Kapitel 3: Automatische Klassifikation von Dokumenten

3.1 Einfache Klassifikatoren

In bestimmten Anwendungen möchte man Dokumente automatisch klassifizieren, also aufgrund ihrer Featurevektoren bestimmten Themen (Klassen, Kategorien) zuordnen. Wenn man von einigen Trainingsdokumenten d1, ..., dn die Klassenzuordnung c(di) $\in$ {C1, ..., Ck} a priori kennt, weil diese Dokumente intellektuell klassifiziert worden sind, kann man weitere Dokumente mit a priori unbekannter Klasse durch statistische Ähnlichkeit mit den Trainingsdaten der verschiedenen Klassen mit einer bestimmten Wahrscheinlichkeit einer Klasse zuordnen. Beispielsweise kann man Dokumente, in denen die Wörter Theorem und Homomorphismus mit hohem (tf*idf-) Gewicht auftreten, mit hoher Wahrscheinlichkeit der Klasse Mathematik zuordnen, während bei Dokumenten mit hoch gewichteten Termen Tor, Elfmeter und Schiedsrichter vieles für die Klasse Sport spricht. Verfahren dieser Art gehören zur Familie des *Supervised Learning*, also der *Lernverfahren mit Trainingsdaten*. Wenn man keinerlei Trainingsdaten hat, kann man Verfahren des *Unsupervised Learning* anwenden, insbesondere die statistischen Verfahren der *Cluster-Analyse*. Im folgenden wird nur Klassifikation mit Trainingsdaten betrachtet.

Gegeben sind n Trainingsdokumente d1, ..., dn $\in [0,1]^m$ über einem m-dimensionalen Featureraum mit bekannten Klassen c(di) $\in$ {C1, ..., Ck} aus einer Menge von k verschiedenen Themen. Ein Klassifikator ist eine Funktion c: $[0,1]^m \rightarrow$ {C1, ..., Ck}.

Automatische Klassifikation ist in IRS-Anwendungen auf vielfältige Weise nützlich:
- *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)

Gütemaße für Klassifikatoren

Zur Bewertung der Güte eines Klassifikators für eine Klasse C stellt man folgende Kontingenztafel auf:

	#Dokumente d mit c(d)=C	#Dokumente d mit c(d)≠C
#Dokumente $\in$ C	a	c
#Dokumente $\notin$ C	b	d

Dann sind die folgenden Gütemaße wie folgt definiert:
Präzision (Precision) = a / (a+b)
Ausbeute (Ausbeute) = a / (a+c)

Genauigkeit (Accuracy) = (a+d) / (a+b+c+d)
Fehler (Error) = 1 – Genauigkeit

Für einen Klassifikator mit mehr als 2 Klassen, berechnet Durchschnittswerte über alle Klassen. Dabei unterscheidet man Makrodurchschnitte (Macro Averages), bei denen direkt die Werte des jeweiligen Gütemaßes gemittelt werden (z.B. die Präzisionswerte), und Mikrodurchschnitte (Micro Averages), bei denen für das jeweilige Gütemaß die Zähler und Nenner separat über alle Klassen aufsummiert werden.

Zur experimentellen Bestimmung der Güte eines Klassifikators werden Benchmark-Daten mit bekannten Klassen verwendet. Man unterteilt die Daten in p Partitionen, trainiert auf einer Partition und testet die Klassifikatorgüte auf den anderen p-1 Partitionen. Diese Schema wird durch Wahl der Trainingspartition systematisch variiert, und der Mittelwert für alle p Möglichkeiten ist die experimentelle Güte. Dieses Prinzip nennt man Kreuzvalidierung (Cross Validation). Ein Sonderfall ist die Leave-one-out-Validierung, bei der man 2 unterschiedliche große Partitionen wählt, eine Trainingspartition, die alle Dokumente außer einem enthält, und eine Testpartition, die genau ein Dokument enthält.

kNN-Klassifikator

Der einfachste Klassifikator ist das sog. k-Nearest-Neighbor-Verfahren, kurz kNN. Es bestimmt zu einem zu klassifizierenden Dokument zunächst die k nächsten Nachbarn unter den Trainingsdaten gemäß einer Ähnlichkeitsfunktion über dem zugrundeliegenden Featureraum, z.B. der Cosinus-Ähnlichkeit. Dann ermittelt das Verfahren die Klassen dieser k nächsten Nachbarn und ordnet schließlich das neue Dokument der am häufigsten auftretenden Klasse zu. Beim letzten Schritt können die Klassenhäufigkeiten mit den Abständen der entsprechenden Trainingsdokumente zu dem neuen Dokument gewichtet werden.

Ordne d derjenigen Klasse Cj zu für die

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

maximal ist.
Falls man nur binär klassifiziert, also nur testen will, ob ein Dokument zu einem gegebenen Thema passt oder nicht, ordnet man d zu, wenn $f(\vec{d},C_j)$ einen bestimmten Schwellwert δ überschreitet (z.B. δ=0.5).

Klassifikator nach 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 d derjenigen Klasse C_j zu, für
die Cosinus-Ähnlichkeit $\cos(d,c_j)$ maximal ist.

Satz:

Für $\alpha=\beta=1$ maximiert cj 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 W elementarer Ereignisse,
- einer Familie E von Teilmengen von Ω mit $\Omega \in E$, die unter $\cup$, $\cap$ und - mit abzählbar vielen Operanden abgeschlossen ist (bei endlichem Ω ist in der Regel $E=2^{\Omega}$), und
- einem Wahrscheinlichkeitsmaß P: E $\rightarrow$ [0,1] mit $P[\Omega]=1$ und $P[\cup_i Ai] = \Sigma_i P[Ai]$ für abzählbar viele, paarweise disjunkte Ai

Eigenschaften von P:
$P[A] + P[\neg A] = 1$
$P[\varnothing] = 0$
$P[A \cup B] = P[A] + P[B] - P[A \cap B]$
$P[\Omega] = 1$

Eine **Zufallsvariable** X über einem Wahrscheinlichkeitsraum (W, E, P) ist eine Funktion X: $\Omega \rightarrow$ M mit M $\subseteq$ R, so daß $\{e \mid X(e) \leq x\} \in E$ für alle $x \in M$.

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

Einige wichtige Wahrscheinlichkeitsverteilungen sind:

- Diskrete Gleichverteilung über $M=\{x \mid x \in N \wedge a \leq x \leq b\} \subseteq N$ mit der Dichte
  $$P[X = k] = f_X(k) = \frac{1}{b-a+1} \quad \text{für } a \leq k \leq b,\ 0 \text{ sonst}$$
- Diskrete Bernoulli-Verteilung über $M=\{0,1\}$ mit der Dichte $f_X(k) = \begin{cases} p & \text{für } k=1 \\ 1-p & \text{für } k=0 \end{cases}$
- Diskrete Binomialverteilung über $M=\{x \mid x \in N \wedge 0 \leq x \leq n\} \subseteq N$ mit der Dichte
  $$P[X = k] = f_X(k) = \binom{n}{k} p^k (1-p)^{n-k}$$
- Diskrete Poisson-Verteilung über den natürlichen Zahlen mit der Dichte
  $$P[X = k] = f_X(k) = e^{-\lambda} \frac{\lambda^k}{k!}$$
- Diskrete geometrische Verteilung über den natürlichen Zahlen mit der Dichte
  $$P[X = k] = f_X(k) = (1-p)^k\ p$$

- Kontinuierliche Gleichverteilung über M=[a,b] $\subseteq$ R für a,b$\in$R mit der Dichte
- $f_X(x) = \frac{1}{b-a}$ *für* $a \le x \le b$, 0 *sonst*
- Kontinuierliche Exponentialverteilung über den nichtnegativen reellen Zahlen mit der Dichte
  $f_X(x) = \lambda e^{-\lambda x}$ *für* $x \ge 0$, 0 *sonst*
- Kontinuierliche Pareto-Verteilung über M=={x | x$\in$R $\wedge$ x>b} mit der Dichte
  $f_X(x) \to \frac{a}{b}\left(\frac{b}{x}\right)^{a+1}$ *für* $x > b$, 0 *sonst*
- Kontinuierliche Normalverteilung über den reellen Zahlen mit der Dichte
  $f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

Dabei sind a, b, p, λ, μ, σ (nichtnegative) Parameter.

Erwartungswert und Momente:
Für eine diskrete bzw. kontinuierliche Zufallsvariable X mit Dichte f(x) ist der Erwartungswert E[X] definiert durch $\sum_{x \in M} x \cdot f(x)$ bzw. $\int_{x \in M} x \cdot f(x)\,dx$. Das i-te Moment ist $E[X^i] = \sum_{x \in M} x^i \cdot f(x)$ bzw. $\int_{x \in M} x^i \cdot f(x)\,dx$. Die Varianz Var[X] ist $E[(X-E[X])^2] = E[X^2] - E[X]^2$.

Zwei Ereignisse A, B eines W.raums heißen **unabhängig**, wenn gilt $P[A \cap B] = P[A]\,P[B]$.
Die **bedingte Wahrscheinlichkeit P[A | B]** von A unter der Bedingung (Hypothese) B ist definiert als: $P[A \mid B] = \frac{P[A \cap B]}{P[B]}$

Satz von der totalen Wahrscheinlichkeit:

Für eine Partitionierung von W in Ereignisse B1, ..., Bn gilt: $P[A] = \sum_{i=1}^{n} P[A \mid Bi]\,P[Bi]$

Satz von Bayes:

Für Ereignisse A, B eines W.raums gilt: $P[A \mid B] = \frac{P[B \mid A] \cdot P[A]}{P[B]}$

Bedingte Wahrscheinlichkeiten der Form P[A|B] werden oft als A-Posteriori-Wahrscheinlichkeiten (nach Eintreten von B) bezeichnet; die unbedingte Wahrscheinlichkeit P[A] heißt dann A-Priori-Wahrscheinlichkeit.

<u>Beispiele:</u>
Beim Würfeln mit 1 Würfel ist P[X=5 | X ist ungerade] = (1/6) / (1/2) = 1/3.
Beim Würfeln mit 2 Würfeln ist P[X und Y zeigen einen Pasch | X+Y ≥] = (2/36) / (4/36) = 1/2.
Auf einem binären Übertragungskanal treten Bitfehler mit folgenden Wahrscheinlichkeiten auf:
P[0 empfangen | 0 gesendet] = 0.8,
P[1 empfangen | 0 gesendet] = 0.2,
P[0 empfangen | 1 gesendet] = 0.3,
P[1 empfangen | 1 gesendet] = 0.7,
und die Häufigkeiten von 0 und 1 seien wie folgt gegeben:
P[0 gesendet] = 0.4, P[1 gesendet] = 0.6.

Dann ist die Wahrscheinlichkeit, dass eine empfangene 1 auch wirklich eine 1 ist:
P[1 gesendet | 1 empfangen] =

$$\frac{P[1 \text{ empfangen} \mid 1 \text{ gesen det}] \cdot P[1 \text{ gesen det}]}{P[1 \text{ empfangen}]} = \frac{P[1 \text{ empfangen} \mid 1 \text{ gesen det}] \cdot P[1 \text{ gesen det}]}{\sum_{x \in \{0,1\}} P[1 \text{ empfangen} \mid x \text{ gesen det}] \cdot P[x \text{ gesen det}]} = 0.42/0.5 = 0.84.$$

3.3 Naive-Bayes-Klassifikator

Bei diesem Verfahren schätzt man die bedingte Wahrscheinlichkeit, dass ein Dokument zur Klasse Cj gehört unter der Vorbedingung, dass es einen bestimmten Featurevektor hat. Durch den Satz von Bayes lässt sich dies zurückführen auf die Wahrscheinlichkeit, dass das Dokument diesen Featurevektor hat, wenn es zur Klasse Cj gehört. Diese Wahrscheinlichkeit lässt sich anhand der Trainingsdaten schätzen.

Einfache Variante: binäre Features

Bei dieser Variante wird nur das Vorkommen oder Nichtvorkommen eines Terms im Dokument berücksichtigt; die Häufigkeit bleibt unberücksichtigt. Der Featurevektor des Dokuments di ist also ein binärer Vektor (der Dimension |F|=m), den wir zur besseren Abgrenzung mit Xi bezeichnen.

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

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

$= \Pi_{i=1}^{m} P[\, X_i \mid d \in c_k\,]\, P[\, d \in c_k\,]$ unter der Annahme, dass die Features paarweise unabhängig sind

bzw. der Linked-Independence-Annahme

$$\frac{P[X \mid d \in c_k]}{P[X \mid d \notin c_k]} = \prod_i \frac{P[X_i \mid d \in c_k]}{P[X_i \mid d \notin c_k]}$$

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

Die Unabhängigkeitsannahme für Features ist eigentlich realitätsfern; sie erklärt das Attribut „naiv“ im Namen des Verfahrens. Da es bei der Klassifikation aber nur auf eine relative Schätzung der Zugehörigkeitswahrscheinlichkeiten zu den verschiedenen Klassen ankommt, funktioniert das Verfahren trotz dieser groben Annahme erstaunlich gut.

Bessere Variante: Bag-of-Words-Modell

Hier werden die Häufigkeiten der Terme im Dokument berücksichtigt, nicht jedoch deren Positionen zueinander (daher Bag-of-Words). Man postuliert ein generierendes Modell, nach dem Dokumente erzeugt werden, und schätzt die Parameter dieses Modells anhand der Trainingsdaten. Eine simple Variante nimmt dazu an, dass man für jeden Term separat „würfelt“ – mit einem zweiseitigen Würfel bzw. einer Münze, ob er auf Wortposition 1, 2, usw. erscheint; dies führt auf eine Binomialverteilung für die Termhäufigkeit. Eine präzisere Variante berücksichtigt die Nebenbedingung, dass die Summe aller Termhäufigkeit gleich der Dokumentlänge sein muss (nach Elimination von Stoppwörtern). Man „würfelt“ dabei für jede Wortposition – quasi mit einem m-seitigen Würfel -, welcher Term

dort erscheint. Dies führt auf eine sog. Multinomialverteilung. Die Wahrscheinlichkeit der verschiedenen Würfelseiten sind die Parameter dieser Verteilung und werden anhand der Trainingsdaten geschätzt.

Schätze $P[\, d \in c_k \,|\, d \;\; hat \;\; \vec{f}\,] \sim P[\, \vec{f}\,|\, d \in c_k \,]\, P[\, d \in c_k \,]$ mit Termhäufigkeitsvektor 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

bzw. präziser:

$= \binom{length(d)}{f_1 \; f_2 \ldots f_m} p_{1k}^{\,f_1} \; p_{2k}^{\,f_2} \ldots p_{mk}^{\,f_m} \;\; p_k$ mit Multinomialverteilung und der Restriktion $\sum_{i=1}^{m} f_i \;\; = length(d)$

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

Beispiel für Naive-Bayes-Klassifikation mit Bag-of-Words-Modell

3 Klassen: c1 – Algebra, c2 – Analysis, c3 – Stochastik

8 Terme, 6 Trainingsdokumente d1, ..., d6: je 2 in jeder Klasse

$\Rightarrow$ p1=2/6, p2=2/6, p3=2/6

	Gruppe	Homomorphismus	Vektor	Integral	Limes	Varianz	Wahrscheinlichkeit	Würfel
	f1	f2	f3	f4	f5	f6	f7	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	Analysis	Stochastik
	k=1	k=2	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

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

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

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

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

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

Resultat: Ordne d7 der Klasse C3 (Stochastik) zu

Verbesserte Parameterschätzung mit Glättung (Smoothing)
Aufgrund der Trainingsdaten werden manche der pik-Parameter mit dem Wert 0 geschätzt, so dass bestimmte Klassenzuordnungen gar nicht mehr in Frage kommen. Dies schneidet den Klassifikator zu stark auf die – mit einer gewissen Willkür behafteten – Trainingsdaten zu; diesen Effekt nennt man Overfitting. Zur Abhilfe schätzt man alle Parameter mit einem von 0 verschiedenen Wert, indem man 0-Werte durch kleine Werte ersetzt und dafür alle anderen Werte geringfügig reduziert. Die einfachste und bekannteste Technik für solche Parameterglättungen ist das Laplace-Smoothing (das sich mathematisch als Korrektur eines sog. Maximum-Likelihood-Schätzers mit einer A-Priori-Gleichverteilung für den zu schätzenden Parameter ergibt). Dabei setzt man für für ein elementares Ereignis, das unter n Versuchen mit m möglichen Ausgängen j-mal beobachtet wurde, den Schätzwert für die Ereigniswahrscheinlichkeit p auf p = (j+1) / (n+m).

3.4 Feature-Selektion

Bei der Klassifikation von Textdokumenten kann die Dimensionalität des Featureraums sehr groß sein, und man möchte aus Effizienzgründen lieber mit einem Raum niedrigerer Dimensionalität arbeiten. Außerdem möchte man das inhärente Rauschen im Featureraum unterdrücken, also nur die wichtigsten, für die jeweiligen Klassen charakteristischen, Terme berücksichtigen. Dies kann man mit Hilfe informationstheoretischer Maße realisieren, z.B. der **relativen Entropie** zwischen Termen und Klassen, die in der Literatur auch als **Mutual-Information-Maß (MI-Maß)** bezeichnet wird:

$MI(X_i, c_j) = \sum_{X \in \{X_i, \bar{X}_i\}} \sum_{C \in \{c_j, \bar{c}_j\}} P[X \wedge C]\, log \frac{P[X \wedge C]}{P[X]\, P[C]}$ mit binären Variablen X_i=1, wenn der entsprechende Term in einem zufälligen Dokument vorkommt, und 0 sonst.

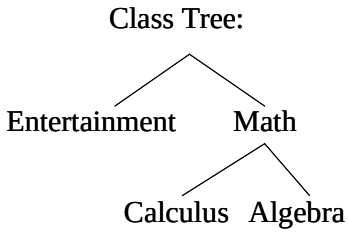
Dies ist ein Sonderfall der sog. **Kullback-Leibler-Distanz**, ein Maß für die Unterschiedlichkeit zweier Wahrscheinlichkeitsverteilungen, insbesondere zwischen einer zweidimensionalen Verteilung von Term und Klasse und einer Verteilung, bei der Term und Klasse unabhängig voneinander sind. Unabhängigkeit würde bedeuten, dass der Term in allen Klassen gleichwahrscheinlich ist, so dass er dann für die Klassifikation wenig geeignet wäre. Man wählt daher bei der Feature-Selektion gerade diejenigen m' << m Terme für eine Klasse aus, deren Kreuzentropie bzw. Kullback-Leibler-Divergenz am größten ist. Man beachte, dass diese Auswahl pro Klasse getroffen werden kann.

Wenn man für alle Klassen denselben reduzierten Featureraum verwenden will, wählt man diejenigen Features mit den größten Werten der mit den Klassenhäufigkeiten gewichteten MI-Maße:

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

Beispiel:

	film	hit	chart	theorem	limit	integral	group	vector
	f1	f2	f3	f4	f5	f6	f7	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



training docs:
d1, d2, d3, d4
→ Entertainment
d5, d6, d7, d8
→ Calculus
d9, d10, d11, d12
→ Algebra

Die Güte des Terms „chart“ als Diskriminator zwischen den Klassen Entertainment und Math ist:
MI(chart, Entertainment) = 3/12 log (3/12 / (4*4/144)) + 1/12 log (1/12 / (4*8/144)) +
1/12 log (1/12 / (8*4/144)) + 7/12 log (7/12 / (8*8/144))

Ergänzende Literatur

- David D. Lewis: Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. ECML Conference, 1998
- Kamal Nigam, Andrew McCallum, Sebastian Thrun, Tom M. Mitchell: Text Classification from Labeled and Unlabeled Documents using EM. Machine Learning Vol. 39 No. 2/3, 2000
- Yiming Yang, Xin Liu: A Re-Examination of Text Categorization Methods. SIGIR Conf. 1999
- Yiming Yang, Jan O. Pedersen: A Comparative Study on Feature Selection in Text Categorization. ICML Conference, 1997
- Arnold O. Allen: Probability, Statistics, and Queueing Theory with Computer Science Applications, Academic Press, 1990
- M. Greiner, G. Tinhofer: Stochastik für Studienanfänger der Informatik, Hanser-Verlag, 1996
- Thomas M. Cover, Joy A. Thomas: Elements of Information Theory, Wiley&Sons, 1991