ECL: Textklassifizierung mit Naive Bayes

Katja Markert

Institut für Computerlinguistik
Universität Heidelberg
markert@cl.uni-heidelberg.de

January 9, 2023

Heute

1. Bis jetzt: Zählen und Verarbeitung von *n*-grams
2. Jetzt: Naive Bayes: ein Bag of Words Textklassifikator aus der Kategorie “ überwachtes maschinelles Lernen”

Übersicht

Übersicht

Was ist eine Klassifikationsaufgabe?

Klassifikation

Gegeben eine Instanz oder eine Menge von Instanzen, weise der Instanz ein Konzept/Klasse/Kategorie aus einer fixen, diskreten Menge von Konzepten zu. Tue dies auf der Basis von Merkmalen (features/explanatory variables).

- Foul
- Elfmeter
- Ecke ✓

- Latein ✓
- Deutsch
- Englisch

Merkmale:

	e	ch	ing	L/D/E
t1	50	10	0	?
t2	10	0	15	?
t3	1	1	2	?

Terminologie

- **Instanz:** einzelnes Beispiel im Datensatz
- **Attribute/Feature/explanatory variable/Merkmal:** ein Aspekt einer Instanz. Beispiel: *ing*, *bank*, Länge, ...
- **Wert (Value):** Wert eines Merkmals. Beispiel: ja, nein für das Merkmal *ing*, oder 0,1,2,3,4 ... 100 ... für das Merkmal *ing*
- **Konzept/Klasse/Response Variable/Antwortvariable:** was man lernen muss. Beispiel: Latein, Deutsch, Englisch ...

Überwachtes Lernen

Grundidee

Ein Algorithmus, der einem Input einen Output zuweist, kann schwer zu entwerfen sein. Bei manchen Aufgaben ist es aber einfach, Beispielinputs mit bekannter Klasse zu erhalten. Man will dann den Algorithmus lernen.

Supervised Learning/Überwachtes Lernen

- Generiere Trainingsset, auf dem man die Klasse pro Instanz kennt.
- Wähle Merkmale und einen Modelltyp
- Lerne ein Vorhersagemodell (classifier) aus dem Trainingsset = schätze Modell aus Daten.
- Verifiziere das Modell mit Testdaten (auf dem die Klassen mithilfe des gelernten Modells “geraten” werden sollen)

Textklassifikation Definition

- (Repräsentation) eines Testdokumentes d
- Eine fixe Menge an Klassen $C = \{c_1, c_2, \ldots, c_j\}$
- Bestimme Kategorie von $d : \gamma(d) \in C$, wobei γ eine Klassifikationsfunktion ist, die Dokumente auf Klassen abbildet

Mit überwachtem maschinellem Lernen:

- Ein Trainingsset D von Dokumenten je mit einem Label in C $(d_1, c_1), \ldots (d_m, c_m)$
- Bestimme eine Lernmethode, die einen Klassifizierer γ lernt
- Bestimme Kategorie von $d : \gamma(d) \in C$

Häufigste Textklassifikation: Topic Classification

MedLine Artikel

Mesh Subject Categories

- Blood Supply
- Chemistry
- Drug Therapy
- Epidemiology
- Embryology
- ...

Manuelle Klassifikation: Library of Congress, PubMed, Yahoo directory (damals) ....

Übersicht

“Bag of words” für Textklassifikation: Intuition

About hotels, restaurants or movies?

A good budget hotel’ .... Price includes breakfast with really nice food. Rooms are modern and of a reasonable size. The centre of Leeds is about a 15 min walk at the most. Hotel has bar area.

budget	1
hotel	2
price	1
rooms	1
breakfast	1
food	1
...	...

BOW in Multinomial Naive Bayes

Fehlende Grafik

Naive Bayes: Multinomiales Model

Intuition

Benutze BoW Modell mit Worten als Merkmalen, Worthäufigkeiten als Merkmalswerten und Bayes Regel. Generatives Modell.

Für Dokument d, bestimme wahrscheinlichste Klasse $c \in C$.

$$
\begin{aligned}
C_{MAP} &= \operatorname*{argmax}_{c \in C} P(c|d) && (1)\\
&= \operatorname*{argmax}_{c \in C} \frac{P(d|c)P(c)}{P(d)} && (2)\\
&= \operatorname*{argmax}_{c \in C} P(d|c)P(c) && (3)\\
&= \operatorname*{argmax}_{c \in C} P(w_1, w_2, \ldots w_{n_d}|c)P(c) && (4)\\
&= \operatorname*{argmax}_{c \in C} P(X_1 = w_1, X_2 = w_2, \ldots, X_{n_d} = w_{n_d}|c)P(c) && (5)
\end{aligned}
$$

MAP = maximum a posteriori; w_1 Wort in Vokabular in Position 1 im Dokument

Eigenschaften, die Klassifikation haben sollte

- **Generalisierung** zu Daten, die nicht im Trainingsset sind (**ungesehene Daten**)
- Fähigkeit zur **Diskriminierung** (verwechsle ähnliche Inputs mit verschiedenen Outputs nicht)
- Das volle Modell auf der letzten Folie modelliert alle Abhängigkeiten zwischen Wörtern (an bestimmten Positionen) sowie zwischen Wörtern und Klasse
- Das volle Modell entspricht einem table look up und kann nicht generalisieren

Modellsimplifizierung: Extremfall

Man könnte annehmen, dass die Antwortvariable überhaupt nicht von den Merkmalen abhängt!

$$c_{\text{MFC}} = \underset{c \in C}{\operatorname{argmax}} P(c)$$

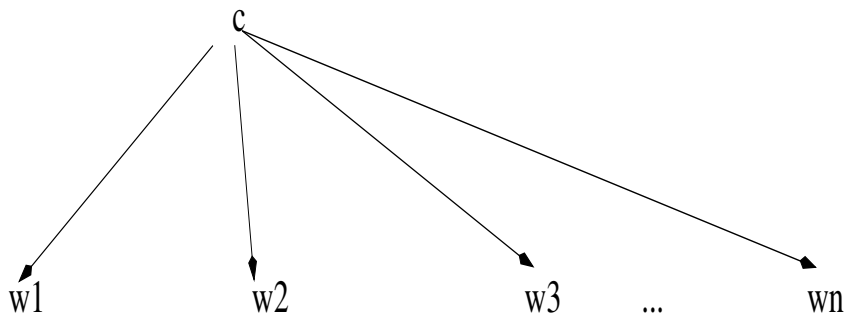
Most frequent class

Dies entspricht einem Algorithmus, der immer die häufigste Klasse jeder Testinstanz zuweist (maximum a posteriori). Oft **Baseline**, die der richtige Klassifizierer schlagen sollte

1. Wann ist dieser Algorithmus schwer zu schlagen?
2. Wann sollte man ihn vielleicht verwenden?

Geringere Simplifizierung: Naive Bayes

Berücksichtige nur die Dependenzen zwischen den Merkmalen (Wörtern) und der Klasse, aber nicht zwischen den verschiedenen Merkmalen.



Naive Bayes: Unabhängigkeitsannahmen

$$P(X_1 = w_1, X_2 = w_2, \ldots X_{n_d} = w_{n_d}|c)$$

- **Conditional Independence:** Nimm an, dass die Worte voneinander unabhängig sind, wenn die Klasse c gegeben ist.

$$\begin{aligned} P(X_1 = w_1, \ldots, X_{n_d} = w_{n_d}|c) &= P(X_1 = w_1|c) \cdot P(X_2 = w_2|c) \cdot \\ &\cdot \quad \ldots \cdot P(X_{n_d} = w_{n_d}|c) \end{aligned}$$

- **Bag of Words** Annahme: Nimm an, dass Wortposition egal ist

$$P(X_k = w|c) = P(X_l = w|c)$$

für alle Klassen c, alle Positionen l, k and alle Worttypen w.

Training des multinomialen NB

1. Extrahiere Vokabular V aus Trainingskorpus
2. Berechne $P(c)$ für alle c aus dem Trainingskorpus:

$$P(c) = \frac{\#docs\ of\ class\ c}{total\ \#\ of\ docs\ in\ training}$$

3. Berechne $P(w_i|c) := P(X_l = w_i|c)$ für alle w_i im Vokabular und alle c:
   1. Konkateniere alle Trainingsdokumente der Klasse c zu einem langen Dokument
   2. $$P(w_i|c) = \frac{n_i^c}{n^c}$$
      wobei n_i^c Frequenz von w_i in langem Dokument und n^c Länge des langen Dokuments
   3. Mit Smoothing:
      $$P(w_i|c) = \frac{n_i^c + 1}{n^c + |V|}$$

Training des multinomialen NB

1. Extrahiere Vokabular V aus Trainingskorpus
2. Berechne $P(c)$ für alle c aus dem Trainingskorpus:

$$P(c) = \frac{\#docs\ of\ class\ c}{total\ \#\ of\ docs\ in\ training}$$

3. Berechne $P(w_i|c) := P(X_l = w_i|c)$ für alle w_i im Vokabular und alle c:
   1. Konkateniere alle Trainingsdokumente der Klasse c zu einem langen Dokument
   2. $$P(w_i|c) = \frac{n_i^c}{n^c}$$
      wobei n_i^c Frequenz von w_i in langem Dokument und n^c Länge des langen Dokuments
   3. Mit Smoothing:
      $$P(w_i|c) = \frac{n_i^c + 1}{n^c + |V|}$$

Training des multinomialen NB

1. Extrahiere Vokabular V aus Trainingskorpus
2. Berechne $P(c)$ für alle c aus dem Trainingskorpus:

$$P(c) = \frac{\#docs\ of\ class\ c}{total\ \#\ of\ docs\ in\ training}$$

3. Berechne $P(w_i|c) := P(X_l = w_i|c)$ für alle w_i im Vokabular und alle c:
   1. Konkateniere alle Trainingsdokumente der Klasse c zu einem langen Dokument
   2. $$P(w_i|c) = \frac{n_i^c}{n^c}$$
      wobei n_i^c Frequenz von w_i in langem Dokument und n^c Länge des langen Dokuments
   3. Mit Smoothing:

$$P(w_i|c) = \frac{n_i^c + 1}{n^c + |V|}$$

Testen des multinomialen NB

$$C_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i \in positions} P(w_i|c)$$

wobei wir über alle Positionen im Testdokument iterieren

Multinomialer Naive Bayes: Beispiel Training

	docID	words in doc	in c=China?
Training set	1	Chinese Bejing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
Testing	5	Chinese Chinese Chinese Tokyo Japan	?

Lernphase: extrahiere Vokabular, schätze $P(c)$ und die bedingten Wahrscheinlichkeiten $p(w|c)$ im Trainingsset

$p(China = yes) = \frac{3}{4}$, $p(China = no) = \frac{1}{4}$

$p(Chinese|China = yes) := P(X_l = Chinese|China = yes) = \frac{5+1}{8+6} = \frac{3}{7}$

$p(Tokyo|China = yes) = \frac{0+1}{8+6} = \frac{1}{14}$

$p(Japan|China = yes) = \frac{0+1}{8+6} = \frac{1}{14}$

$p(Chinese|China = no) = \frac{1+1}{3+6} = \frac{2}{9}$

$p(Tokyo|China = no) = \frac{1+1}{3+6} = \frac{2}{9}$

$p(Japan|China = no) = \frac{1+1}{3+6} = \frac{2}{9}$

Multinomialer Naive Bayes: Beispiel Testing

Testphase für Testdokument 5 *Chinese Chinese Chinese Tokyo Japan*.

$$C_{NB} = \operatorname*{argmax}_{c \in C} P(c) \prod_{i \in positions} P(w_i|c) = \operatorname*{argmax}_{c \in C} [\log P(c) + \sum_{i \in positions} \log P(w_i|c)]$$

$$P(China = yes|d) \propto \frac{3}{4} \cdot (\frac{3}{7})^3 \cdot \frac{1}{14} \cdot \frac{1}{14} = 0.0003$$
$$P(China = no|d) \propto \frac{1}{4} \cdot (\frac{2}{9})^3 \cdot \frac{2}{9} \cdot \frac{2}{9} = 0.0001$$

Zusammenfassung Multinomialer Naive Bayes

- Benutzt Unabhängigkeitsannahmen
- BoW: ignoriert Wortpositionen beim Schätzen
- Training: sieht alle Trainingsdokumente einer Klasse als ein langes Dokument
- In Training und Testing: Wortfrequenzen wichtig!
- Ignoriert beim Testen Vokabular, das nicht im Testdokument vorkommt

Multivariate Bernoulli Naive Bayes

Immer noch BoW und Unabhängigkeitsannahmen, aber nun nur Wortvorkommen ohne Häufigkeit

Fehlende Grafik

Binomialer Naive Bayes: Die Herleitung

$$
\begin{aligned}
C_{NB_{Bi}} &= \operatorname*{argmax}_{c \in C} P(c|d) && (6)\\
&= \operatorname*{argmax}_{c \in C} \frac{P(d|c)P(c)}{P(d)} && (7)\\
&= \operatorname*{argmax}_{c \in C} P(d|c)P(c) && (8)\\
&= \operatorname*{argmax}_{c \in C} P(e_1, \ldots, e_{|v|}|c)P(c) && (9)\\
&= \operatorname*{argmax}_{c \in C} P(c) \prod_{w_i \in V} P(e_i|c) && (10)
\end{aligned}
$$

wobei e_1 = yes/no, je nachdem ob w_1 im Dokument vorkommt

Wie smoothed man jetzt?

Multivariate Bernoulli Naive Bayes: Beispiel Training

	docID	words in doc	in c=China?
Training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
Testing	5	Chinese Chinese Chinese Tokyo Japan	?

$p(China = yes) = \frac{3}{4}$, $p(China = no) = \frac{1}{4}$

$p(Chinese|China = yes) := p(Chinese = yes|China = yes) = \frac{3+1}{3+2} = \frac{4}{5}$

$p(Chinese = no|China = yes) = \frac{0+1}{3+2} = \frac{1}{5}$

$p(Japan|China = yes) = p(Tokyo|China = yes) = \frac{0+1}{3+2} = \frac{1}{5}$

$p(Bejing|China = yes) = p(Macao|China = yes) =$

$p(Shanghai|China = yes) = \frac{1+1}{3+2} = \frac{2}{5}$

$p(Chinese|China = no) = \frac{1+1}{1+2} = \frac{2}{3}$

$p(Japan|China = no) = p(Tokyo|China = no) = \frac{1+1}{1+2} = \frac{2}{3}$

$p(Bejing|China = no) = p(Macao|China = no) =$

$p(Shanghai|China = no) = \frac{0+1}{1+2} = \frac{1}{3}$

Multivariate Bernoulli Naive Bayes: Beispiel Training

	docID	words in doc	in c=China?
Training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
Testing	5	Chinese Chinese Chinese Tokyo Japan	?

$p(China = yes) = \frac{3}{4}$, $p(China = no) = \frac{1}{4}$

$p(Chinese|China = yes) := p(Chinese = yes|China = yes) = \frac{3+1}{3+2} = \frac{4}{5}$

$p(Chinese = no|China = yes) = \frac{0+1}{3+2} = \frac{1}{5}$

$p(Japan|China = yes) = p(Tokyo|China = yes) = \frac{0+1}{3+2} = \frac{1}{5}$

$p(Bejing|China = yes) = p(Macao|China = yes) =$

$p(Shanghai|China = yes) = \frac{1+1}{3+2} = \frac{2}{5}$

$p(Chinese|China = no) = \frac{1+1}{1+2} = \frac{2}{3}$

$p(Japan|China = no) = p(Tokyo|China = no) = \frac{1+1}{1+2} = \frac{2}{3}$

$p(Bejing|China = no) = p(Macao|China = no) =$

$p(Shanghai|China = no) = \frac{0+1}{1+2} = \frac{1}{3}$

Multivariate Bernoulli Naive Bayes: Beispiel Training

	docID	words in doc	in c=China?
Training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
Testing	5	Chinese Chinese Chinese Tokyo Japan	?

$p(China = yes) = \frac{3}{4}$, $p(China = no) = \frac{1}{4}$

$p(Chinese|China = yes) := p(Chinese = yes|China = yes) = \frac{3+1}{3+2} = \frac{4}{5}$
$p(Chinese = no|China = yes) = \frac{0+1}{3+2} = \frac{1}{5}$
$p(Japan|China = yes) = p(Tokyo|China = yes) = \frac{0+1}{3+2} = \frac{1}{5}$
$p(Bejing|China = yes) = p(Macao|China = yes) =$
$p(Shanghai|China = yes) = \frac{1+1}{3+2} = \frac{2}{5}$
$p(Chinese|China = no) = \frac{1+1}{1+2} = \frac{2}{3}$
$p(Japan|China = no) = p(Tokyo|China = no) = \frac{1+1}{1+2} = \frac{2}{3}$
$p(Bejing|China = no) = p(Macao|China = no) =$
$p(Shanghai|China = no) = \frac{0+1}{1+2} = \frac{1}{3}$

Multivariate Bernouilli Naive Bayes: Beispiel Testing

Testphase für Testdokument 5 *Chinese Chinese Chinese Tokyo Japan*.

$$C_{NB_{Bi}} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{w_i \in V} P(e_i|c) = \underset{c \in C}{\operatorname{argmax}} [\log P(c) + \sum_{w_i \in V} \log P(e_i|c)]$$

$$\begin{aligned}
P(China = yes|d) &\propto P(c) \cdot P(Chinese|China = yes) \cdot P(Japan|China = yes) \\
&\cdot P(Tokyo|China = yes) \\
&\cdot (1 - P(Bejing|China = yes)) \cdot (1 - P(Shanghai|China = ye \\
&\cdot (1 - P(Macao|China = yes)) \\
&= \frac{3}{4} \cdot \frac{4}{5} \cdot \frac{1}{5} \cdot \frac{1}{5} \cdot (1 - \frac{2}{5}) \cdot (1 - \frac{2}{5}) \cdot (1 - \frac{2}{5}) \\
&= 0.005
\end{aligned}$$

Auf gleiche Weise $P(China = no|d) = 0.022$

Bernoulli vs Multinomial

	multinomial	binomial
Zufallsvariable	$X = w$, wenn w an pos X	$e_w = 1$, wenn w in Doc
doc rep	Sequenz von Worthäuf.	Sequenz von 0, 1
Mehrfachvork..	ja	ignoriert
Doklänge	gut für länger	nur für kurze
$\|V\|$	auch für groß	besser klein
Schätzung f. *the*	$p(X = the\|c) = 0.05$	$p(e_{the} = 1\|c) = 1.0$

Zusammenfassung NB

- NB sehr schnell, braucht wenig Speicher
- Multinomial: gut beim Ignorieren irrelevanter Merkmale
- NB gut, wenn Merkmale gleich wichtig
- NB gute Baseline für Textklassifikation

Eine Warnung: Jurafsky und Martin besprechen multinomial, aber nicht Bernoulli, sondern statt dem letzteren eine Mischform aus multinomial und Bernoulli, die multinomial binary genannt wird.

Übersicht

Annahmen

- Trainingsdaten: Korpus mit Instanzen, von denen wir Klassifikation wissen, und aus denen wir Klassifizierer lernen
- Development Set: Korpus mit Instanzen, von denen wir Klassifikation wissen. Zum Schätzen von Hyperparametern (z.B. Smoothing)
- Testset: Korpus mit Instanzen, von denen “wir” Klassifikation wissen, aber nicht während des Trainings benutzen. Nur zur Evaluation.

Accuracy

Unter der Annahme, dass jede Instanz nur zu einer Klasse gehört:

- Sei N die Anzahl aller Instanzen (im Testset)
- Sei G die Anzahl aller richtig klassifizierten Instanzen
- $accuracy = \frac{G}{N}$
- $error\ rate = 1 - accuracy$

Probleme?

Contingency table/confusion matrix

Wenn man eine einzelne Klasse K betrachtet:

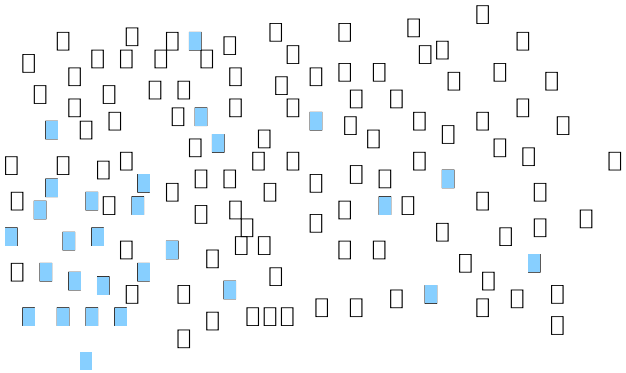
	wirklich K	nicht K	
klass. als K	A = tp	B = fp	# positives= $A+B$
klass. als $nicht-K$	C =fn	D = tn	# negatives = $C+D$
	$\|K\| = A+C$	$\|nicht\ K\| = B+D$	A+B+C+D=N

tp : true positive
fp: false positive
tn: true negative
fn: false negative

- Accuracy: $\frac{tp+tn}{tp+fp+tn+fn}$
- $error\ rate = 1 - Accuracy$
- Precision_K: $\frac{tp}{tp+fp} = \frac{A}{A+B}$
- Recall_K: $\frac{tp}{tp+fn} = \frac{A}{A+C}$

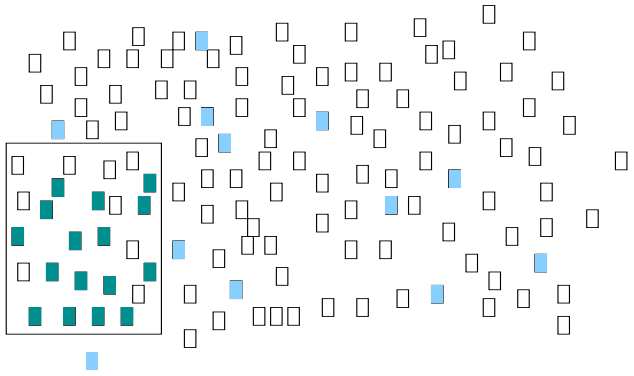
Beispiel

Wir wollen Hate Speech klassifizieren aus einer Menge von Tweets.
(Frage: Ist uns Precision oder Recall wichtiger?)



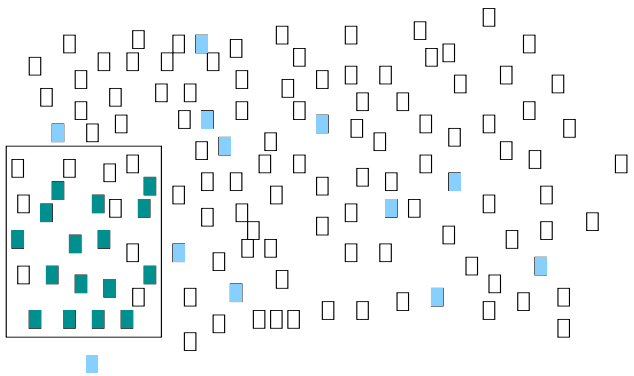
- Alle Tweets: A+B+C+D = 130
- Wirklich Hate Speech: A+C = 28

Recall and Precision: System 1



- System 1 klassifiziert 25 Tweets als Hate Speech: A+B = 25
- Richtige und gefundene Hate Speech: A = 16
- Wirklich Hate Speech: A+C = 28

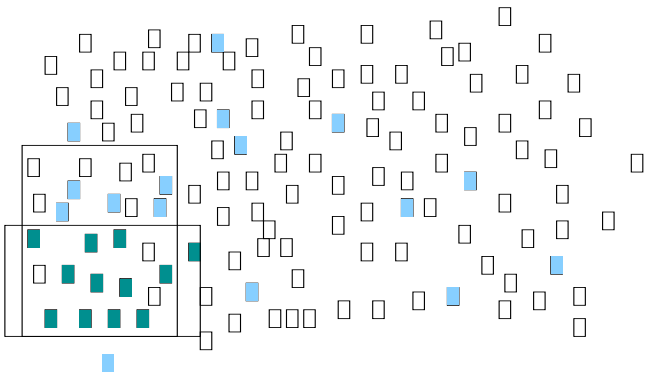
Recall and Precision: System 1



$$R_1 = \frac{A}{A+C} = \frac{16}{28} = .57$$
$$P_1 = \frac{A}{A+B} = \frac{16}{25} = .64$$

- System 1 klassifiziert 25 Tweets als Hate Speech: A+B = 25
- Richtige und gefundene Hate Speech: A = 16
- Wirklich Hate Speech: A+C = 28

Recall and Precision: System 2



- System 2 klassifiziert 15 Tweets als Hate Speech: A+B = 15
- Richtige und gefundene Hate Speech: A = 12
- Wirklich Hate Speech: A+C = 28
- Berechnen Sie Precision und Recall von System 2

Recall/Precision tradeoff

- Was ist der Recall, wenn man alle Tweets als Hate Speech klassifiziert?
- Wenn man nie etwas als Hate Speech klassifiziert?
- Und die Precision?
- Wie kombiniert man Precision P und Recall R in einer Zahl?

F-measure (harmonisches Mittel)

$$\text{F-measure} = \frac{2 \cdot P \cdot R}{P + R}$$

Warum nimmt man nicht einfach das arithmetische Mittel $\frac{P+R}{2}$?

Recall/Precision tradeoff

- Was ist der Recall, wenn man alle Tweets als Hate Speech klassifiziert?
- Wenn man nie etwas als Hate Speech klassifiziert?
- Und die Precision?
- Wie kombiniert man Precision P und Recall R in einer Zahl?

F-measure (harmonisches Mittel)

$$\text{F-measure} = \frac{2 \cdot P \cdot R}{P + R}$$

Warum nimmt man nicht einfach das arithmetische Mittel $\frac{P+R}{2}$?

Recall/Precision tradeoff

- Was ist der Recall, wenn man alle Tweets als Hate Speech klassifiziert?
- Wenn man nie etwas als Hate Speech klassifiziert?
- Und die Precision?
- Wie kombiniert man Precision P und Recall R in einer Zahl?

F-measure (harmonisches Mittel)

F-measure= $\frac{2 \cdot P \cdot R}{P+R}$

Warum nimmt man nicht einfach das arithmetische Mittel $\frac{P+R}{2}$?

Mehr als zwei Klassen: any-of Klassifikation

- Für jede Klasse $c \in C$: baue binären Klassifizierer γ_c, um c von allen anderen Klassen zu unterscheiden
- Geg. Testdoc d, evaluiere die Klassenzugehörigkeit mit jedem γ_c. Dann gehört d zu jeder Klasse c für die γ_c "ja" zurückgibt.

Beispiel für ein *any-of* Datenset: Reuters-21578 dataset

- 21,578 docs
- ModApte split: 9,603 training, 3,299 test docs
- 118 Kategorien: Artikel kann in mehr als einer Kategorie sein
- Nur um die 10 Kategorien sind groß

class	#train	#test	class	#train	#test
earn	2877	1087	trade	369	119
acquisitions	1650	179	interest	347	131
money-fx	538	179	ship	197	89
grain	433	149	wheat	212	71
crude	389	189	corn	182	56

- das durschnittliche Dokument gehört zu 1.24 Klassen

Ein typisches Reuters Dokument

```
<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-
NEWID="798">
<DATE> 2-MAR-1987 16:51:43.42</DATE>
<TOPICS><D>livestock</D><D>hog</D></TOPICS>

<TITLE>AMERICAN PORK CONGRESS KICKS OFF TOMORROW</TITLE>

<DATELINE> CHICAGO, March 2 - </DATELINE><BODY>The American
Pork Congress kicks off tomorrow,
March 3, in Indianapolis with 160 of the nations pork produc
from 44 member states determining industry positions
on a number of issues, according to the National
Pork Producers Council, NPPC.
Delegates to the three day Congress will be considering
26 resolutions concerning various issues, including the futu
direction of farm policy and the tax law
 as it applies to the agriculture sector. The delegates will
endorse concepts of a national PRV (pseudorabies virus)
```

Mehr als zwei Klassen: one-of Klassifizierer

Bei one-of Klassifikation schließen sich die Klassen gegenseitig aus

- Für jede Klasse $c \in C$: baue Klassifizierer γ_c wie auf any-of
- Geg. Testdok. d, evaluiere die Klassenzugehörigkeit mit jedem γ_c. Dann gehört d zu der Klasse mit dem maximalem Score.

Evaluation bei mehreren Klassen: Precision/Recall per Klasse

One-of-Problem: Klassifiziere Emails als $\{urgent, normal, spam\}$.

Konfusionsmatrix und precision/recall per Klasse:

		wirklich			
		urgent	normal	spam	
System	urgent	8	10	1	$P_u = \frac{8}{19} = 0.42$
	normal	5	60	50	$P_n = \frac{60}{115} = 0.52$
	spam	3	30	200	$P_s = \frac{200}{233} = 0.86$
		$R_u = \frac{8}{16} = 0.5$	$R_n = \frac{60}{100} = 0.6$	$R_s = \frac{200}{251} = 0.79$	

Auch F-measure kann per Klasse normal berechnet werden

Macro-averaged Precision/Recall

Macro-averaging

Wir nehmen an alle m Klassen sind gleich wichtig und mitteln Precision/Recall einfach über alle Klassen.

$$Macro-averaged\ precision = \frac{\sum_{i=1}^{m} P_i}{m}$$

$$Macro-averaged\ recall = \frac{\sum_{i=1}^{m} R_i}{m}$$

$$Macro-averaged\ Fmeasure = \frac{\sum_{i=1}^{m} F_i}{m}$$

Im Beispiel:

$$Macro-averaged\ precision = \frac{0.42+0.52+0.86}{3} = 0.6$$

$$Macro-averaged\ recall = \frac{0.5+0.6+0.79}{3} = 0.63$$

Micro-averaged Precision/Recall

Eine Tabelle per Klasse:

	wirklich	
	urg.	nicht
kl. urg.	8	11
kl. nicht	8	340

	wirklich	
	norm.	nicht
kl. norm.	60	55
kl. nicht	40	212

	wirklich	
	spam	nicht
kl. spam	200	33
kl. nicht	51	83

Pooled Matrix:

	wirklich	
	true	nicht
kl. true	268	99
kl. nicht	99	635

Micro-averaging mit Pooled Matrix

Pooled Matrix:

	wirklich	
	true	nicht
kl. true	268	99
kl. nicht	99	635

$$Micro-averaged\ Precision = \frac{268}{268+99} = 0.73$$

$$Micro-averaged\ Recall = \frac{268}{268+99} = 0.73$$

Allgemein für m Klassen:

$$Micro-averaged\ Precision = \frac{\sum_{i=1}^{m} tp_i}{\sum_{i=1}^{m} tp_i + \sum_{i=1}^{m} fp_i}$$

$$Micro-averaged\ Recall = \frac{\sum_{i=1}^{m} tp_i}{\sum_{i=1}^{m} tp_i + \sum_{i=1}^{m} fn_i}$$

Zwischenfazit Evaluationsmaße

- Accuracy für alle Evaluationen, bei denen sich Klassen gegenseitig ausschließen. Hat Probleme mit unbalancierter Klassenverteilung.
- Precision/Recall erlauben klassenspezifische Evaluation
- F-measure erlaubt eine (konservative) klassenspezifische Kombination von Precision und Recall
- Macro-averaged Precision/Recall sieht alle **Klassen** als gleich wichtig an und mittelt einfach deren Precision/Recall. Kleine, oft schlecht erkannte Klassen ziehen das Macro-average herunter.
- Micro-averaged Precision/Recall sehen alle **Instanzen** als gleich wichtig an. Wird errechnet aus allen tp/fn/fp in einer pooled matrix. Große Klassen geben den meisten Ausschlag.

Algorithmusvergleich

Wie vergleichen wir zwei Klassifikationsalgorithmen A und B?

Lösung

Berechne die Performanz von A und B auf dem gleichen Testdatensatz und berechne Unterschied in Performanz.

Beispiel: Algorithmus *A* hat eine accuracy von 80% bei der Hate Speech Erkennung, Algorithmus *B* eine accuracy von 60% auf dem gleichen Datensatz.

Heisst dies, dass *A* besser als *B* ist? Wovon hängt die Antwort auf diese Frage ab?

Grundidee Signifikanztests

- Performanzunterschiede hängen vom Sampling des Testdatensatzes ab sowie von seiner Größe.
- **Nullhypothese** H_0: Kein Unterschied zwischen den Algorithmen
- Idee: Ein Ergebnis/Performanzunterschied, der unter der Nullhypothese sehr unwahrscheinlich ist, ist gute Evidenz, dass die Nullhypothese verworfen werden kann

p-Value

p-value/p-Wert

Unter der Annahme, dass die Null-Hypothese richtig ist, ist der p-Wert die Wahrscheinlichkeit, dass das Sampleresultat so extrem (oder extremer) ausfallen würde wie das tatsächlich beobachtete Ergebnis.

- Idee: wenn wir das Experiment mit vielen Samples (der gleichen Größe) wiederholen würden, wie oft würden wir, wenn die Nullhypothese wahr ist, einen solchen oder einen noch extremeren Performanzunterschied bekommen
- Je kleiner der p-Wert, desto stärker die Evidenz gegen die Nullhypothese
- Ist der p-wert kleiner als α, so sagt man, dass das Ergebnis **statistisch signifikant** auf dem Niveau α ost.

Algorithmusvergleich: Signifikanztest des Maßes Accuracy mit McNemar Test

Zwei Algorithmen A1 und A2:

	A1 korrekt	A1 inkorrekt
A2 korrekt	a= 120	b= 100
A2 inkorrekt	c=40	d=400

Nullhypothese: $b = c$

$$\chi^2 = \frac{(b-c)^2}{b+c} = \frac{(100-40)^2}{100+40} = \frac{3600}{140} = 25.7$$

Kann mit χ^2 Verteilung auf p-Wert getestet werden.

Bootstrapping Test (optional)

- McNemar test für Accuracy gut, nicht leicht anwendbar auf precision/recall/F-measure
- Immer anwendbar: Bootstrapping test (siehe J&M, Kapitel 4)

Baselines und Upper Bounds

Baselines

Einfache Algorithmen, die mein Algorithmus immer schlagen sollte.
Beispiel:

- Zufällige Klassenauswahl
- Most Frequent Class Baseline
- Einfache regelbasierte Baselines
- ...

Upper Bounds

Abschätzung, wie schwierig mein Task ist. Oft menschliche Performanz bzw. Übereinstimmung zwischen zwei Menschen.

Wichtigkeit der Korpusgrößen: Diskussion

- Was passiert, wenn mein Trainingsset klein ist?
- Wie kann ich abschätzen, ob mein Trainingsset zu klein ist? (Learning Curves)
- Was passiert, wenn mein Testset klein ist?
- Wie kann ich mein Korpus optimal ausnutzen? Ist es am besten, immer mit 80-10-10 zu Training/Development/Test zu splitten?

Crossvalidation/Kreuzvalidierung

Cross-Validation

- Teile Daten zufällig in k Teilmengen (folds) von gleicher Größe (z.B. $k = 10$)
- Trainiere Modell auf $k - 1$ folds, nimm ein fold fürs Testing
- Wiederhole k mal
- Berechne durschnittliche Performanz auf k Testsets.

Crossvalidation

Beispiel: 100 Instanzen, 5-fold crossvalidation

test set	training set

$i_1, i_2, i_3, i_4, \ldots i_{20}$
$i_{21}, i_{22}, i_{23}, i_{24} \ldots i_{40}$
$i_{41}, i_{42}, i_{43}, i_{44}, \ldots i_{60}$
$i_{61}, i_{62}, i_{63}, i_{64} \ldots i_{80}$
$i_{81}, i_{82}, i_{83}, i_{84} \ldots i_{100}$

Fehlerrate: 20%

Crossvalidation

Beispiel: 100 Instanzen, 5-fold crossvalidation

test set	training set

$i_1, i_2, i_3, i_4, \ldots i_{20}$
$i_{21}, i_{22}, i_{23}, i_{24} \ldots i_{40}$
$i_{41}, i_{42}, i_{43}, i_{44}, \ldots i_{60}$
$i_{61}, i_{62}, i_{63}, i_{64} \ldots i_{80}$
$i_{81}, i_{82}, i_{83}, i_{84} \ldots i_{100}$

Fehlerrate: 25%

Crossvalidation

Beispiel: 100 Instanzen, 5-fold crossvalidation

test set	training set

$i_1, i_2, i_3, i_4, \ldots i_{20}$
$i_{21}, i_{22}, i_{23}, i_{24} \ldots i_{40}$
$i_{41}, i_{42}, i_{43}, i_{44}, \ldots i_{60}$
$i_{61}, i_{62}, i_{63}, i_{64} \ldots i_{80}$
$i_{81}, i_{82}, i_{83}, i_{84} \ldots i_{100}$

Fehlerrate: 15%

Crossvalidation

Beispiel: 100 Instanzen, 5-fold crossvalidation

test set	training set

$i_1, i_2, i_3, i_4, \ldots i_{20}$
$i_{21}, i_{22}, i_{23}, i_{24} \ldots i_{40}$
$i_{41}, i_{42}, i_{43}, i_{44}, \ldots i_{60}$
$i_{61}, i_{62}, i_{63}, i_{64} \ldots i_{80}$
$i_{81}, i_{82}, i_{83}, i_{84} \ldots i_{100}$

Fehlerrate: 10%

Crossvalidation

Beispiel: 100 Instanzen, 5-fold crossvalidation

test set	training set

$i_1, i_2, i_3, i_4, \ldots i_{20}$
$i_{21}, i_{22}, i_{23}, i_{24} \ldots i_{40}$
$i_{41}, i_{42}, i_{43}, i_{44}, \ldots i_{60}$
$i_{61}, i_{62}, i_{63}, i_{64} \ldots i_{80}$
$i_{81}, i_{82}, i_{83}, i_{84} \ldots i_{100}$

Fehlerrate: 30%

durchschnittliche Fehlerrate: 20%

Bias oder “Size is not everything”

- **Sampling Bias/Stichprobenverzerrung:** Korpus so gesammelt, dass manche Instanzen der Gesamtpopulation durch die Samplingmethode eine größere Wahrscheinlichkeit haben, im Korpus zu landen. Kein random sampling. Kann zu systematischer Über- bzw Unterrepräsentation führen.

  Die Wörter mit dem höchsten PMI mit der Klasse *hate speech* in zwei Trainingskorpora (Wiegand et al., 2019)

Founta	Waseem
bitch	commentator
niggas	comedian
motherfucker	football
fucking	announcer
nigga	pedophile
idiot	mankind
asshole	sexist
fuck	sport

- **Labeling Bias** der Annotierer

Fazit Evaluation

- Verschiedene Maße: accuracy, precision/recall/F-measure per class, macro-averaged vs. micro-averaged precision/recall/F-measure
- Anwendbar auf beliebig viele Klassen
- Signifikanztests
- Performanz auf dem Trainingsset ist kein guter Indikator für die Performanz auf ungesehenen Daten.
- Das Testset kann ein **unabhängiges Sample** sein oder per **Kreuzvalidierung** entstehen.
- Kreuzvalidierung erlaubt maximale Ausnutzung des Korpus und Varianzabschätzung.
- Größe, Sampling und Labeling des Trainings- und Testsets spielen eine wichtige Rolle (Bias!)

Fazit Evaluation

- Verschiedene Maße: accuracy, precision/recall/F-measure per class, macro-averaged vs. micro-averaged precision/recall/F-measure
- Anwendbar auf beliebig viele Klassen
- Signifikanztests
- Performanz auf dem Trainingsset ist kein guter Indikator für die Performanz auf ungesehenen Daten.
- Das Testset kann ein **unabhängiges Sample** sein oder per **Kreuzvalidierung** entstehen.
- Kreuzvalidierung erlaubt maximale Ausnutzung des Korpus und Varianzabschätzung.
- Größe, Sampling und Labeling des Trainings- und Testsets spielen eine wichtige Rolle (Bias!)

Übersicht

Gesamtablauf

- Sammle und annotiere Korpus
- Lösche irrelevante Teile der Texte (Bilder, Text-markup, Online-Kommentare unter Text etc.), wenn nötig
- Teile in Trainings/Validierung/Testkorpus oder benutze Kreuzvalidierung
- Entscheide Klassifikationsmodell
- Trainiere:
  - Für ein BoW Model muss man natürlich Worte bestimmen $\longrightarrow$ Tokenisierung
  - Evtl. weitere Normalisierungen wie stemming
  - **Merkmalsselektion**
  - Schätzungen, inklusive Erlernen von Metaparametern auf validation set
- Teste auf Testset

Merkmale

BoW:

- Nur Worte
- bis jetzt alle Wörter im Text ODER wir haben offen gelassen, wie man das Vokabular auswählt
- Warum könnte es besser sein, nicht alle Wörter zu benutzen?
- Wie könnte man ein Vokabular auswählen?

Warum Merkmalsselektion?

- Textkollektionen haben sehr große Merkmalsanzahl
- Einige Klassifizierer können mit großer Merkmalsanzahl nicht umgehen: NB aber schon
- Reduziert Trainingszeit
- Kann Generalisierung verbessern
  - eliminiert Rauschen
  - vermeidet overfitting
- Bernoulli NB besonders anfällig für Rauschen
- Eine Option: eliminiere alle Funktionswörter (stop words)

Grundidee

Overfitting

- Algorithmus passt sich den Trainingsdaten zu stark an.
- Kann zu mangelnder Generalisierung führen.
- Verschiedene Möglichkeiten, dies zu vermeiden: Smoothing, Merkmalsselektion, Algorithmusanpassungen ...

Feature selection/Merkmalsselektion

Berechne ein Maß $A(w, c)$ für jedes Wort und jede Klasse c und selektiere die k Terme mit den höchsten Werten von $A(w, c)$

Wir konzentrieren uns auf zwei Maße und zeigen diese für zwei Klassen.

Häufigkeit als Merkmalsselektion

- Benutze einfach die k häufigsten Wörter
- Exkludiert informative Worte wie *super-entertaining*
- Warum ist es in der Praxis oft trotzdem ein gutes Verfahren?

Mutual Information zur Merkmalsselektion

Wieviel trägt ein Term zu korrekter Klassifizierung bei?

- Idee: vergleiche beobachtete Wahrscheinlichkeiten mit erwarteten Wahrscheinlichkeiten, wenn Term und Klasse unabhängig werden
- Formel:

$$I(T;C) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} P(T = e_t, C = e_c) \log \frac{P(T = e_t, C = e_c)}{P(T = e_t) P(C = e_c)}$$

Ein Assoziationsmaß: Mutual Information

$$I(T;C) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} P(T = e_t, C = e_c) \log \frac{P(T = e_t, C = e_c)}{P(T = e_t)P(C = e_c)}$$

Beispiel mit der Klasse `poultry` und dem Term *export* im reuters-Korpus

	$e_c = e_{poultry} = 1$	$e_c = e_{poultry} = 0$	Marginals
$e_t = e_{export} = 1$	49	27,652	27, 701
$e_t = e_{export} = 0$	141	774,106	774, 247
	190	801,758	801,948

Einige Beispielberechnungen

- $P(e_c = 1, e_t = 1) = \frac{49}{801948}$
- $P(e_c = 1) = \frac{190}{801948}$
- $P(e_t = 0) = \frac{774{,}347}{801{,}948}$

Ein Assoziationsmaß: Mutual Information

Zusammen:

$$
\begin{aligned}
I(T;C) &= \frac{49}{801,948} \log \frac{801,948 \cdot 49}{(49+27,652)(49+141)} \\
&+ \frac{141}{801,948} \log \frac{801,948 \cdot 141}{(141+774,106)(49+141)} \\
&+ \frac{27,652}{801,948} \log \frac{801,948 \cdot 27,652}{(49+27,652)(27,652+774,106)} \\
&+ \frac{774,107}{801,948} \log \frac{801,948 \cdot 774,106}{(141+774,106)(27,652+774,106)} \\
&= 0.0001105
\end{aligned}
$$

Mutual information für drei Reuters-Klassen

UK class	sports class	poultry class
london 0.192	soccer 0.0682	poultry 0.0013
uk 0.0755	cup 0.0515	meat 0.0008
british 0.0596	match 0.0441	chicken 0.0006
stg 0.0555	matches 0.0408	agriculture 0.0005
britain 0.0469	played 0.0388	avian 0.00004
plc 0.0357	league 0.0386	broiler 0.0003
england 0.0238	beat 0.0301	

Daten aus Manning et al: Introduction to IR

χ^2 Merkmalsselelektion (Optional)

- Bestimme Unabhängigkeit der beiden Ereignisse: Vorkommen des terms (Merkmals) und Vorkommen der Klasse
- Vergleiche wirklich beobachtetes Vorkommen mit erwartetem Vorkommen, wenn die beiden Ereignisse unabhängig wären
- Beruht ebenfalls auf der 2x2 Matrix

	$e_c = e_{poultry} = 1$	$e_c = e_{poultry} = 0$	Marginals
$e_t = e_{export} = 1$	49	27,652	27, 701
$e_t = e_{export} = 0$	141	774,106	774, 247
	190	801,758	801,948

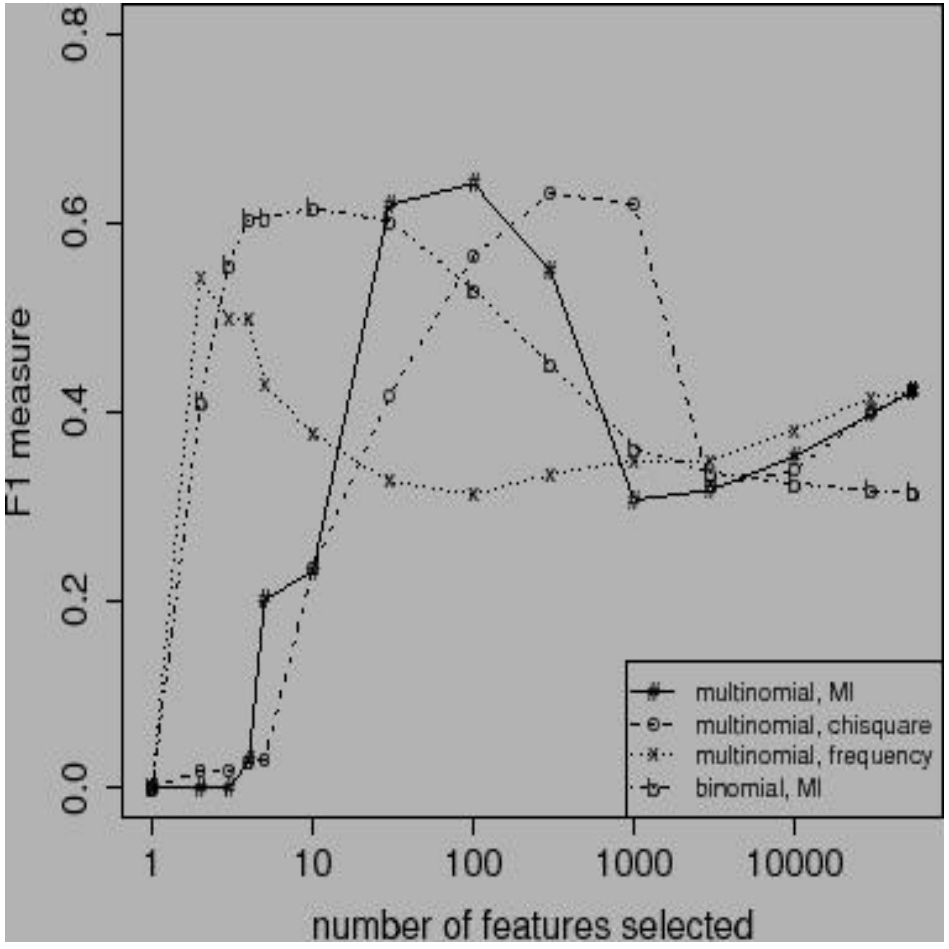
bzw

	$e_c = e_{poultry} = 1$	$e_c = e_{poultry} = 0$	Marginals
$e_t = e_{export} = 1$	N11	N10	N1.
$e_t = e_{export} = 0$	N01	N00	N0.
	N.1	N.0	N

Hierzu müssen Sie Kapitel 13.5.2 in Manning et al, Introduction to IR lesen. Online auch unter `https://nlp.stanford.edu/IR-book/`.

Effekt auf Performanz

Für 5 Klassen yes/no (F-measure über 5 Klassen) aus Manning, Raghavan, Schuetze: Introduction to Information retrieval (Figure 13.8)
100K Docs im Training und 100K im Testing



Übersicht

Sentimentklassifikation: Daumen hoch?

Original text

A good budget hotel' .... Price includes breakfast. Rooms are modern and of a reasonable size. The centre of Leeds is about a 15 min walk at the most. Hotel has bar area.

Würden Sie andere Merkmale als für Themenklassifikation benutzen?

Typisches one-of Datenset: 50,000 IMDB Movie reviews (25K pos, 25k neg)
`https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews`

Weitere Beispiele

- Hate Speech Classification:
  - https://hatespeechdata.com/
  - https://cloud.gate.ac.uk/shopfront/displayItem/gate-hate-generic
  - Hateful Memes Dataset: https://ai.facebook.com/blog/hateful-memes-challenge-and-data-set/
- Bias Klassifikation:
  https://www.thebipartisanpress.com/analyze-bias/
- Spamklassifikation
- Autorenidentifikation (Federalist Papers, Forensische Linguistik)
- Alters- und Geschlechtsidentifikation

Fazit

In der Praxis

- ist das Preprocessing (Tokenisierung etc) wichtig
- braucht man Merkmalsselektion sowie gutes Smoothing
- muss man den Algorithmus auf mehrere Klassen anpassen
- vorsichtig evaluieren
- nicht nur Wortmerkmale benutzen

References

- Jurafsky and Martin: 3rd online edition, Chapter 4
- *Manning, Raghavan and Schuetze: Introduction to Information Retrieval. Chapter 13
- Aufgabenblatt 6

Eine Warnung: Jurafsky und Martin besprechen multinomial, aber nicht Bernoulli, sondern statt dem letzteren eine Mischform aus multinomial und Bernoulli, die multinomial binary genannt wird.

Referenzen

- Wang, S. and Manning, C. D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In ACL 2012, 90–94.
- Wiegand et al (2019): Detection of Abusive Language: the Problem of Biased Datasets. In NAACL 2019.
- Dietterich (1998): Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*
- Berg-Kirkpatrick, T., Burkett, D., and Klein, D. (2012). An empirical investigation of statistical significance in NLP. In EMNLP 2012, 995–1005.
- Moore, D; Notz, W (2019): Statistics: Concepts and Controversies.