Text mining (2019)

Text classification

Marco Kuhlmann
Department of Computer and Information Science

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)

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

Sentiment analysis

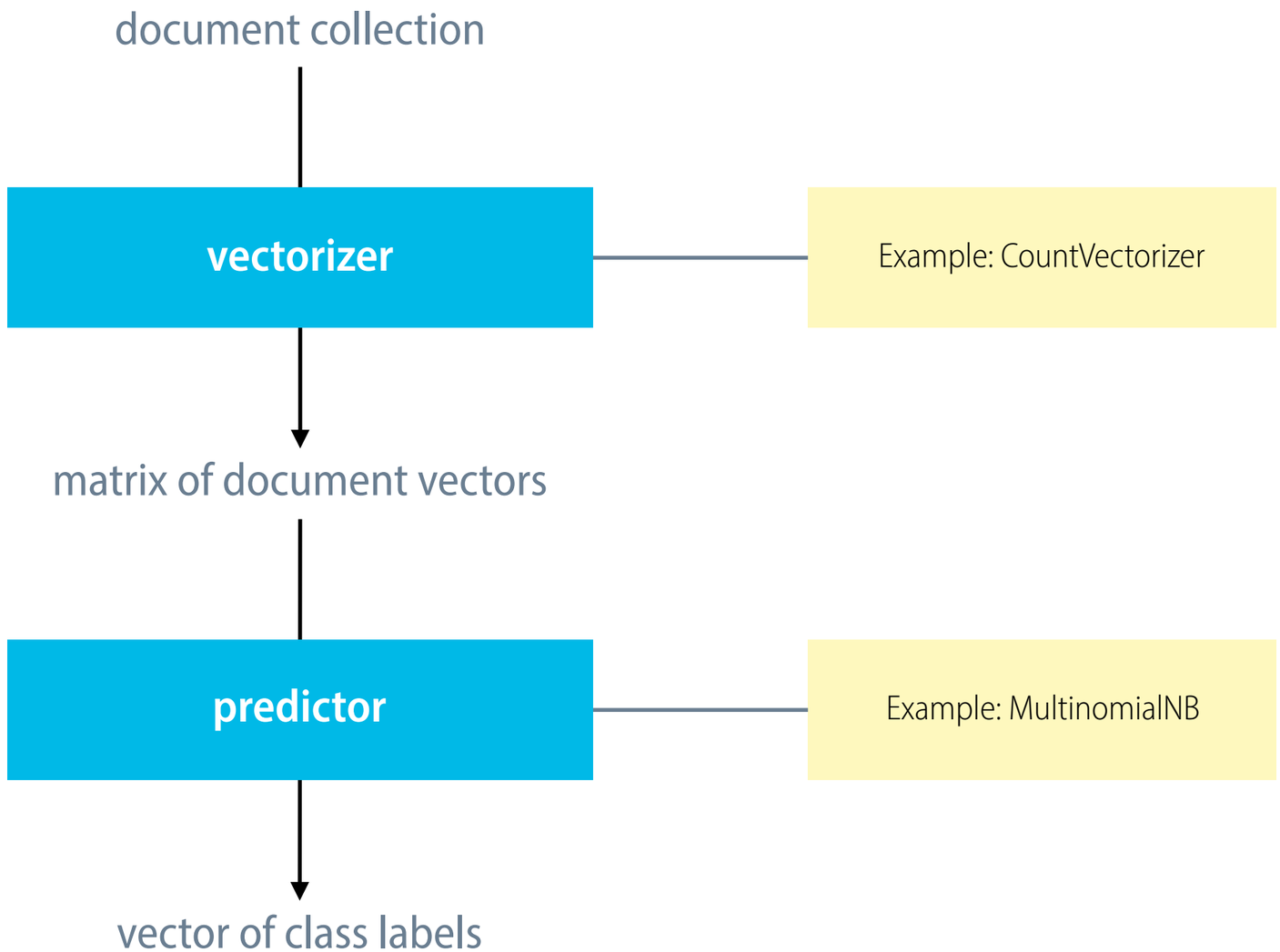
The gorgeously elaborate continuation of "The Lord of the Rings" trilogy is so huge that a column of words cannot adequately describe co-writer/director Peter Jackson's expanded vision of J.R.R. Tolkien's Middle-earth

positive

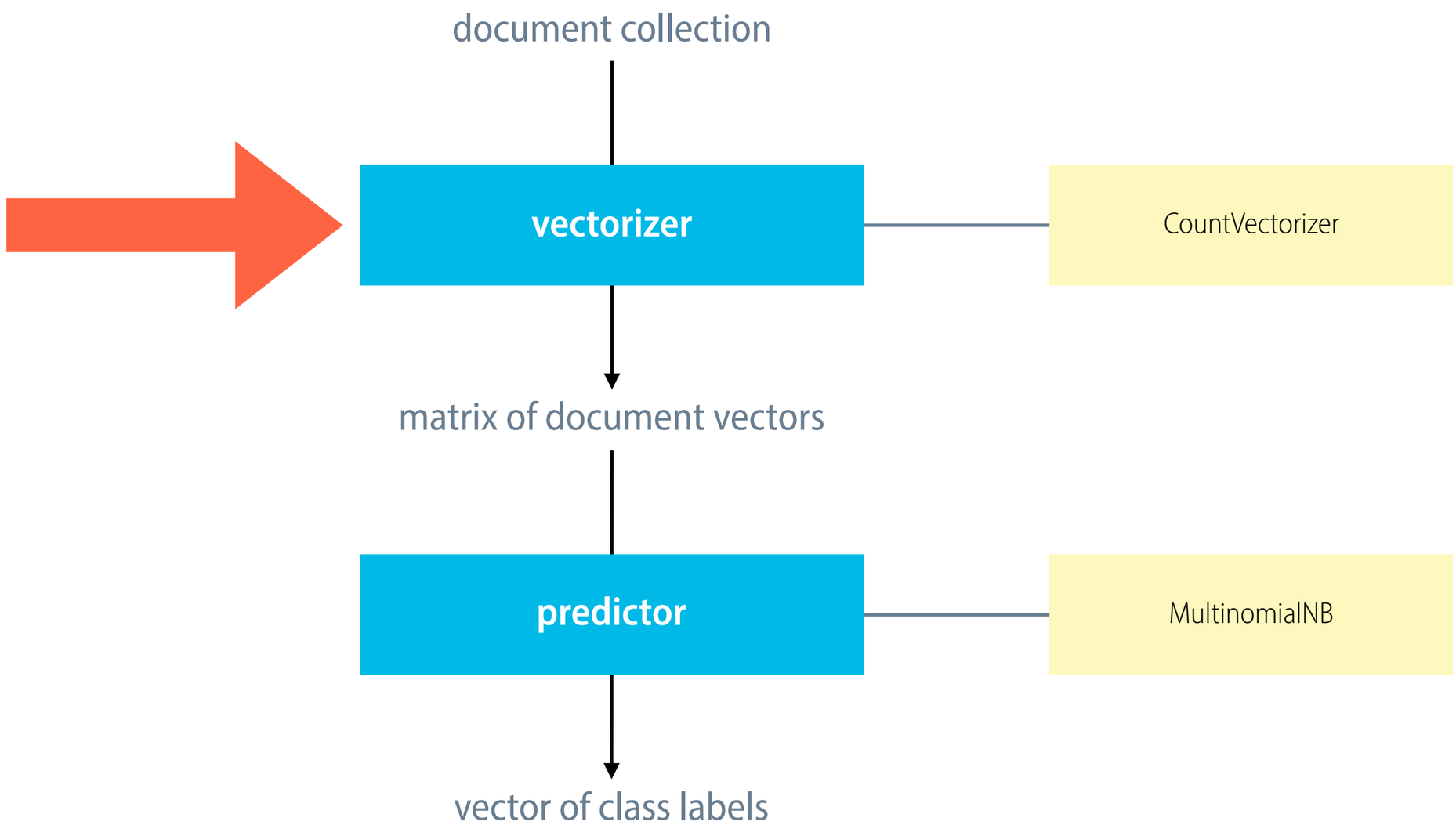
… is a sour little movie at its core; an exploration of the emptiness that underlay the relentless gaiety of the 1920's, as if to stop would hasten the economic and global political turmoil that was to come.

negative

The standard text classification pipeline



The standard text classification pipeline



Reminder: Documents as tf–idf vectors

	Scandal in Bohemia	Final problem	Empty house	Norwood builder	Dancing men	Retired colourman
Adair	0,0000	0,0000	0,0692	0,0000	0,0000	0,0000
Adler	0,0531	0,0000	0,0000	0,0000	0,0000	0,0000
Lestrade	0,0000	0,0000	0,0291	0,1424	0,0000	0,0000
Moriarty	0,0000	0,0845	0,0528	0,0034	0,0000	0,0000

Documents as count vectors – 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

Documents as count vectors – 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

Documents as count vectors – the bag of words

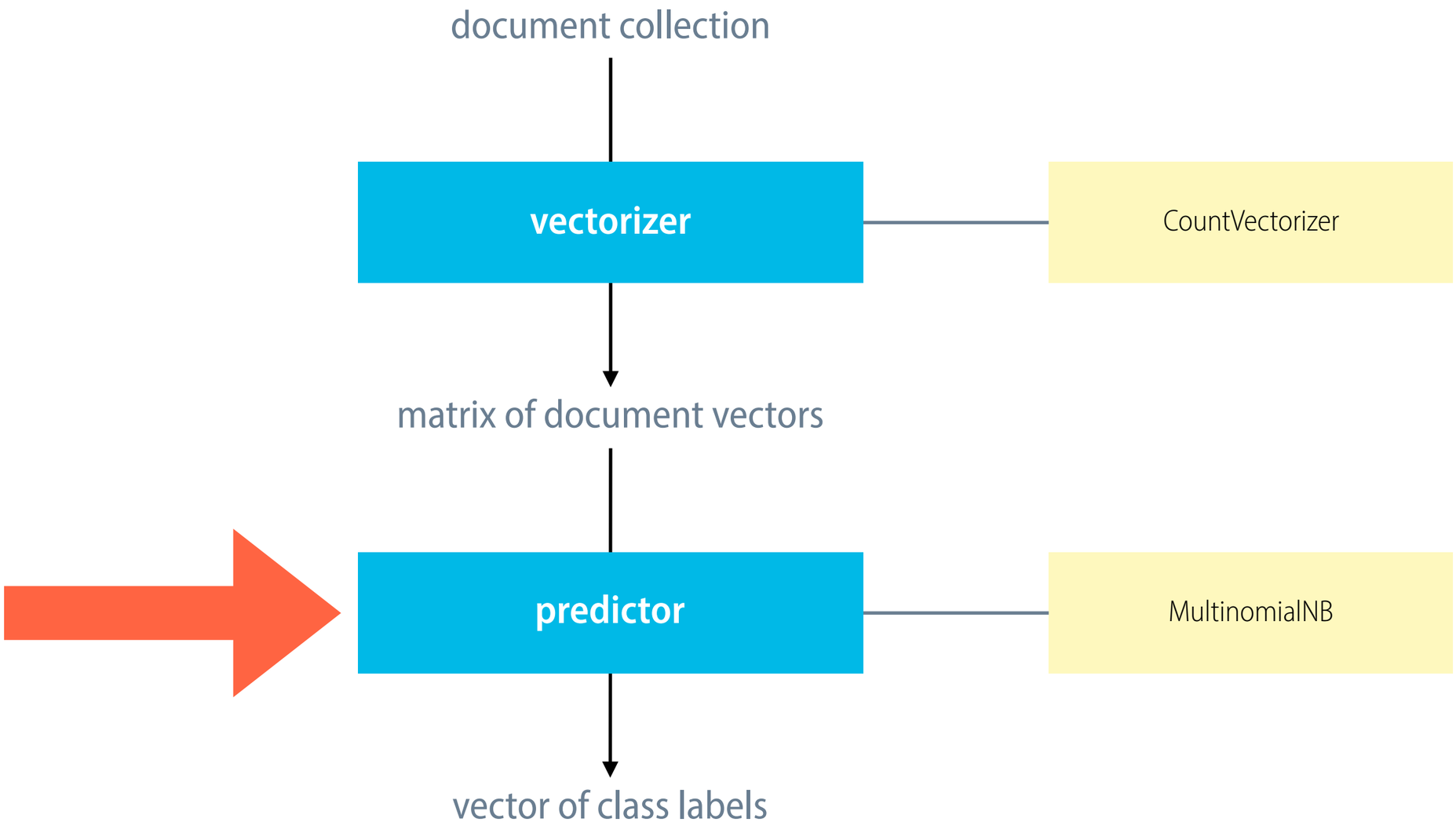
Word	Count
of	4
the	3
words	1
vision	1
trilogy	1
…	

positive

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

negative

The standard text classification pipeline



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 a document x and its gold-standard class y and apply some learning algorithm.

  maximise likelihood/minimise cross-entropy of the training data

- **Testing**

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

Two challenges in text classification

- Standard document representations such as the bag of words easily yield tens of thousands of features.

  computational challenge, data sparsity

- Many document collections are highly imbalanced with respect to the represented classes.

  frequency bias, problems for evaluation

This lecture

- Introduction to text classification
- Evaluation of text classifiers
- The Naive Bayes classifier
- The Logistic Regression 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.

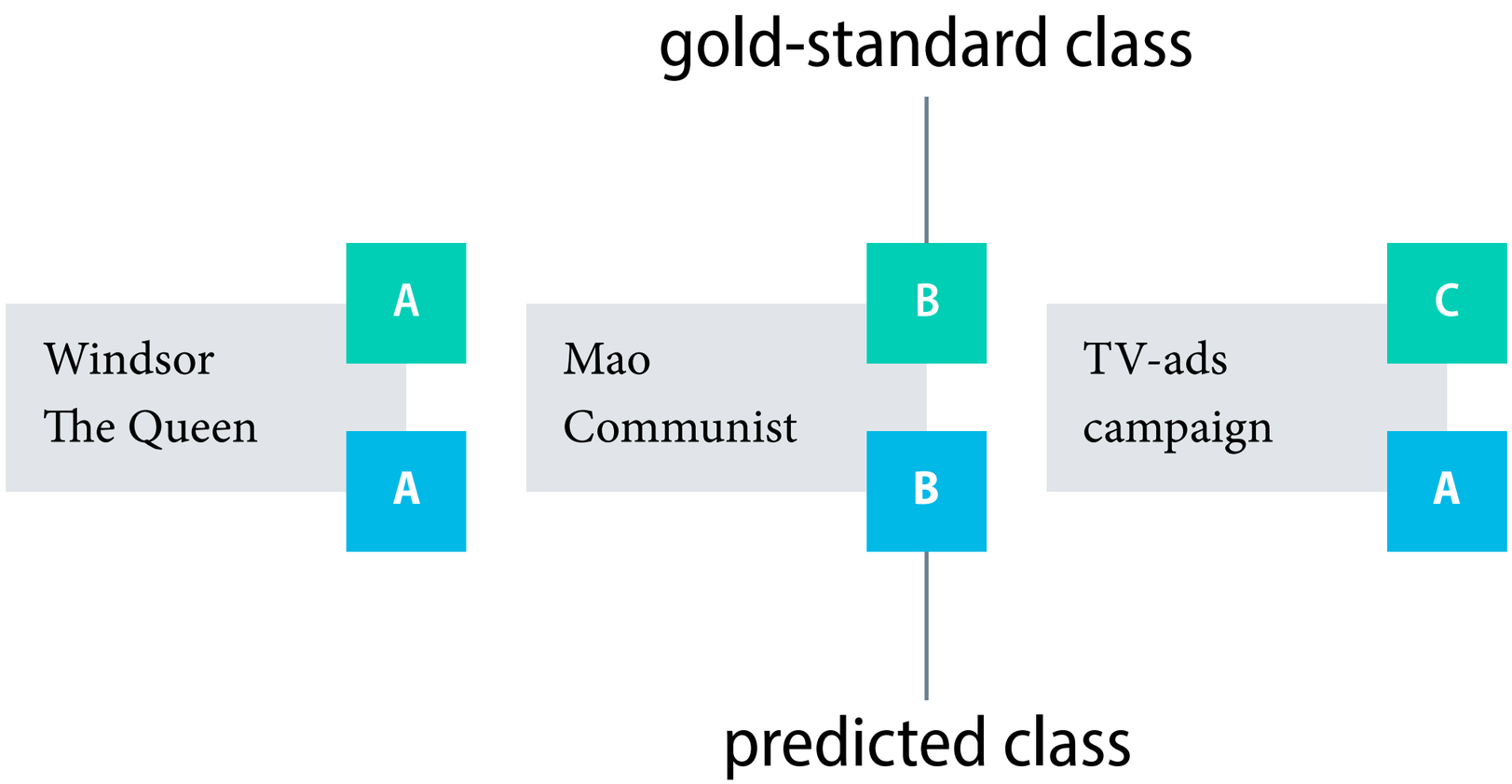
  typically part of a larger gold-standard data set

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

- We do so in order to estimate how well the classifier will perform on new, previously unseen documents.

  assume that all samples are drawn from the same distribution

Evaluation of text classifiers



Accuracy

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

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

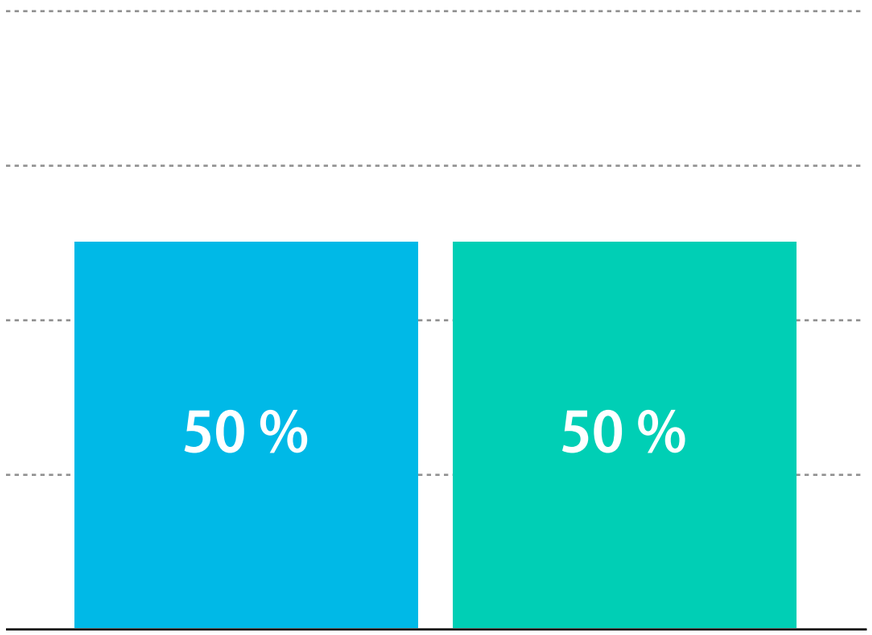
Accuracy

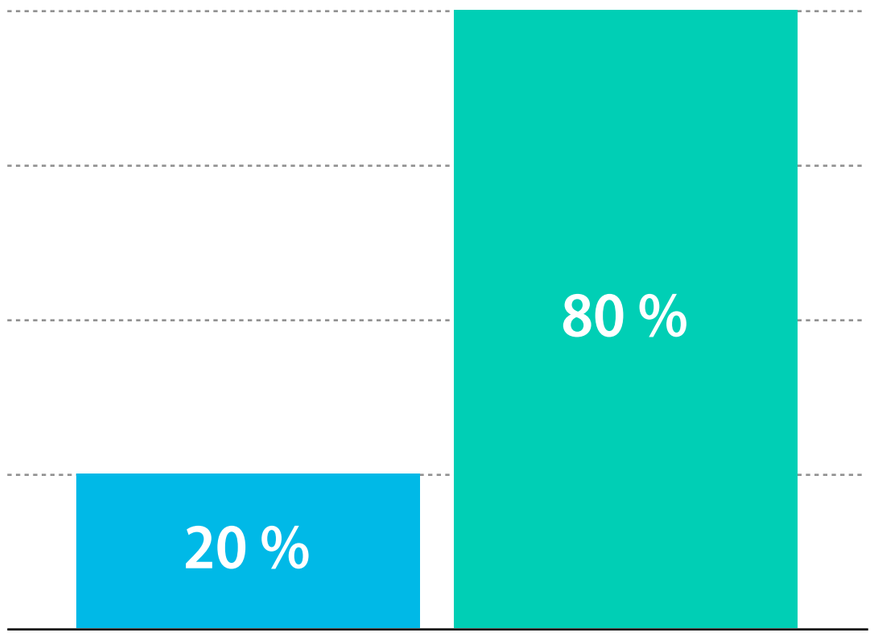
Document	Gold-standard class	Predicted class
Chinese Beijing Chinese	China	China
Chinese Chinese Shanghai	China	China
Chinese Macao	China	China
Tokyo Japan Chinese	Japan	China

accuracy for this example: $3/4 = 75\%$

Accuracy and imbalanced data sets

Is 80% accuracy good or bad?





The role of baselines

- Evaluation measures are no absolute measures of performance.

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

- Instead, we should ask for a classifier’s performance relative to other classifiers, or other points of comparison.

  ‘Logistic Regression has a higher accuracy than Naive Bayes.’

- When other classifiers are not available, a simple baseline is to always predict the **most frequent class** in the training data.

Grading criteria

Soundness and correctness Is the technical approach sound and well-chosen? Are the claims made in the report supported by proper experiments, and are the results of these experiments correctly interpreted?

0 Troublesome. There may be some ideas worth salvaging here, but the work should really have been done or evaluated differently. The claims made in the report have no support in the experimental results.

3 Fairly reasonable work. The approach is not bad, the methods are appropriate, and at least the main claims are probably correct. The report contains a discussion of the possibilities and limitations of the technical approach.

5 The approach is very apt, and the claims are convincingly supported. The report contains a well-developed discussion of the possibilities and limitations of the technical approach, including the reliability and validity of the results.

The role of baselines

- Evaluation measures are no absolute measures of performance.

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

- Instead, we should ask for a classifier’s performance relative to other classifiers, or other points of comparison.

  ‘Logistic Regression has a higher accuracy than Naive Bayes.’

- When other classifiers are not available, a simple baseline is to always predict the **most frequent class** in the training data.

Confusion matrix

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

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 class *c*, how often is it correct?

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

  When the document has class *c*, 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

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

Precision and recall with respect to the positive class

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

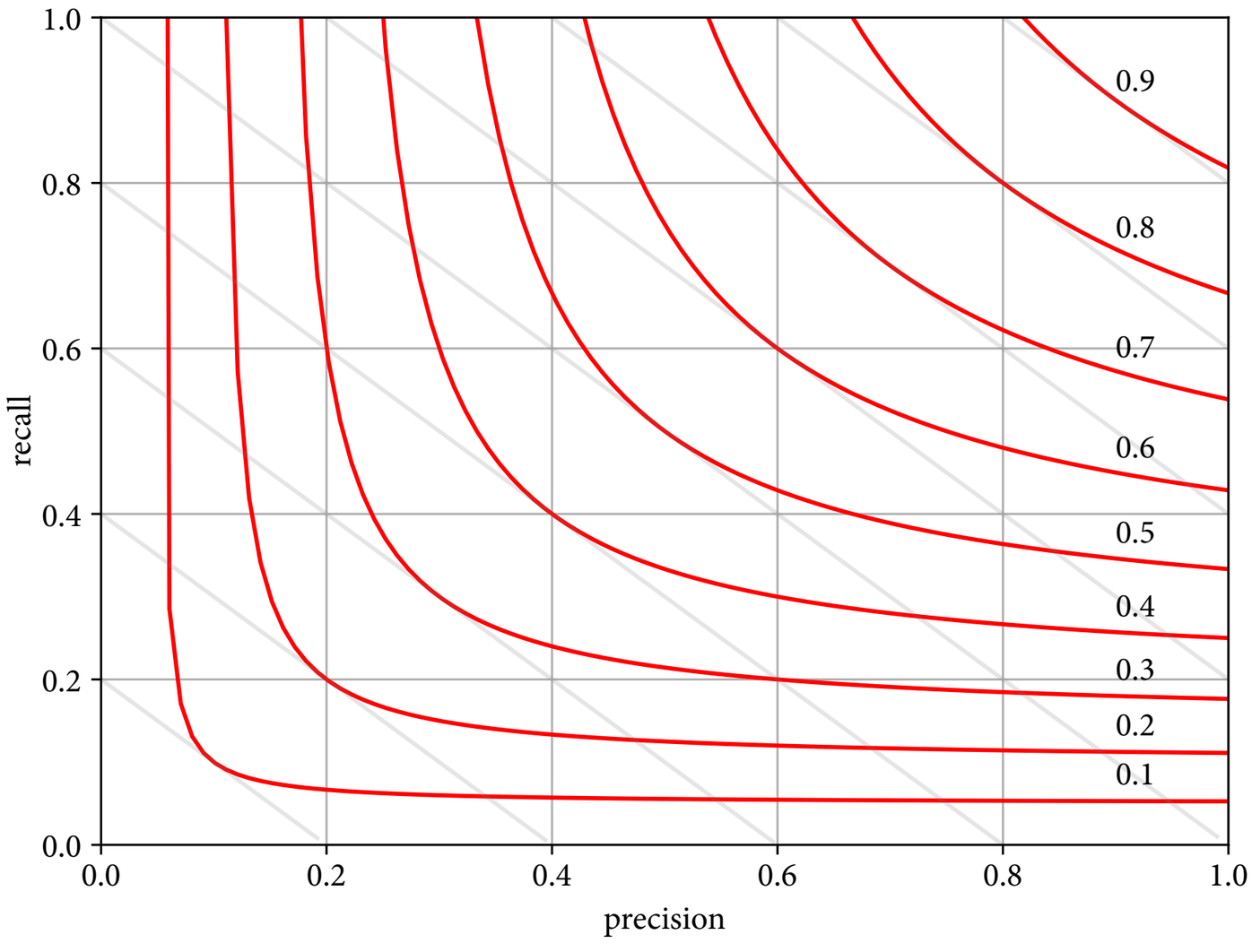
$$\text{recall} = \frac{\#\text{ true positives}}{\#\text{ true positives} + \#\text{ false negatives}}$$

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

Classification report in scikit-learn

	precision	recall	f1-score	support
C	0.63	0.04	0.07	671
KD	0.70	0.02	0.03	821
L	0.92	0.02	0.04	560
M	0.36	0.68	0.47	1644
MP	0.36	0.25	0.29	809
S	0.46	0.84	0.59	2773
SD	0.57	0.12	0.20	1060
V	0.59	0.15	0.24	950
accuracy			0.43	9288
macro avg	0.57	0.26	0.24	9288
weighted avg	0.52	0.43	0.34	9288

This lecture

- Introduction to text classification
- Evaluation of text classifiers
- The Naive Bayes classifier
- The Logistic Regression classifier

The Naive Bayes classifier

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.

Naive Bayes decision rule, informally

?

elaborate
gorgeously

positive

elaborate
gorgeously

score(**pos**) =

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

70%

negative

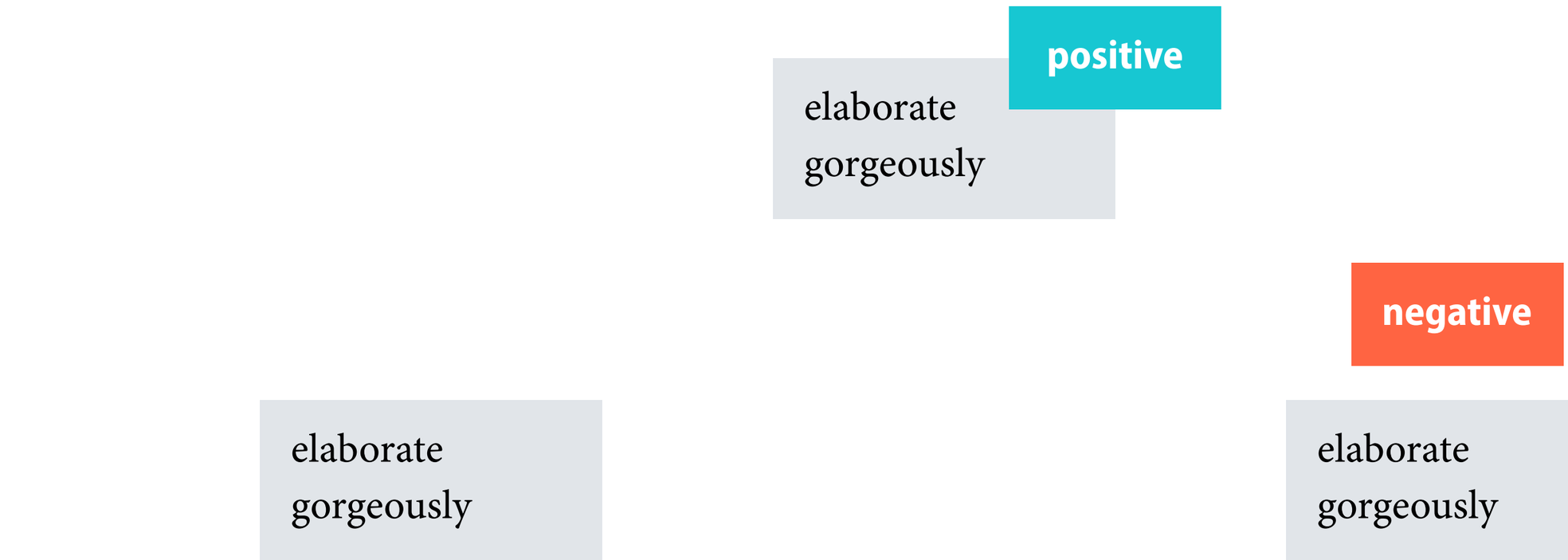
elaborate
gorgeously

score(**neg**) =

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

30%

Naive Bayes decision rule, informally



The role of Bayes' rules

- 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.

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

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 decision rule

choose that class c which 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
for the document

count of the word w
in the document

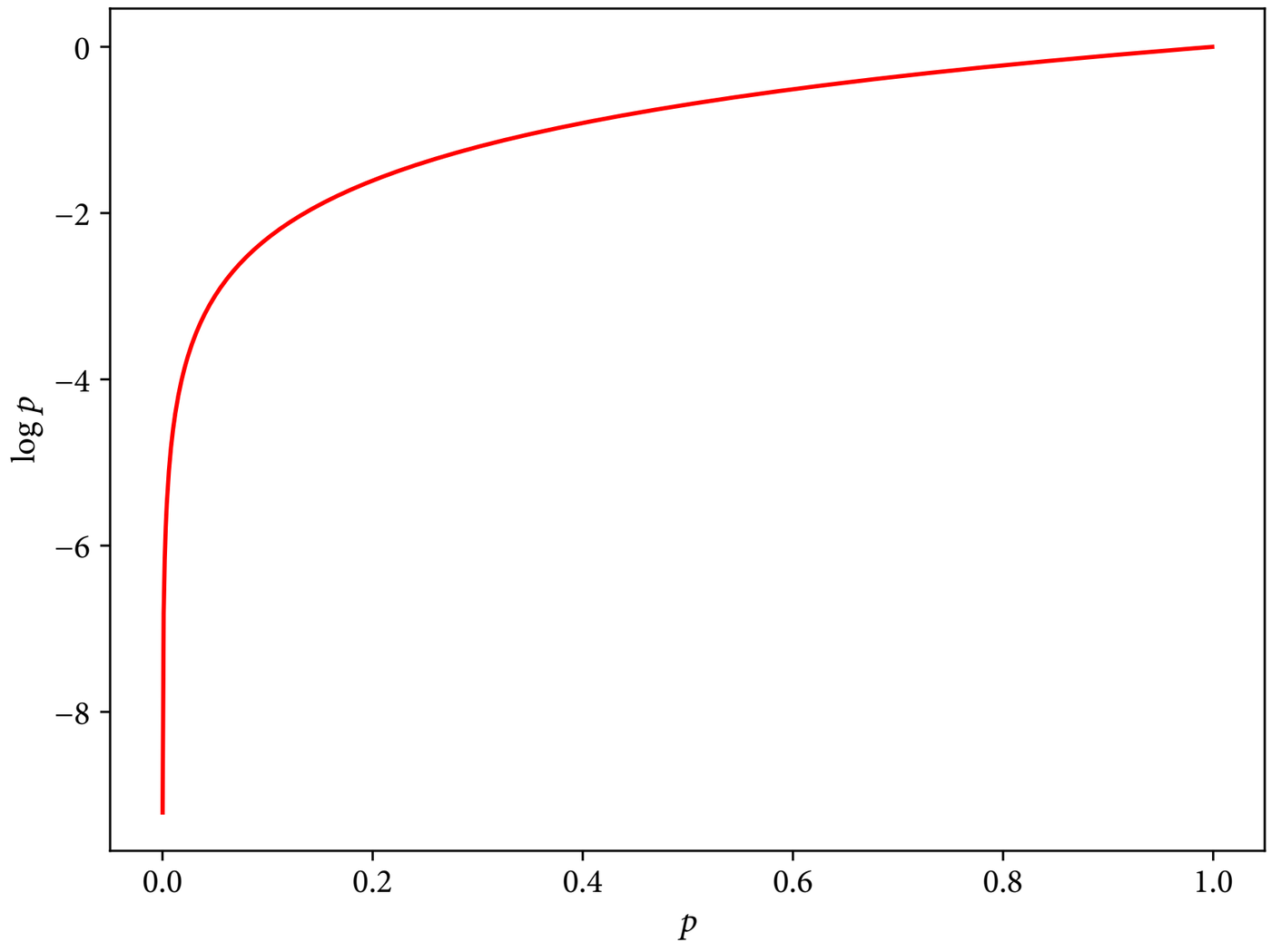
Log probabilities

- In order to avoid underflow, implementations use the logarithms of probabilities instead of the probabilities themselves.

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

- Note that in this case, instead of multiplying probabilities, we have to add their logarithms.

Log probabilities



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

Maximum Likelihood Estimation (MLE)

- One standard technique for estimating the probabilities of a model is **Maximum Likelihood Estimation** (MLE).

  Find probabilities that maximise the probability of the training data.

- Depending on the type of model under consideration, MLE can be computationally challenging.

- 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.

  For example, ‘gorgeously’ may only occur in positive documents.

- 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.

Additive smoothing

- In **additive smoothing**, we add a constant value $\alpha \geq 0$ to all counts before computing relative frequencies.

  For $\alpha = 1$, this technique is also known as Laplace smoothing.

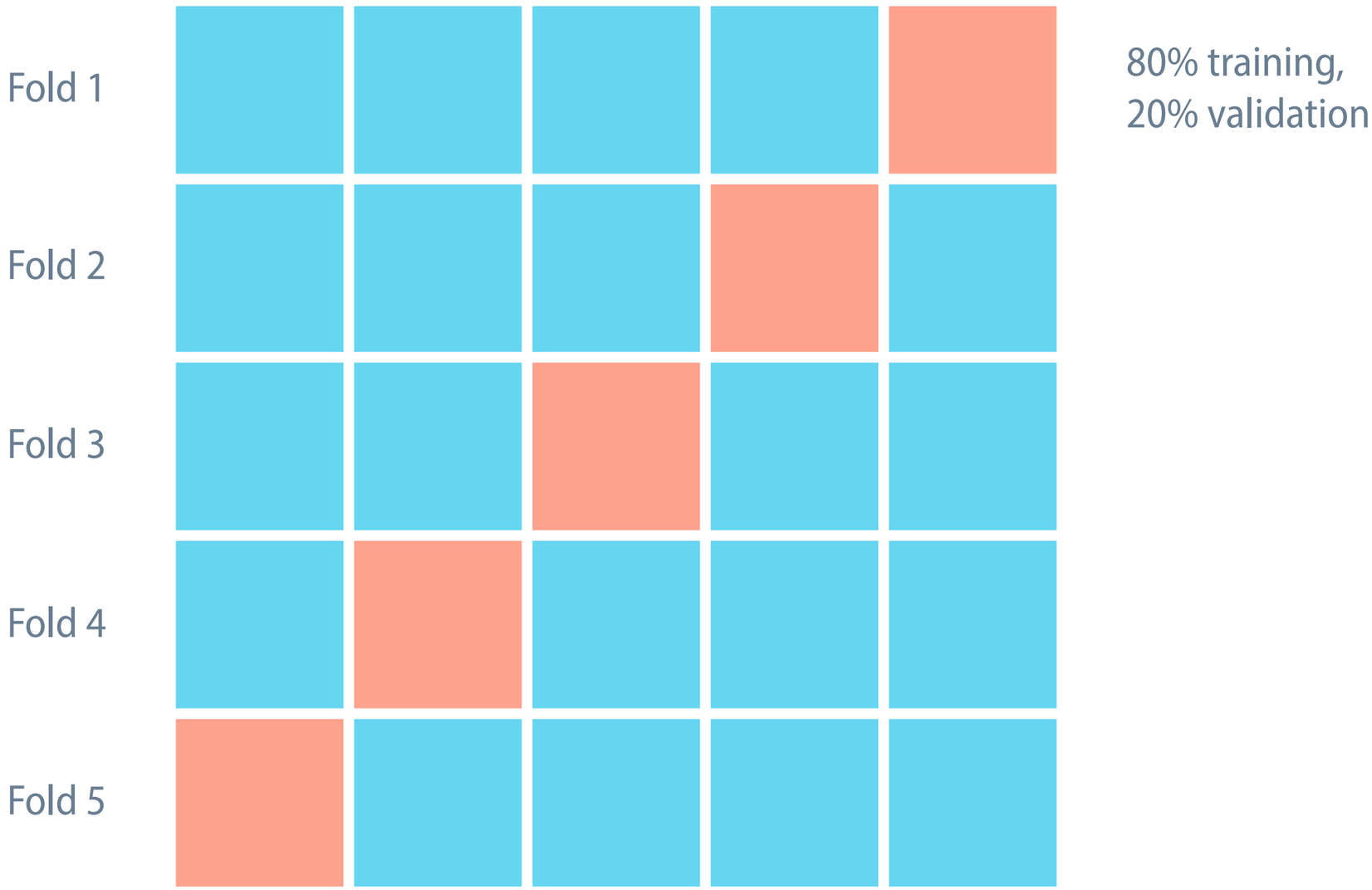
- Effectively, we hallucinate α extra occurrences of each word, including words that never occurred.

- The smoothing constant α is a hyperparameter of the model that needs to be tuned on validation data.

Hyperparameters and cross-validation

- A **hyperparameter** is a parameter of a machine learning model that is not set by the learning algorithm itself.
- To tune hyperparameters, we need a separate validation set, or need to use **cross-validation:**
  - Split the data into k parts.
  - In each fold, use $k - 1$ parts for training, 1 part for validation.
  - Take the mean performance over the k folds.

Five-fold cross validation



Additive smoothing

- In **additive smoothing**, we add a constant value $\alpha \geq 0$ to all counts before computing relative frequencies.

  For $\alpha = 1$, this technique is also known as Laplace smoothing.

- Effectively, we hallucinate α extra occurrences of each word, including words that never occurred.

- The smoothing constant α is a hyperparameter of the model that needs to be tuned 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

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

no smoothing here!

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

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

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 is assuming a uniform prior

Let N_c be the total number of word occurrences in class c. Then

$$P(w \mid c) = \frac{\#(w, c) + \alpha}{N_c + \alpha V}$$

can be written as a linear interpolation of the maximum-likelihood estimate and the uniform distribution over word types:

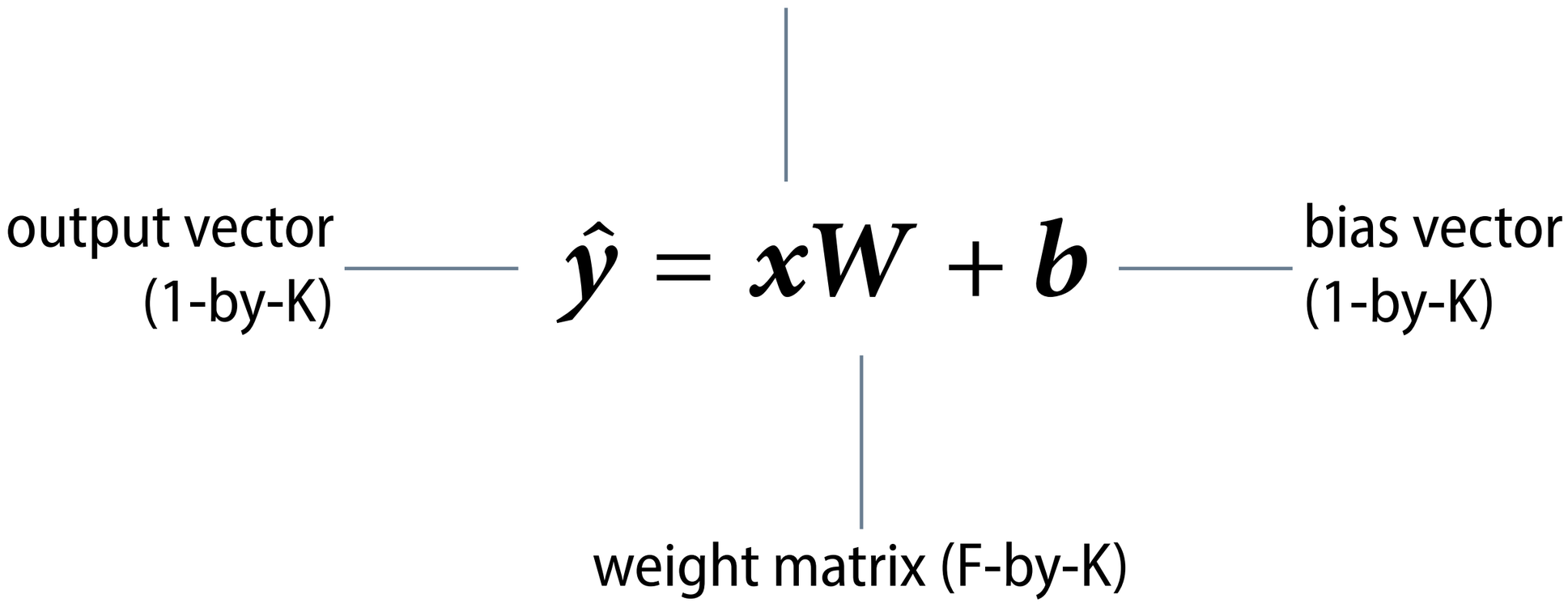
$$P(w \mid c) = \lambda \frac{\#(w, c)}{N_c} + (1 - \lambda)\frac{1}{V} \qquad \text{where} \qquad \lambda = \frac{N_c}{N_c + \alpha V}$$

This lecture

- Introduction to text classification
- Evaluation of text classifiers
- The Naive Bayes classifier
- The Logistic Regression classifier

The Logistic Regression classifier

The multi-class linear model



F = number of features (input variables), K = number of classes (output variables)

The logistic model

- The **logistic model** extends the linear model by adding a pointwise logistic function f:

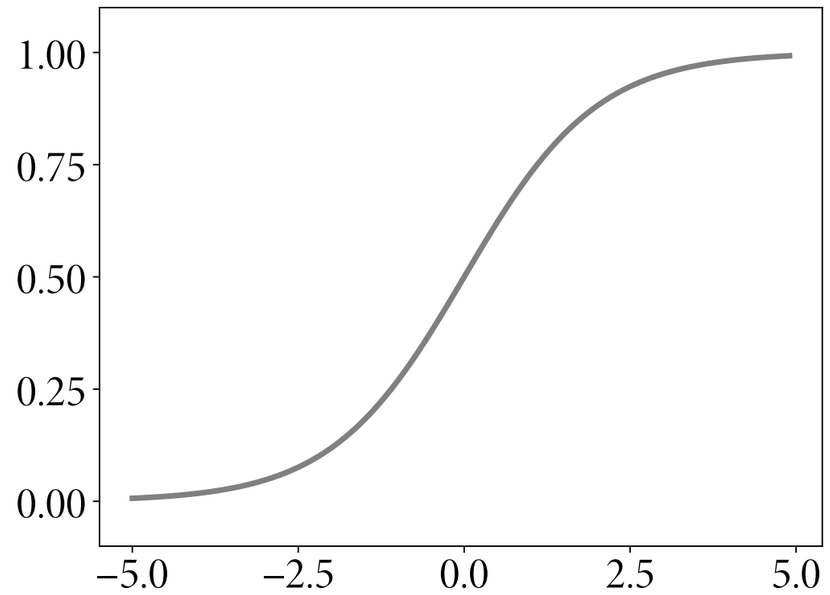
$$\hat{\boldsymbol{y}} = f(\boldsymbol{z}) \qquad \text{where} \qquad \boldsymbol{z} = \boldsymbol{x}\boldsymbol{W} + \boldsymbol{b}$$

logit

- The term **logistic regression** refers to the procedure of learning the parameters of the logistic model.

  That model is then used for classification. Confusing terminology!

The standard logistic function for binary classification



$$\hat{y} = \frac{1}{1 + \exp(-z)}$$

The softmax function for k-ary classification

- The **softmax function** converts a vector of activations into a vector of probabilities (a finite probability distribution).

$$\text{softmax}(\boldsymbol{z})[i] \;=\; \frac{\exp(\boldsymbol{z}[i])}{\sum_k \exp(\boldsymbol{z}[k])}$$

- The softmax function is the natural generalisation of the standard logistic function to k-ary classification problems.

Training a logistic regression classifier

- We present the model with training samples of the form $(\boldsymbol{x}, y)$ where $\boldsymbol{x}$ is a feature vector and y is the gold-standard class.

- The output of the model is a vector of conditional probabilities $P(k \mid \boldsymbol{x})$ where k ranges over the possible classes.

- We want to train the model so as to maximise the likelihood of the training data under this probability distribution.

  Same training principle as for Naive Bayes – but no closed-form solution.

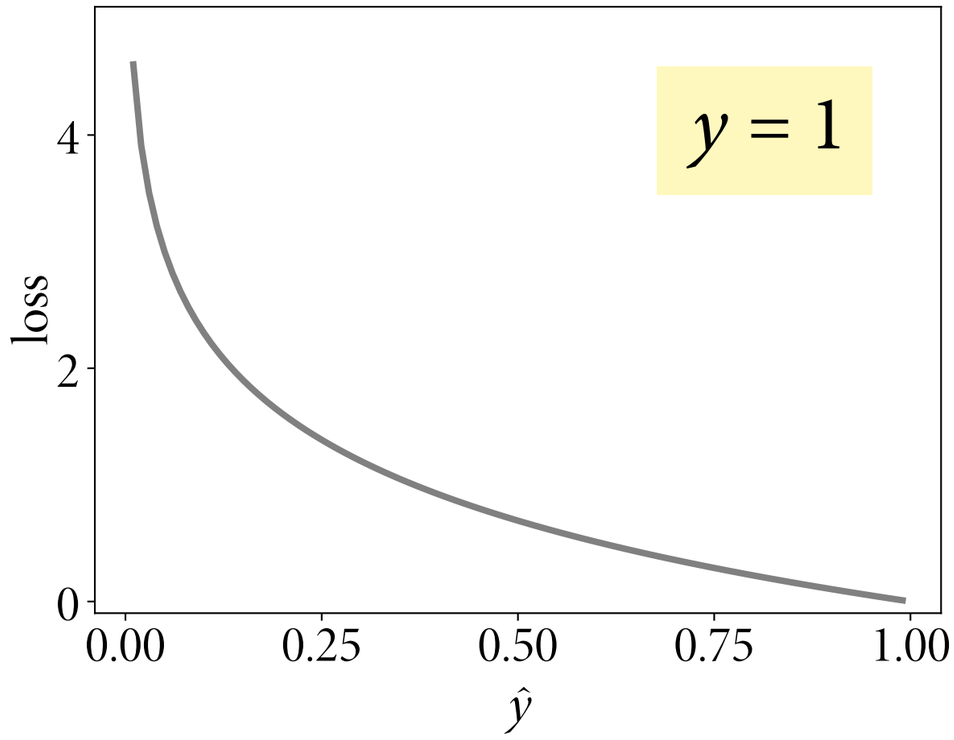
Gradient descent

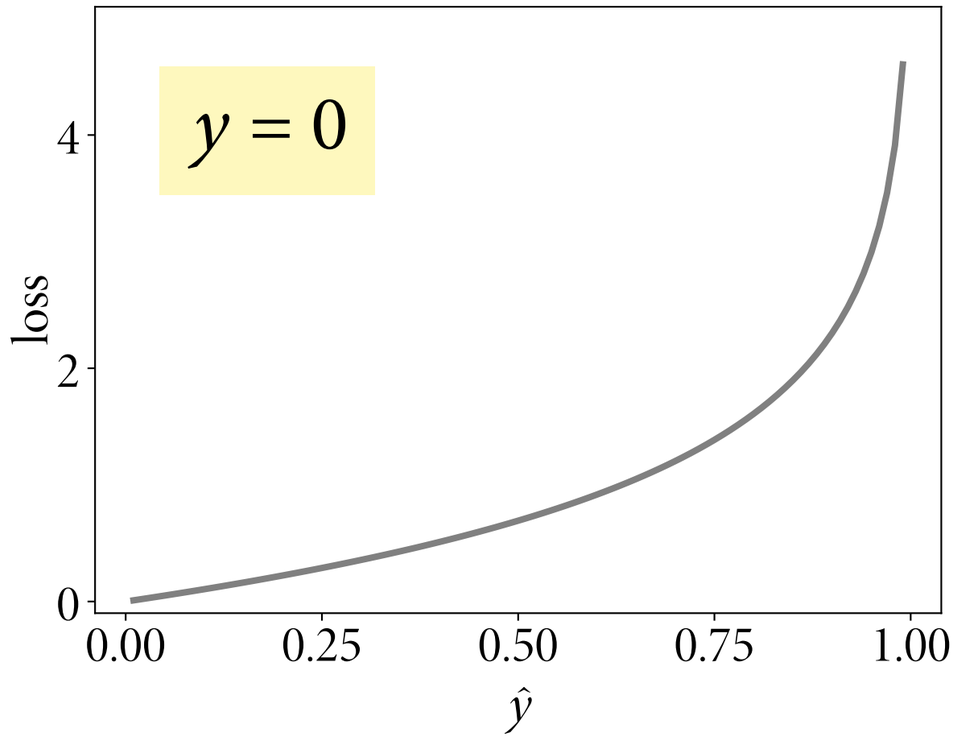
'Follow the gradient into the valley of error.'

- **Step 0:** Start with an arbitrary value for the parameters $\boldsymbol{\theta}$.
- **Step 1:** Compute the gradient of the loss function, $\nabla L(\boldsymbol{\theta})$.
- **Step 2:** Update the parameters $\boldsymbol{\theta}$ as follows: $\boldsymbol{\theta} \coloneqq \boldsymbol{\theta} - \eta \nabla L(\boldsymbol{\theta})$

  The hyperparameter η is the learning rate.
- Repeat step 1–2 until the error is sufficiently low.

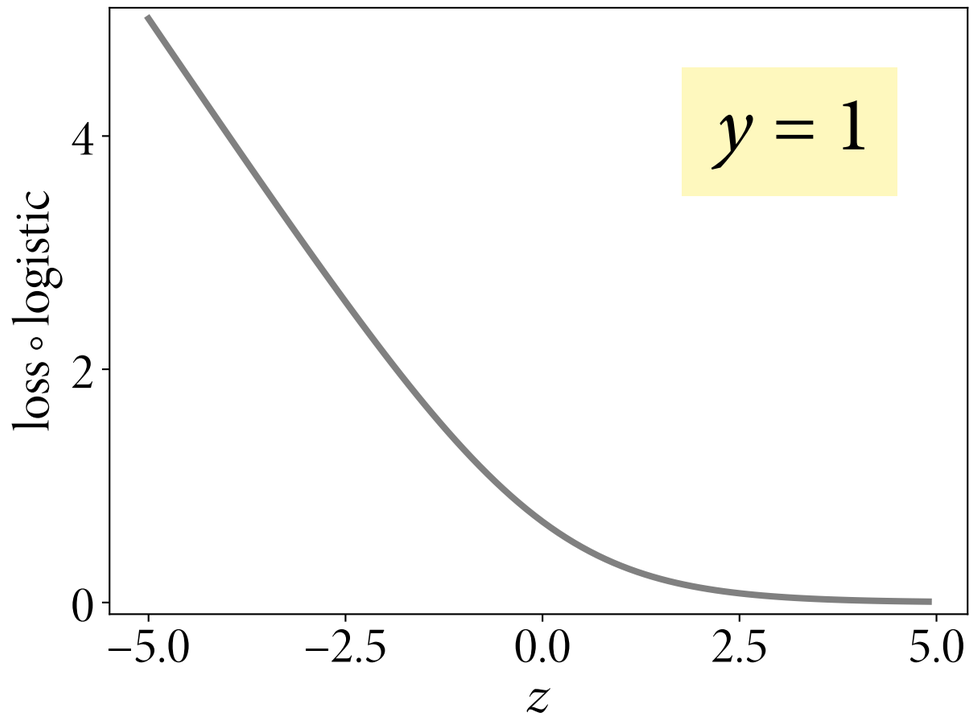
Cross-entropy loss function

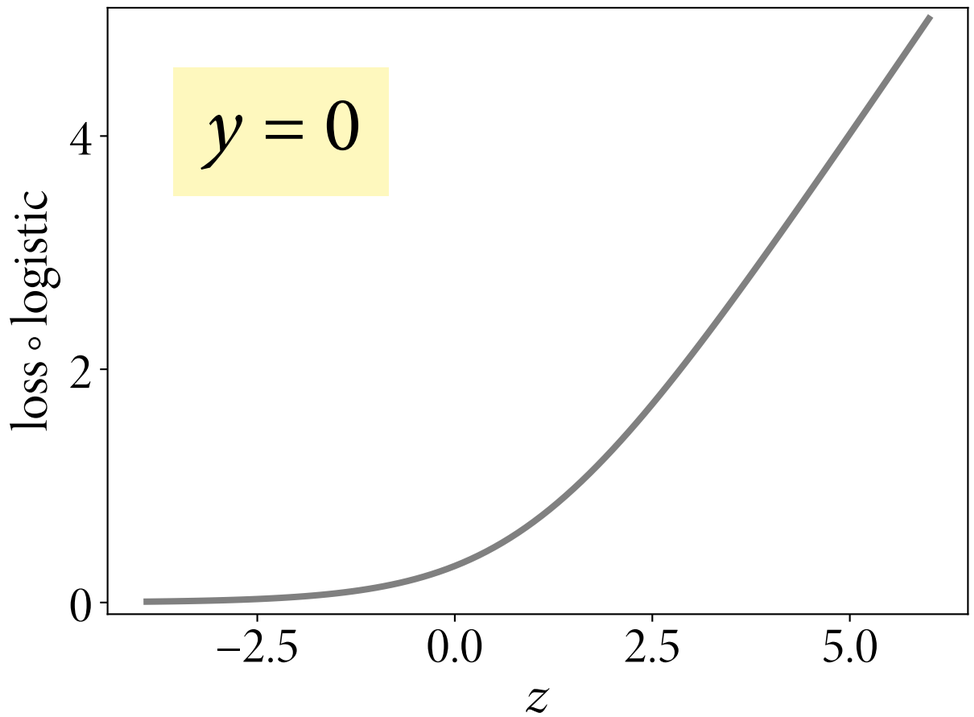




$$L(\boldsymbol{\theta}) = -\big(y \log \hat{y} + (1 - y) \log(1 - \hat{y})\big)$$

Cross-entropy loss for logistic units





$$\frac{dL(\boldsymbol{\theta})}{dz} = \hat{y} - y$$

Cross-entropy loss function for the softmax function

For a logistic model with parameters $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{b})$, the **cross-entropy loss** for a sample $(\boldsymbol{x}, y)$ is defined as

$$L(\boldsymbol{\theta}) = -\sum_k [k = y] \log \operatorname{softmax}(\boldsymbol{x}\boldsymbol{W} + \boldsymbol{b})[k]$$

where we write $[k = y]$ for the function that returns 1 if $k = y$, and 0 otherwise (Iverson bracket).

Regularisation

- The term **regularisation** refers to all changes to a model that are intended to decrease generalisation error but not training error.
- A standard for logistic regression is **L2-regularisation**, where we penalise parameter vectors based on their lengths:

$$\boldsymbol{\theta} := \boldsymbol{\theta} - \eta\left(\nabla L(\boldsymbol{\theta}) + \rho\|\boldsymbol{\theta}\|_2\right)$$

regularisation strength

Logistic Regression and Naive Bayes

- Learning a Naive Bayes classifier involves fitting a probability model that optimizes the joint likelihood $P(\text{class}, \text{document})$.

  $P(\text{class} \mid \text{document})$ is obtained via Bayes' rule.

- Logistic regression fits the same probability model to directly optimize the conditional $P(\text{class} \mid \text{document})$.

  lower aymptotic error

- Naive Bayes and logistic regression form a **generative–discriminative pair**.

This lecture

- Introduction to text classification
- Evaluation of text classifiers
- The Naive Bayes classifier
- The Logistic Regression classifier

Project idea

- There are many other text classification methods than Naive Bayes and Logistic regression.

  support vector machines, neural networks, …

- Pick a method that you find interesting and compare its performance to that of other methods on a standard data set.

- In the report, make it clear why you chose this method, and that you have understood how it works.

Beyond the bag-of-words assumption

- The linear sequence of the words in a text carries meaning.

  The brown dog on the mat saw the striped cat through the window.

  The brown cat saw the striped dog through the window on the mat.

- We can try to capture at least parts of this sequence by representing texts as ***n*-grams**, contiguous sequences of *n* words.

  unigram (*not*), bigram (*not bad*)

- **Recurrent neural networks** have been designed to explicitly capture long-range dependencies between words.