Language Technology (2021)

Text classification

Marco Kuhlmann
Department of Computer and Information Science

LINKÖPING UNIVERSITY

Text classification

- **Text classification** is the task of categorising text documents into predefined classes.
- The term ‘document’ is applied to everything from tweets over press releases to complete books.

Topic classification

UK	China	Elections	Sports
congestion London	Olympics Beijing	recount votes	diamond baseball
Parliament Big Ben	tourism Great Wall	seat run-off	forward soccer
Windsor The Queen	Mao Communist	TV-ads campaign	team captain

Adapted from Manning et al. (2008)

Sentiment analysis

The gorgeously elaborate continuation of “The Lord of the Rings” trilogy is so huge that a column of words cannot adequately describe co-writer/director Peter Jackson’s expanded vision of J.R.R. Tolkien’s Middle-earth

positive

… is a sour little movie at its core; an exploration of the emptiness that underlay the relentless gaiety of the 1920’s, as if to stop would hasten the economic and global political turmoil that was to come.

negative

Forensic linguistics

‘I realized the faxed copy I just received was an outline of the manifesto, using much of the same wording, definitely the same topics and themes. … I invented [the language analysis] for this case and really, forensics linguistics took off after that.’

James Fitzgerald, profiler

Sources: Wikipedia, Newsweek

Text classification using handwritten rules

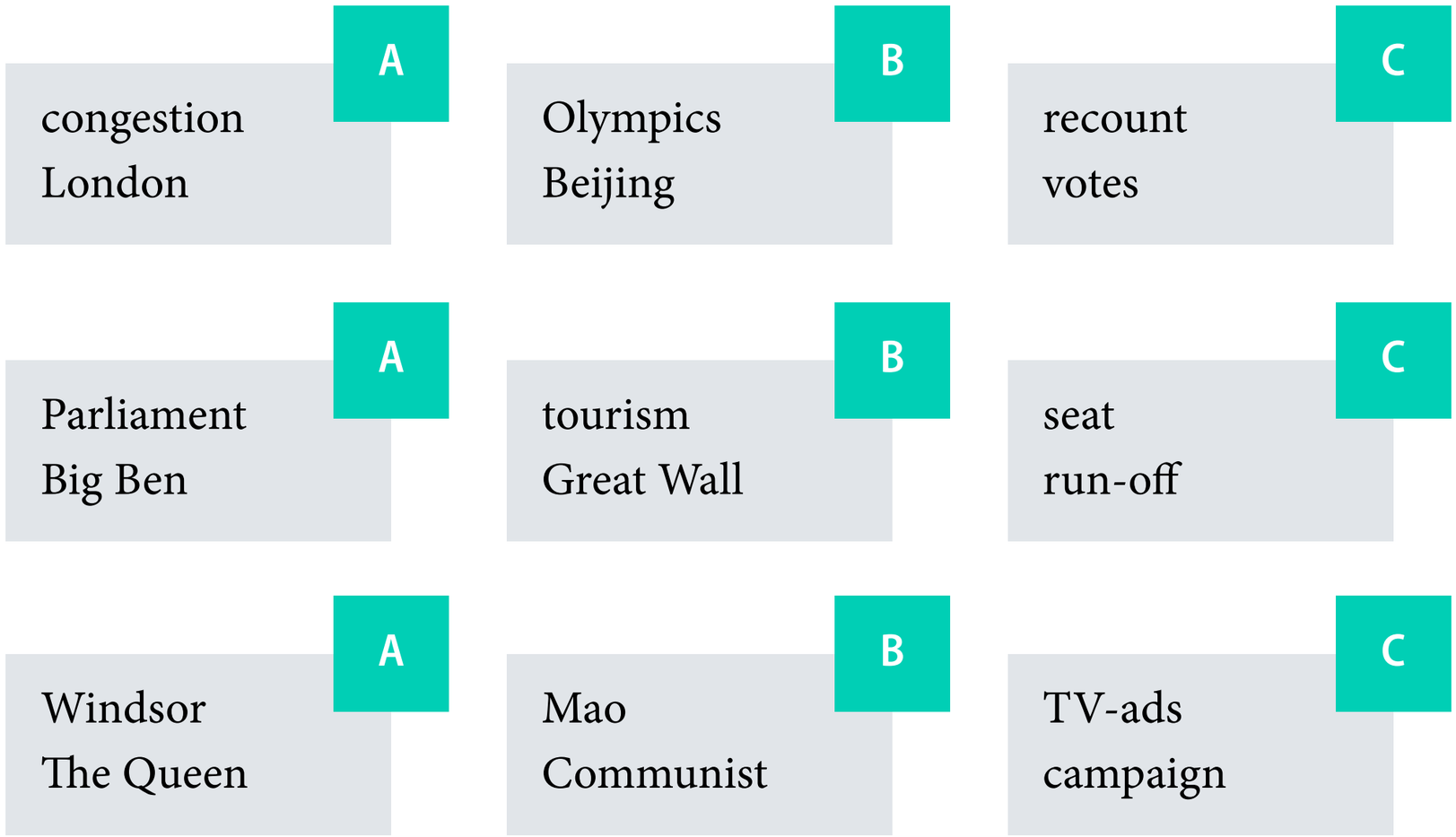
- We can assign a class to a document via handwritten rules.

  spam filtering using complex regular expressions

- Handwritten rules can have high precision, but to develop and maintain them requires expert knowledge and is costly.

- In spite of these shortcomings, handwritten rules still define the state of the art in many domains.

Text classification as supervised machine learning



Text classification as supervised machine learning

training set
learning

A	B	C
congestion London	Olympics Beijing	recount votes
Parliament Big Ben	tourism Great Wall	seat run-off

test set
evaluation

A	B	C
Windsor The Queen	Mao Communist	TV-ads campaign

Training and testing

- **Training**

  When we train a classifier, we present it with the document x and the correct class y and apply some learning algorithm.

  this section: Naive Bayes

- **Testing**

  When we evaluate a classifier, we present it with x and compare the predicted class for this input with the correct class y.

General machine learning methodology

- To train a machine learning system, we apply some learning algorithm to optimise performance on the training set.
- What we really want to optimise is the system's performance on new, previously unseen data.

  How well does the system generalise?

- Because we cannot measure system performance on unseen data, we *estimate* this performance using the test set.

Recurring questions

- How does the model work?

  often some kind of mathematical formula or algorithm

- How can we learn the model?

  estimate probabilities, learn weights of a neural network, …

- How can we evaluate the model?

  typically some evaluation measure, such as classification accuracy

This lecture

- Introduction to text classification
- Evaluation of text classifiers
- Text classification with Naive Bayes
- Learning a Naive Bayes classifier

Evaluation of text classifiers

Evaluation of text classifiers

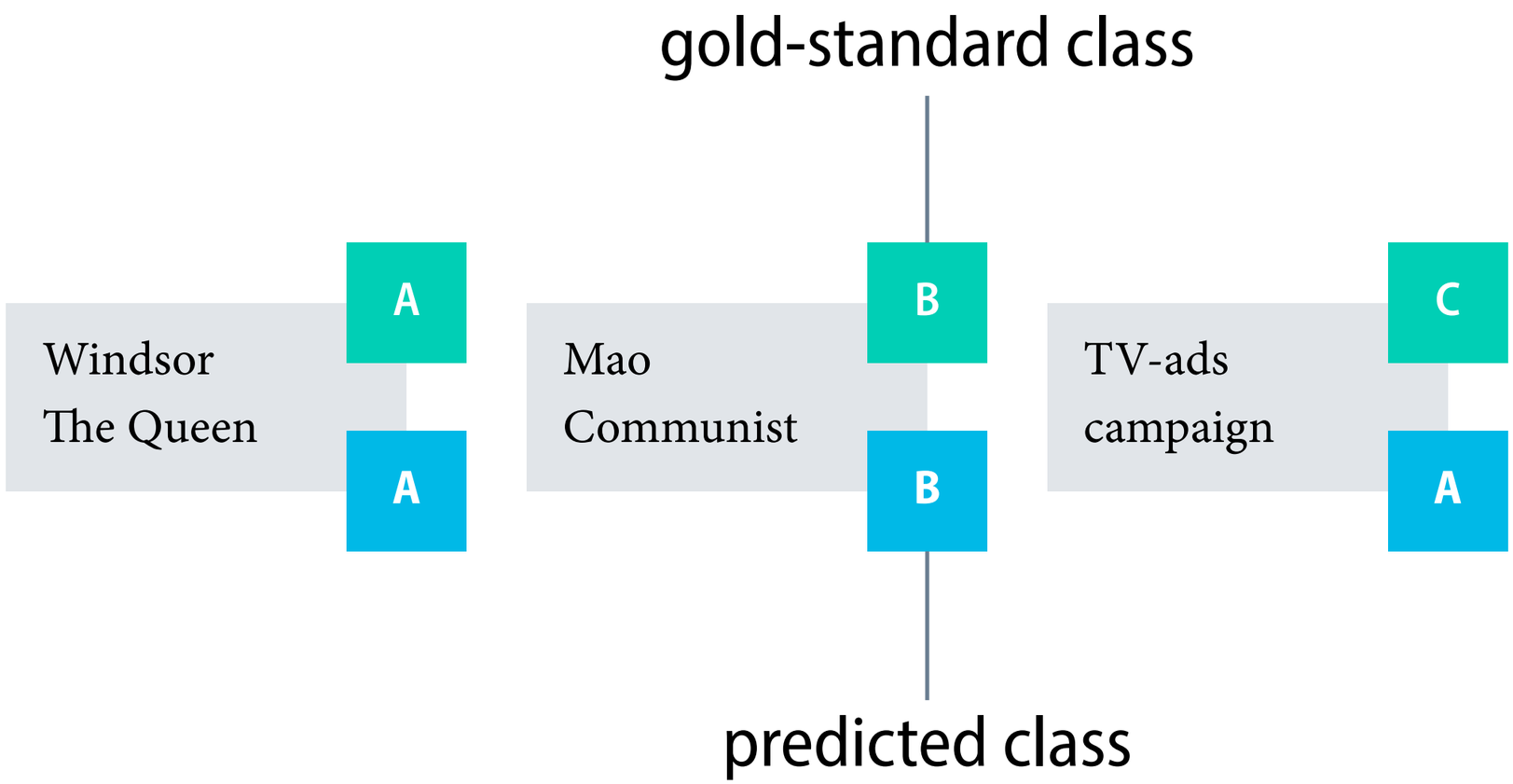
- We require a **test set** consisting of a number of documents, each of which has been tagged with its correct class.

  typically part of a larger gold-standard data set

- To evaluate a classifier, we apply it to the test set and compare the predicted classes with the gold-standard classes.

- The result of this comparison allows us to estimate how well the classifier will perform on new, previously unseen documents.

Evaluation of text classifiers



Accuracy

The **accuracy** of a classifier is the proportion of documents for which the classifier predicts the gold-standard class:

$$\text{accuracy} = \frac{\text{number of correctly classified documents}}{\text{number of all documents}}$$

Accuracy

Document	Gold-standard class	Predicted class
Chinese Beijing Chinese	China	China
Chinese Chinese Shanghai	China	China
Chinese Macao	China	China
Tokyo Japan Chinese	Japan	China

accuracy for this example: 3/4 = 75%

Why accuracy is problematic

We want to classify texts written by patients of a geriatric hospital and predict whether they suffer from a rare neurodegenerative disease. We evaluate on a test set consisting of 10,000 texts, 10 texts written by patients that suffer from the disease and 9,990 texts written by patients that do not suffer from the disease.

It is trivial to achieve a very high accuracy on this task: Always predict that the patient does not suffer from the disease.

Confusion matrix

	classifier ‘positive’	classifier ‘negative’
gold standard ‘positive’	true positives	false negatives
gold standard ‘negative’	false positives	true negatives

Confusion matrix

	classifier ‘positive’	classifier ‘negative’
gold standard ‘positive’	0	10
gold standard ‘negative’	0	9,990

Accuracy

	classifier ‘positive’	classifier ‘negative’
gold standard ‘positive’	true positives	false negatives
gold standard ‘negative’	false positives	true negatives

Accuracy

	classifier ‘positive’	classifier ‘negative’
gold standard ‘positive’	0	10
gold standard ‘negative’	0	9,990

Precision and recall

- **Precision** and **recall** 'zoom in' on how good a system is at identifying documents of a specific class *c*.

- **Precision** is the proportion of correctly classified documents among all documents for which the system predicts class *c*.

  When the system predicts the disease, how often is it correct?

- **Recall** is the proportion of correctly classified documents among all documents with gold-standard class *c*.

  If the patient has the disease, how often does the system predict it?

Precision with respect to the positive class

	classifier ‘positive’	classifier ‘negative’
gold standard ‘positive’	true positives	false negatives
gold standard ‘negative’	false positives	true negatives

Precision with respect to the positive class

	classifier ‘positive’	classifier ‘negative’
gold standard ‘positive’	0	10
gold standard ‘negative’	0	9,990

Recall with respect to the positive class

	classifier ‘positive’	classifier ‘negative’
gold standard ‘positive’	true positives	false negatives
gold standard ‘negative’	false positives	true negatives

Recall with respect to the positive class

	classifier ‘positive’	classifier ‘negative’
gold standard ‘positive’	0	10
gold standard ‘negative’	0	9,990

Precision and recall with respect to the positive class

$$\text{precision} = \frac{\#\text{ true positives}}{\#\text{ true positives} + \#\text{ false positives}}$$

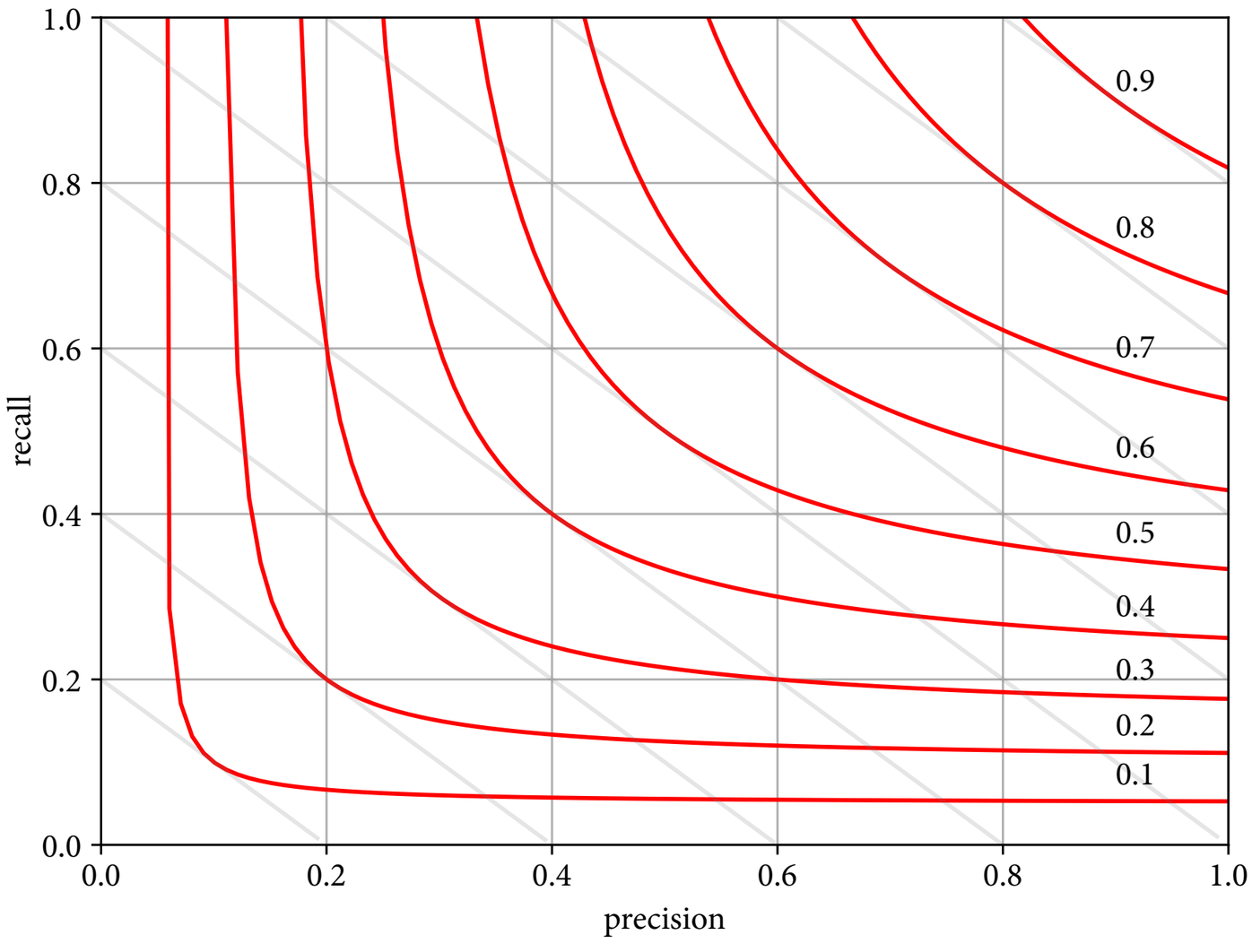
$$\text{recall} = \frac{\#\text{ true positives}}{\#\text{ true positives} + \#\text{ false negatives}}$$

F1-measure

A good classifier should balance between precision and recall.
The **F1-measure** is the harmonic mean of the two values:

$$F1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

F1-measure


1.0
0.8
0.6
0.4
0.2
0.0
recall
0.0
0.2
0.4
0.6
0.8
1.0
precision
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

Accuracy with three classes

	A	B	C
A	58	6	1
B	5	11	2
C	0	7	43

Precision with respect to class B

	A	B	C
A	58	6	1
B	5	11	2
C	0	7	43

Recall with respect to class B

	A	B	C
A	58	6	1
B	5	11	2
C	0	7	43

Precision and recall with respect to class A

	A	B	C
A	58	6	1
B	5	11	2
C	0	7	43

The role of baselines

- The evaluation measures as such do not really tell us much.

  Whether ‘80% recall’ is good or not depends on the task at hand.

- Instead, we should ask for a classifier’s performance relative to a reference result, a **baseline**.

  ‘The accuracy of our system is 5 points higher than that of the baseline.’

- A simple baseline for classification is to always predict the class which occurred most often in the training data.

  Most Frequent Class

This lecture

- Introduction to text classification
- Evaluation of text classifiers
- Text classification with Naive Bayes
- Learning a Naive Bayes classifier

Text classification with Naive Bayes

Naive Bayes

- The **Naive Bayes classifier** is a simple but surprisingly effective probabilistic text classifier that builds on Bayes' rule.

- It is called 'naive' because it makes strong (unrealistic) independence assumptions about probabilities.

- It uses a representation of texts as **bags of words**, that is, it does not pay attention to word order.

The bag of words

the gorgeously elaborate continuation of the lord of the rings trilogy is so huge that a column of words cannot adequately describe co-writer/director peter jackson's expanded vision of j.r.r. tolkien's middle-earth

positive

… is a sour little movie at its core an exploration of the emptiness that underlay the relentless gaiety of the 1920's as if to stop would hasten the economic and global political turmoil that was to come

negative

The bag of words

a adequately cannot
co-writer/director column
continuation describe
elaborate expanded gorgeously
huge is j.r.r. jackson lord
middle-earth of of of of peter
rings so that the the the tolkien
trilogy vision words

positive

… 1920's a an and as at come
core economic emptiness
exploration gaiety global
hasten if is its little movie of of
political relentless sour stop
that that the the the the to to
turmoil underlay was would

negative

The bag of words

Word	Count
of	4
the	3
words	1
vision	1
trilogy	1
…	

positive

Word	Count
the	4
to	2
that	2
of	2
would	1
…	

negative

Naive Bayes classification rule, informally

elaborate
gorgeously

?

positive

elaborate
gorgeously

score(**pos**) =

P(**pos**) P(elaborate|**pos**) P(gorgeously|**pos**)

70%

negative

elaborate
gorgeously

score(**neg**) =

P(**neg**) P(elaborate|**neg**) P(gorgeously|**neg**)

30%

Naive Bayes classification rule, informally

elaborate
gorgeously

positive

elaborate
gorgeously

score(**pos**) =

P(**pos**) P(elaborate|**pos**) P(gorgeously|**pos**)

70%

negative

elaborate
gorgeously

score(**neg**) =

P(**neg**) P(elaborate|**neg**) P(gorgeously|**neg**)

30%

Bayes' rule

- For classification, we would like to know $P(\text{class}|\text{document})$.

  $P(\text{disease}|\text{symptom})$

- But a Naive Bayes classifier contains $P(\text{document}|\text{class})$.

  $P(\text{symptom}|\text{disease})$

- The classifier uses **Bayes' rule** to convert between the two.

  $P(\text{class}|\text{document}) \propto P(\text{class})\ P(\text{document}|\text{class})$

Bayes' rule

Thomas Bayes
(1702–1761)

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

Formal definition of a Naive Bayes classifier

C	a set of possible classes
V	a set of possible words; the model's **vocabulary**
$P(c)$	probabilities that specify how likely it is for a document to belong to class c (one probability for each class)
$P(w\|c)$	probabilities that specify how likely it is for a document to contain the word w, given that the document belongs to class c (one probability for each class–word pair)

Naive Bayes classification rule, formally

choose the class c that maximises
the term to the right of the 'arg max'

$$\hat{c} = \arg\max_{c \in C} P(c) \cdot \prod_{w \in V} P(w \mid c)^{\#(w)}$$

predicted class

count of the word w

Implementing the Naive Bayes classification rule

- **Problem 1:** takes long time to loop over a large vocabulary

  Solution: loop over the words in the document instead

- **Problem 2:** words not in the vocabulary

  Solution: skip unknown words (this is what the model says!)

- **Problem 3:** underflow as one multiplies probabilities

  Solution: use log probabilities instead

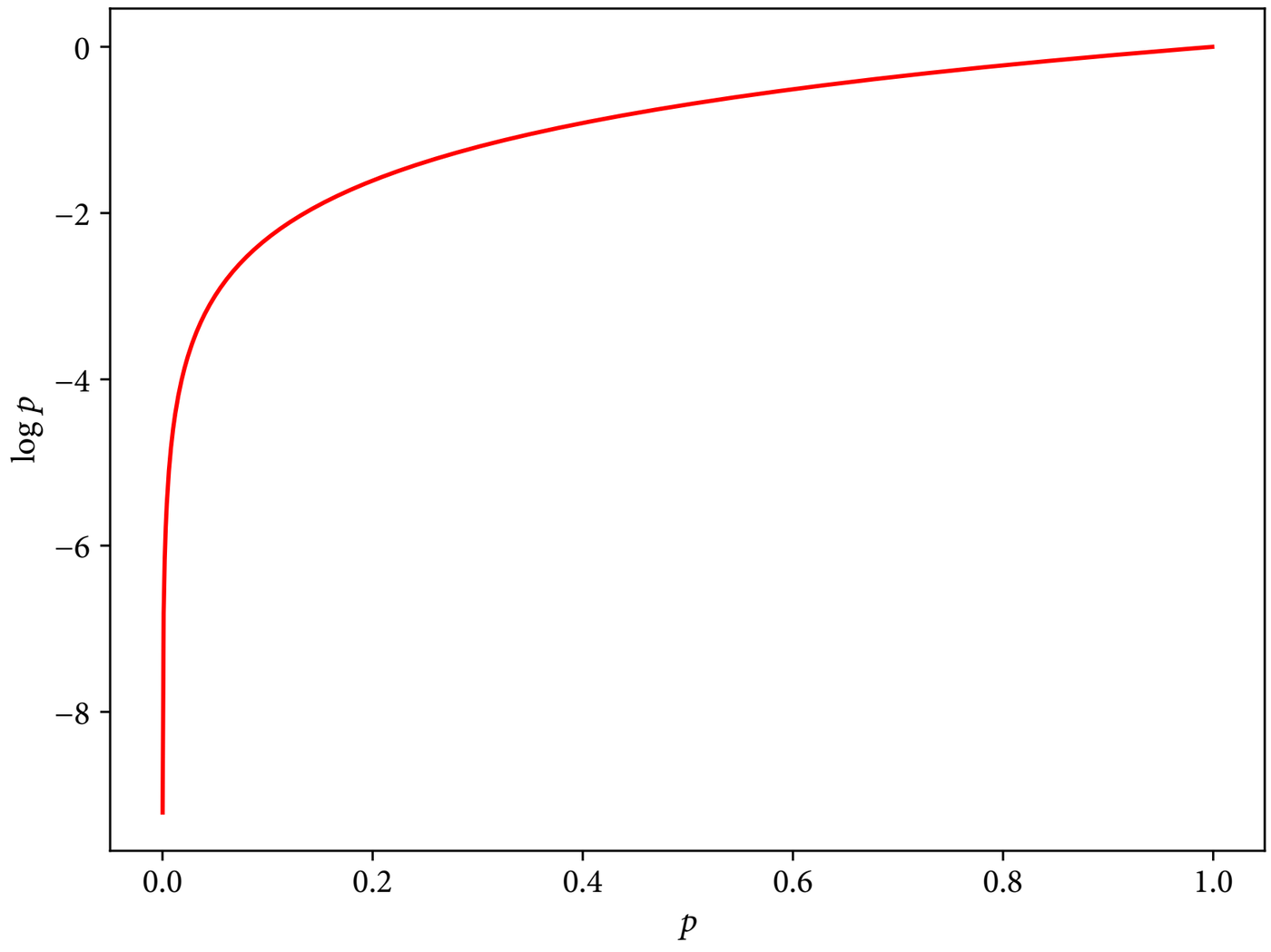
Log probabilities

- In order to avoid underflow we can use the logarithms of probabilities instead of the probabilities themselves.

  $P(w|c)$ becomes $\log P(w|c)$

- In this case, instead of multiplying probabilities, we have to add their logarithms.

Log probabilities



This lecture

- Introduction to text classification
- Evaluation of text classifiers
- Text classification with Naive Bayes
- Learning a Naive Bayes classifier

Learning a Naive Bayes classifier

Learning a Naive Bayes classifier

A

congestion
London

B

Olympics
Beijing

C

recount
votes

A

Parliament
Big Ben

B

tourism
Great Wall

C

seat
run-off

A

Windsor
The Queen

B

Mao
Communist

C

TV-ads
campaign

Learning a Naive Bayes classifier

training set
learning

A	B	C
congestion London	Olympics Beijing	recount votes
Parliament Big Ben	tourism Great Wall	seat run-off

---

test set
evaluation

A	B	C
Windsor The Queen	Mao Communist	TV-ads campaign

Learning a Naive Bayes classifier

A

congestion
London

B

Olympics
Beijing

C

recount
votes

A

Parliament
Big Ben

B

tourism
Great Wall

C

seat
run-off

$P(c)$

class probabilities

$P(w|c)$

word probabilities

Learning a Naive Bayes classifier

Probability	Value
…	?
$P(\text{hasten}\,|\,\textbf{pos})$	?
$P(\text{huge}\,|\,\textbf{pos})$	?
$P(\text{if}\,|\,\textbf{pos})$	?
$P(\text{its}\,|\,\textbf{pos})$	?
…	positive

Probability	Value
…	?
$P(\text{hasten}\,|\,\textbf{neg})$	?
$P(\text{huge}\,|\,\textbf{neg})$	?
$P(\text{if}\,|\,\textbf{neg})$	?
$P(\text{its}\,|\,\textbf{neg})$	?
…	negative

Maximum Likelihood Estimation (MLE)

- One of the simplest techniques for estimating probabilities is **Maximum Likelihood Estimation** (MLE).

  Find probabilities that maximise the probability of the training data.

- For the special case of the Naive Bayes classifier, MLE amounts to equating probabilities with relative frequencies.

MLE for the Naive Bayes classifier

- To estimate the class probabilities $P(c)$:

  Compute the percentage of documents with class c among all documents in the training set.

- To estimate the word probabilities $P(w|c)$:

  Compute the percentage of occurrences of the word w among all word occurrences in documents with class c.

MLE for the Naive Bayes classifier

$\#(c)$ number of documents with gold-standard class c

$\#(w, c)$ number of occurrences of w in documents with class c

$$P(c) = \frac{\#(c)}{\sum_{x \in C} \#(x)} \qquad P(w \mid c) = \frac{\#(w, c)}{\sum_{x \in V} \#(x, c)}$$

MLE of word probabilities

The gorgeously elaborate continuation of "The Lord of the Rings" trilogy is so huge that a column of words cannot adequately describe co-writer/director Peter Jackson's expanded vision of J.R.R. Tolkien's Middle-earth

positive

31 tokens

… is a sour little movie at its core; an exploration of the emptiness that underlay the relentless gaiety of the 1920's, as if to stop would hasten the economic and global political turmoil that was to come.

negative

37 tokens

MLE of word probabilities

Word	Count
of	4
The	2
words	1
vision	1
trilogy	1
…	

positive

Word	Count
the	4
to	2
that	2
of	2
would	1
…	

negative

MLE of word probabilities

Probability	Estimated value
$P(\text{of}\,|\,\textbf{pos})$	4/31
$P(\text{The}\,|\,\textbf{pos})$	2/31
$P(\text{words}\,|\,\textbf{pos})$	1/31
$P(\text{vision}\,|\,\textbf{pos})$	1/31
$P(\text{trilogy}\,|\,\textbf{pos})$	1/31
…	**positive**

Probability	Estimated value
$P(\text{the}\,|\,\textbf{neg})$	4/37
$P(\text{to}\,|\,\textbf{neg})$	2/37
$P(\text{that}\,|\,\textbf{neg})$	2/37
$P(\text{of}\,|\,\textbf{neg})$	2/37
$P(\text{would}\,|\,\textbf{neg})$	1/37
…	**negative**

Smoothing

- If we equate word probabilities with relative frequencies, some probabilities may be zero.
- This is a problem because we multiply probabilities in the decision rule for Naive Bayes.

  Slogan: Zero probabilities destroy information.

- To avoid zero probabilities, we can use techniques for **smoothing** the probability distribution.

Add-one smoothing

- A very simple smoothing method is **add-one smoothing**, where we add 1 to all counts before computing relative frequencies.

  also known as Laplace smoothing.

- Effectively, we hallucinate one extra occurrence of each word.

MLE with add-one smoothing

$\#(c)$ number of documents with gold-standard class c

$\#(w, c)$ number of occurrences of w in documents with class c

$$P(c) = \frac{\#(c)}{\sum_{x \in C} \#(x)}$$

no smoothing here!

$$P(w \mid c) = \frac{\#(w, c) + 1}{\sum_{x \in V} [\#(x, c) + 1]}$$

1 extra occurrence of each word

MLE with add-one smoothing

$\#(c)$ number of documents with gold-standard class c

$\#(w, c)$ number of occurrences of w in documents with class c

$$P(c) = \frac{\#(c)}{\sum_{x \in C} \#(x)} \qquad P(w \mid c) = \frac{\#(w, c) + 1}{\left[\sum_{x \in V} \#(x, c)\right] + |V|}$$

Estimating word probabilities

The gorgeously elaborate continuation of "The Lord of the Rings" trilogy is so huge that a column of words cannot adequately describe co-writer/director Peter Jackson's expanded vision of J.R.R. Tolkien's Middle-earth

positive

31 tokens

… is a sour little movie at its core; an exploration of the emptiness that underlay the relentless gaiety of the 1920's, as if to stop would hasten the economic and global political turmoil that was to come.

negative

37 tokens

Vocabulary

1920's J.R.R. Jackson's Lord Middle-earth Peter Rings The Tolkien's a adequately an and as at cannot co-writer/director column come continuation core describe economic elaborate emptiness expanded exploration gaiety global gorgeously hasten huge if is its little movie of political relentless so sour stop that the to trilogy turmoil underlay vision was words would

53 unique words

MLE with add-one smoothing

Word	Modified count
of	4 + 1
The	2 + 1
words	1 + 1
vision	1 + 1
trilogy	1 + 1
…	positive

Word	Modified count
of	2 + 1
The	0 + 1
words	0 + 1
vision	0 + 1
trilogy	0 + 1
…	negative

MLE with add-one smoothing

Probability	Estimated value
$P(\text{of}\mid\textbf{pos})$	$(4+1)/(31+53)$
$P(\text{The}\mid\textbf{pos})$	$(2+1)/(31+53)$
$P(\text{words}\mid\textbf{pos})$	$(1+1)/(31+53)$
$P(\text{vision}\mid\textbf{pos})$	$(1+1)/(31+53)$
$P(\text{trilogy}\mid\textbf{pos})$	$(1+1)/(31+53)$
…	positive

Probability	Estimated value
$P(\text{of}\mid\textbf{neg})$	$(2+1)/(37+53)$
$P(\text{The}\mid\textbf{neg})$	$(0+1)/(37+53)$
$P(\text{words}\mid\textbf{neg})$	$(0+1)/(37+53)$
$P(\text{vision}\mid\textbf{neg})$	$(0+1)/(37+53)$
$P(\text{trilogy}\mid\textbf{neg})$	$(0+1)/(37+53)$
…	negative

Additive smoothing

- Instead of hallucinating 1 extra occurrence of each word, we can hallucinate k extra occurrences: **additive smoothing**.
- Note that k may be any non-negative number; in particular, it could be smaller than 1!

  The constant k is a parameter that we have to tune on validation data.

MLE with additive smoothing

$\#(c)$ number of documents with gold-standard class c

$\#(w, c)$ number of occurrences of w in documents with class c

$$P(c) = \frac{\#(c)}{\sum_{x \in C} \#(x)}$$

no smoothing here!

$$P(w \mid c) = \frac{\#(w, c) + k}{\sum_{x \in V} [\#(x, c) + k]}$$

k extra occurrences of each word

MLE with additive smoothing

$\#(c)$ number of documents with gold-standard class c

$\#(w, c)$ number of occurrences of w in documents with class c

$$P(c) = \frac{\#(c)}{\sum_{x \in C} \#(x)} \qquad P(w \mid c) = \frac{\#(w, c) + k}{\left[\sum_{x \in V} \#(x, c)\right] + k \cdot |V|}$$

More advanced smoothing techniques

- Additive smoothing and add-one smoothing in particular often works well in the context of text classification.
- In other contexts, more advanced smoothing techniques work considerably better than additive smoothing.

  Witten–Bell smoothing, Kneser–Ney smoothing for ngram-models

This lecture

- Introduction to text classification
- Evaluation of text classifiers
- Text classification with Naive Bayes
- Learning a Naive Bayes classifier