Text Classification

Marcel Bollmann

Department of Computer and Information Science (IDA)

LINKÖPING UNIVERSITY

Adapted from slides by Marco Kuhlmann.

Today's lecture

1. Introduction
   - Examples
   - Machine Learning
2. Evaluation
   - Accuracy
   - Confusion Matrix
   - Precision/Recall/F1
   - The Importance of Baselines
3. Naive Bayes
   - Bag of Words
   - Decision Rule
   - Training via MLE
   - Smoothing

Introduction to Text Classification

1

What is text classification?

Definition

Text classification is the task of categorizing text documents into predefined classes.

- "Text documents" can refer to text of any granularity.
  - social media posts
  - newspaper articles
  - entire books
  - …

Example: Sentiment analysis

I love it so much! The mic works great!!!! I use it for online live classes, cosplay, and to look cute!! The lightup feature really works great! The app also works great too! The sound sounds amazing too! I just wish it had a case for when I travel.

positive

Not durable. The cord came apart from the audio adjuster. The saddest part is that happens only two months after it was purchased, and no force was applied. Definitely, I will not purchase and I do not recommend the item.

negative

Adapted from Amazon

Example: Topic classification

It took them an hour of huffing and puffing, but Arsenal did something at Stamford Bridge they hadn't managed since September – they scored an away goal in the Premier League.

- ✗ Business
- ✗ Politics
- ✗ Technology
- ✓ Sports
- ✗ Entertainment

Quote source: The Guardian

Example: Natural language inference

Premise	A man inspects the uniform of a figure in some East Asian country.
Hypothesis	The man is sleeping.
Label	**Contradiction**

Premise	Soccer game with multiple males playing.
Hypothesis	Some men are playing a sport.
Label	**Entailment**

Premise	An older and younger man smiling.
Hypothesis	Two men are smiling and laughing at the cats playing on the floor.
Label	**Neutral**

Source: NLP-progress

Text classification as machine learning

training set learning	Hudson River (A)	Paul Klee (B)	Niche Retail (C)
	Badger Mountain (A)	Linda Chapman (B)	Polk Brothers (C)
test set evaluation	Shelf Glacier (A)	Charles Keating (B)	Aarti Industries (C)

Inspired by DBpedia14: A ∼ Natural Place; B ∼ Artist; C ∼ Company

Training and testing

training set
learning

Hudson River (A)	Paul Klee (B)	Niche Retail (C)
Badger Mountain (A)	Linda Chapman (B)	Polk Brothers (C)

When we train a classifier, we present it with a **document** x and its **gold-standard class** y and apply some **learning algorithm**.

Training and testing

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

test set
evaluation

Document	Gold-standard class	Predicted class
Shelf Glacier	A	A
Charles Keating	B	B
Aarti Industries	C	A

General machine learning methodology

- In **supervised machine learning**, we apply some learning algorithm to optimize performance on the **training data**.
  - supervised = the training data is labelled with the "correct" class
- The goal is to **optimize** performance on **new, unseen data**.
  - "How well does the system generalize?"
- We **estimate** this performance using separate **test data**.

Recurring questions

- How does this method work?
  - algorithm, mathematical formula, …
- How can we evaluate this method?
  - accuracy, precision/recall, …
- How does this method use data?
  - estimate probabilities, learn weights of a neural network, …

Evaluation of Text Classifiers

2

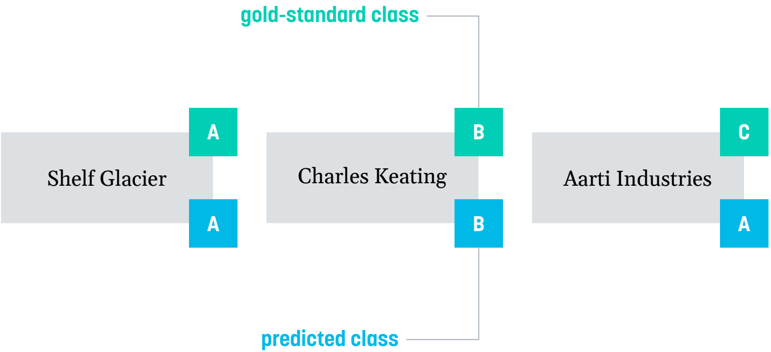
Evaluation of text classifiers

- We need a **test set** of documents with **gold-standard labels**.
  - gold-standard = assumed to be correct; often produced or verified manually
- We **apply** the classifier to our test set and **compare** the predicted classes with the gold-standard classes.

💡 Idea: This estimates how well the classifier **generalizes** to new documents.

  - This is why it's important that documents in the test set are *not* in the training set!

Accuracy


gold-standard class
A
Shelf Glacier
A
B
Charles Keating
B
C
Aarti Industries
A
predicted class

Accuracy

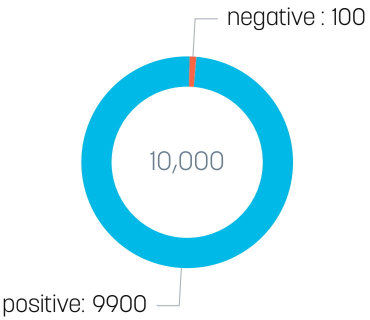
- **Accuracy** is the proportion of documents for which the classifier was “correct”.

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

Accuracy can be problematic

- If the class labels are **unbalanced**, accuracy can be misleading.

- A classifier that **always predicts "positive"** would achieve **99% accuracy** if the test data looks as seen on the right.
  - Not a very useful classifier…



Confusion matrix

		predicted	
		positive	negative
gold-standard	positive	0	100
	negative	0	9,900

Confusion matrix

		predicted	
		positive	negative
gold-standard	positive	true positive	false negative
	negative	false positive	true negative

Confusion matrix with three classes

		predicted		
		A	B	C
gold-standard	A	58	6	1
	B	5	11	2
	C	0	7	43

Documents labelled "B" in the gold-standard
where the model predicted "C"

Accuracy

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

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

Precision and recall

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

Precision

When the model predicts class x, how often is it correct?

Recall

When the document has class x, how often does the model predict it?

- The proportion of correctly classified documents among all documents for which **the model predicts** class x.

- The proportion of correctly classified documents among all documents for which **the gold-standard class** is x.

Precision and recall with two classes

- Precision and recall are always computed **with respect to a class**.
- In a two-class setting, they are usually defined with respect to the **positive class**.
  - assumes two classes 'positive' and 'negative'

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

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

Precision with respect to class B

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

$$\text{precision} = \frac{\text{\# of true positives for "B"}}{\text{\# of all documents predicted to be "B"}}$$

Recall with respect to class B

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

$$\text{recall} = \frac{\text{\# of true positives for “B”}}{\text{\# of all documents labelled “B” in the gold-standard}}$$

F1-measure

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

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

The importance of baselines

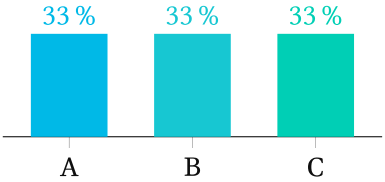
- Evaluation metrics are **no absolute measures** of performance.
  - What is “good” depends on the task!

If you think:	You should ask yourself:
“This classifier performs very well!”	*“...compared to what?”*

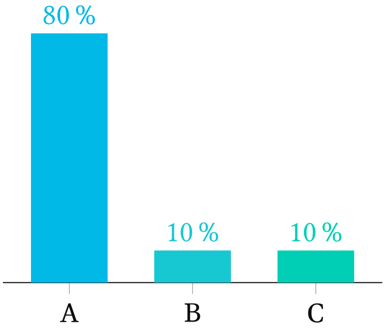
- We should judge a classifier’s performance by **comparing it** against something else.
  - “Logistic regression achieves better accuracy than Naive Bayes.”
- The point of comparison is often called the **baseline**.

Most-frequent-class baseline

- A simple baseline is to always predict the **most frequent class** in the training data.



A classifier with 80% accuracy
could be pretty good here!



A classifier with 80% accuracy
is not better than the MFC baseline...

Important concepts

- accuracy, precision, recall, F1-score
- confusion matrix
- baselines, most-frequent-class baseline

The Naive Bayes Classifier

3

Naive Bayes

- The **Naive Bayes classifier** is a simple but effective probabilistic text classifier.
  - Bayes' rule: $P(A|B) = \dfrac{P(B|A) \cdot P(A)}{P(B)} \propto P(B|A) \cdot P(A)$

- It is called **"naive"** because it makes strong (unrealistic) independence assumptions about probabilities.
  - i.e., the exact probability values it outputs should not be taken too seriously

- Before we can use it, we need to decide **how to represent our documents**.

Bag of words

- Simply counting the number of times each word occurs is also called a **bag of words**.
  - Word order doesn't matter!

It is a truth universally acknowledged, that a single man in possession of a good fortune must be in want of a wife.

Sentiment analysis with bag-of-words

I love it so much! The mic works great!!!! I use it for online live classes, cosplay, and to look cute!! The lightup feature really works great! The app also works great too! The sound sounds amazing too! I just wish it had a case for when I travel.

positive

Not durable. The cord came apart from the audio adjuster. The saddest part is that happens only two months after it was purchased, and no force was applied. Definitely, I will not purchase and I do not recommend the item.

negative

Sentiment analysis with bag-of-words

a also amazing and app case classes cosplay cute feature for for great great great had I I I I it it it just lightup live look love mic much online really so sound sounds the the the the to too too travel use when wish works works works

positive

adjuster after and and apart applied audio came cord definitely do durable force from happens I I is it item months no not not not only part purchase purchased recommend saddest that the the the the two was was will

negative

Sentiment analysis with bag-of-words: vectors

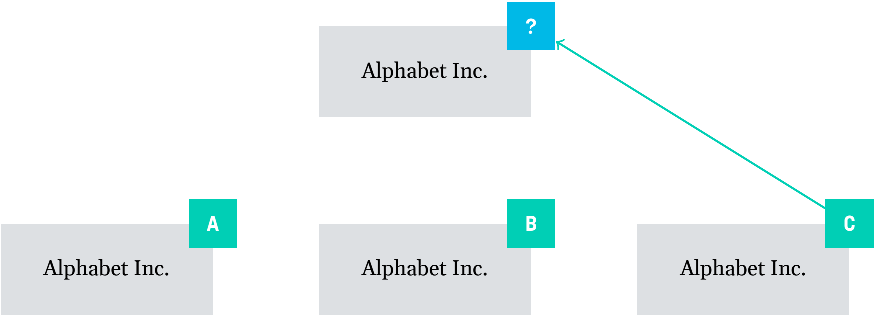
word	count
I	4
the	4
great	3
it	3
works	3
for	2
too	2
...	

positive

word	count
the	4
not	3
and	2
I	2
was	2
adjuster	1
after	1
...	

negative

Naive Bayes decision rule (informally)



$\text{score}(A) =$
$P(\text{A})\ P(\text{Alphabet}|\text{A})\ P(\text{Inc.}|\text{A})$

10%

$\text{score}(B) =$
$P(\text{B})\ P(\text{Alphabet}|\text{B})\ P(\text{Inc.}|\text{B})$

20%

$\text{score}(C) =$
$P(\text{C})\ P(\text{Alphabet}|\text{C})\ P(\text{Inc.}|\text{C})$

70%

The role of Bayes’ rule

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

- But a Naive Bayes classifier learns $P(\textbf{document} \mid \textbf{class})$.
  - This is easy to compute using maximum likelihood estimation (MLE).

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

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

Naive Bayes decision rule (formally)

choose the class c which maximizes the term to the right of the $\arg\max$

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

predicted class

words in the vocabulary

count of the word w in the document

Implementing the decision rule

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

1. If the vocabulary is large, it can **take a long time** to loop over it.
   - Loop only over the words in the document instead.
2. Some words in the document may be **missing** from the vocabulary.
   - Just skip them; that's what the equation says!
3. Multiplying many very small values can result in **underflow**.
   - Can use the logarithm of probabilities instead.

Log probabilities

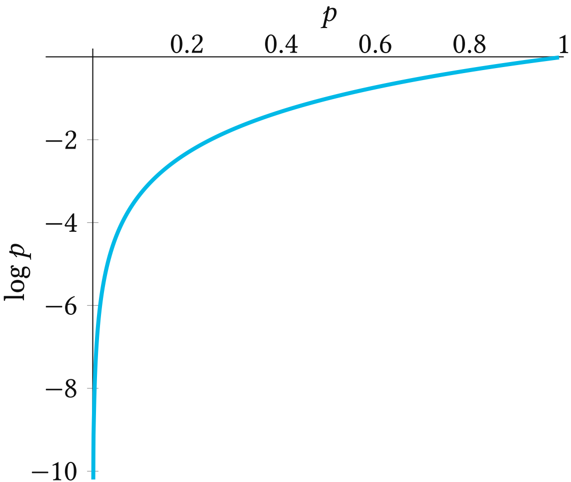
- To avoid underflow, we can use the **logarithms of probabilities** instead of the probabilities themselves.

$$P(w|c) \text{ becomes } \log P(w|c)$$

- Instead of multiplying probabilities, we have to **add** their logarithms.

$$\log(a \cdot b) = \log a + \log b$$

Log probabilities



How do we train a Naive Bayes classifier?

Reminder: Machine learning methodology

training set learning	Hudson River (A)	Paul Klee (B)	Niche Retail (C)
	Badger Mountain (A)	Linda Chapman (B)	Polk Brothers (C)
test set evaluation	Shelf Glacier (A)	Charles Keating (B)	Aarti Industries (C)

Inspired by DBpedia14: A ∼ Natural Place; B ∼ Artist; C ∼ Company

Training a Naive Bayes classifier

Hudson River (A)	Paul Klee (B)	Niche Retail (C)
Badger Mountain (A)	Linda Chapman (B)	Polk Brothers (C)

$$P(c)$$

class probabilities

$$P(w|c)$$

word probabilities

Word probabilities in Naive Bayes

probability	value
…	…
$P(\text{great} \mid \textbf{pos})$	?
$P(\text{works} \mid \textbf{pos})$	?
$P(\text{not} \mid \textbf{pos})$	?
$P(\text{it} \mid \textbf{pos})$	?
…	

positive

probability	value
…	…
$P(\text{great} \mid \textbf{neg})$	?
$P(\text{works} \mid \textbf{neg})$	?
$P(\text{not} \mid \textbf{neg})$	?
$P(\text{it} \mid \textbf{neg})$	?
…	

negative

Maximum Likelihood Estimation (MLE)

- **Maximum likelihood estimation (MLE)** is a simple technique for estimating probabilities.
  - Find probabilities that maximize the likelihood (= probability) of the training data.

- For Naive Bayes: probabilities $\sim$ **relative frequencies**

MLE for Naive Bayes

- 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 Naive Bayes

number of documents with class c

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

number of occurrences of w in documents with class c

$$P(w|c) = \frac{\boxed{\#(w, c)}}{\sum_{x \in V} \#(x, c)}$$

MLE of word probabilities

a also amazing and app case classes cosplay cute feature for for great great great had I I I I it it it just lightup live look love mic much online really so sound sounds the the the the to too too travel use when wish works works works

positive

49 tokens

adjuster after and and apart applied audio came cord definitely do durable force from happens I I is it item months no not not not only part purchase purchased recommend saddest that the the the the two was was will

negative

40 tokens

MLE of word probabilities: counting

word	count
...	...
great	3
works	3
not	0
it	3
...	

positive

word	count
...	...
great	0
works	0
not	3
it	1
...	

negative

MLE of word probabilities: taking the percentage

probability	value
...	...
$P(\text{great} \mid \textbf{pos})$	$3/49$
$P(\text{works} \mid \textbf{pos})$	$3/49$
$P(\text{not} \mid \textbf{pos})$	$0/49$
$P(\text{it} \mid \textbf{pos})$	$3/49$
...	

positive

probability	value
...	...
$P(\text{great} \mid \textbf{neg})$	$0/40$
$P(\text{works} \mid \textbf{neg})$	$0/40$
$P(\text{not} \mid \textbf{neg})$	$3/40$
$P(\text{it} \mid \textbf{neg})$	$1/40$
...	

negative

Smoothing

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

- If $\boxed{P(w|c)}$ corresponds to word frequencies, some probabilities **may be zero**.
- This is a problem, since we're multiplying them!
  - Slogan: Zero probabilities destroy information.
- Use **smoothing** techniques to ensure that probabilities are **always non-zero**!

Add-one smoothing

- **Add-one smoothing** adds 1 to all word counts.
  - Also known as **Laplace smoothing**.
  - Effectively, we "hallucinate" an extra occurrence of every word.

Class probabilities:

$$P(c) = \frac{\#(c)}{\sum_{x \in C} \underbrace{\#(x)}_{\text{no smoothing here}}}$$

Word probabilities:

$$P(w|c) = \frac{\#(w,c) + 1}{\sum_{v \in V} [\#(v,c) + \underbrace{1}_{\text{one extra occurrence of each word}}]}$$

Add-one smoothing

Class probabilities:

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

Word probabilities:

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

$$= \frac{\#(w,c)+1}{\sum_{v \in V} [\#(v,c)] + \boxed{|V|}}$$

number of words in the vocabulary

Vocabulary

a adjuster after also amazing and apart app applied audio came case classes cord cosplay cute definitely do durable feature for force from great had happens I is it item just lightup live look love mic months much no not online only part purchase purchased really recommend saddest so sound sounds that the to too travel two use was when will wish works

positive + negative

63 types

MLE before smoothing

probability	value
…	…
$P(\text{great} \mid \textbf{pos})$	$3/49$
$P(\text{works} \mid \textbf{pos})$	$3/49$
$P(\text{not} \mid \textbf{pos})$	$0/49$
$P(\text{it} \mid \textbf{pos})$	$3/49$
…	

positive

probability	value
…	…
$P(\text{great} \mid \textbf{neg})$	$0/40$
$P(\text{works} \mid \textbf{neg})$	$0/40$
$P(\text{not} \mid \textbf{neg})$	$3/40$
$P(\text{it} \mid \textbf{neg})$	$1/40$
…	

negative

MLE with add-one smoothing

probability	value
…	…
$P(\text{great} \mid \textbf{pos})$	$(3+1)/(49+63)$
$P(\text{works} \mid \textbf{pos})$	$(3+1)/(49+63)$
$P(\text{not} \mid \textbf{pos})$	$(0+1)/(49+63)$
$P(\text{it} \mid \textbf{pos})$	$(3+1)/(49+63)$
…	

positive

probability	value
…	…
$P(\text{great} \mid \textbf{neg})$	$(0+1)/(40+63)$
$P(\text{works} \mid \textbf{neg})$	$(0+1)/(40+63)$
$P(\text{not} \mid \textbf{neg})$	$(3+1)/(40+63)$
$P(\text{it} \mid \textbf{neg})$	$(1+1)/(40+63)$
…	

negative

Other smoothing techniques

- **Additive smoothing**: Instead of adding 1, we can add k extra occurrences.
  - k can be any real, non-negative number
  - Includes numbers smaller than one!
- Additive smoothing often works well in text classification.
- There are more **advanced smoothing techniques** that can work better in other contexts.
  - Witten–Bell smoothing, Kneser–Ney smoothing

Important concepts

- Naive Bayes classifier, log probabilities
- maximum likelihood estimation
- class probabilities vs. word probabilities
- additive smoothing, add-one smoothing

LINKÖPING
UNIVERSITY