linear separierbare Datenobjekte durch eine Hyperebene von einander zu trennen -> Instanzraum wird in einen höher dimensionalen Raum abgebildet
- die Hyperebene wird maximal diskriminierende Hyperebene genannt

Maximal diskriminierende Hyperebene

- für die Objekte der beiden Klassen wird die konvexe Hülle gebildet
- die Ebene, die am weitesten von allen Objekten entfernt ist, heißt maximal diskriminierende Hyperebene
- liegt orthogonal zu den kürzesten Linien, die die beiden konvexen Hüllen verbindet
- die Objekte mit dem geringsten Abstand zu der maximal diskriminierenden Hyperebene werden support vectors / Stützvektoren genannt

[www.imtech.res.in/raghava/rbpred/svm.jpg]

Nichtlineare Klassengrenzen



Hauptkomponentenanalyse
Principal Component Analysis

Problem

- der Datensatz bezogen auf die Anzahl der Attribute ist meist zu groß
- wähle aus n Attributen die k ‚wesentlichen' Attribute aus
- Idee: Transformation in einen Sekundärraum mit k Attributen
- Methode: Karhunen-Loeve-Transformation

[Sch09]

Karhunen-Loeve-Transformation

X – Datenobjekt mit n-Attributen

A – $n^x n$ Orthonormalmatrix mit $a_1,\ldots,a_n$ als Spaltenverktoren

Cov – Kovarianzmatrix zu X

$X = A\ Y \rightarrow Y = A^{-1}\ X = A^T\ X$

Beispiel

[Sch09]

$$Cov = \begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix}$$

$$\text{charakteristische Polynom}: \begin{vmatrix} 4-\lambda & 1 \\ 1 & 4-\lambda \end{vmatrix} = \lambda^2 - 8\lambda + 15 \rightarrow \begin{matrix} \lambda_1 = 5 \\ \lambda_2 = 3 \end{matrix}$$

$$\text{Eigenvektoren}: Cov \cdot x = \lambda_i x \rightarrow \begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \lambda_i \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\lambda_1 : x_1 = x_2 \rightarrow e_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \xrightarrow{normiert} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$\longrightarrow$$

$$\lambda_2 : x_1 = -x_2 \rightarrow e_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \xrightarrow{normiert} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix}$$

$$A = (e_1 \quad e_2) = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

$$\text{Primärvektor } x = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$y = A^T x = \begin{pmatrix} 0 \\ \sqrt{2} \end{pmatrix} \rightarrow \text{Attribut im Sekundärraum } a = 1$$

Umsetzung in Software

- Software

	Naive Bayes	Entscheidungsbaum	KNN	SVM	k-NN	Quelle
MS SQL Server 2008	x	x	x			msdn
Salford Systems - CART 6.0 ProEX		x	x	x		salford-systems.com
KNIME	x	x	x	x		www.knime.org
RapidMiner		x	x	x		rapid-i.com
Auton Lab - Auton Web					x	www.autonlab.org/autonweb

Vor-/Nachteile der Algorithmen

Entscheidungsbaum

- Regel leicht ersichtlich, aber der Gesamtüberblick schwierig (Baum zu tief) -> Pruning (siehe Literatur)
- Attribute mit sehr viel möglichen Werten ergeben Mehrfachverzweigungen
- zur Klassifikation relevante Attribute befinden sich in Wurzelnähe

naive Bayes

- arbeitet auf großen Datenmengen gut, mit hoher Genauigkeit
- nimmt Unabhängigkeit der Attribute an;

  wenn fälschlicher Weise -> unzureichende Ergebnisse
- Geschwindigkeit ähnlich wie Entscheidungsbaum, KNN
- kontinuierliche Daten -> als Verteilungsfkt. wird meist fälschlicher Weise Normalverteilung vorausgesetzt

KNN

- tolerant gegenüber Ausreißern
- Attributkombinationen die nicht in der Trainingsmenge vorkommen, stellen kein Problem dar
- erlernte Gewichte sind kaum interpretierbar -> Klassifikationsergebnis nicht erklärbar
- lange Trainingsphase
- kategorielle Daten müssen in metrische Daten umgewandelt werden
- Netz ist individuell bezüglich Anzahl verborgener Schichten und Anzahl Neuronen zu erstellen

k - nearst neighbour

- für metrische und kategorielle Daten geeignet (durch Anpassung der Distanzfunktion)
- keine Lernphase, Daten werden zum Zeitpunkt der Klassifikation ausgewertet, aber gesamte Trainingsmenge muss für die Klassifikation eines Objektes durchlaufen werden
- Aufwand steigt mit steigenden k

SVM

- schnelle Klassifikation, weil nur Stützvektoren erforderlich
- effektiv auf Daten mit vielen Attributen
- finden der maximal diskriminierenden Hyperebene langwierig

Quellen

[Los02] Loss, Dirk: Data Mining: Klassifikations- und Clusteringverfahren, 2002, http://www.dirk-loss.de/DM-class-cluster_dloss.pdf

[Wit01] Witten, Ian H., Frank, Eibe: Data Mining – Praktische Werkzeuge und Techniken für das maschinelle Lernen, 2001, Hanser Verlag, ISBN: 3-446-21533-6

[Fuh05] Prof. Dr.-Ing. Fuhr, Norbert: Informationssysteme, 2005, www.is.informatik.uni-duisburg.de/courses/dm_ws05, Grundlage: Witten & Frank

[Wer00] Prof. Dr. Werner, Heinrich: der Forschungsgruppe Neuronale Netzwerke der Universität Kassel angehörig, Neuronale Netzwerke – Vorlesungsmaterial, 2000

[Sch09] Prof. Dr. Schönherr, Siegfried: HTWK Leipzig, Mustererkennung – Vorlesungsmaterial, 2009

outlook	temperature	humidity	windy	play
overcast	cool	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	high	false	yes
overcast	hot	normal	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
rainy	mild	high	false	yes
rainy	mild	normal	false	yes
rainy	mild	high	true	no
sunny	cool	normal	false	yes
sunny	mild	normal	true	yes
sunny	mild	high	false	no
sunny	hot	high	false	no
sunny	hot	high	true	no