# NewsWeeder - ein Nachrichtenfiltersystem

Jochen Hagemann

# Übersicht

- Klassifizierung von Filtersystemen
- Bedienung von NewsWeeder
- Datenrepräsentation
- Lernverfahren
  - tf-idf & Cosinus-Ähnlichkeitsmaß
  - MDL
- Vergleich der Verfahren
- Schlußfolgerung

## Klassifizierung von Filtersystemen

Einteilung in zwei Gruppen:

**Kollaborativ** Artikel werden aufgrund vorhergegangener Bewertungen vieler Leser eingestuft.
*Die Benutzer leisten die eigentliche Arbeit*

**Inhaltsbasiert** Artikel werden vom System ausgewertet und aufgrund eines Benutzerprofils eingestuft

- Benutzer muß sein Profil selbst anlegen (Schlüsselwörter, Regeln, ...)
- Filtersystem erlernt das Benutzerprofil

NewsWeeder verfolgt beide Ansätze - Hier aber **nur** inhaltsbasierter Ansatz

# Bedienung von NewsWeeder

NewsWeeder arbeitet nach folgender Prozedur:

- Der Benutzer wählt ein Nachrichtenbrett aus
- Es erscheint eine Überschriftenliste von vorbewerteten Artikeln
- Der Benutzer wählt eine Reihe von Artikeln aus und ließt diese nacheinander
- Nach dem Lesen eines Artikels muß eine Bewertung zwischen 1 und 5 abgegeben werden
- NewsWeeder sammelt die Bewertungen und paßt das Profil den Benutzerinteressen an

**Ziel:** Die Bewertung der Artikel besser vorherzusagen

# Datenrepräsentation

Jedes Textdokument wird zunächst in **Token** zerlegt
Beispiele für Token:

- gewöhnliche Wörter
- spezielle Zeichen ($ - Dollar, % -Prozent)
- kurze Wortfolgen mit besonderer Bedeutung: URLs und Eigennamen wie “Bayern München” (zusätzlich zu “Bayern” und “München”)

danach: Abstraktion vom Textdokument durch Bildung von **Tokenzahl-Vektoren** (Bag-of-Words-Modell) Jede Komponente entspricht dabei einem Token aus dem gesamten Vokabular

Probleme:

- Vektoren haben die Dimension 100.000 oder höher
- Reihenfolgeinformation geht verloren

## Lernverfahren - Motivation

Die Lernverfahren können auf folgende Daten zurückgreifen:

- eine Menge bereits gelesener Artikel mit den entsprechenden Bewertungen
- ein neuer noch ungelesener Artikel

Die Frage:"Wie ist der neue Artikel zu bewerten ?" zerfällt bei gewählter Repräsentation in

- Welche Token sind besonders aussagekräftig ?
- Wann sind Vektoren ähnlich ?

Die Lernverfahren tf-idf zusammen mit dem Cosinus-Ähnlichkeitsmaß und MDL sollen diese Fragen beantworten !

# td-idf

tf-idf ist eine Methode aus dem Bereich IR und basiert auf zwei empirischen Beabachtungen

**term-frequency** $tf_{t,d}$ Je öfter ein Token t in einem Dokument d vorkommt, umso wahrscheinlicher ist es, daß t maßgeblich ist für das Thema mit dem sich das Dokument beschäftigt

**document-frequency** $df_t$ Je öfter ein Token t in allen möglichen Dokumenten vorkommt, umso weniger unterscheidet t zwischen den Themen der Dokumente

Das Gewicht, welches einem Token in einem bestimmten Dokument zugemessen wird, entsteht durch Multiplikation von tf und dem inversen von df. (Deshalb tf-idf)

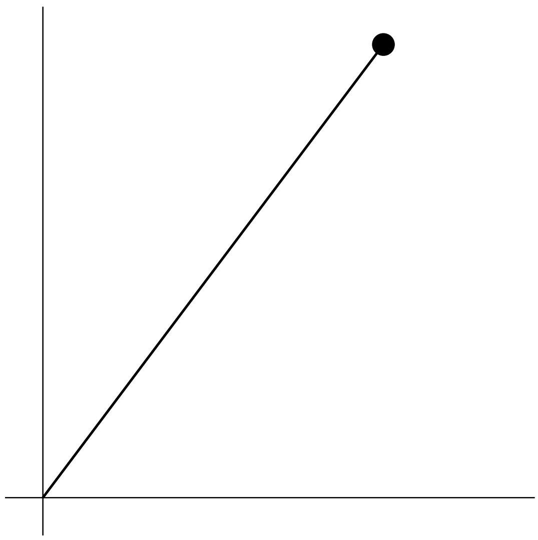
NewsWeeder berechnet das Gewicht eines Tokens t im Dokument d zu:

$$w(t,d) = tf_{t,d} * log(|N|/df_t)$$

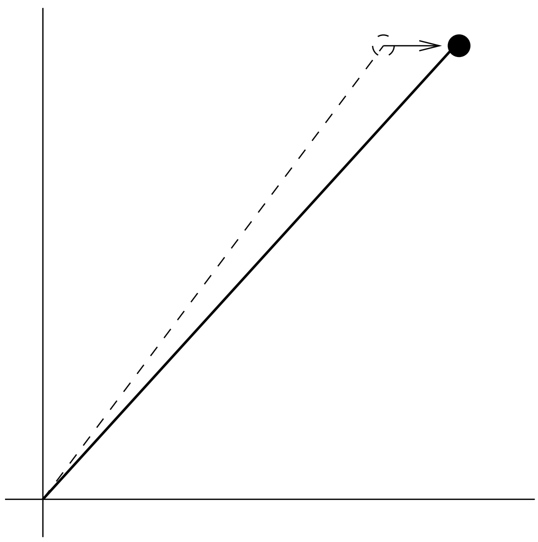
wobei N die Menge aller Dokumente darstellt.

# tf-idf anschaulich

Vektor vor . . .



. . . und nach der Anwendung von tf-idf

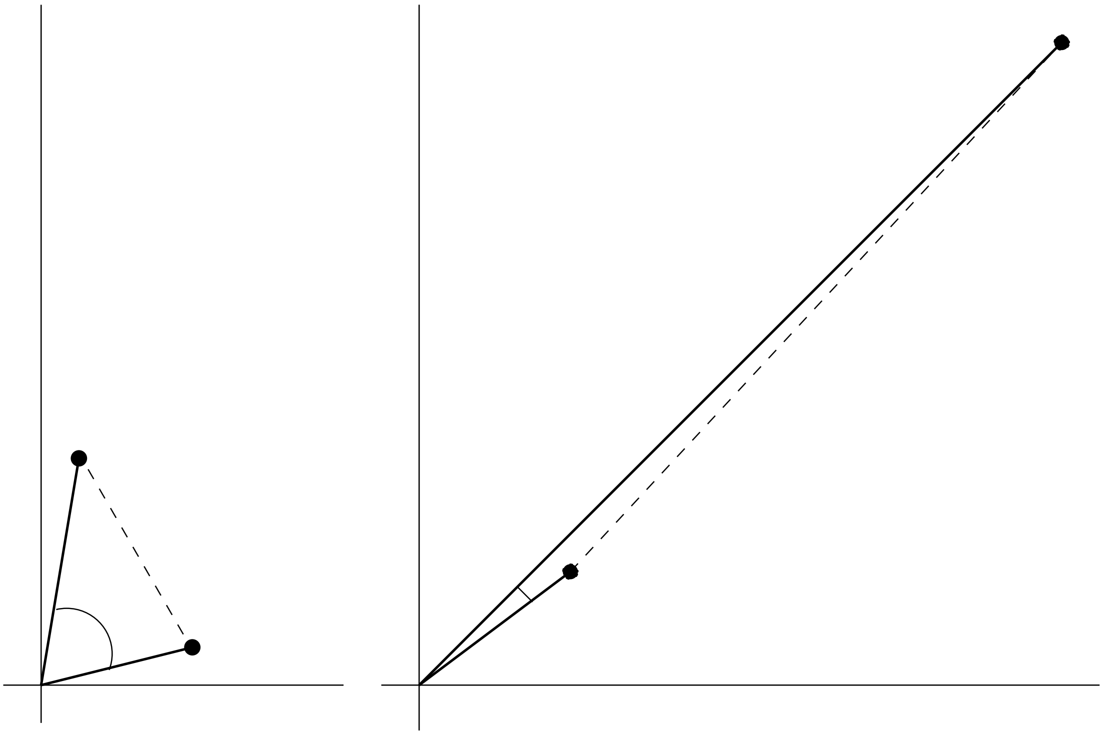


tf-idf hebt bestimmte Dimensionen hervor und verschiebt so den Vektor im Raum

## Das Cosinus-Vergleichskriterium

Annahme: Vektoren sind ähnlich, wenn

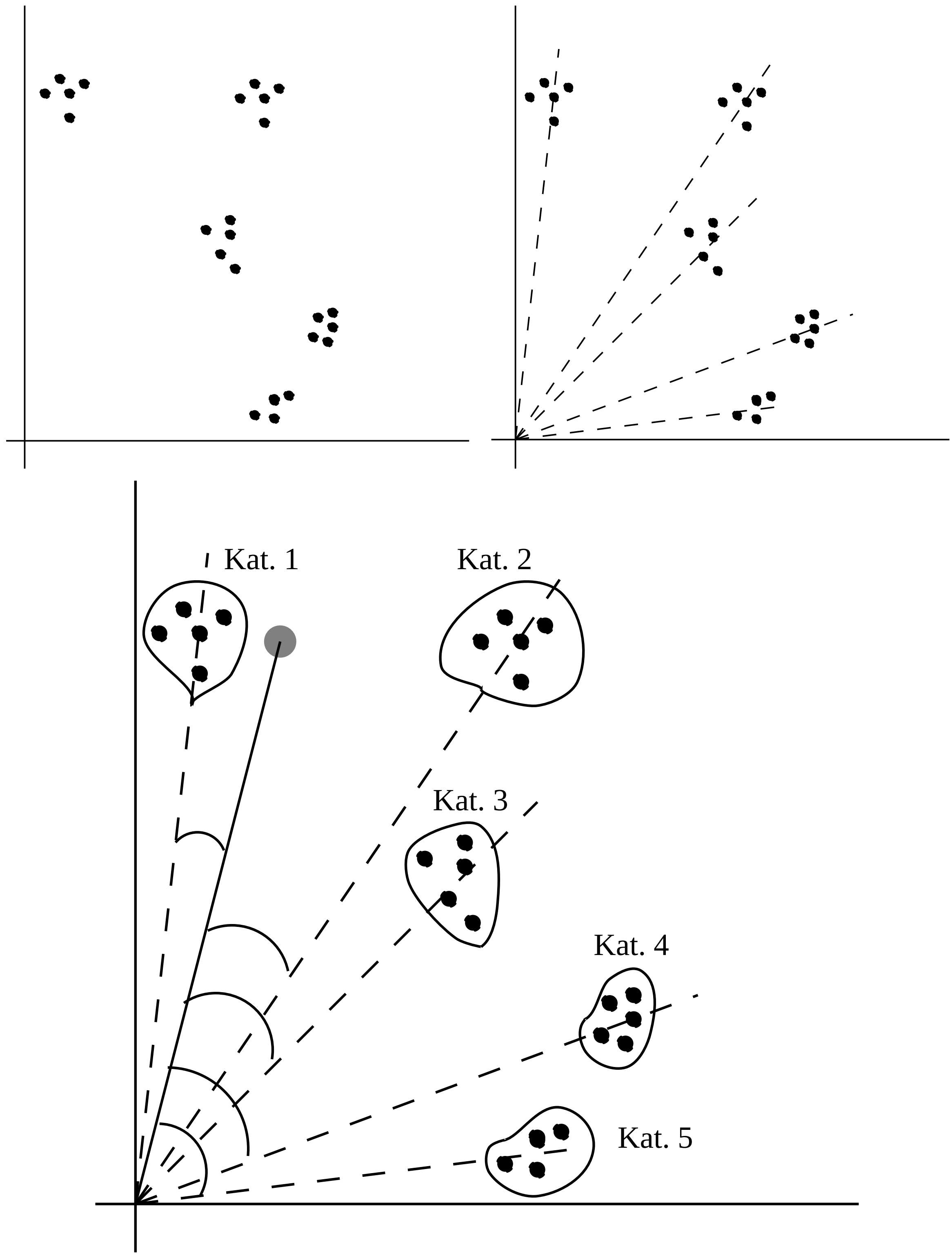
- die Differenz klein ist
- der Winkel zwischen ihnen klein ist



Es ist sinnvoller, den Winkel zwischen den Vektoren als Ähnlichkeitsmaß zu betrachten
Cosinus folgt aus Punktprodukt:

$$\mathbf{A} \cdot \mathbf{B} = AB cos(\mathbf{A}, \mathbf{B}) \Leftrightarrow cos(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{AB}$$

# Bewertungsvorhersage



# Das MDL-Prinzip

MDL steht für Minimum Description Length und "stellt einen Informationstheoretischen Rahmen zur Verfügung, mit dessen Hilfe ein Kompromiß zwischen Modellkomplexität und Trainingsfehler geschlossen werden kann."

Herleitung aus der Regel von Bayes:

$$p(H|D) = \frac{p(D|H)\,p(H)}{p(D)}$$

Ziel: Finde die Hypothese, die $p(H|D)$ **maximiert**
Da $p(D)$ unabhängig von H ist, kann man stattdessen folgenden Ausdruck **minimieren**:

$$-log_2(p(D|H) - log_2(p(H))$$

Nach Shannon beträgt die optimale Kodierungslänge einer Tokens X $-log_2(p(X))$ Bits.

$p(X) = \frac{1}{2} : -log_2(\frac{1}{2}) = 1$

$p(Y) = \frac{1}{8} : -log_2(\frac{1}{8}) = 3$

# Regel von Bayes anschaulich

$$p(H|D) = \frac{p(D|H)\,p(H)}{p(D)}$$

D (Daten, Beobachtung, Ursache...) :
*Die Temperatur beträgt 30 °C*

H (Hypothese, Schlußfolgerung, Wirkung...) :
*Es ist August*

$p(H|D)$ **:** Die Wahrscheinlichkeit, mit der die Aussage “Es ist 30°C, also haben wir August” zutrifft

$p(D|H)$ **:** Dir Wahrscheinlichkeit, daß irgendwann im August 30°C herrschen

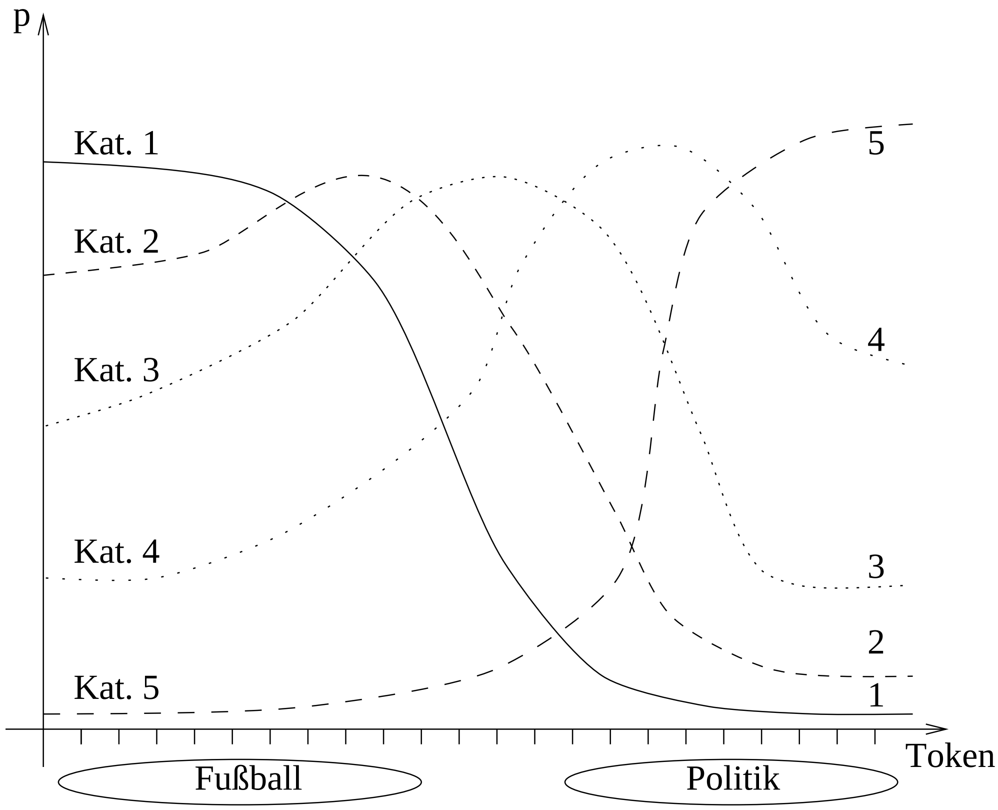
$p(H)$ **:** Die Wahrscheinlichkeit, daß überhaupt einmal August ist (= 1/12)

$p(D)$ **:** Die Wahrscheinlichkeit, daß an einem beliebigen Tag 30°C herrschen

# Anwendung des MDL-Prinzips I

Nach ausreichender Trainingsphase lassen sich Aussagen über die Wahrscheinlichkeitsverteilung der Token in jeder Kategorie machen. Für jede Kategorie wird ein Kodierungsmodell aufgestellt.

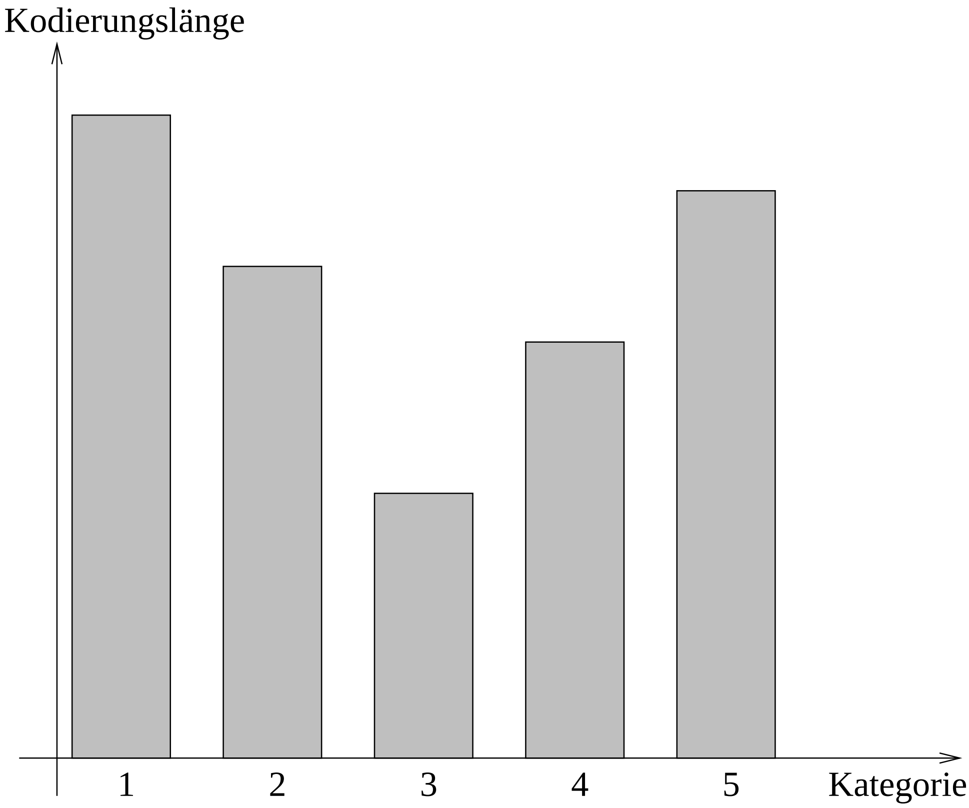
Bsp:
Nutzer interessiert sich für Fußball und **nicht** für Politik



# Anwendung des MDL-Prinzips II

Die Bewertung eines neuen Artikels erfolgt durch die Anwendung der Kodierungsmodelle auf seine Repräsentation und Beobachtung der sich ergebenden Kodierungslänge.

Bsp:
"Helmut Kohl Ehrentorwart beim FC Oggersheim"




Kompakteste Darstellung durch das Kodierungsmodell der Kategorie 3

# Wahrscheinlichkeitsmodell für die Tokenverteilung

Probleme:

- Abhängigkeiten zwischen den Token: Das Vorkommen eines Tokens kann die Auftretenswahrscheinlichkeit eines anderen Token beeinflussen
- Abhängigkeiten von der Textlänge: Bestimmte Token treten in kurzen Texten öfter auf, als in langen

Parameter für das Wahrscheinlichkeitsmodell:

| | |
|---|---|
| $c_i$ | Kategorie |
| $T_d$ | Tokenvektor |
| $l_d$ | Länge des Dokuments |
| $D_{train}$ | Trainingsdaten |

| | |
|---|---|
| $t_{i,d}$ | 1, wenn Token i in Dokument d vorkommt |
| $t_i$ | Anzahl der Dokumente die Token i enthalten |
| $r_{i,l}$ | Korrelationsgröße zwischen $T_i$ und $l_d$ |

# Vergleich der Verfahren & Schlußfolgerung

Nur zwei Benutzer lieferten genügend Daten für eine Analyse. Als Datenbasis dienten die 10% der Artikel mit den höchsten Bewertungsvorhersagen (Top-10%)

tf-idf Performance-Analyse:

- Liefert beste Ergebnisse, wenn die 300 häufigsten Token entfernt werden
- Der Anteil wirklich interessanter Artikel unter den Top-10% lag bei 37% bzw. 49%

MDL Performance-Analyse:

- Die Entfernung der N häufigsten Token verändert die Ergebnisse (erwartungsgemäß) nicht
- Der Anteil wirklich interessanter Artikel unter den Top-10% lag bei 44% bzw. 59%

**Schlußfolgerung:** MDL liefert bessere Ergebnisse als tf-idf. Die Trefferquote liegt bei ca. 50%

# Trefferquote unter MDL

| Nutzer-bewertung | Vorhergesagte Bewertung | | | | | | Gesamt |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | Skip | |
| 1 | 0 | 2 | 1 | 1 | 0 | 1 | 5 |
| 2 | 2 | 24 | 23 | 16 | 1 | 29 | 95 |
| 3 | 0 | 20 | 77 | 44 | 1 | 28 | 170 |
| 4 | 0 | 2 | 26 | 104 | 1 | 54 | 187 |
| 5 | 0 | 0 | 2 | 5 | 2 | 2 | 11 |
| Skip | 0 | 14 | 19 | 4 | 1 | 361 | 399 |
| Gesamt | 2 | 62 | 148 | 174 | 6 | 475 | 867 |