

Web Mining: Übung 3

Lösungsvorschlag



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Aufgabe 1:

Einteilung der Testdaten



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ Verwendeter Testdatensatz: Co-training Experiments for COLT 98
<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-51/www/co-training/data/>
- ▶ Testdaten sind in Unterordnern sortiert:
Art/Kategorie/VolleURL
- ▶ Art kann „fulltext“ oder „inlinks“
- ▶ „fulltext“: HTML-Dateien
- ▶ „inlinks“: Link-Text(e), die auf die Seiten verwiesen haben.
Beispiel: `Link-Text`
- ▶ Zur einfacheren Verarbeitung soll die Ordnerhierarchie entfallen und die Informationen in den Dateinamen übernommen werden.

Aufgabe 1:

BASH: Neusortierung/Entfernung der Hierarchie

Teil 1: Die dunklen Zeichen



- ▶ Mit einem Bash-Skript soll die Hierarchie flach gezogen werden
- ▶ Das Namensschema der Dateien soll lauten:
$$\text{fulltext|inlinks} \underbrace{\text{---}}_{\text{Trenner}} \text{Kategorie} \underbrace{\text{---}}_{\text{Trenner}} \text{VolleURL}$$
- ▶ Die URL soll dabei außerdem angepasst werden:
 - ▶ Doppelpunkte werden entfernt, weil sich Windows daran stört
 - ▶ `http://www.` wird ersetzt, um die URLs zu verkürzen. Windows stört sich auch an langen Dateipfaden
- ▶ Am Ende sehen die Dateinamen so aus:
$$\underbrace{\text{fulltext}}_{\text{Art}} \text{---} \underbrace{\text{course}}_{\text{Kategorie}} \text{---} \underbrace{\text{BEG}}_{\text{http...}} \text{---} \text{cs.washington.edu}^{\wedge} \text{education}^{\wedge} \text{courses}^{\wedge} 431^{\wedge}$$
- ▶ Vorteil des Ersetzens gegenüber dem Wegwerfen einzelner Zeichen/Abschnitte: Es gehen keine Informationen verloren. Bei Bedarf kann der gesamte Link wiederhergestellt werden.

Aufgabe 1:

BASH: Neusortierung/Entfernung der Hierarchie

Teil 2: Angriff der Kategorisierung



- ▶ Die Daten müssen nun noch in Test- und Trainingsdaten eingeteilt werden
- ▶ Das Skript wirft für jede Datei die Münze und dokumentiert das Ergebnis ebenfalls im Dateinamen

- ▶ Beispiel:

```
test--fulltext--course--BEG--cs.wisc.edu^^dyer^cs766.html
```

- ▶ Das Ergebnis der Verteilung scheint zufällig und damit brauchbar:

	Training	Test	Summe
course	233	227	460
non-course	802	840	1642
Summe	1035	1067	2102

- ▶ Mit dem im weiteren Verlauf verwendeten libsvm lassen sich die Daten zwar einfacher und eleganter in Trainings- und Testdaten einteilen, aber da es nun mal in der Aufgabenstellung stand ...

Aufgabe 2: Aufbereitung der Daten

Dokument zu Wort-Liste

Beispieldokument



TECHNISCHE
UNIVERSITÄT
DARMSTADT

CSE 590D (Autumn 1995): Transcript-Based Education/WWW

Welcome to the CSE 590D Home Page!

This is the World Wide Web ("the Web" for short) hypermedia document for CSE 590D and contains information about the class. Keep in mind that this document is not static, and that new information will be added from time to time.

Copyright Notice: The material in this course web is subject to copyright. While it may be viewed by the public, it should not be installed at any web site other than the one at the University of Washington.

Reading for October 10 -- G. McCalla: "The Central Importance of Student Modelling in Intelligent Tutoring."

Reading for October 17 -- (presented by Sandi Youngquist)

Meeting of October 23 -- Discussion with Paul Barton-Davis about Internet services.

Reading for October 31 -- C. Laborde and J-M Laborde: "Problem Solving in Geometry: From Microworlds to Intelligent Computer Environments" (presented by Tessa Lau)

Reading for November 7 -- B. Bartels: "Promoting mathematics connections with concept mapping" (plus presentation by Gary Anderson)

No meeting November 14 --

Reading for November 21 --

The readings for this meeting are all online (on the WWW).

Aufgabe 2: Aufbereitung der Daten

Dokument zu Wort-Liste

Reduzierung der Wörter



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ Text aus dem html-Body wird in Kleinbuchstaben transformiert und an Leerstellen in einzelne Wörter unterteilt
- ▶ Emailadressen und URLs, die mit `http[s]://` beginnen, werden entfernt
- ▶ Wörter, die auf Satzzeichen, Klammern, Anführungszeichen etc. enden, werden um dieses reduziert
- ▶ Ebenso Wörter, die mit Klammern, Anführungszeichen o.ä. beginnen

Beispiel:

(autumn wird zu autumn
page! wird zu page
public, wird zu public
1995): wird zu 1995

Aufgabe 2: Aufbereitung der Daten

Stopwords und Stemming



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ Die verbliebenen Wörter werden um Stopwords aus einer Liste reduziert
 - ▶ Vorher: 381 Wörter
 - ▶ Nachher: 231 Wörter
- ▶ Anschließendes Stemming der Wörter mit dem Porter Stemmer
<https://tartarus.org/martin/PorterStemmer/>
 - ▶ Von den 231 Wörtern verbleiben 160 verschiedene Wortstämme

Stemming-Beispiele:

welcome → welcom

contains → contain

information → inform

added → ad

Aufgabe 2: Aufbereitung der Daten

TF-IDF-Vektor



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ normalisierte TF des Terms t in Dokument d : $tf(t, d) = \frac{f(t, d)}{\max\{f(w, d): w \in d\}}$
- ▶ inv. Document Frequency: $idf_i = \log \frac{N}{n_i}$
- ▶ Gewichtung der einzelnen Wörter: $w_{i,j} = tf_{i,j} \cdot idf_i$

Werte der Beispieldatei im Trainingsset:

TF-IDF-Vektor:

```
{'cse': 0.7920241942426215, '590d': 1.2435691877385933, 'home':  
0.04761829630391385, 'page': 0.029480629516760913, 'autumn':  
0.502504280189381, '1995': 0.1530192780213716, ...}
```


Aufgabe 2: Aufbereitung der Daten

Sparse-Darstellung

- ▶ Die 10 häufigsten Wörter nach Dokumenthäufigkeit:
[(`'page'`, 234), (`'comput'`, 234), (`'home'`, 198), (`'scienc'`, 197), (`'univers'`, 189), (`'depart'`, 149), (`'1996'`, 133), (`'inform'`, 128), (`'research'`, 128), (`'system'`, 127)]
- ▶ Sparse-Format:
+1 1:0.029480629516760913 2:0.014740314758380457
3:0.04761829630391385 5:0.026334571412992794
8:0.1424731521859943
- ▶ Bedeutung:
 - ▶ page hat eine TF-IDF-Gewichtung von 0.029480629516760913
 - ▶ comput hat eine TF-IDF-Gewichtung von 0.014740314758380457
 - ▶ home hat eine TF-IDF-Gewichtung von 0.04761829630391385
 - ▶ scienc ist nicht im Dokument vorhanden
 - ▶ univers hat eine TF-IDF-Gewichtung von 0.026334571412992794
 - ▶ usw.

Aufgabe 3:

Analyse der erhobenen Daten

Teil 1: Statistische Analyse – Konfusionsmatrix



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ Trainingsdaten: 279, Testdaten: 280, Gesamt: 559
- ▶ Konfusionsmatrix:

		Vorhersage		Summe
		course	non-course	
Kategorie	course (+1)	12	49	61
	non-course (-1)	42	177	219
	Summe	54	226	280

- ▶ Accuracy: $A = \frac{TP+TN}{N} = \frac{12+177}{280} \approx 67,5\%$
- ▶ Trainingszeit: 0,0267 Sekunden (-t 0 -c 50, 851 Iterationen)
Testzeit: 0,0236 Sekunden

Aufgabe 3:

Analyse der erhobenen Daten

Teil 1: Statistische Analyse – Baseline



- ▶ Baseline: Tippt immer auf die am häufigsten vorkommende Kategorie
- ▶ Insgesamt sind 428 von 559 Datensätzen non-course (-1)
Ermittelt mit `cat data.txt | grep '-1' | wc -l`
- ▶ In den Trainingsdaten sind $428 - 219 = 209$ von 279 Datensätzen non-course
- ▶ Die Baseline Konfusionsmatrix sieht also so aus:

		Vorhersage		
		course	non-course	Summe
Kategorie	course (+1)	0	61	61
	non-course (-1)	0	219	219
Summe		0	280	280

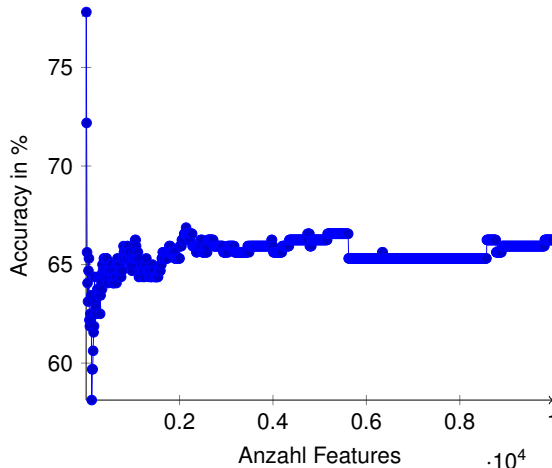
- ▶ Accuracy: $A = \frac{TP+TN}{N} = \frac{219}{280} \approx 78,21\%$
- ▶ Dieses simple Modell ist wesentlich besser als das trainierte Modell

Aufgabe 3.3: Variation der Feature-Anzahl

Siehe Datei 3.3result.csv

Genauigkeit

- ▶ Accuracy fällt zunächst stark ab, nähert sich dann $\approx 72\%$ an
- ▶ Sinnvoll ist also entweder die Betrachtung von weniger als ≈ 50 Features (danach fällt Genauigkeit ab)
- ▶ ...oder ≈ 2500 , weil danach die Genauigkeit nicht wesentlich steigt

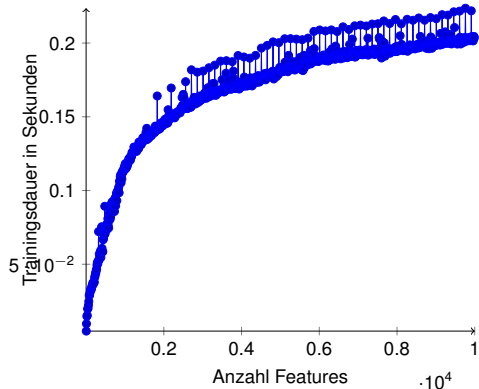


Aufgabe 3.3:

Variation der Feature-Anzahl

Trainingsdauer

- ▶ Trainingsdauer steigt von Ausreißern abgesehen weniger als linear an
- ▶ Selbst mit vielen Features beträgt die Dauer jedoch weniger als eine Sekunde
- ▶ → Trainingsdauer eher nicht maßgeblich für Wahl der Anzahl



Aufgabe 3.3: Variation der Feature-Anzahl Testdauer

- ▶ Die Testdauer ist nicht linear
- ▶ Abnehmende Grenzrate
- ▶ Auch bei vielen Features weit unter einer Sekunde

