SEMINAR

Lernende Agenten im WWW

Sommersemester 1997

Prof. Dr. Katharina Morik
Lehrstuhl für Künstliche Intelligenz
Fachbereich Informatik
Universität Dortmund

SEMINARTHEMA

Klassifikation von Texten mit TF/IDF und Relevance Feedback

Ralf Klinkenberg

Literatur:

J. J. Rocchio, Jr. (1971):

Relevance Feedback in Information Retrieval. In Gerard Salton (Editor), *The SMART Retrieval System: Experiments in Automatic Document Processing*, pp. 313-323, Prentice-Hall, Englewood Cliffs, NJ, USA.

Thorsten Joachims (1997):

A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. To appear in *Proceedings of the 1997 International Conference on Machine Learning (ICML '97)*.

Repräsentation von Texten

- Für die weitere Verarbeitung im Rahmen von Textklassifikationsverfahren werden Texte durch **Dokumentvektoren** repräsentiert (Vektorraummodell).
- Der Dokumentvektor ist ein Vektor aus numerischen Werten, die Ausprägungen von Attributen des Dokumentes sind.
- Einfache Textrepräsentation:
  Attribute = alle im Text vorkommenden Wörter,
  Werte = Häufigkeit der Wörter im Text.

Attributselektion

- **Ziel:** gezielte Auswahl von Attributen für effizientere Verarbeitung und zuverlässigere Klassifikation.
- **Selektion nach Wortarten:**
  z.B. nur Substantive und Adjektive verwenden.
- **Stoppwortelimination:**
  z.B. Worte wie „der“, „die“, „das“, „und“ etc. entfernen.
- **Grundformreduktion:**
  nur Grund- oder Stammformen verwenden.
- **Statistische Maße:**
  Relevanz der einzelnen Attribute ermitteln.

Attributgewichtung mit TF/IDF

Mittels der **Termfrequenz** und der **Dokumentfrequenz** der Wörter eines Dokumentes aus einer Kollektion wird eine Gewichtung der Wörter in der Vektorraumdarstellung durchgeführt. Ein **Ähnlichkeitsmaß** erlaubt die Ermittlung (semantisch) ähnlicher Dokumente.

Termfrequenz:

TF*(w,d)* = Vorkommenshäufigkeit von Wort *w* in Dokument *d*

Dokumentfrequenz:

DF*(w)* = Anzahl der Dokumente, in denen *w* vorkommt

Inverse Dokumentfrequenz (*n* = Anzahl der Dokumente)**:**

$$\mathrm{IDF}(w) = \log\left(\frac{n}{\mathrm{DF}(w)}\right)$$

TF/IDF-Gewicht:

$$\mathrm{TFIDF}(w,d) = \mathrm{TF}(w,d) \cdot \mathrm{IDF}(w)$$

Ähnlichkeitsmaß für zwei Dokumente *d₁*, *d₂*:

$$S(d_1,d_2) = \cos\left(\overline{\mathrm{TFIDF}}(d_1), \overline{\mathrm{TFIDF}}(d_2)\right) = \frac{\overline{\mathrm{TFIDF}}(d_1) \cdot \overline{\mathrm{TFIDF}}(d_2)}{\left\|\overline{\mathrm{TFIDF}}(d_1)\right\| \cdot \left\|\overline{\mathrm{TFIDF}}(d_2)\right\|}$$

Normalisierung der Vektorlänge:

$$\overline{\mathrm{nTFIDF}}(d) = \frac{\overline{\mathrm{TFIDF}}(d)}{\left\|\overline{\mathrm{TFIDF}}(d)\right\|}$$

Relevance Feedback
(Rocchio Algorithmus)

- Das **Lernen** des Textklassifikators wird erreicht durch Kombinieren der Dokumentvektoren in **Prototypvektoren** $\vec{c}_j$ für jede Klasse C_j.
- Die normalisierten Dokumentvektoren der positiven sowie der negativen Trainingsbeispiele der jeweiligen Klasse werden getrennt summiert.
- Der **Prototypvektor** ergibt sich als gewichtete Differenz dieser Summen.

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

Negative Komponenten des Vektors $\vec{c}_j$ werden auf Null gesetzt; α und β sind nicht-negative Konstanten, *D* die Menge aller Trainingsdokumente.

Klassifikation eines neuen Dokuments *d'*:

Die Prototypvektoren werden benutzt, um die Klasse eines neuen Dokuments *d'* vorherzusagen:

$$c_{d'} = \arg\max_{C_j} \left(\cos\left(\vec{c}_j, \vec{d}' \right) \right)$$

argmax f(x) liefert das Argument *x*, für das *f(x)* maximal ist.