Kapitel 3: Automatische Klassifikation von Dokumenten

Automatische Klassifikation von Dokumenten

Ziel:
Organisation von Dokumenten in (hierarchischen) Ontologien mit möglichst geringem intellektuellem Aufwand

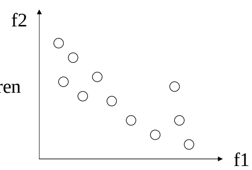
Techniken:
- Klassifikation mit Training (Supervised Learning)
  - kNN-Verfahren
  - Rocchio-Verfahren
  - Naives Bayes-Verfahren
  - Support Vector Machines (SVM)

  ...
- Klassifikation ohne Training (Unsupervised Learning)
  - Verfahren der Clusteranalyse

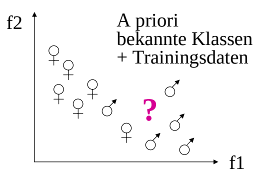
Orthogonal dazu gibt es flache vs. hierarchische Verfahren

Klassifikationsproblem (Kategorisierung)

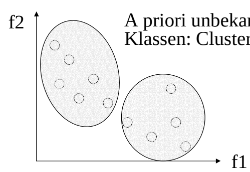
gegeben:
Featurevektoren



bestimme
Klassenzugehörigkeit
von Feature-Vektoren



A priori
bekannte Klassen
+ Trainingsdaten



A priori unbekannte
Klassen: Clustering

Anwendung von Klassifikationsverfahren im IR

- *Filtern:* teste eintreffende Dokumente (z.B. Mail, News), ob sie in eine interessante Klasse fallen
- *Übersicht:* organisiere Query-/Crawler-Resultate, Verzeichnisse, Feeds, etc.
- *Query-Expansion:* ordne Query einer Klasse zu und ergänze dementsprechende Suchterme
- *Relevanz-Feedback:* klassifiziere Treffer und lasse Benutzer relevante Klassen identifizieren, um bessere Query zu generieren
- *Query-Effizienz:* beschränke (Index-)Suche auf relevante Klasse(n)

Klassifikationsvarianten:
- mit Termen, Termhäufigkeiten, Linkstruktur als Features
- binär: Gehört ein Dokument zu einer Klasse c oder nicht?
- mehrstellig: In welche von k Klassen passt ein Dokument am besten?
- hierarchisch: Iteration der Klassifikation über ontologischen Baum

Bewertung der Klassifikationsgüte

empirisch durch automatische Klassifikation von Dokumenten, die nicht zu den Trainingsdaten gehören

Für binäre Klassifikation bzgl. Klasse C:
a = #Dok., die zu C klassifiziert wurden und zu C gehören
b = #Dok., die zu C klassifiziert wurden, aber nicht zu C gehören
c = #Dok., die nicht zu C klassifiziert wurden, aber zu C gehören
d = #Dok., die nicht zu C klassifiziert wurden und nicht zu C gehören

Genauigkeit (acccuracy) = $\frac{a+d}{a+b+c+d}$

Präzision (precision) = $\frac{a}{a+b}$ Ausbeute (recall) = $\frac{a}{a+c}$

Für mehrstellige Klassifikation bzgl. Klassen C1, ..., Ck:
- Makrodurchschnitt über k Klassen oder
- Mikrodurchschnitt über k Klassen

3.1 Einfache Klassifikatoren: k-Nearest-Neighbor-Verfahren (kNN)

Schritt 1:
Finde unter den Trainingsdokumenten aller Klassen die k (z.B. 10-100) bzgl. der (Cosinus-) Ähnlichkeit nächsten Nachbarn eines neuen Dokuments $\vec{d}$

Schritt 2:
Ordne $\vec{d}$ derjenigen Klasse Cj zu, für die die Funktion

$$f(\vec{d}, C_j) = \sum_{\vec{v} \in kNN(\vec{d})} sim(\vec{d}, \vec{v}) * \begin{cases} 1 & falls\ \vec{v} \in C_j \\ 0 & sonst \end{cases}$$

maximal wird

Bei binärer Klassifikation ordne $\vec{d}$ der Klasse C zu, falls $f(\vec{d}, C)$ über einem Schwellwert δ (δ >0.5) liegt.

Klassifikationsverfahren von Rocchio

Schritt 1:
Repräsentiere die Trainingsdokumente einer Klasse Cj
- mit tf*idf-Vektorkomponenten – durch den **Prototypvektor**:

$$\vec{c}_j := \alpha \frac{1}{|C_j|} \sum_{\vec{d} \in C_j} \frac{\vec{d}}{\|\vec{d}\|} - \beta \frac{1}{|D - C_j|} \sum_{\vec{d} \in D - C_j} \frac{\vec{d}}{\|\vec{d}\|}$$

mit geeigneten Koeffizienten α und β (z.B. α=16, β=4)

Schritt 2:
Ordne ein neues Dokument $\vec{d}$ derjenigen Klasse Cj zu, für die Cosinus-Ähnlichkeit $cos(\vec{c}_j, \vec{d})$ maximal ist.

Satz:
Für α=β=1 maximiert $\vec{c}_j$ die Funktion:

$$f(\vec{c}_j) = \frac{1}{|C_j|} \sum_{\vec{d} \in C_j} \cos(\vec{c}_j, \vec{d}) - \frac{1}{|D - C_j|} \sum_{\vec{d} \in D - C_j} \cos(\vec{c}_j, \vec{d})$$

3.2 Grundlagen aus der Wahrscheinlichkeitsrechnung

Ein **Wahrscheinlichkeitsraum** ist ein Tripel (Ω, E, P) mit
- einer Menge Ω elementarer Ereignisse,
- einer Familie E von Teilmengen von Ω mit Ω∈E, die unter ∩, ∪ und – mit abzählbar vielen Operanden abgeschlossen ist (bei endlichem Ω ist in der Regel $E=2^{\Omega}$), und
- einem W.maß P: E → [0,1] mit P[Ω]=1 und $P[\cup_i Ai] = \sum_i P[Ai]$ für abzählbar viele, paarweise disjunkte Ai

Eigenschaften von P:

P[A] + P[¬A] = 1   P[∅] = 0

P[A ∪ B] = P[A] + P[B] – P[A ∩ B]   P[Ω] = 1

Zufallsvariable

Eine **Zufallsvariable** X über einem W.raum (Ω, E, P) ist eine Funktion X: Ω → M mit M⊆R, so daß {e | X(e) ≤ x} ∈E für alle x ∈M.

F_X: M → [0,1] mit $F_X(x)$ = P[X ≤ x] heißt ***Verteilungsfunktion*** von X; bei abzählbarer Menge M heißt f_X: M → [0,1] mit $f_X(x)$ = P[X = x] ***Dichtefunktion*** von X, ansonsten ist $f_X(x)$ durch $F'_X(x)$ gegeben.

Momente

Für eine diskrete Zufallsvariable X mit Dichtefunktion f_X sind

$E[X] = \sum_{k \in M} k\, f_X(k)$ der ***Erwartungswert*** von X

$E[X^i] = \sum_{k \in M} k^i\, f_X(k)$ das ***i. Moment*** von X

$V[X] = E[(X - E[X])^2] = E[X^2] - E[X]^2$ die ***Varianz*** von X

Für eine kontinuierliche Zufallsvariable X mit Dichtefunktion f_X sind

$E[X] = \int_{-\infty}^{+\infty} x\, f_X(x)\, dx$ der ***Erwartungswert*** von X

$E[X^i] = \int_{-\infty}^{+\infty} x^i\, f_X(x)\, dx$ das ***i. Moment*** von X

$V[X] = E[(X - E[X])^2] = E[X^2] - E[X]^2$ die ***Varianz*** von X

Erwartungswerte sind additiv, $E[X+Y] = E[X] + E[Y]$
Verteilungen nicht

Wichtige diskrete Verteilungen

- Gleichverteilung über {1, 2, ..., m}:

$$P[X=k] = f_X(k) = \frac{1}{m} \quad for 1 \le k \le m$$

- Binomialverteilung (Münzwurf n-mal wiederholt; X: #Köpfe):

$$P[X=k] = f_X(k) = \binom{n}{k} p^k (1-p)^{n-k}$$

- Poisson-Verteilung (mit Rate λ):

$$P[X=k] = f_X(k) = e^{-\lambda} \frac{\lambda^k}{k!}$$

- Geometrische Verteilung (# Münzwürfe bis zum ersten Kopf):

$$P[X=k] = f_X(k) = (1-p)^k\, p$$

- 2-Poisson-Mix (mit $a_1+a_2=1$):

$$P[X=k] = f_X(k) = a_1 e^{-\lambda_1} \frac{\lambda_1^k}{k!} + a_2 e^{-\lambda_2} \frac{\lambda_2^k}{k!}$$

Kontinuierliche Verteilungen

- Gleichverteilung über dem Intervall [a,b]

$$f_X(x) = \frac{1}{b-a} \quad für\ a \le x \le b\ (0\ sonst)$$

- Exponentialverteilung (z.B. Zeit bis zum nächsten Ereignis eines Poisson-Prozesses) mit Rate $\lambda = \lim_{\Delta t \to 0}$ (# Ereignisse in Δt) / Δt :

$$f_X(x) = \lambda e^{-\lambda x} \quad für\ x \ge 0\ (0\ sonst)$$

- Hyperexponential-Verteilung: $f_X(x) = p\lambda_1 e^{-\lambda_1 x} + (1-p)\lambda_2 e^{-\lambda_2 x}$
- Pareto-Verteilung: $f_X(x) \to \frac{a}{b}\left(\frac{b}{x}\right)^{a+1}$ $für\ x > b,\ 0\ sonst$

Beispiel einer „Heavy-tailed“-Verteilung mit $f_X(x) \to \frac{c}{x^{\alpha+1}}$

Normalverteilung

• Normalverteilung N(μ,σ²) (Gauß-Verteilung; approximiert Summen unabhängiger, identisch verteilter Zufallsvariablen): $f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

• Verteilungsfunktion von N(0,1): $\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$

Sei X normalverteilt mit Erwartungswert μ und Varianz σ².
Dann ist $Y := \frac{X-\mu}{\sigma}$
normalverteilt mit Erwartungswert 0 und Varianz 1.

Zentraler Grenzwertsatz

Satz:
Seien X1, ..., Xn unabhängig und identisch verteilte Zufallsvariablen mit Erwartungswert μ und Varianz σ².
Die Verteilungsfunktion Fn der Zufallsvariablen Zn := X1 + ... + Xn konvergiert gegen eine Normalverteilung N(nμ, nσ²)
mit Erwartungswert nμ und Varianz nσ²:

$$lim_{n\to\infty} P[\, a \le \frac{Z_n - n\mu}{\sqrt{n}\sigma} \le b \,] = \Phi(b) - \Phi(a)$$

Korollar:
$\bar{X} := \frac{1}{n} \sum_{i=1}^{n} X_i$
konvergiert gegen eine Normalverteilung N(μ, σ²/n)
mit Erwartungswert nμ und Varianz σ²/n .

Satz von Bayes

Zwei Ereignisse A, B eines W.raums heißen **unabhängig**, wenn gilt P[A ∩ B] = P[A] P[B].

Die **bedingte Wahrscheinlichkeit P[A | B]** von A unter der Bedingung (Hypothese) B ist definiert als: $P[A|B] = \frac{P[A \cap B]}{P[B]}$

Satz von der totalen Wahrscheinlichkeit:
Für eine Partitionierung von Ω in Ereignisse B1, ..., Bn gilt:

$$P[A] = \sum_{i=1}^{n} P[A|Bi]\, P[Bi]$$

Satz von Bayes: $\underbrace{P[A|B]}_{\text{A-Posteriori-W. von A}} = \frac{P[B|A]\, P[A]}{P[B]}$

3.3 Naives Bayes-Verfahren mit binären Features Xi

Schätze: $P[\, d \in c_k | d \text{ hat } \vec{X} \,] = \frac{P[\, d \text{ hat } \vec{X} | d \in c_k \,]\, P[\, d \in c_k \,]}{P[\, d \text{ hat } \vec{X} \,]}$

$\sim P[\, X | d \in c_k \,]\, P[\, d \in c_k \,]$

$= \Pi_{i=1}^{m} P[\, X_i | d \in c_k \,]\, P[\, d \in c_k \,]$ bei Featureunabhängigkeit bzw. Linked Dependence: $\frac{P[X | d \in c_k]}{P[X | d \notin c_k]} = \Pi_i \frac{P[X_i | d \in c_k]}{P[X_i | d \notin c_k]}$

$= \Pi_{i=1}^{m} p_{ik}^{\;Xi} (1 - p_{ik})^{1-Xi}\, p_k$ mit empirisch zu schätzenden $p_{ik}=P[X_i=1|c_k]$, $p_k=P[c_k]$

$$\Rightarrow \log P[c_k | d] \sim \sum_{i=1}^{m} X_i \log \frac{p_{ik}}{(1-p_{ik})} + \sum_{i=1}^{m} \log(1-p_{ik}) + \log p_k$$

für binäre Klassifikation mit Quote $P[d \in c_k] / P[d \notin c_k]$ statt P[...] weitere Vereinfachung möglich

Naives Bayes-Verfahren mit Bag-of-Words-Modell

Schätze: $P[\, d \in c_k | d \text{ hat } \vec{f} \,] \sim P[\, \vec{f} | d \in c_k \,]\, P[\, d \in c_k \,]$

mit Termhäufigkeitsvektor $\vec{f}$

$= \Pi_{i=1}^{m} P[\, f_i | d \in c_k \,]\, P[\, d \in c_k \,]$ bei Featureunabhängigkeit

$= \Pi_{i=1}^{m} \binom{length(d)}{f_i} p_{ik}^{\;f_i} (1 - p_{ik})^{length(d) - f_i}\, p_k$

mit Binomialverteilung für jedes Feature

bzw. besser: $= \binom{length(d)}{f_1\, f_2 \ldots f_m} p_{1k}^{\;f_1}\, p_{2k}^{\;f_2} \ldots p_{mk}^{\;f_m}\, p_k$

mit Multinomialverteilung der Featurevektoren und $\sum_{i=1}^{m} f_i = length(d)$

mit $\binom{n}{k_1\, k_2 \ldots k_m} := \frac{n!}{k_1!\, k_2! \ldots k_m!}$

Beispiel für das naive Bayes-Verfahren (1)

3 Klassen: c1 – Algebra, c2 – Analysis, c3 – Stochastik
8 Terme, 6 Trainingsdokumente d1, ..., d6: je 2 in jeder Klasse

⇒ p1=2/6, p2=2/6, p3=2/6

	Gruppe f1	Homomorphismus f2	Vektor f3	Integral f4	Limes f5	Varianz f6	Wahrscheinlichkeit f7	Würfel f8
d1:	3	2	0	0	0	0	0	1
d2:	1	2	3	0	0	0	0	0
d3:	0	0	0	3	3	0	0	0
d4:	0	0	1	2	2	0	1	0
d5:	0	0	0	1	1	2	2	0
d6:	1	0	1	0	0	0	2	2

	Algebra k=1	Analysis k=2	Stochastik k=3
p1k	4/12	0	1/12
p2k	4/12	0	0
p3k	3/12	1/12	1/12
p4k	0	5/12	1/12
p5k	0	5/12	1/12
p6k	0	0	2/12
p7k	0	1/12	4/12
p8k	1/12	0	2/12

Beispiel für das naive Bayes-Verfahren (2)

Klassifikation von d7: (0 0 1 2 0 0 3 0)

$$P[\,\vec{f}\,|d \in c_k\,]\;P[\,d \in c_k\,] = \binom{length(d)}{f_1\ f_2 \cdots f_m}\, p_{1k}{}^{f_1}\, p_{2k}{}^{f_2} \cdots p_{mk}{}^{f_m}\;\; p_k$$

für k=1 (Algebra): $= \binom{6}{1\ 2\ 3}\left(\frac{3}{12}\right)^1 0^2\, 0^3\;\; \frac{2}{6} \qquad = 0$

für k=2 (Analysis): $= \binom{6}{1\ 2\ 3}\left(\frac{1}{12}\right)^1\left(\frac{5}{12}\right)^2\left(\frac{1}{12}\right)^3\;\frac{2}{6} \qquad = 20 * \frac{25}{12^6}$

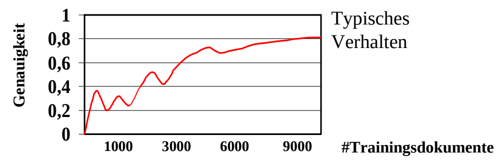
für k=3 (Stochastik): $= \binom{6}{1\ 2\ 3}\left(\frac{1}{12}\right)^1\left(\frac{1}{12}\right)^2\left(\frac{4}{12}\right)^3\;\frac{2}{6} \qquad = 20 * \frac{64}{12^6}$

Resultat: Ordne d7 der Klasse C3 (Stochastik) zu

Typisches Verhalten des naiven Bayes-Verfahrens

Reuters Benchmark (siehe trec.nist.gov):
12902 kurze Artikel (Wirtschaftsnachrichten)
aus 90 Kategorien (acq, corn, earn, grain, interest, money-fx, ship, ...)

- Verwende die (bzw. einen Teil der) ältesten 9603 Artikel zum Trainieren des Klassifikators
- Verwende die neuesten 3299 Artikel zur Evaluation der Klassifikationsgenauigkeit



max. Genauigkeit liegt je nach Kategorie zwischen 50 und 90 Prozent

Verbesserung des naiven Bayes-Verfahrens

1) geglättete Schätzung der p_{ik} durh Laplace-Smoothing:

$1/(m + \sum_{d \in C_k} \mathrm{length}(d))$ statt 0 für in den Trainingsdokumenten einer Klasse überhaupt nicht auftretende Features

2) Anreicherung des Trainingsmaterials durch unbenannte, automatisch klassifizierte Dokumente zur besseren Schätzung der p_{ik}

mit unterschiedlicher Gewichtung der intellektuell und der automatisch klassifizierten „Trainingsdokumente“

3) Berücksichtigung von Abhängigkeiten zwischen Features durch Verallgemeinerung auf Bayessche Netze

Erweiterung um Semisupervised Learning

Motivation:
- Klassifikator nur so gut wie seine Trainingsdaten
- Trainingsdaten teuer wegen intellektueller Klassifikation
- Trainingsdaten sind im Featureraum nur dünnbesetzt

→ Verwendung zusätzlicher nichtklassifizierter Daten zum impliziten Lernen von Korrelationen

Beispiel:
- Klassifikator für Thema „cars“ wurde auf Dokumenten trainiert, die „car“ enthalten, aber nicht „automobile“.
- In den nichtklassifizierten Daten sind „car“ und „automobile“ stark korreliert.
- Testdokumente enthalten „autombobile“, aber nicht „car“.

Simples Iteratives Labeling

Sei D^K die Menge der Dok. mit bekannten Klassen (Trainingsdaten) und sei D^U die Menge der Dok. mit unbekannten Klassen.

<u>Algorithm:</u>
train classifier with D^K as training data
classify docs in D^U
repeat
  re-train classifier with D^K and the now labeled docs in D^U
  classify docs in D^U
until labels do not change anymore (or changes are marginal)

<u>Robustheitsproblem:</u>
einige wenige Dokumente aus D^U können den Klassifikator zu einem „Drift“ verleiten
→ bessere, aber komplexere Iterationsverfahren basierend auf dem Expectation-Maximization-Verfahren für Parameterschätzung

3.4 Feature-Selektion

Zur Entscheidung zwischen Klassen einer Stufe werden geeignete Features ausgewählt
(aus Effizienzgründen und zur Vermeidung von Overfitting).

Beispiel:
Terme wie „Definition“, „Theorem“, „Lemma“ sind gute Diskriminatoren zwischen Arts, Entertainment, Science, etc.;
sie sind schlechte Diskriminatoren zwischen den Unterklassen von Mathematics wie z.B. Algebra, Stochastics, etc.

→ Betrachtung statistischer bzw. informationstheoretischer Maße zur Selektion geeigneter Features

Feature-Selektion auf der Basis der Mutual Information (MI)

Mutual Information (Relative Entropie, Kullback-Leibler-Distanz):
Zur Entscheidung für Klasse cj wähle diejenigen binären Features Xi (Termvorkommen) mit dem größten Wert von

$$MI(X_i, c_j) = P[X_i \wedge c_j] \; log \frac{P[X_i \wedge c_j]}{P[X_i] \, P[c_j]}$$

oder

$$MI(X_i, c_j) = \sum_{X \in \{X_i, \bar{X}_i\}} \sum_{C \in \{c_j, \bar{c}_j\}} P[X \wedge C] \; log \frac{P[X \wedge C]}{P[X] \, P[C]}$$

und für die Entscheidung bzgl. Klassen c1, ..., ck:

$$MI(X_i) = \sum_{j=1}^{k} P[c_j] \, MI(X_i, c_j)$$

Berechnung in Zeit O(n)+O(mk)
für n Trainingsdokumente,
m Terme und k Klassen

Beispiel für Feature-Selektion

	Film f1	Hit f2	Chart f3	Theorem f4	Limes f5	Integral f6	Gruppe f7	Vektor f8
d1:	1	1	0	0	0	0	0	0
d2:	0	1	1	0	0	0	1	0
d3:	1	0	1	0	0	0	0	0
d4:	0	1	1	0	0	0	0	0
d5:	0	0	0	1	1	1	0	0
d6:	0	0	0	1	0	1	0	0
d7:	0	0	0	0	1	0	0	0
d8:	0	0	0	1	0	1	0	0
d9:	0	0	0	0	0	0	1	1
d10:	0	0	0	1	0	0	1	1
d11:	0	0	0	1	0	1	0	1
d12:	0	0	1	1	1	0	1	0

Klassenbaum:

- Unterhaltung
- Mathematik
  - Analysis
  - Algebra

Trainingsdokumente:
d1, d2, d3, d4
→ Unterhaltung
d5, d6, d7, d8
→ Analysis
d9, d10, d11, d12
→ Algebra

Beispielrechnung für Feature-Selektion auf der Basis des MI-Maßes

Unterhaltung (d1-d4) vs. Mathematik (d5-d12):

MI(Film) = 2/12 log [2/12 / (2/12 * 1/3)] + 0 log 0 +
2/12 log [2/12 / (2/12 * 1/3)] + 8/12 log[8/12 / (10/12 * 2/3)]

MI(Chart) = 3/12 log [3/12 / (4/12 * 1/3)] + 1/12 log [1/12 / (4/12 * 2/3)] +
1/12 log [1/12 / (8/12 * 1/3)] + 7/12 log[7/12 / (8/12 * 2/3)]

MI(Theorem) = 0 log 0 + 6/12 log [6/12 / (6/12 * 2/3)] +
4/12 log [4/12 / (6/12 * 1/3)] + 2/12 log[2/12 / (6/12 * 2/3)]

Analysis (d5-d8) vs. Algebra (d9-d12):

MI(Film) = 0 log 0 + 0 log 0 +
4/8 log [4/8 / (8/8 * 1/2)] + 4/8 log[4/8 / (8/8 * 1/2)]

MI(Theorem) = 3/8 log [3/8 / (6/8 * 1/2)] + 3/8 log [3/8 / (6/8 * 1/2)] +
1/8 log [1/8 / (2/8 * 1/2)] + 1/8 log[1/8 / (2/8 * 1/2)]

MI(Vektor) = 0 log 0 + 3/8 log [3/8 / (3/8 * 1/2)] +
4/8 log [4/8 / (5/8 * 1/2)] + 1/8 log[1/8 / (5/8 * 1/2)]