Lernende Suchmaschinen

Qingchui Zhu

PG 520 - Intelligence Service (WiSe 07 / SoSe 08)

Verzeichnis

Problemstellung und Zielsetzung

Problemstellung

- Anfragen liefern potentiell sehr viele Antworten
- Aber nur wenige davon sind für den Nutzer interessant
- Z. B. betrachten bei Internet-Suchmaschinen mehr als 90% aller Nutzer nur die ersten 10 Antworten
- Wie können potentiell relevante Antworten gehäuft am Anfang der Rangliste auftauchen?

Zielsetzung

Man entwickelt eine lernende Suchmaschine, die diese Probleme lösen kann.

Was ist eine lernende Suchmaschine?

Eine lernende Suchmaschine ist eine Suchmaschine mit künstlicher Intelligenz, die selbständig lernt, für wen etwas interessant sein könnte, indem sie die Benutzer befragt.

Beispiel 1.1 'ein Suchergebnis'

Die Frage 'support vector machine' (Figure1)

1. **Kernel Machines**
   http://svm.first.gmd.de/
2. Support Vector Machine
   http://jbolivar.freeservers.com/
3. **SVM-Light Support Vector Machine**
   http://ais.gmd.de/ ~ thorsten/svm_light/
4. An Introduction to Support Vector Machines
   http://www.support − vector.net/
5. Support Vector Machine and Kernel Methods References
   http://svm.research.bell − labs.com/SVMrefs.html
6. Archives of SUPPORT-VECTOR-MACHINES@JISCMAIL.AC.UK
   http://www.jiscmail.ac.uk/lists/SUPPORT–VECTOR–MACHINES.html
7. **Lucent Technologies: SVM demo applet**
   http://svm.research.bell − labs.com/SVT/SVMsvt.html
8. Royal Holloway Support Vector Machine
   http://svm.dcs.rhbnc.ac.uk/
9. Support Vector Machine - The Software
   http://www.support − vector.net/software.html
10. Lagrangian Support Vector Machine Home Page
    http://www.cs.wisc.edu/dmi/lsvm

Trio (q,r,c)

Definition Trio (q,r,c)

- q: Anfrage
- r: Die Rangliste der Suchergebnisse
- c: URL-Liste, die der Benutzer gewählt hat

Bei Beispiel 1.1

- q: 'support vector machine'
- r: Liste (1,2,3,4,5,6,7,8,9,10)
- c: (1,3,7)

Welche Information sind relevant?

Statistik: die Anzahl der angeklickten Seiten? Nein

Niemand will alle Suchergebnisse lesen, wenn sie zu viel sind. Bei Internet-Suchmaschinen betrachten mehr als 90% aller Nutzer nur die ersten 10 Antworten.

Man kann nur relativ relevant bestimmen. (bei Beispiel 1.1)

- Definition r^*: partielle Information der gesuchte Sortierung der Suchergebnisse
- link3 $<_{r^*}$ link2 link7 $<_{r^*}$ link2
  link7 $<_{r^*}$ link4
  link7 $<_{r^*}$ link5
  link7 $<_{r^*}$ link6

Darstellung der Relation r^*

- Für eine Suchergebnisliste gibt es m Dokumente $D = \{d_1, \ldots, d_m\}$
- Die Relation r^* kann man mit einer Matrix M darstellen. ($M \subset D \times D$)
  $(d_i <_{r^*} d_j) \Longrightarrow (M_{ij} = 1 \text{ und } M_{ji} = 0)$
- Bei Beispiel 1.1

$$M = \begin{pmatrix} - & * & * & * & * & * & * & * & * & * \\ * & - & 1 & * & * & * & 1 & * & * & * \\ * & 0 & - & * & * & * & * & * & * & * \\ * & * & * & - & * & * & 1 & * & * & * \\ * & * & * & * & - & * & 1 & * & * & * \\ * & * & * & * & * & - & 1 & * & * & * \\ * & 0 & * & 0 & 0 & 0 & - & * & * & * \\ * & * & * & * & * & * & * & - & * & * \\ * & * & * & * & * & * & * & * & - & * \\ * & * & * & * & * & * & * & * & * & - \end{pmatrix}.$$

(* unbekannt)

Kendall's τ

Definition Kendall's τ

- r_a und r_b : zwei Rangliste mit m Dokumente
- P : Anzahl der übereinstimmenden Paare
- Q : Anzahl der nicht übereinstimmenden Paare
- Q+P := $\binom{m}{2}$
- $\tau := \frac{P-Q}{P+Q} = 1 - \frac{2Q}{\binom{m}{2}}$

Beispiel für 5 Dokumente (m=5)

- $r_a : d_1 <_{r_a} d_2 <_{r_a} d_3 <_{r_a} d_4 <_{r_a} d_5$
- $r_b : d_3 <_{r_b} d_2 <_{r_b} d_1 <_{r_b} d_4 <_{r_b} d_5$
- Q = 3 $((d_2, d_3), (d_1, d_2), (d_1, d_3))$
- $P + Q = \binom{5}{2} = 10; \tau = 1 - \frac{6}{10} = 0,4$

Präzision

- Definition $r_{f(q)}$: Rangliste der Suchergebnisse nach Anfrage q
- Definition $\tau(r_{f(q)}, r^*)$: Präzision der Suchergebnisse
  Minimierung $Q \Leftrightarrow$ Maximierung τ
  und $(-1 \leq \tau(r_{f(q)}, r^*) \leq 1)$
- Wie groß ist $\tau(r_{f(q)}, r^*) \Leftrightarrow$ Wie groß ist die Präzision der Suchergebnisse

Präzision

- Maximierung $\tau(r_{f(q)}, r^*)$ ist equivalent zu Minimierung der Summe der Rangnummer der relevanten Dokumente. $\Rightarrow$ $AvgPrec(r_{f(q)}, r^*) = \frac{1}{R}\Sigma_{i=1}^{R}\frac{i}{p_i}$ Maximierung

  (R ist Anzahl der relevanten Dokumente; p_i ist die Rangnummer).
- Untere Schranke für $AvgPrec(r_{f(q)}, r^*)$

$$AvgPrec(r_{f(q)}, r^*) \geq \frac{1}{R}\left[Q + \binom{R+1}{2}\right]^{-1}\left(\sum_{i=1}^{R}\sqrt{i}\right) \qquad (1)$$

- Ziel : Maximierung $\tau p(f) = \int \tau(r_{f(q)}, r^*) d\ \Pr(q, r^*)$
  $Pr(q, r^*)$: eine Instanz von r^* für identische q

Beweis zu Formel (1)

- r_{rel} : die Rangliste, in der alle relevante Dokumente am Anfang auftauchen
- r_{sys} : Lernende Rangliste
- $p_1, \cdots p_R$: relevante und steigend sortierte Dokumente in r_{sys}.

$$AvgPrec(r_{sys}, r_{rel}) = \frac{1}{R} \sum_{i=1}^{R} \frac{i}{p_i} \tag{2}$$

$$p_1 + \cdots + p_R = Q + \binom{R+1}{2} \tag{3}$$

Beispiel 2,2 für 5 Dokumente mit 3 relevante Dokumente

- $r_{rel} : d_1 <_{r_{rel}} d_2 <_{r_{rel}} d_3 <_{r_{rel}} \square <_{r_{rel}} \square$
- $r_{f(q)} : d_1 <_r d_2 <_r \square <_r \square <_r d_3$
- $AvgPrec(r_{f(q)}, r_{rel}) = \frac{1}{3}(\frac{1}{1} + \frac{2}{2} + \frac{3}{5}) = 0,867$
- $p_1 + p_2 + p_3 = 1 + 2 + 5 = 8$

Beweis zu Formel (1)

Das Optimierungsproblem (Minimierung Gl.(2) mit Nebenbedingung Gl.(3)) in der sog. Lagrange-Funktion $L(p_1 + \cdots + p_R, \lambda)$ zusammenfassen:

$$L(p_1 + \cdots + p_R, \lambda) = \frac{1}{R} \sum_{i=1}^{R} \frac{i}{p_i} + \lambda \left(\Sigma_{i=1}^{R} p_i - Q - \binom{R+1}{2} \right)$$

$$\frac{\delta L(p_1+\cdots+p_R,\lambda)}{\delta p_i} = -iR^{-1} p_i^{-2} + \lambda \doteq 0$$

$$\frac{\delta L(p_1+\cdots+p_R,\lambda)}{\delta \lambda} \doteq 0$$

Im Prinzip kann man die Minimierung (Formel (1)) errechnen.

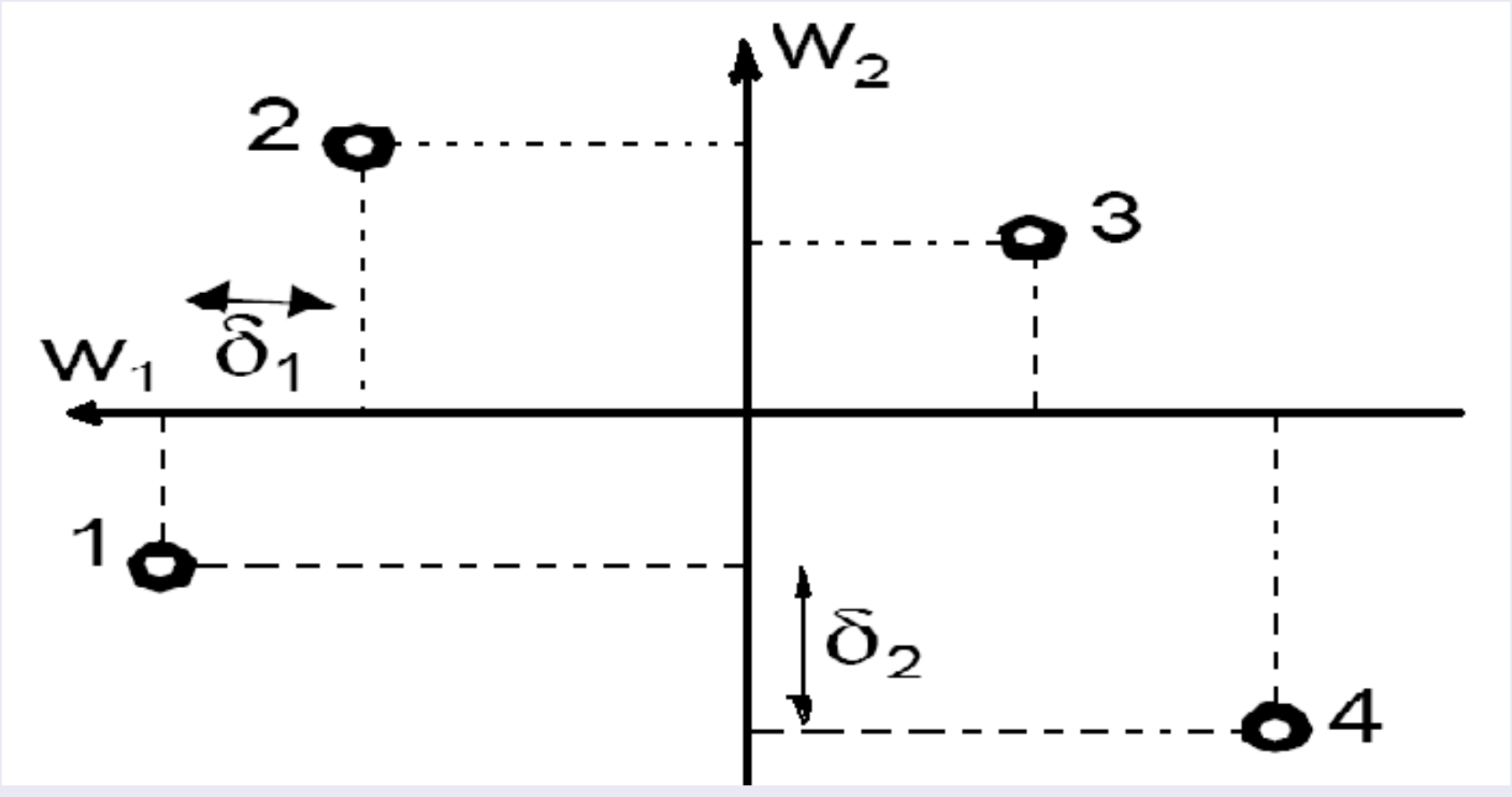
gewichtete Funktion

Für identische q und ihre r^*: $(q_1, r_1^*), \cdots, (q_n, r_n^*)$
wählt Lerner L eine Funktion f, so dass
$\tau_s(f) = \frac{1}{n} \sum_{i=1}^{n} \tau(r_{f(q_i)}, r_i^*)$ maximiert wird.
Man muss ein Algorismus und eine Funktionsfamilie F erzeugen, so dass Lerner $f \in F$ ein maximales $\tau_s(f)$ finden kann.

Man definiert eine gewichtete Funktion:
$(d_i, d_j) \in f_{\bar{\omega}}(q) \Leftrightarrow \bar{\omega}\phi(q, d_i) > \bar{\omega}\phi(q, d_j)$
(Definition $\phi(q, d)$: siehe Folie 16)

Mit unterschiedlichen Gewichtsvektoren ω werden die Dokument unterschiedlich sortiert (siehe Figure2).

Figure2



Mit Gewichtsvektor ω_1 werden die 4 Punkte (1,2,3,4) sortiert;
Mit Gewichtsvektor ω_2 werden die 4 Punkte (2,3,1,4) sortiert.

Gewichtsvektoren bei Suchergebnissen

$\phi(q, d)$ ist ein Diagramm mit der Dimension, die die Eigenschaft zwischen Anfrage q und Dokument d beschreibt. Das folgende Beispiel illustriert die Anwendung der Gewichtsvektoren bei Suchergebnissen:
Beispiel-Frage: 'side effect of drugs on memory and cognitive abilities, not aging'

t_i	ω_i	d_{1i}	d_{2i}	d_{3i}	d_{4i}
side effect	2	1	0,5	1	1
drugs	2	1	1	1	1
memory	1	1		1	
cognitive ability	1		1	1	0,5
not aging	-2		1		
Retrivalgewicht		5	2	6	4,5

Entsprechend den Retrivalgewichten werden die Dokumente in der Reihenfolge $d_3; d_1; d_4; d_2$ ausgegeben.

Das Optimierungsproblem der Suchergebnisse

Maximierung $\tau_s(f)$ ist equivalent zu maximalen Erfüllung der folgenden Ungleichung.

$\forall(d_i, d_j) \in r_i^* : \bar{\omega}\phi(q_1, d_i) > \bar{\omega}\phi(q_1, d_j)$

$\ldots$

$\forall(d_i, d_j) \in r_n^* : \bar{\omega}\phi(q_n, d_i) > \bar{\omega}\phi(q_n, d_j)$

Aber es ist NP-Schwer Problem.

Das Optimierungsproblem der Suchergebnisse

Lösung: Einführen SVM Algorithmus

Min. $V(\bar{\omega}, \bar{\xi}) = \frac{1}{2}\bar{\omega} \cdot \bar{\omega} + C \sum \xi_{i,j,k}$

mit Nebenbedingung:

$\forall (d_i, d_j) \in r_i^* : \bar{\omega}\phi(q_1, d_i) \geq \bar{\omega}\phi(q_1, d_j) + 1 - \xi_{i,j,1}$

$\ldots$

$\forall (d_i, d_j) \in r_n^* : \bar{\omega}\phi(q_n, d_i) \geq \bar{\omega}\phi(q_n, d_j) + 1 - \xi_{i,j,n}$

$\forall i, j, k : \xi_{i,j,k} \geq 0$

C ist positiv Konstante, die den Ausgleich zwischen der Minimierung von $\frac{1}{2}\bar{\omega} \cdot \bar{\omega}$ und der korrekten Klassifizierung der Trainingsbeispiele regelt.

Das Optimierungsproblem der Suchergebnisse

- $\bar{\omega}(\phi(q_n, d_i) - \phi(q_n, d_j)) \geq 1 - \xi_{i,j,n}$
- $(d_i, d_j) \in f_{\bar{\omega}^*}(q)$
  $\iff \bar{\omega}^* \phi(q, d_i) > \bar{\omega}^* \phi(q, d_j)$
  $\iff \sum \alpha^*_{k,l} \phi(q_k, d_l) \phi(q, d_i) > \sum \alpha^*_{k,l} \phi(q_k, d_l) \phi(q, d_j)$
- **Grundlegende Idee:**
  Der Vektorraum und die zugehörigen Trainingsdaten sind durch eine Kernelfunktion $\sum \alpha^*_{k,l} \phi(q_k, d_l) \phi(q, d_i)$ in einen Raum mit so hoher Dimension zu überführen, dass sich die Trainingsdaten dort linear trennen lassen.

Wie kann man zwei Rangordnungen A und B vergleichen?

- Man kombiniert A und B zu C
- top ℓ Links von C enthalten top k_a Links von A und top k_b Links von B
- $\mid k_a - k_b \mid \leq 1$
- das bedeutet: Die Benutzer können fast gleiche Anzahl Dokument aus A und B lesen.
- Vergleich der angeklickten Anzahl des Links aus A und B

Beispiel

Ranking A:

1. Kernel Machines
   http : //svm.first.gmd.de/
2. SVM-Light Support Vector Machine
   http : //ais.gmd.de/ ∼ thorsten/svm_light/
3. Support Vector Machine and Kernel ... References
   http : //svm.....com/SVMrefs.html
4. Lucent Technologies: SVM demo applet
   http : //svm.....com/SVT/SVMsvt.html
5. Royal Holloway Support Vector Machine
   http : //svm.dcs.rhbnc.ac.uk/
6. Support Vector Machine - The Software
   http : //www.support−vector.net/software.html
7. Support Vector Machine - Tutorial
   http : //www.support−vector.net/tutorial.html
8. Support Vector Machine
   http : //jbolivar.freeservers.com/

Ranking B:

1. Kernel Machines
   http : //svm.first.gmd.de/
2. Support Vector Machine
   http : //jbolivar.freeservers.com/
3. An Introduction to Support Vector Machines
   http : //www.support − vector.net/
4. Archives of SUPPORT-VECTOR-MACHINES ...
   http : //www.jiscmail.ac.uk/lists/SUPPORT...
5. SVM-Light Support Vector Machine
   http : //ais.gmd.de/ ∼ thorsten/svm_light/
6. Support Vector Machine - The Software
   http : //www.support−vector.net/software.html
7. Lagrangian Support Vector Machine Home Page
   http : //www.cs.wisc.edu/dmi/lsvm
8. A Support ... - Bennett, Blue (ResearchIndex)
   http : //citeseer.../bennett97support.html

Combined Results:

A B	1. **Kernel Machines**
	http : //svm.first.gmd.de/
B	2. Support Vector Machine
	http : //jbolivar.freeservers.com/
A	3. **SVM-Light Support Vector Machine**
	http : //ais.gmd.de/ ∼ thorsten/svm_light/
B	4. An Introduction to Support Vector Machines
	http : //www.support − vector.net/
A	5. Support Vector Machine and Kernel Methods References
	http : //svm.research.bell − labs.com/SVMrefs.html
B	6. Archives of SUPPORT-VECTOR-MACHINES@JISCMAIL.AC.UK
	http : //www.jiscmail.ac.uk/lists/SUPPORT−VECTOR−MACHINES.html
A	7. **Lucent Technologies: SVM demo applet**
	http : //svm.research.bell − labs.com/SVT/SVMsvt.html
A	8. Royal Holloway Support Vector Machine
	http : //svm.dcs.rhbnc.ac.uk/
B A	9. Support Vector Machine - The Software
	http : //www.support − vector.net/software.html
B	10. Lagrangian Support Vector Machine Home Page
	http : //www.cs.wisc.edu/dmi/lsvm

Beispiel

- jeweils 7 links aus A und B
- 3 Links werden aus A angeklickt
- 1 Link wird aus B angeklickt
- A ist deswegen besser als B.