Generative Models of Text Classification

NTV 2026 / HC2a

Today's lecturer: Sandro Pezzelle

s.pezzelle@uva.nl

Quick updates & announcements

- Week 1 is over!
- Coming soon: feedback on T1
- T2 and E2 are available on Canvas
- Stay on top of self-study activities

Fraud: Check Canvas!

What is considered fraud in NTV:

- Any sort of plagiarism
- Using any GenAI tools
- At the exam: Helping peers/seeking help, consulting unauthorized sources, etc.

Be aware of the (severe) consequences!

Module 1 highlights

What is NLP, why is it important, what are the tasks and applications, and the *engineering* approach (solving tasks) we take here

Module 1 highlights

What is NLP, why is it important, what are the tasks and applications, and the *engineering* approach (solving tasks) we take here

We can use fundamental statistical models/laws to capture statistical properties of corpora (e.g., document length, word frequency, sentiment)

Module 1 highlights

What is NLP, why is it important, what are the tasks and applications, and the *engineering* approach (solving tasks) we take here

We can use fundamental statistical models/laws to capture statistical properties of corpora (e.g., document length, word frequency, sentiment)

Choosing a model family: we match properties of data against properties of available choices (e.g., data type, shape of probability mass function, central tendency, symmetry, skewness, spread, etc.)

Module 1 highlights

What is NLP, why is it important, what are the tasks and applications, and the *engineering* approach (solving tasks) we take here

We can use fundamental statistical models/laws to capture statistical properties of corpora (e.g., document length, word frequency, sentiment)

Choosing a model family: we match properties of data against properties of available choices (e.g., data type, shape of probability mass function, central tendency, symmetry, skewness, spread, etc.)

Fitting parameters: maximum likelihood estimation (e.g., grid search, or closed-form solution when possible)

Module 1 highlights

What is NLP, why is it important, what are the tasks and applications, and the *engineering* approach (solving tasks) we take here

We can use fundamental statistical models/laws to capture statistical properties of corpora (e.g., document length, word frequency, sentiment)

Choosing a model family: we match properties of data against properties of available choices (e.g., data type, shape of probability mass function, central tendency, symmetry, skewness, spread, etc.)

Fitting parameters: maximum likelihood estimation (e.g., grid search, or closed-form solution when possible)

Fitting conditional models suffers from data sparsity.

Outline

This class motivates **text classification problems** and introduces a **generative model of text classification based on Bayesian networks** (the so-called *naive Bayes* model).

Suggested reading:

- Lecture notes by Wilker
- or Appendix B of textbook (2026 version)
  in an earlier version (2025) this was Chapter 4

Background:

- Bayesian networks (module 1 of the PGMs course)

Goals

ILOs. After this class the student can

- design text classifiers based on probabilistic inference (under a *naive Bayes* model)
- estimate parameters of *naive Bayes* via maximum likelihood estimation
- evaluate the performance of text classifiers
- recognise limitations of a *naive Bayes* model in light of the linguistic phenomena of relevance to an NLP problem

Table of contents

Text Classification

Motivation

In many situations, we may want to *categorise* text.

Here are some examples:[1]

labels string · *classes* 20 values	**text** string · *lengths* 2 2.42k
pt	os chefes de defesa da estónia, letónia, lituânia, alemanha, itália, espanha e eslováquia assinarão o acordo para fornecer pessoal e financiamento para o…
bg	размерът на хоризонталната мрежа може да бъде по реда на няколко километра (km) за на симулация до около 100 km за на симулация .
zh	很好，以前从不去评价，不知道浪费了多少积分，现在知道积分可以换钱，就要好好评价了，后来我就把这段话复制走了，既能赚积分，还省事，走到哪复制到哪，最重要的是，不用认真的评论了，不用想还差多少字…
th	สำหรับ ของเก่า ที่ จริงจัง ลอง honeychurch ของเก่า ที่ ไม่ 29 สำหรับ เฟอร์นิเจอร์ และ เงิน ไท ร้อง บริษัท ที่ 122 สำหรับ ลาย คราม
ru	Он увеличил давление .
pl	S Jak sobie życzysz: Widzisz, jak Hitler zabija Żydów?

Language identification

[1] Snippets of datasets from *https://huggingface.co/datasets*

Motivation

In many situations, we may want to *categorise* text.

Here are some examples:[1]

text string · *lengths* 2 — 41.5k	**label** string · *classes* 2 values
hey I am looking for Xray baggage datasets can you provide me with the same	not_spam
"Get rich quick! Make millions in just days with our new and revolutionary system! Don't miss out on this amazing opportunity!"	spam
URGENT MESSAGE: YOU WON'T BELIEVE WHAT WE HAVE TO OFFER!!! Hey you! Yeah, you with the eyes reading this right now. Do you want to be the coolest cat on the…	spam
[Google AI Blog: Contributing Data to Deepfake Detection Research] (https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html…	not_spam

Spam detection

[1] Snippets of datasets from *https://huggingface.co/datasets*

Motivation

In many situations, we may want to *categorise* text.

Here are some examples:[1]

label int64 1 4	title string · *lengths* 6 115	description string · *lengths* 20 985
3	Record labels cut deals with file-sharing companies	The major record labels, which refused initially to deal with the emerging online song file-sharing technologies, have turned over …
4	Microsoft and labels in talks abou t copy protection and Longhorn	Microsoft and a group of recording labels are in discussion about how the next generation of the Windows Operating System, codename…
1	Sudan atrocities need more than la bel	The world was astounded when Secretary of State Colin Powell label ed the humanitarian crisis in Darfur, Sudan, as genocide. While t…
3	FDA Sees Changes to Antidepressant Labels	WASHINGTON (Reuters) - The U.S. Food and Drug Administration plans to update antidepressant labels to reflect studies that suggest a…

Topic classification

[1] Snippets of datasets from *https://huggingface.co/datasets*

Motivation

In many situations, we may want to *categorise* text.

Here are some examples:[1]

text string · *lengths* 1 2.18k	**label** int64 0 2	**sentiment** string · *classes* 3 values
Cooking microwave pizzas, yummy	2	positive
Any plans of allowing sub tasks to show up in the widget?	1	neutral
I love the humor, I just reworded it. Like saying 'group therapy' instead`a 'gang banging'. Keeps m…	2	positive
naw idk what ur talkin about	1	neutral
That sucks to hear. I hate days like that	0	negative
Umm yeah. That`s probably a pretty good note to self because eeeeeewwwwwww.	2	positive

Sentiment classification

[1]Snippets of datasets from *https://huggingface.co/datasets*

Motivation

In many situations, we may want to *categorise* text.

Here are some examples:[1]

premise string · *lengths* 7 — 402	hypothesis string · *lengths* 1 — 295	label class label 3 classes
A person on a horse jumps over a broken down airplane.	A person is training his horse for a competition.	1 neutral
A person on a horse jumps over a broken down airplane.	A person is at a diner, ordering an omelette.	2 contradiction
A person on a horse jumps over a broken down airplane.	A person is outdoors, on a horse.	0 entailment

Natural language inference

[1] Snippets of datasets from *https://huggingface.co/datasets*

A detour on the concept of *inference*

Generally, inference is a process by which a conclusion is reached (*inferred*) from multiple observations through inductive reasoning.

Humans are good at:

- **Logical inference:** deriving new, valid conclusions from known premises or statements based on established logical rules
- **Semantic inference:** deriving new, implicit facts, relationships, or knowledge from existing data (see NLI task above)
- **Pragmatic inference:** deriving implied, non-literal meaning from language by combining semantic content with context, shared knowledge, and social conventions ('reading between the lines')

A detour on the concept of *inference*

Generally, inference is a process by which a conclusion is reached (*inferred*) from multiple observations through inductive reasoning.

In this course, we make inferences (predictions) from texts using **statistics** and **machine learning**—one of the key goals of NLP!

We use models to:

- Infer the language of a document, its topic, whether it is spam, which sentiment it conveys, and so on.

A detour on the concept of *inference*

Generally, inference is a process by which a conclusion is reached (*inferred*) from multiple observations through inductive reasoning.

In this course, we make inferences (predictions) from texts using **statistics** and **machine learning**—one of the key goals of NLP!

This implies:

- Analyzing sample data to draw conclusions, make predictions, or generalize findings about an unknown, larger population
- Using parameter estimation and hypothesis testing to bridge the gap between observed data and population characteristics, accounting for uncertainty and sampling error

Text classification

Text classification is a form of text analysis that may vary from simple pattern recognition to sophisticated language understanding, with the goal of categorising text against a finite set of labels.

Some tasks might require no real understanding of language or only the ability to recognise superficial patterns of language use

- language identification depends largely on script and vocabulary
- URLs, excessive use of block capitals, and canned phrases are common indicators of spams
- the words that are most frequently used tend to correlate strongly with the topics of the documents that contain them

Text classification

Text classification is a form of text analysis that may vary from simple pattern recognition to sophisticated language understanding, with the goal of categorising text against a finite set of labels.

Other tasks clearly require a deeper understanding of language and of what is being said:

- the sentiment expressed in a passage may be faintly related to or downright contrary to the sentiment stereotypically expressed by individual words in the passage;
- natural language inference requires syntactic and semantic understanding of text, a lot of world knowledge, and sophisticated logical reasoning.

Data for text classification

We focus on problems where the set $\mathcal{Y}$ of *possible* labels is finite and relatively small.[2]

Data points take the form of **labelled text**. To annotate a sample text x, a person reads it and categorises it as $y \in \mathcal{Y}$, following task-specific guidelines.

Example (from the textbook, Ch 4):[3]

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

Short movie reviews annotated for (binary) *opinion polarity*.

[2] Tasks with very large or arguably unbounded label spaces are better tackled by a different technology (namely, language models), as we shall see later in the course.

[3] As you know from ML, a separation into training/validation/test is useful for empirical research.

A note on disagreement: Noise?

For some problems, disagreement between annotators is considered a **form of annotation error**, or the data point is assumed to be noisy.

For example, in language identification, either the annotator does not recognise the language (say, for lack of knowledge), or the text is so short or incoherent that it cannot be categorised. This may overlook genuine ambiguity (e.g., text written by a multilingual speaker accustomed to code-switching).

A note on disagreement: Richness?

Other problems welcome disagreement, as it reveals different perspectives in the population at large. For example, sentiment, opinion on social matters, and natural language understanding are largely open to **diversity of views and interpretation**.

There is no right or wrong recipe. As with everything in data analysis, it depends on our needs.

Task(s)

When we think of text classification as *model designers*, there are two related but different *tasks* that we need to tackle.

The NLP task is to map an input text *x* to the class *y* class that 'better captures' its category (note the hidden assumption of a 1-1 mapping).

The statistical task is to give a probabilistic account of the non-deterministic mapping between *x* and all *possible* classes. This involves:

- a (parametric) model of the mapping
- and good fit of the model to observed data.

Bridging the tasks

Having a model able to assign probability $P(Y = y|X = x)$ to each and every class y in the label set $\mathcal{Y}$ given the input text x, a **decision rule** bridges the statistical and the NLP task.

The decision rule is a recipe that tells us to **choose** the output $y^\star$ that optimises a criterion of choice.

The most common decision rule in NLP outputs the **most probable class** (also known as the *mode* of the conditional distribution):[4]

$$y^\star = \underset{y \in \mathcal{Y}}{\text{argmax}} \; P(Y = y|X = x)$$

[4] Others do exist, and some are rather popular in language generation problems. We will get to that later in the course.

A form of conditional categorical model?

Our job as designers is to figure out how to associate *any* given text x with a probability distribution $P(Y|X = x)$ for a random variable Y with outcome space $\mathcal{Y}$ (label set).

A Categorical model over the label set, given the input x?
This seems easy enough! Or is it?

A form of conditional categorical model?

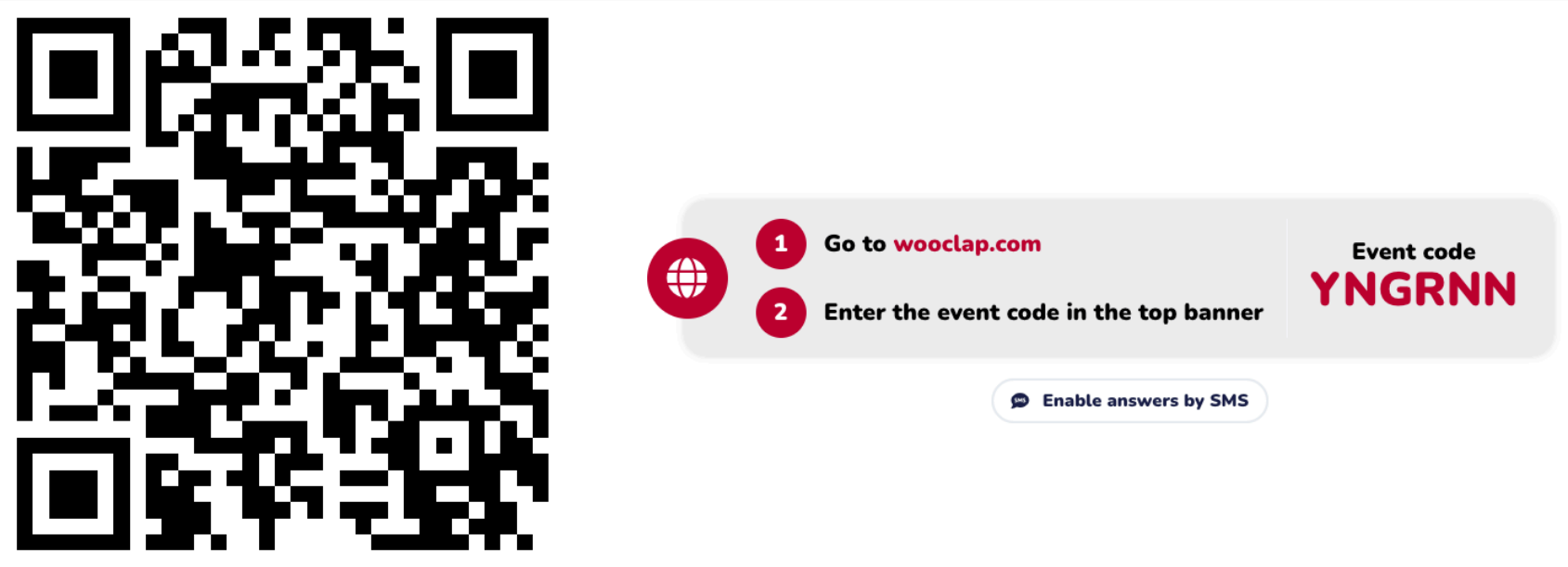
Our job as designers is to figure out how to associate *any* given text x with a probability distribution $P(Y|X = x)$ for a random variable Y with outcome space $\mathcal{Y}$ (label set).

A Categorical model over the label set, given the input x?
This seems easy enough! Or is it?

	Count per outcome of Y	
Condition x	$-$	$+$
just plain boring	1	0
entirely predictable and lacks energy	1	0
no surprises and very few laughs	1	0
very powerful	0	1
the most fun film of the summer	0	1
predictable with no fun	0	0

This *naive* form of modelling will never generalise to new text

Let’s move to Wooclap!



Naive Bayes Classifier

Generative modelling: the NBC

Why *generative* modelling? We learn a complete statistical description of how the data is *generated*, rather than just learning a decision boundary

We ask: If we wanted to generate a *Positive* review, what is the distribution of words we would likely use?

Generative modelling

Rather than attempting to realise the map $x \to P(Y|X = x)$ for all possible texts directly, let's model **the joint distribution of possible texts and labels**.

Then, whenever we are given some text x, we use $P(X = x, Y)$ to **infer** $P(Y|X = x)$ on demand.

Generative modelling

Rather than attempting to realise the map $x \to P(Y|X = x)$ for all possible texts directly, let's model **the joint distribution of possible texts and labels**.

Then, whenever we are given some text x, we use $P(X = x, Y)$ to **infer** $P(Y|X = x)$ on demand.

The model of choice is a **Bayesian network** (BN; see PGMs, module 1):

What's the chain rule factorisation of $P(X, Y)$ under this BN?

Generative modelling

Rather than attempting to realise the map $x \to P(Y|X = x)$ for all possible texts directly, let's model **the joint distribution of possible texts and labels**.

Then, whenever we are given some text x, we use $P(X = x, Y)$ to **infer** $P(Y|X = x)$ on demand.

The model of choice is a **Bayesian network** (BN; see PGMs, module 1):

What's the chain rule factorisation of $P(X, Y)$ under this BN?

$$P(X, Y) = P(Y)P(X|Y)$$

- A tabular representation for $P(Y)$ is simple enough.
- But, a tabular representation for $P(X|Y)$ is as challenging as a tabular representation for $P(Y|X)$, so we are back where we started from.

(Toy) tabular representation for $P(Y)$

Class $y \in \mathcal{Y}$	$P(Y = y)$
positive	0.40
negative	0.30
neutral	0.30

Estimated via Maximum Likelihood Estimation (MLE):[5]

$$P(Y = c) \overset{\text{MLE}}{=} \frac{\textbf{count}_Y(c)}{N}$$

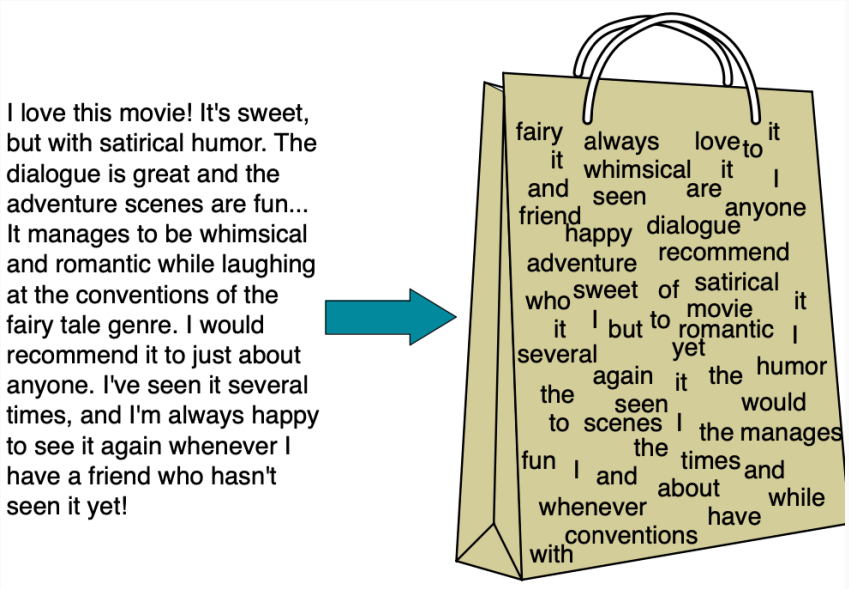
Marginal probabilities: column sums up to 1.0

[5] The function $\textbf{count}_Y(c)$ returns the number of times the rv Y was observed to taken on class c in the training dataset.

Basic text representation

We often think of text as a **sequence of tokens** (words, punctuations, etc.). That is, x is can be thought of as some sequence $\langle w_1, \ldots, w_\ell \rangle$ of length ℓ, where each w_i is a symbol in a finite vocabulary $\mathcal{W}$.

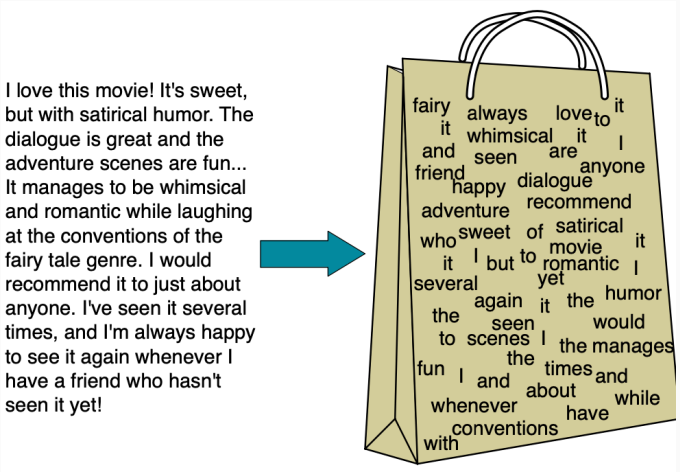
Let's instead reimagine x as a simpler data structure:



A 'bag of words' (Figure B.1 from the textbook).

What just happened?

Bag of words



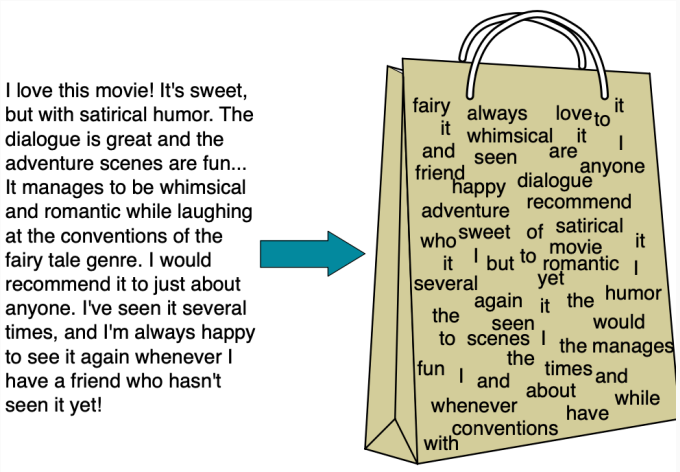
A ‘bag of words’ (Figure B.1 from the textbook).

In a **bag of words** representation, we retain information about *which* tokens x is made of, *how many times* these tokens occur, but we give up on the relative order in which they occur.

This **view** of the text is rather lossy; a lot of linguistic information depends crucially on word order (syntactic function, semantic role, etc.).

If our text classifiers were constrained to ‘seeing’ the text x only through this view, would they be limited in any way?

Bag of words



A 'bag of words' (Figure B.1 from the textbook).

In a **bag of words** representation, we retain information about *which* tokens x is made of, *how many times* these tokens occur, but we give up on the relative order in which they occur.

This **view** of the text is rather lossy; a lot of linguistic information depends crucially on word order (syntactic function, semantic role, etc.).

If our text classifiers were constrained to 'seeing' the text x only through this view, would they be limited in any way? Yes, they would struggle with tasks that depend on more than label/word and label/word-count associations. We embrace this for now!

Word order matters

Consider these two sentences (built with the very same words):

- Harsh reviewers gave us very good feedback.
- Good reviewers gave us very harsh feedback.

Word order matters

Consider these two sentences (built with the very same words):

- Harsh reviewers gave us very good feedback.
- Good reviewers gave us very harsh feedback.

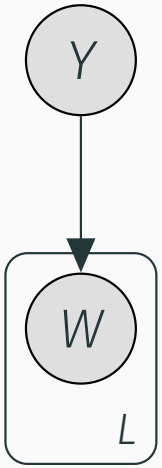
Would you give these sentences the same sentiment?

We'll get back to this later.

Conditional independence

We now revisit our BN model and introduce a key conditional independence assumption: given the text's category Y, we assume the tokens $W_1, \ldots, W_L$ in the text X are drawn independently from the same conditional distribution $P(W|Y)$ over the vocabulary of known tokens $\mathcal{W}$.

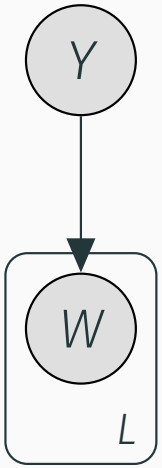
What's the chain rule factorisation of $P(X, Y)$ under this BN?



Conditional independence

We now revisit our BN model and introduce a key conditional independence assumption: given the text's category Y, we assume the tokens $W_1, \ldots, W_L$ in the text X are drawn independently from the same conditional distribution $P(W|Y)$ over the vocabulary of known tokens $\mathcal{W}$.

What's the chain rule factorisation of $P(X, Y)$ under this BN?



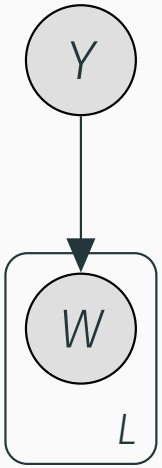
$$P(X, Y) = P(Y) \prod_{i=1}^{L} P(W_i|Y)$$

- A tabular representation for $P(Y)$ is simple enough.
- Is it feasible to work with a tabular representation for $P(W|Y)$?

Conditional independence

We now revisit our BN model and introduce a key conditional independence assumption: given the text's category Y, we assume the tokens $W_1, \ldots, W_L$ in the text X are drawn independently from the same conditional distribution $P(W|Y)$ over the vocabulary of known tokens $\mathcal{W}$.

What's the chain rule factorisation of $P(X, Y)$ under this BN?



$$P(X, Y) = P(Y) \prod_{i=1}^{L} P(W_i|Y)$$

- A tabular representation for $P(Y)$ is simple enough.
- Is it feasible to work with a tabular representation for $P(W|Y)$? We have $|\mathcal{Y}|$ conditions (say K) and $|\mathcal{W}|$ outcomes (say V), that is, a table of size $K \times V$. That seems alright for computation and for estimation (via MLE).

(Toy) tabular representation for $P(W|Y)$

c	P(good|c)	P(movie|c)	P(director|c)	P(with|c)	P(a|c)
Pos	0.30	0.10	0.10	0.05	0.05
Neg	0.05	0.10	0.10	0.05	0.05
Neu	0.10	0.15	0.15	0.10	0.05

Estimated via MLE (often modified by Laplace / add-α smoothing):[6]

$$P(W = w|Y = c) \overset{\text{MLE}}{=} \frac{\textbf{count}_{YW}(c, w) + \alpha}{\sum_{t=1}^{V}(\textbf{count}_{YW}(c, t) + \alpha)}$$

Conditional probabilities per class: **rows** sum up to 1.0 (not columns)

[6] The function $\textbf{count}_{YW}(c, w)$ returns the number of times the random variables (Y, W) were jointly observed to take class c and token w, resp., in the training dataset.

Note on visualizing conditional probs in tabular format

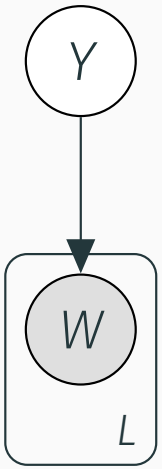
Representing conditional probabilities in this tabular format (classes as *rows* instead of *columns*) is a convention, e.g., the one followed in Koller's textbook (seen in PGM)

Flipping the table only changes how you read it, not what the numbers mean: each entry is still $P(w|c)$, and the sum of probabilities is 1.0 class-wise (not word-wise)!

Inferring posterior distributions

Suppose we indeed can represent and estimate $P(Y)$ and $P(W|Y)$.

Now we observe a test sample x, but we do not observe a class for it.
How can we assign probability to a possible class $y \in \mathcal{Y}$ given x?

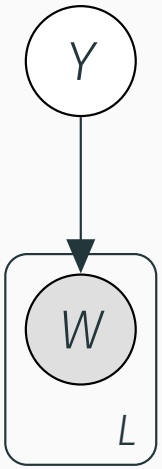


Inferring posterior distributions

Suppose we indeed can represent and estimate $P(Y)$ and $P(W|Y)$.

Now we observe a test sample x, but we do not observe a class for it. How can we assign probability to a possible class $y \in \mathcal{Y}$ given x?

Posterior inference (via Bayes rule):



$$P(Y = y|X = x) = \frac{P(X = x, Y = y)}{P(X = x)}$$
$$= \frac{P(Y = y)P(X = x|Y = y)}{P(X = x)}$$
$$\overset{\text{ind.}}{=} \frac{P(Y = y)\prod_{i=1}^{L} P(W = w_i|Y = y)}{P(X = x)}$$

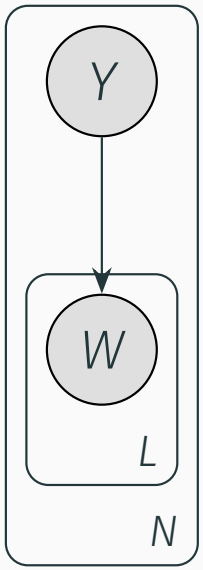
with the denominator (evidence) obtained via marginalisation:

$$P(X = x) = \sum_{c \in \mathcal{Y}} P(X = x, Y = c) \overset{\text{ind.}}{=} \sum_{c \in \mathcal{Y}} P(Y = c) \prod_{i=1}^{L} P(W = w_i|Y = c)$$

Estimating the model

The model has 2 tabular CPDs (see toy examples above), which we estimate via MLE (possibly with Laplace smoothing to avoid 0s).

The sufficient statistics (counts) necessary for MLE are collected using the training data:



Class probabilities:

$$P(Y = c) = \frac{\text{count}_Y(c)}{N}$$

Class-conditioned token probabilities (standard implementation form):

$$P(W = w|Y = c) = \frac{\text{count}_{YW}(c, w) + \alpha}{\alpha V + \text{count}_Y(c)}$$

We grid-search for a good value of α using performance (likelihood or a notion of classification accuracy) on a validation set as optimisation criterion (see T2).

A toy example

1. “A good movie with a good director”
    - $P(Pos) \times P(a|Pos) \times P(good|Pos) \times P(movie|Pos) \ldots$

A toy example

1. “A good movie with a good director”
   - $P(Pos) \times P(a|Pos) \times P(good|Pos) \times P(movie|Pos) \ldots$
   - $P(S_1|Pos) \approx 4.5 \times 10^{-7}$

A toy example

1. “A good movie with a good director”
   - $P(Pos) \times P(a|Pos) \times P(good|Pos) \times P(movie|Pos) \ldots$
   - $P(S_1|Pos) \approx 4.5 \times 10^{-7}$
   - $P(S_1|Neg) \approx 2 \times 10^{-8}$

A toy example

1. “A good movie with a good director”
   - $P(Pos) \times P(a|Pos) \times P(good|Pos) \times P(movie|Pos) \ldots$
   - $P(S_1|Pos) \approx 4.5 \times 10^{-7}$
   - $P(S_1|Neg) \approx 2 \times 10^{-8}$
   - $P(S_1|Neu) \approx 5 \times 10^{-8}$

A toy example

1. “A good movie with a good director”
   - $P(Pos) \times P(a|Pos) \times P(good|Pos) \times P(movie|Pos) \ldots$
   - $P(S_1|Pos) \approx 4.5 \times 10^{-7}$
   - $P(S_1|Neg) \approx 2 \times 10^{-8}$
   - $P(S_1|Neu) \approx 5 \times 10^{-8}$
   - After normalization (calculating posterior probabilities):

A toy example

1. “A good movie with a good director”
   - $P(Pos) \times P(a|Pos) \times P(good|Pos) \times P(movie|Pos) \ldots$
   - $P(S_1|Pos) \approx 4.5 \times 10^{-7}$
   - $P(S_1|Neg) \approx 2 \times 10^{-8}$
   - $P(S_1|Neu) \approx 5 \times 10^{-8}$
   - After normalization (calculating posterior probabilities):
   - $P(Pos|S_1) \approx \mathbf{0.86}$; $P(Neg|S_1) \approx 0.04$; $P(Neu|S_1) \approx 0.10$

A toy example

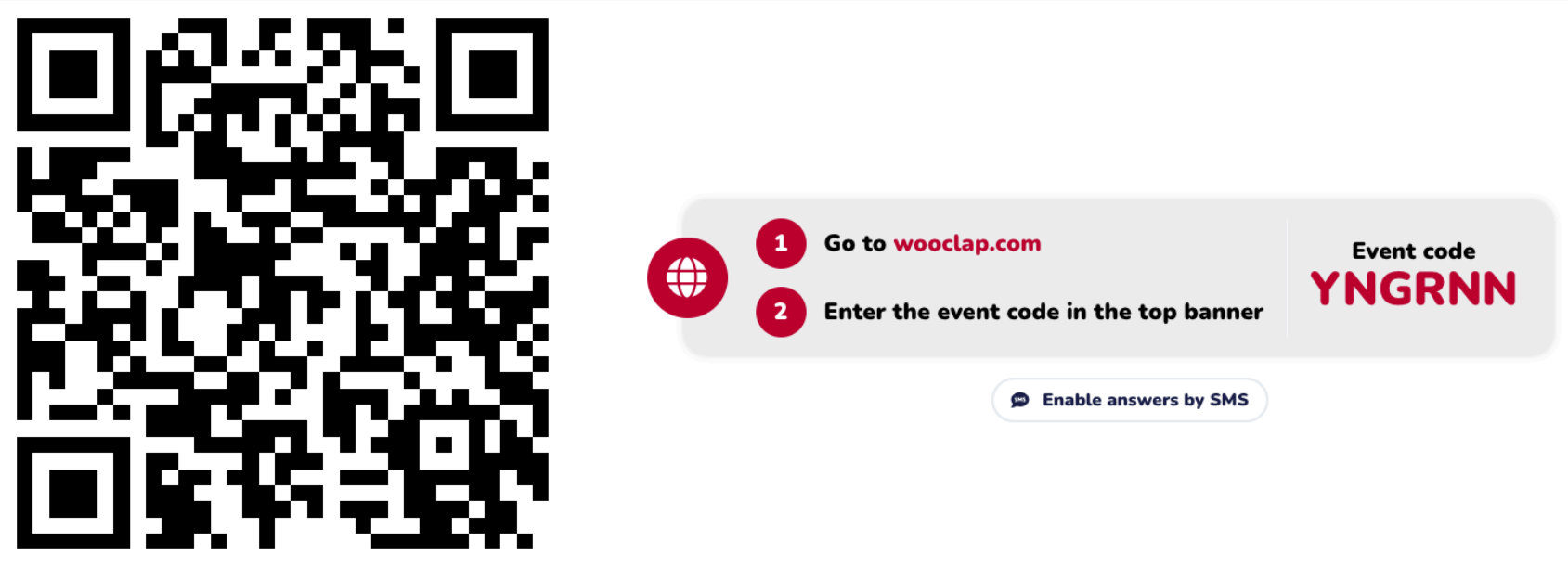
1. “A good movie with a good director”
   - $P(Pos) \times P(a|Pos) \times P(good|Pos) \times P(movie|Pos) \ldots$
   - $P(S_1|Pos) \approx 4.5 \times 10^{-7}$
   - $P(S_1|Neg) \approx 2 \times 10^{-8}$
   - $P(S_1|Neu) \approx 5 \times 10^{-8}$
   - After normalization (calculating posterior probabilities):
   - $P(Pos|S_1) \approx \mathbf{0.86}$; $P(Neg|S_1) \approx 0.04$; $P(Neu|S_1) \approx 0.10$
2. “A movie with a good director”
   - $P(Pos) \times P(a|Pos) \times P(movie|Pos) \times P(with|Pos) \times P(a|Pos) \ldots$

A toy example

1. "A good movie with a good director"
   - $P(Pos) \times P(a|Pos) \times P(good|Pos) \times P(movie|Pos) \ldots$
   - $P(S_1|Pos) \approx 4.5 \times 10^{-7}$
   - $P(S_1|Neg) \approx 2 \times 10^{-8}$
   - $P(S_1|Neu) \approx 5 \times 10^{-8}$
   - After normalization (calculating posterior probabilities):
   - $P(Pos|S_1) \approx \mathbf{0.86}$; $P(Neg|S_1) \approx 0.04$; $P(Neu|S_1) \approx 0.10$
2. "A movie with a good director"
   - $P(Pos) \times P(a|Pos) \times P(movie|Pos) \times P(with|Pos) \times P(a|Pos) \ldots$
   - After normalization (calculating posterior probabilities):
   - $P(Pos|S_1) \approx \mathbf{0.625}$; $P(Neg|S_1) \approx 0.0625$; $P(Neu|S_1) \approx 0.3125$

Word count matters!

Let’s move to Wooclap!



Classifying

Equipped with posterior inference, we can return to the NLP task and make *decisions*

Classifying

Equipped with posterior inference, we can return to the NLP task and make *decisions*

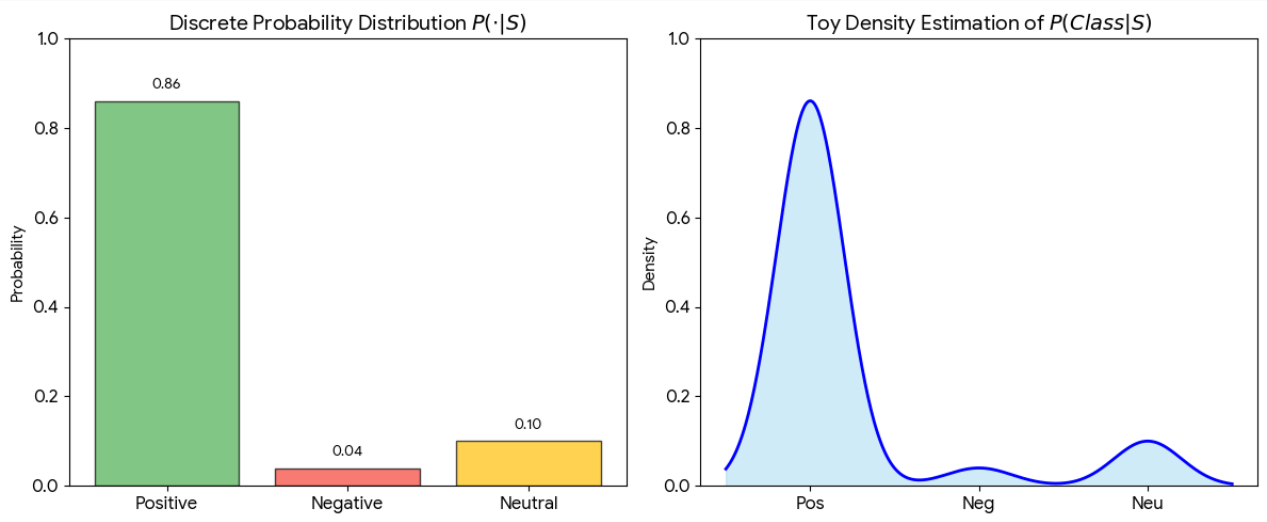
We use **argmax** to select the argument (in this case the class c) that maximizes a function (in this case the probability $P(c|d)$)

Classifying

Equipped with posterior inference, we can return to the NLP task and make *decisions*

We use **argmax** to select the argument (in this case the class c) that maximizes a function (in this case the probability $P(c|d)$)

A movie with a good director

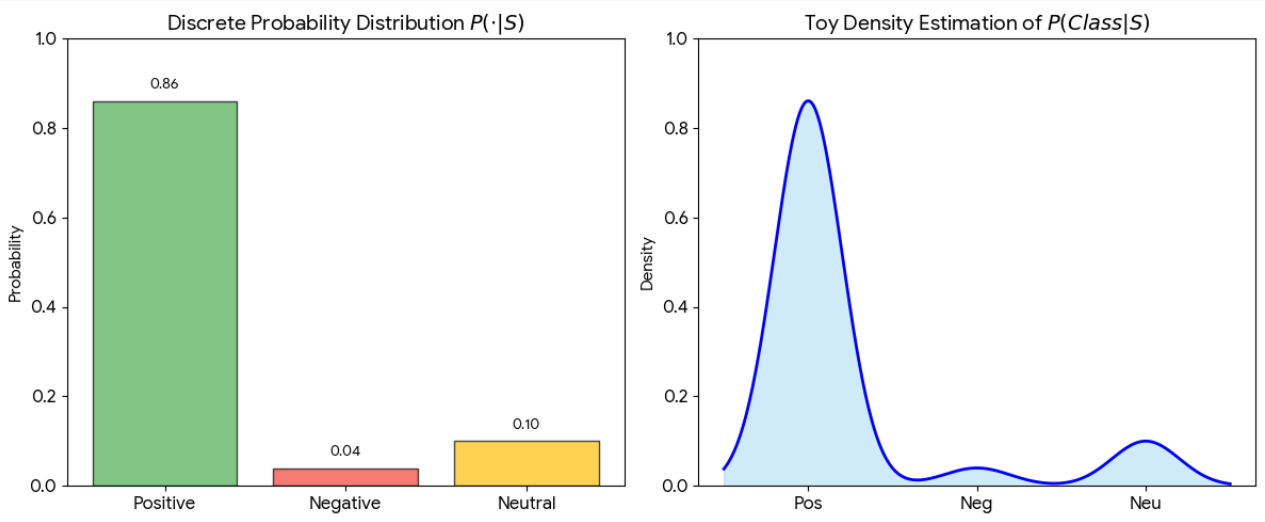


Classifying

Equipped with posterior inference, we can return to the NLP task and make *decisions*

We use **argmax** to select the argument (in this case the class c) that maximizes a function (in this case the probability $P(c|d)$)

A movie with a good director



$$\hat{c} = \arg\max \{P(Pos|S), P(Neg|S), P(Neu|S)\}$$
$$\hat{c} = \arg\max(0.86, 0.04, 0.10) = Pos$$

Evaluating a model

We use Precision, Recall, and F-measure/F1 (see in TTTV)

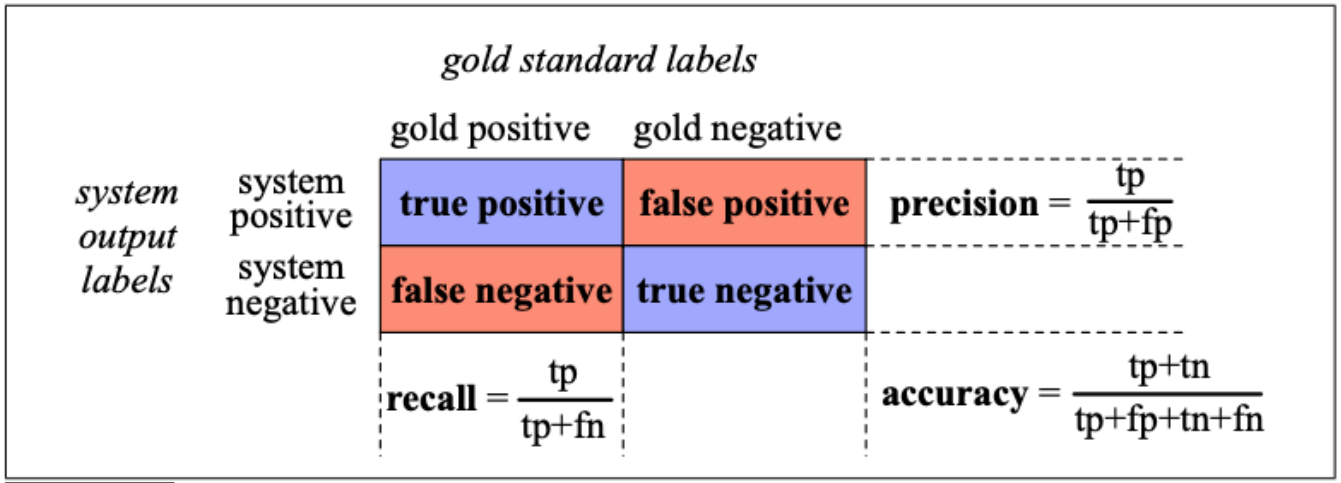


Figure B.4 A confusion matrix for visualizing how well a binary classification system performs against gold standard labels.

See Lecture notes by Wilker and Appendix B of textbook (2026 version) for further details

Count vectors

There's another way to represent the same model, which is a little more computer-science-friendly:

The count vector view of a bag of words (Fig B.1 from textbook).

Think of the bag of words representation of a text x as a V-dimensional vector $\phi(x)$, where the tth coordinate $\phi_t(x)$ is the number of times the token whose id is t occurs in x.

NB and count vectors

We can now rewrite the factorisation as

$$P(X, Y) \overset{\text{ind.}}{=} P(Y) \prod_{t=1}^{V} P(W = t|Y)^{\phi_t(x)}$$

That is,

- the product ranges over the vocabulary (not over the sequence)
- the exponent captures the number of occurrences in x of each known token
- as $\boldsymbol{\phi}(x)$ is rather sparse (many 0s), many terms in the product evaluate to 1.

The standard view is easier to express on paper (e.g., in an exercise), the vectorised view is sometimes more convenient in a computer programme (e.g., if you are using numpy or scipy data structures) and it will be helpful in the move towards linear, generalised linear and non-linear models.

Limitations

- The Bag of Words assumption: It cannot distinguish between sentences with identical words but different meanings
- Loss of context and negation: Negations completely flip the sentiment of a sentence, but a unigram model often misses this because it sees *good* and *not* as two independent features
- The strong independence assumption: In real language, words are highly dependent
- Zero-frequency problem (novel words): Even with smoothing, the model doesn't *understand* the new word; it just gives it a tiny default value
- Sensitivity to document length: Because it multiplies many small probabilities together, longer documents result in much smaller raw products than shorter ones

What’s next?

What's next?

- Using bigrams, trigrams, etc.
- Generalised Linear Models (GLMs): next HC