INF4080 – 2022 FALL

NATURAL LANGUAGE PROCESSING

Jan Tore Lønning

(Mostly Text) Classification, Naive Bayes

Lecture 3, 8 Sept.

Today - Classification

- Motivation
- Classification
- Naive Bayes classification
- NB for text classification
  - The multinomial model
  - The Bernoulli model
- Experiments: training, test and cross-validation
- Evaluation

Motivation

Positive or negative movie review?

- *unbelievably disappointing*

- *Full of zany characters and richly applied satire, and come great plot twists*

- *this is the greatest screwball comedy ever filmed*

- *It was pathetic. The worst part about it was the boxing scenes.*

From Jurafsky & Martin

What is the subject of this article?

MEDLINE Article


Brain and Cognition

Syntactic frame and verb bias in aphasia: Plausibility judgments of undergoer-subject sentences

?

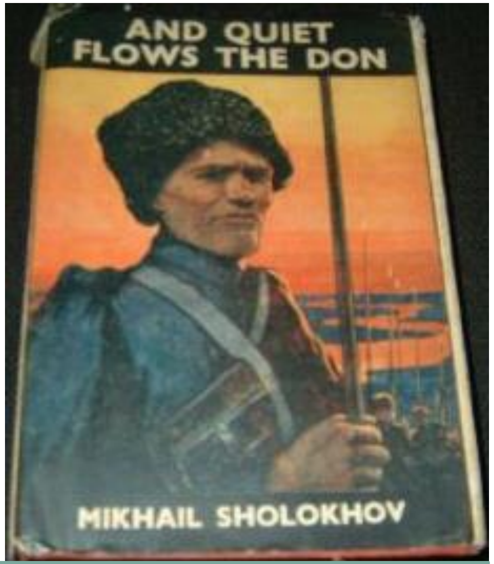
MeSH Subject Category Hierarchy

- Antagonists and Inhibitors
- Blood Supply
- Chemistry
- Drug Therapy
- Embryology
- Epidemiology
- …

From Jurafsky & Martin

Did Mikhail Sholokov write *And Quiet Flows the Don?*

- Sholokov, 1905-1984
- *And Quiet Flows the Don*
  - published 1928-1940
- Nobel prize, literature, 1965
- Authorship contested
  - e.g. Aleksandr Solzhenitsyn, 1974
- Geir Kjetsaa (UiO) et al, 1984
  - refuted the contestants
- Nils Lid Hjort, 2007, confirmed Kjetsaa by using sentence length and advanced statistics.
- https://en.wikipedia.org/wiki/Mikhail_Sholokhov



Kjetsaa according to Hjort

In addition to various linguistic analyses and several doses of detective work, quantitative data were gathered and organised, for example, relating to word lengths, frequencies of certain words and phrases, sentence lengths, grammatical characteristics, etc.

Classification

Classification

- Can be rule-based, but mostly machine learned
- Text classification is a sub-class

- Text classification examples:
  - Spam detection
  - Genre classification
  - Language identification
  - Sentiment analysis:
    - Positive-negative

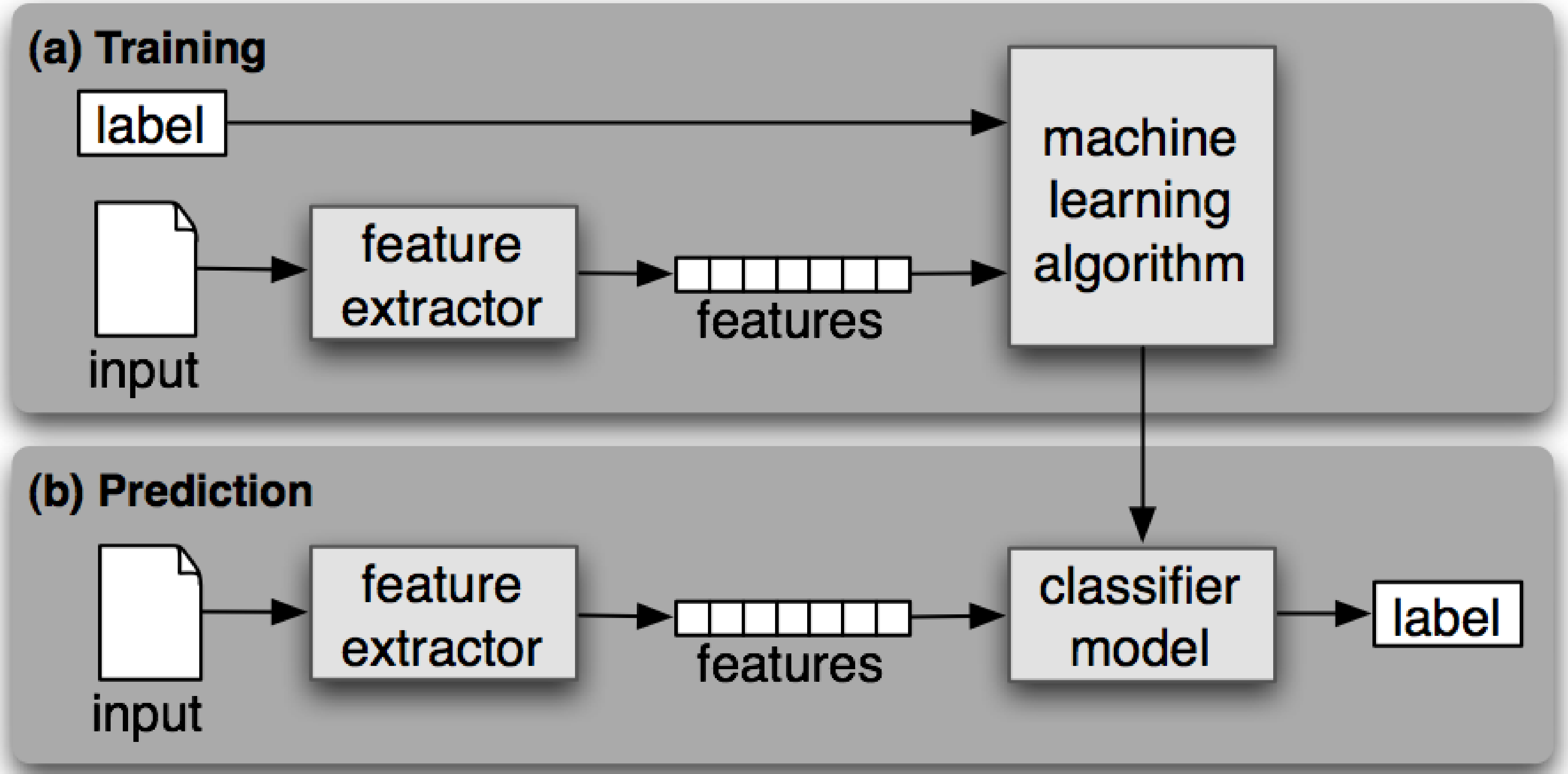
- Other types of classification:
  - Word sense disambiguation
  - Sentence splitting
  - Tagging
  - Named-entity recognition

Machine learning

1. Supervised
    1. Classification (categorical)
    2. Regression (numerical)
2. Unsupervised
3. Reinforcement learning

- Supervised:
    - Given classes
    - Given examples of correct classes
- Unsupervised:
    - Construct classes

Supervised classification



Supervised classification

- Given
  - a well-defined set of observations, O
  - a given set of classes, $C=\{c_1, c_2, \ldots, c_k\}$
- Goal: a classifier, γ, a mapping from O to C
- For supervised training one needs a set of pairs from OxC

Task	O	C
Spam classification	E-mails	Spam, no-spam
Language identification	Pieces of text	Arabian, Chinese, English, Norwegian, …
Word sense disambi-guation	Occurrences of ”bass”	Sense1, …, sense8

Features

- To represent the objects in O, extract a set of features
- Be explicit:
  - Which features
  - For each feature
    - The type
      - Categorical
      - Numeric (Discrete/Continuous)
    - The value space

O: person
Features:
- height
- weight
- hair color
- eye color
- …

O: email
Features:
- length
- sender
- contained words
- language
- …

Cf. First lecture
Classes and features are both attributes of the observations

Supervised classification

- A given set of classes, $C=\{c_1, c_2, \ldots, c_k\}$
- A well defined class of observations, O

- Some features $f_1, f_2, \ldots, f_n$
- For each feature: a set of possible values $V_1, V_2, \ldots, V_n$
- The set of feature vectors: $V= V_1\times V_2\times\ldots\times V_n$
- Each observation in O is represented by some member of V:
  - Written $(f_1=v_1, f_2=v_2, \ldots, f_n=v_n)$, or
  - $(v_1, v_2, \ldots, v_n)$, if we have decided the order

- A classifier, γ, can be considered a mapping from V to C

A variety of ML classifiers

- k-Nearest Neighbors
- Rocchio
- Naive Bayes
- Logistic regression (Maximum entropy)
- Support Vector Machines
- Decision Trees
- Perceptron
- Multi-layered neural nets ("Deep learning")

Naïve Bayes

Example: Jan. 2021 (invented example)

Professor, do you think I will enjoy IN3050?

I can give you a scientific answer using machine learning.

Baseline

- Survey
- Asked all the students of 2020
- 200 answered:
  - 130 yes
  - 70 no
- Baseline classifier:
  - Choose the majority class
  - Accuracy 0.65=65%

Yes,
you will like it.

Start by constructing a baseline

Example: Jan. 2021

Professor, do you think I will enjoy IN3050?

To answer that, I have to ask you some questions.

The 2020 survey (imaginary)

Asked each of the 200 students who completed 2020

- Did you enjoy the course?
  - Yes/no
- Do you like mathematics?
  - Yes/no
- Do you have programming experience?
  - None/some/good (= 3 or more courses)
- Have you taken advanced machine learning courses?
  - Yes/no
- And many more questions, but we have to simplify here

Results of the 2020 survey: a data set

Student no	Enjoy maths	Programming	Adv. ML	Enjoy
1	Y	Good	N	Y
2	Y	Some	N	Y
3	N	Good	Y	N
4	N	None	N	N
5	N	Good	N	Y
6	N	Good	Y	Y
….				

Summary of the 2020 survey

	A	B	C	D	E	F
1	programing	AdvML-course	Like maths	enjoyed	not enjoye	sum
2	good	yes	yes	3	10	13
3	good	yes	no	7	4	11
4	good	no	yes	50	4	54
5	good	no	no	40	4	44
6	some	yes	yes	4	1	5
7	some	yes	no	0	0	0
8	some	no	yes	11	9	20
9	some	no	no	10	24	34
10	none	yes	yes	1	2	3
11	none	yes	no	0	0	0
12	none	no	yes	2	5	7
13	none	no	no	2	7	9
14				130	70	200

Our new 2021 student

	A	B	C	D	E	F
1	programing	AdvML-course	Like maths	enjoyed	not enjoye	sum
2	good	yes	yes	3	10	13
3	good	yes	no	7	4	11
4	good	no	yes	50	4	54
5	good	no	no	40	4	44
6	some	yes	yes	4	1	5
7	some	yes	no	0	0	0
8	some	no	yes	11	9	20
9	some	no	no	10	24	34
10	none	yes	yes	1	2	3
11	none	yes	no	0	0	0
12	none	no	yes	2	5	7
13	none	no	no	2	7	9
14				130	70	200

But what should we say to a student with some programming background, and adv. ML course who does not like maths.?

- We ask our incoming new student the same three question
- From the table we can see e.g. that if:
  - she has good programming
  - no AdvML-course
  - does not like maths
- There is a $\frac{40}{44}$ chance she will enjoy the course

A little more formal

- We consider
  - $P(enjoy = yes \mid prog = good, AdvML = no, Maths = no)$ and
  - $P(enjoy = not \mid prog = good, AdvML = no, Maths = no)$
- and decide on the class which has the largest probability, in symbols
  - $argmax_{y \in \{yes, no\}} P(enjoy = y \mid prog = good, AdvML = no, Maths = no)$

Exponential growth

- In the example:
  - 2 binary features
  - 1 ternary feature
- How many different bins are there?
  -
- We could have considered the bins directly

- With 20 binary features, how many bins?
  -
- With 30,000 binary features (e.g., a vocabulary)?

Hence:

- apply Bayes theorem,
- make a simplifying assumption

Naive Bayes: Decision

$S = \{s_1, s_2, \ldots, s_k\}$ is the set of classes

- Given an observation
  - $\langle f_1 = v_1, f_2 = v_2, \ldots, f_n = v_n \rangle$
- Consider
  - $P(s_m \mid \langle f_1 = v_1, f_2 = v_2, \ldots, f_n = v_n \rangle)$ for each class s_m
- Choose the class with the largest value, in symbols

$$\arg\max_{s_m \in S} P(s_m \mid \langle f_1 = v_1, f_2 = v_2, \ldots, f_n = v_n \rangle)$$

- i.e. choose the class for which the observation is most likely

Naive Bayes: Model

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)}$$

- Bayes formula
  - $$P\left(s_m \mid \left\langle f_1=v_1, f_2=v_2, \ldots, f_n=v_n\right\rangle\right)=\frac{P\left(\left\langle f_1=v_1, f_2=v_2, \ldots, f_n=v_n\right\rangle \mid s_m\right)P(s_m)}{P\left(\left\langle f_1=v_1, f_2=v_2, \ldots, f_n=v_n\right\rangle\right)}$$
- Sparse data, we may not even have seen
  - $$\left\langle f_1=v_1, f_2=v_2, \ldots, f_n=v_n\right\rangle$$
- We assume (wrongly) independence
  - $$P\left(\left\langle f_1=v_1, f_2=v_2, \ldots, f_n=v_n\right\rangle \mid s_m\right)\approx\prod_{i=1}^{n}P\left(f_i=v_i \mid s_m\right)$$
- Putting together, choose
  - $$\arg\max_{s_m\in S} P\left(s_m \mid \left\langle f_1=v_1, f_2=v_2, \ldots, f_n=v_n\right\rangle\right)\approx\arg\max_{s_m\in S} P(s_m)\prod_{i=1}^{n}P\left(f_i=v_i \mid s_m\right)$$

Naive Bayes, Training 1

- Maximum Likelihood
- $$\hat{P}(s_m) = \frac{C(s_m, o)}{C(o)}$$
  - where $C(s_m, o)$ are the number of occurrences of observations o in class s_m
- Observe what we are doing:
  - We are looking for the true probability $P(s_m)$
  - $\hat{P}(s_m)$ is an approximation to this, our best guess from a set of observations
  - Maximum likelihood means that it is the model which makes the set of observations we have seen, most likely

Naive Bayes: Training 2

- Maximum Likelihood
- $$\hat{P}(f_i = v_i \mid s_m) = \frac{C(f_i = v_i, s_m)}{C(s_m)}$$
  - where $C(f_i=v_i, s_m)$ is the number of observations o
    - where the observation o belongs to class s_m
    - and the feature f_i takes the value v_i
  - $C(s_m)$ is the number of observations belonging to class s_m

Back to example

16	programming	enjoyed	num		^P(x | yes)
17	good	yes	100	/130=	0,7692308
18	some	yes	25	/130=	0,1923077
19	none	yes	5	/130=	0,0384615
20					^P(x | no)
21	good	no	22	/70=	0,3142857
22	some	no	34	/70=	0,4857143
23	none	no	14	/70=	0,2
24					
25	advanced	enjoyed	num		^P(x | yes)
26	yes	yes	15	/130=	0,1153846
27	no	yes	115	/130=	0,8846154
28					^P(x | no)
29	yes	no	17	/70=	0,2428571
30	no	no	53	/70=	0,7571429
31					
32	like maths	enjoyed	num		^P(x | yes)
33	yes	yes	71	/130=	0,5461538
34	no	yes	59	/130=	0,4538462
35					^P(x | no)
36	yes	no	31	/70=	0,4428571
37	no	no	39	/70=	0,5571429

- Collect the numbers
- Estimate the probabilities

	A	B	C	D	E	F
1	programing	AdvML-course	Like maths	enjoyed	not enjoye	sum
2	good	yes	yes	3	10	13
3	good	yes	no	7	4	11
4	good	no	yes	50	4	54
5	good	no	no	40	4	44
6	some	yes	yes	4	1	5
7	some	yes	no	0	0	0
8	some	no	yes	11	9	20
9	some	no	no	10	24	34
10	none	yes	yes	1	2	3
11	none	yes	no	0	0	0
12	none	no	yes	2	5	7
13	none	no	no	2	7	9
14				130	70	200

Back to example

16	programming	enjoyed	num		^P(x | yes)
17	good	yes	100	/130=	0,7692308
18	some	yes	25	/130=	0,1923077
19	none	yes	5	/130=	0,0384615
20					^P(x | no)
21	good	no	22	/70=	0,3142857
22	some	no	34	/70=	0,4857143
23	none	no	14	/70=	0,2
24					
25	advanced	enjoyed	num		^P(x | yes)
26	yes	yes	15	/130=	0,1153846
27	no	yes	115	/130=	0,8846154
28					^P(x | no)
29	yes	no	17	/70=	0,2428571
30	no	no	53	/70=	0,7571429
31					
32	like maths	enjoyed	num		^P(x | yes)
33	yes	yes	71	/130=	0,5461538
34	no	yes	59	/130=	0,4538462
35					^P(x | no)
36	yes	no	31	/70=	0,4428571
37	no	no	39	/70=	0,5571429

- $argmax_{c_m \in C} P(c_m) \prod_{i=1}^{n} P(f_i = v_i | c_m)$
- $P(yes) \times P(good|yes) \times P(A{:}\,no|yes) \times P(M{:}\,no|yes) = \frac{130}{200} \times \frac{100}{130} \times \frac{115}{130} \times \frac{59}{130} = 0.2$
- $P(no) \times P(good|no) \times P(A{:}\,no|no) \times P(M{:}\,no|no) = \frac{70}{200} \times \frac{22}{70} \times \frac{53}{70} \times \frac{39}{70} = 0.046$
- So we predict that the student will most probably enjoy the class
- Accuracy on training data: 75%
  - For each of 200 students, consider whether they are classified correctly
  - Compare to Baseline: 65%
  - Best classifier: 80%

Laplace-smoothing

- MLE-estimate: $P(w_i) = \frac{c_i}{N}$
- Laplace-estimate: $P_{\text{Laplace}}(w_i) = \frac{c_i + 1}{N + V}$
- Lidstone-smoothing: add *k*, e.g. 0.5: $\hat{P}(w_i) = \frac{c_i + k}{N + kV}$
- nltk.NaiveBayesClassifier uses Lidstone (0.5) as default

Laplace applied to example

$$P_{\text{Laplace}}(w_i) = \frac{c_i + 1}{N + V}$$

16	programming	enjoyed	num		^P(x | yes)
17	good	yes	100	/130=	0,7692308
18	some	yes	25	/130=	0,1923077
19	none	yes	5	/130=	0,0384615
20					^P(x | no)
21	good	no	22	/70=	0,3142857
22	some	no	34	/70=	0,4857143
23	none	no	14	/70=	0,2
24					
25	advanced	enjoyed	num		^P(x | yes)
26	yes	yes	15	/130=	0,1153846
27	no	yes	115	/130=	0,8846154
28					^P(x | no)
29	yes	no	17	/70=	0,2428571
30	no	no	53	/70=	0,7571429
31					
32	like maths	enjoyed	num		^P(x | yes)
33	yes	yes	71	/130=	0,5461538
34	no	yes	59	/130=	0,4538462
35					^P(x | no)
36	yes	no	31	/70=	0,4428571
37	no	no	39	/70=	0,5571429

- $\hat{P}(prog = good|yes) = \frac{100+1}{130+3}$
- $\hat{P}(prog = some|yes) = \frac{25+1}{130+3}$
- $\hat{P}(prog = none|yes) = \frac{5+1}{130+3}$
- $\hat{P}(adv = yes|yes) = \frac{15+1}{130+2}$

Naive Bayes: Calculation

- $$\arg\max_{s_m \in S} P\left(s_m \mid \left\langle f_1 = v_1, f_2 = v_2, \ldots, f_n = v_n \right\rangle\right) \approx \arg\max_{s_m \in S} P(s_m) \prod_{i=1}^{n} P\left(f_i = v_i \mid s_m\right)$$

- For calculations
  - avoid underflow, use logarithms
  - $$\arg\max_{s_m \in S} P(s_m) \prod_{i=1}^{n} P\left(f_i = v_i \mid s_m\right) =$$

    $$\arg\max_{s_m \in S} \left(\log\left(P(s_m) \prod_{i=1}^{n} P\left(f_i = v_i \mid s_m\right) \right) \right)$$

    $$= \arg\max_{s_m \in S} \left(\log(P(s_m)) + \sum_{i=1}^{n} \log(P\left(f_i = v_i \mid s_m\right)) \right)$$

Properties of Naive Bayes

- A probabilistic classifier
- A multi-class classifier:
  - i.e. can handle more than two classes
- Categorical features natively
  - Can be adopted to numeric features
- NLTK contains an implementation

- The independence assumption is in general: wrong!
  - $P(v_1, v_2, \dots, v_n|c)$ is far from
  - $P(v_1|c) \times P(v_2|c) \cdots \times P(v_n|c)$
- Still NB works reasonably well as a classifier (discriminator)
- It is not prone to overfitting
- Other classifiers may work better

Text classification with NB

Text classification with NB

- Naive Bayes may be applied to various NLP tasks
- Text classification:
  - Goal: classify the text on the basis of the words in the text
  - What are the features?
  - What are the possible values.
- Two possible answers:
  - The Multinomial model
  - The Bernoulli model

1. Multinomial NB text classification

$$\arg\max_{s_m \in S} P\big(s_m \mid \langle f_1 = v_1, f_2 = v_2, \ldots, f_n = v_n \rangle\big) \approx \arg\max_{s_m \in S} P(s_m) \prod_{i=1}^{n} P\big(f_i = v_i \mid s_m\big)$$

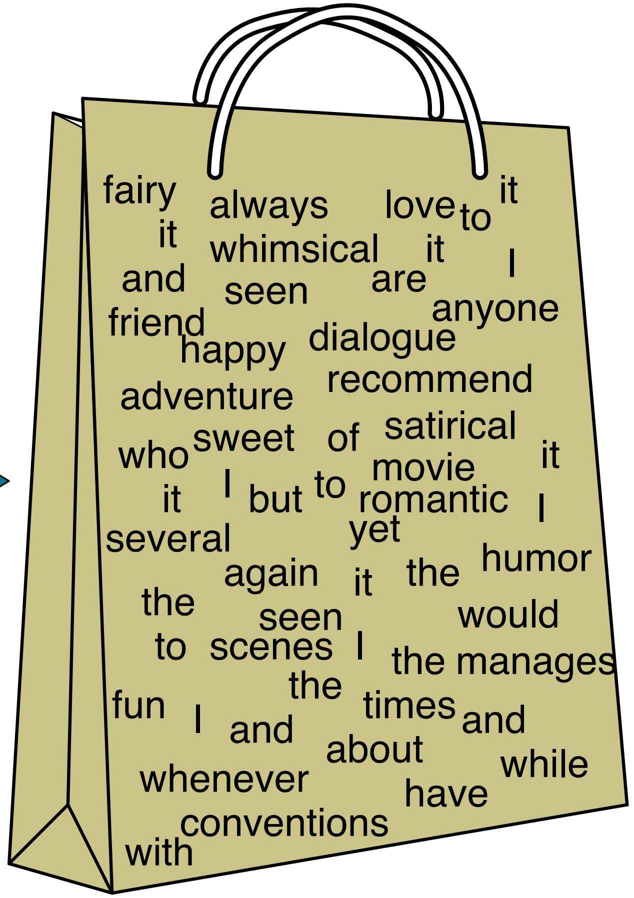
- f_i refers to position i in the text
- v_i is the word occurring in this position
- n is the number of tokens in the text

- Simplifying assumption: a word is equally likely in all positions
- Hence we count how many times each word occurs in the text

$$\arg\max_{s_m \in S} P(s_m) \prod_{i=1}^{n} P\big(f_i = v_i \mid s_m\big) = \arg\max_{s_m \in S} P(s_m) \prod_{i=1}^{n} P\big(v_i \mid s_m\big)$$

The Bag of Words Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



From Jurafsky & Martin

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

Multinomial NB: Training

- $$\hat{P}(s_m) = \frac{C(s_m, o)}{C(o)}$$
  - where $C(s_m, o)$ is the number of occurrences of documents o in class s_m
- $$\hat{P}(w_i \mid s_m) = \frac{C(w_i, s_m)}{\sum_j C(w_j, s_m)}$$
  - where $C(w_i, s_m)$ is the number of occurrences of word w_i in all documents in class s_m
  - $\sum_j C(w_j, s_m)$ is the total number of words in all documents in class s_m

Example: Movie reviews corpus (NLTK)

- 2000 documents
  - (a subset of a larger corpus)
- Two classes: ‘neg’, ‘pos’, 1000 doc.s in each class

```
› from nltk.corpus import movie_reviews

› documents = [(list(movie_reviews.words(fileid)), category)
               for category in movie_reviews.categories()
               for fileid in movie_reviews.fileids(category)]
```

Example: movie reviews, multinomial

- Considered 1900 doc.s for training
- '*pitt*' occurs in 15 'pos' and 6 'neg' reviews
- '*pitt*' occurs 31 times in the 'pos' reviews and 25 times in the negative reviews
- There are 798,742 words in the 'pos' reviews and 705,726 in the 'neg' reviews
- $\hat{P}(w = pitt|pos) = \frac{31}{798\,742}$ $\hat{P}(w = pitt|neg) = \frac{25}{705\,726}$

$3 \times pitt, \quad 2 \times terrible, \quad 0 \times terrific$

'pos'

- $\hat{P}(pos) = \frac{959}{1900}$
- $\hat{P}(w = pitt|pos) = \frac{31}{798\,742}$
- $\hat{P}(w = terrible|pos) = \frac{26}{798\,742}$
- $\hat{P}(pos \mid 3 \times pitt, 2 \times terrible, 0 \times terrific) =$

$$k' \frac{959}{1900} \times \left(\frac{31}{798\,742}\right)^3 \times \left(\frac{26}{798\,742}\right)^2$$
$$= k'\, 3.12 \times 10^{-23}$$

'neg'

$\hat{P}(neg) = \frac{941}{1900}$

$\hat{P}(w = pitt|neg) = \frac{25}{705\,726}$

$\hat{P}(w = terrible|neg) = \frac{104}{705726}$

$\hat{P}(pos \mid 3 \times pitt, 2 \times terrible, 0 \times terrific) =$

$$k' \frac{941}{1900} \times \left(\frac{25}{705\,726}\right)^3 \times \left(\frac{104}{705\,726}\right)^2$$
$$= k'\, 4.78 \times 10^{-22}$$

($k' \approx 1/P(w1 = pitt, w2 = terrible, w3 = terrific)$, the same constant for both classes)

2. NB – Bernoulli model for text classification

- How are words turned into features?
- A vocabulary of words, W
- Each word $w_i$makes a feature f_i
- The possible values for f_i is True and False (1 and 0)
- $f_i = 1$ in a document if and only if it contains w_i.

Bernoulli NB: Decision

$$\arg\max_{s_m \in S} P\left(s_m \mid \left\langle f_1 = v_1, f_2 = v_2, \ldots, f_n = v_n \right\rangle\right) \approx \arg\max_{s_m \in S} P(s_m) \prod_{i=1}^{n} P\left(f_i = v_i \mid s_m\right)$$

- f_i refers to a word in the vocabulary
- v_i is 1 or 0 depending on whether the word occurs in the text or not
- n is the number of words in the vocabulary

- For a document D:

$$\prod_{i=1}^{n} P(f_i = v_i | s_m) = \prod_{f_i \in D} P(f_i = 1 | s_m) \prod_{f_i \notin D} P(f_i = 0 | s_m)$$

Example: movie reviews NLTK (Bernoulli)

- 'pitt' occurs in 15 'pos' and 6 'neg' reviews
- $\hat{P}(pitt = True|pos) = \frac{15}{959}$ $\hat{P}(pitt = True|neg) = \frac{6}{941}$
- $\hat{P}(pitt = False|pos) = \frac{944}{959}$ $\hat{P}(pitt = False|neg) = \frac{935}{941}$

Comparison

Multinomial

- Counts how many times a term is present
- Considers
  - only the present terms
  - ignores absent terms
- *n*: number of words in the document
- Tends to be the better of the two for longer texts

Bernoulli

- Registers whether a term is present or not
- Considers both
  - The present terms
  - The absent terms
- *n*: number of words in the vocabulary
- Compatible on shorter snippets

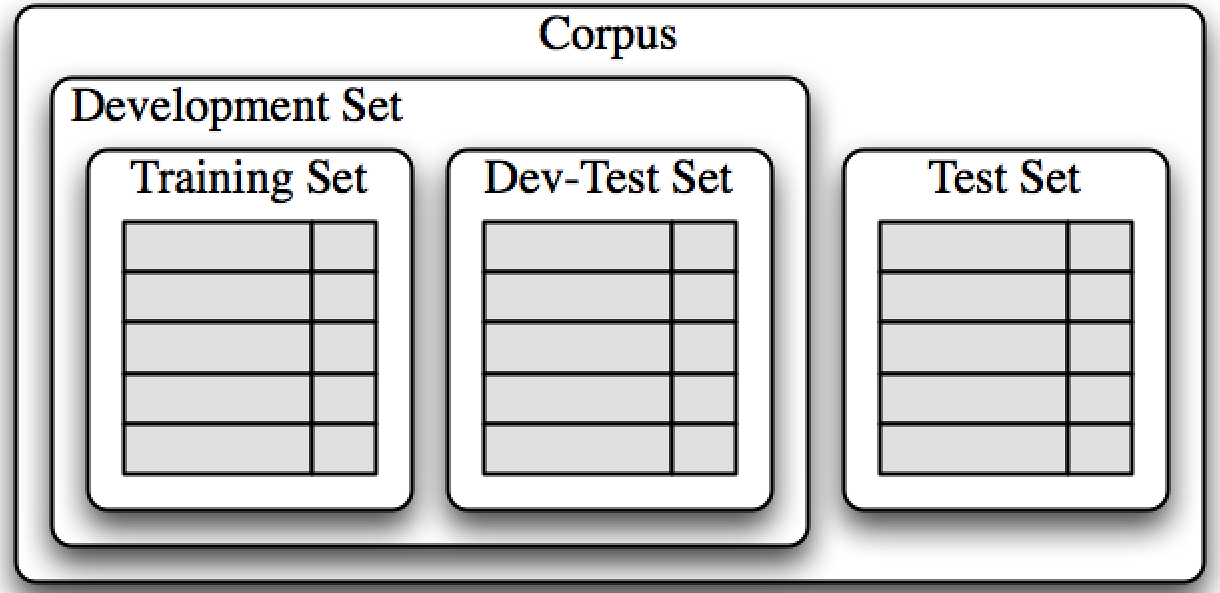
The two models

- Multinomial model
  - Jurafsky and Martin, 3.ed, sec. 4, Sentiment analysis
  - Related to n-gram models
- Bernoulli
  - NLTK book, Sec. 6.1, 6.2, 6.5
    - Including the Movie review example
  - Jurafsky and Martin, 2.ed, sec. 20.2, WSD
- Both
  - Manning, Raghavan, Schütze, *Introduction to Information Retrieval*, Sec. 13.0-13.3

Set-up for experiments

Set-up for experiments

- Before you start: split into development set and test set.
- Hide the test set
- Split development set into Training and Development-Test set
- Use training set for training a learner



- Use Dev(-Test) for repeated evaluation in the test phase
- Finally test on the test set!

Procedure

1. Train classifier on training set
2. Test it on dev-test set
3. Compare to earlier runs, is this better?
4. Error analysis: What are the mistakes (on dev-test set)
5. Make changes to the classifier
6. Repeat from 1

===================

- When you have run empty on ideas, test on test set. Stop!

Cross-validation

- Small test sets → Large variation in results
- N-fold cross-validation:
  - Split the development set into n equally sized bins
    - (e.g. n = 10)
  - Conduct n many experiments:
    - In experiment m, use part m as test set and the n-1 other parts as training set.
  - This yields n many results:
    - We can consider the mean of the results
    - We can consider the variation between the results.
      - Statistics!

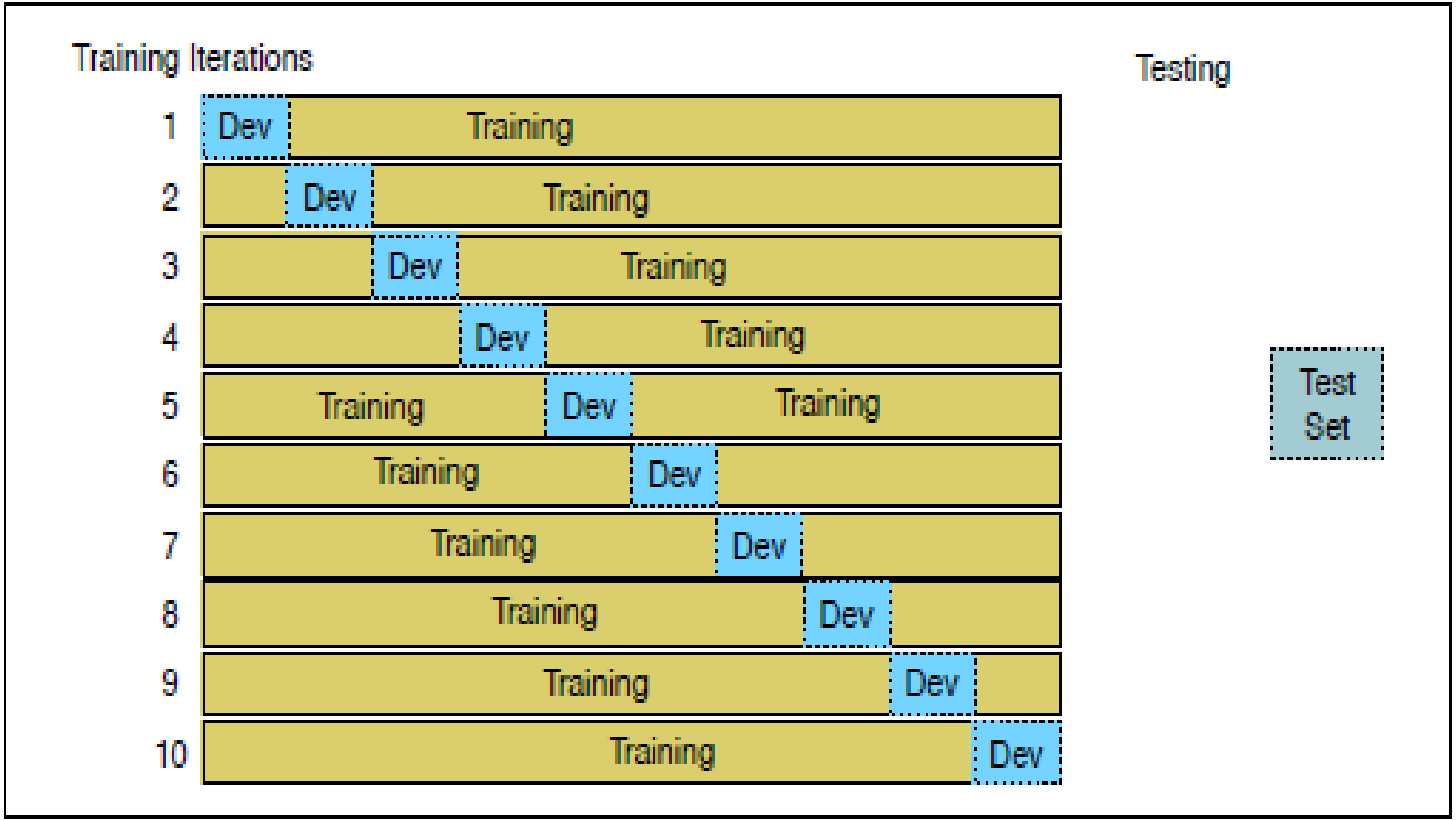


Figure 6.7 10-fold crossvalidation

Evaluation

Evaluation measure: Accuracy

- What does accuracy 0.81 tell us?
- Given a test set of 500 documents:
  - The classifier will classify 405 correctly
  - And 95 incorrectly
- A good measure given:
  - The 2 classes are equally important
  - The 2 classes are roughly equally sized
  - Example:
    - Woman/man
    - Movie reviews: pos/neg

But

- For some tasks, the classes aren't equally important
  - Worse to loose an important mail than to receive yet another spam mail

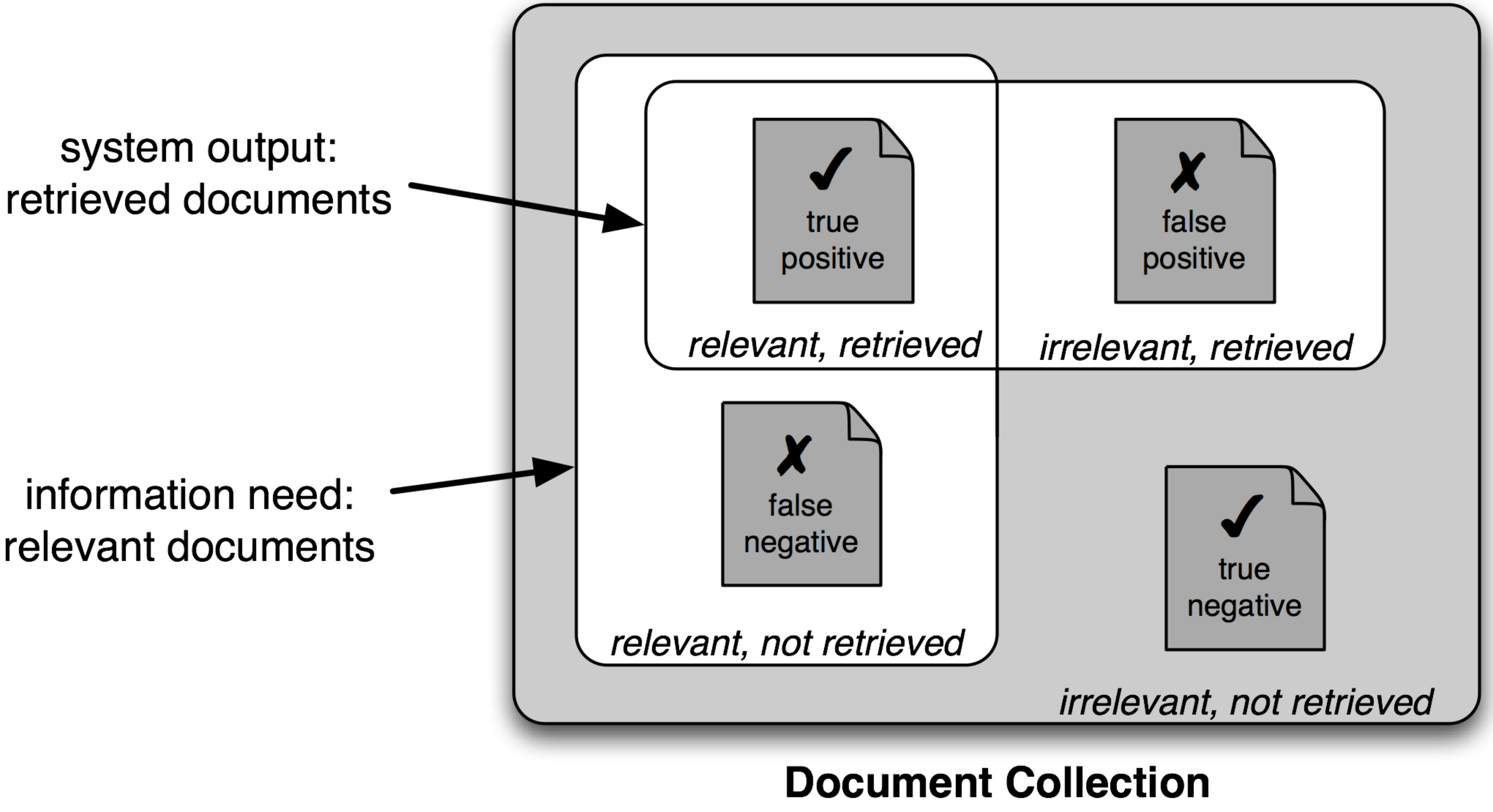
- For some tasks the different classes have different sizes.

Information retrieval (IR)

- Traditional IR, e.g. a library
  - Goal: Find all the documents on a particular topic out of 100 000 documents,
    - Say there are 5
  - The system delivers 10 documents: all irrelevant
    - What is the accuracy?

- For these tasks, focus on
  - The relevant documents
  - The documents returned by the system
- Forget the
  - Irrelevant documents which are not returned

IR - evaluation



Confusion matrix

		gold standard labels		
		gold positive	gold negative	
system output labels	system positive	true positive	false positive	precision = $\frac{tp}{tp+fp}$
	system negative	false negative	true negative	
		recall = $\frac{tp}{tp+fn}$		accuracy = $\frac{tp+tn}{tp+fp+tn+fn}$

Figure 6.4 Contingency table

- Beware what the rows and columns are:
  - NLTKs ConfusionMatrix swaps them compared to this table

Evaluation measures

		Is in C	
		Yes	NO
Classifier	Yes	tp	fp
	No	fn	tn

- Accuracy: (tp+tn)/N
- Precision:tp/(tp+fp)
- Recall: tp/(tp+fn)

- F-score combines P and R
- $F_1 = \frac{2PR}{P+R} \left(= \frac{1}{\frac{\frac{1}{R}+\frac{1}{P}}{2}} \right)$
- F_1 called ''harmonic mean''
- General form
  - $F = \frac{1}{\alpha\frac{1}{P}+(1-\alpha)\frac{1}{R}}$
  - for some $0 < \alpha < 1$

Confusion matrix

		gold labels			
		urgent	normal	spam	
system output	urgent	8	10	1	$\text{precision}_u = \frac{8}{8+10+1}$
	normal	5	60	50	$\text{precision}_n = \frac{60}{5+60+50}$
	spam	3	30	200	$\text{precision}_s = \frac{200}{3+30+200}$
		$\text{recall}_u = \frac{8}{8+5+3}$	$\text{recall}_n = \frac{60}{10+60+30}$	$\text{recall}_s = \frac{200}{1+50+200}$	

Figure 6.5 Confusion matrix for a three-class categorization task, showing for each pair of classes (c_1, c_2), how many documents from c_1 were (in)correctly assigned to c_2

- Precision, recall and f-score can be calculated for each class against the rest

Today - Classification

- Motivation
- Classification
- Naive Bayes classification
- NB for text classification
  - The multinomial model
  - The Bernoulli model
- Experiments: training, test and cross-validation
- Evaluation