Computational Political Science

Session 4

David Broska
Zeppelin University
February 23, 2021

Outline for today

1. **Supervised machine learning**
    - Why do we need a labeled set?
2. **Probability fundamentals review**
    - Conditional probability
    - Rule of product
3. **Bayes theorem**
    - From conditional probabilities to Bayes theorem
    - Example: How good is our spam classifier?
4. **Naive Bayes**
    - Dissecting the Naive Bayes
    - Implementation in quanteda
5. **Prior probabilities**
6. **Coding Exercise**

Supervised machine learning

Supervised machine learning

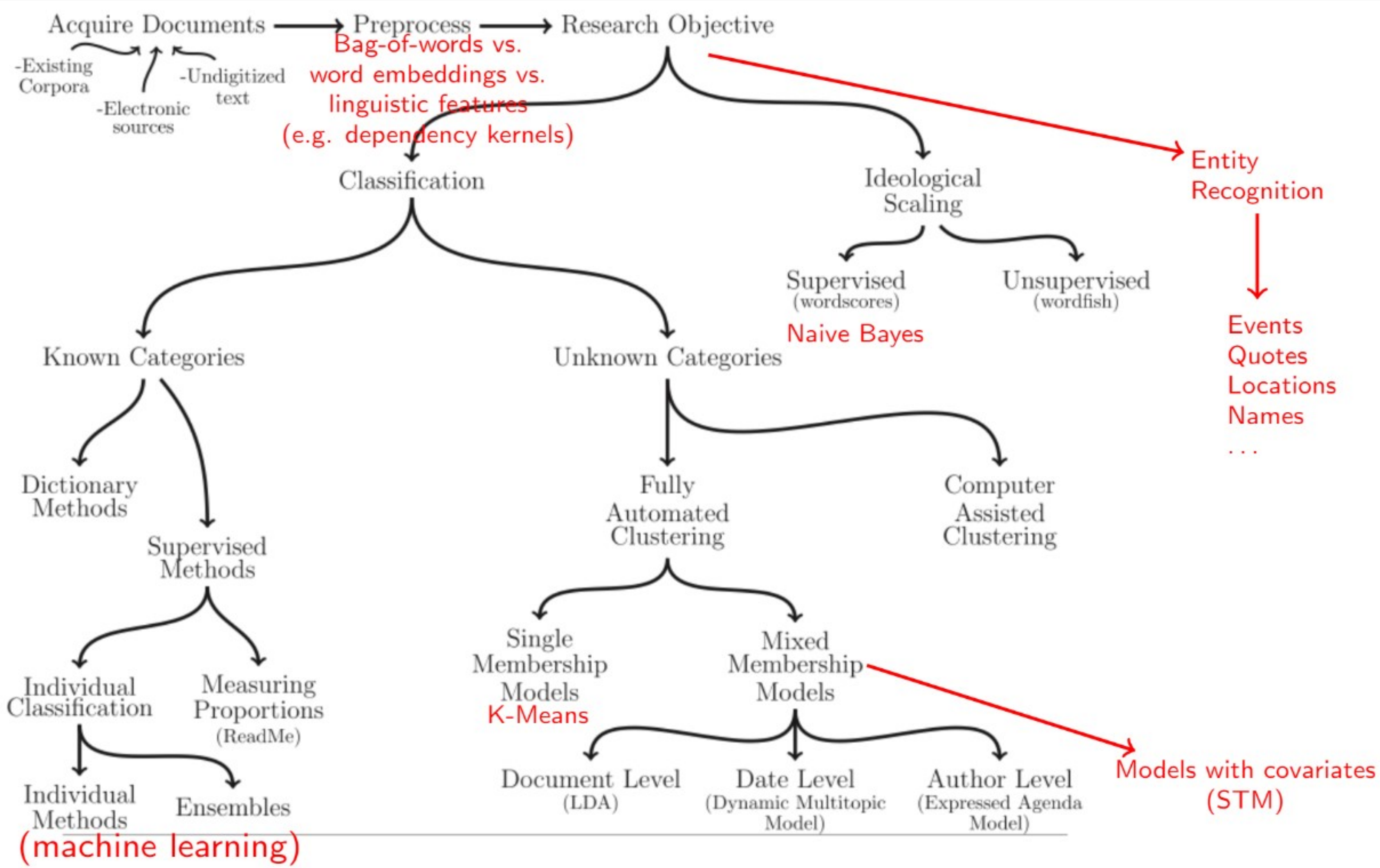


Fig. 1 in Grimmer and Stuart (2013)

Supervised machine learning

The goal is classify documents into pre existing categories.

For example, authors of documents, sentiment of tweets, ideological position of parties based on manifestos, tone of movie reviews...

What we need

- Hand-coded dataset (labeled), to be split into:
    - *Training set*: used to train the classifier
    - *Validation/Test set*: used to validate the classifier
- Method to *extrapolate* from hand coding to unlabeled documents (classifier):
    - Naive Bayes, regularized regression, SVM, CNN, ensemble methods, etc.
- *Performance metric* to choose best classifier and avoid overfitting:
    - Confusion matrix, accuracy, precision, recall...
- Approach to *validate* classifier: cross-validation

Creating a labeled set

How do we obtain a labeled set?

- External sources of annotation
  - Disputed authorship of Federalist papers estimated based on known authors of other documents
- Expert annotation
  - "Canonical" dataset in Comparative Manifesto Project
  - In most projects, undergraduate students (expertise comes from training)
- Crowd-sourced coding
  - Wisdom of crowds: aggregated judgments of non-experts converge to judgments of experts at much lower cost (Benoit et al 2016)
  - Easy to implement with CrowdFlower or MTurk

Labeling example

Code the Content of a Sample of Tweets

Instructions ▴

In this job, you will be presented with tweets about the recent protests related to race and law enforcement in the U.S.

You will have to read the tweet and answer a set of questions about its content.

Read the tweet below paying close attention to detail:

Tweet ID: **447**

El Cid
@JohnGalt2112

Follow

#BlackLivesMatter don't matter unless they are taken by a white cop.

4:23 PM - 13 Dec 2014

Is this tweet related to the ongoing debate about law enforcement and race in the United States?

- ◯ Yes
- ◯ No
- ◯ Don't Know

Principles of supervised learning

Generalization

A classifier or a regression algorithm learns to correctly predict output from given inputs

Crucially, it predicts correctly not only in previously seen samples but also in previously unseen samples.

Overfitting

A classifier or a regression algorithm learns to correctly predict output from given inputs in previously seen samples.

However, it fails to do so in previously unseen samples. This causes poor prediction/generalization.

The goal is to maximize the frontier of precise identification of true condition with accurate recall

Probability fundamentals review

Motivation

Why should we look more closely into conditional probabilities and Bayes theorem?

After all, we could just learn the broad idea of the Naive Bayes classifier and run `textmodel_nb()` in R without knowing the details …

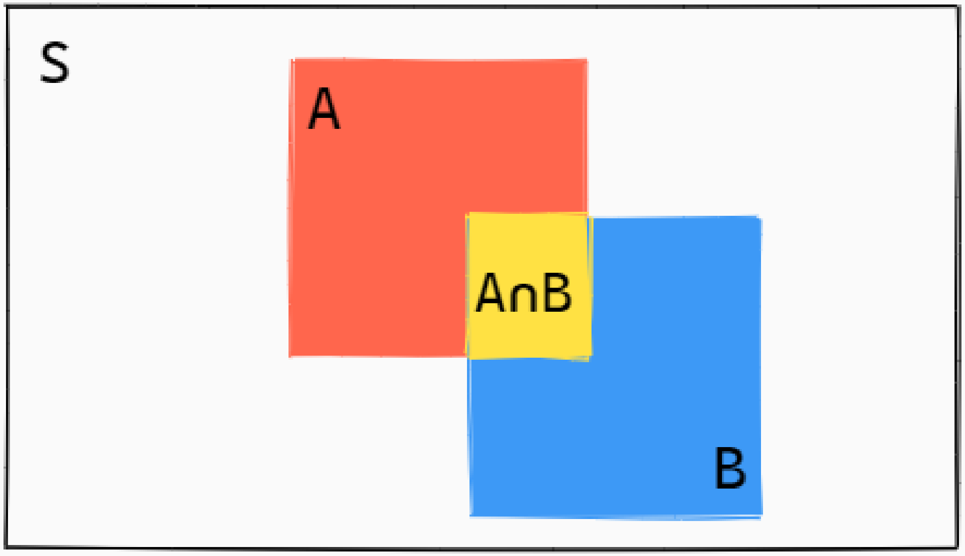
- We introduce classification by the example of the Naive Bayes which requires a better understanding of the model's internal mechanisms
- Although universities mostly teach probability theory for independent events, the most interesting social phenomena are conditional (and not independent of) other factors
- We lay the foundation for interesting and widely used data science applications of Bayesian thinking

Conditional probability

Definition

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \text{ when } P(B) > 0$$

where A and B are events in a sample space S.



Intuitively, what this formula does is *restricting the sample space* to events where B occurs, and counting those where both A and B occur.

Example

What is the probability of event A given B?

Let A be the event that a the roll of a dice results in an *odd* number: $A = \{1, 3, 5\}$

Let B be the event that the outcome is *smaller or equal to three*: $B = \{1, 2, 3\}$

$$P(A|B) = \frac{|A \cap B|}{|B|} = \frac{|\{1,3\}|}{|\{1,2,3\}|} = \frac{2}{3}$$

More generally, we can rewrite this in terms of probabilities

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{\frac{|A \cap B|}{|S|}}{\frac{|B|}{|S|}} = \frac{\frac{2}{6}}{\frac{3}{6}} = \frac{2}{3}$$

where $S = \{1, 2, 3, 4, 5, 6\}$ is the sample space.

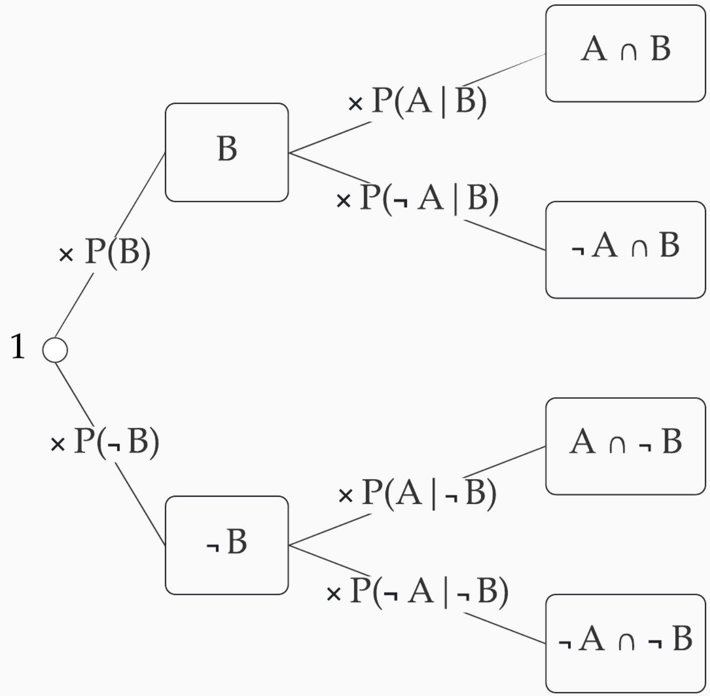
Rule of product

The rule of product allows finding the probability of two (or more) events occurring together

Rule of product for dependent events

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

Rewriting the conditional probability allows finding $P(A \cap B)$ using a tree diagram:



Rule of product

The rule of product allows finding the probability of two (or more) events occurring together

Rule of product for independent events

Two events A and B are independent only if

$$P(A \cap B) = P(A)P(B)$$

Independence means that conditional probability of one event given another is the same as the original (prior) probability:

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

Rule of product

Rule of product for independent events (continued)

Sometimes the independence of two events is clear because they do not have a physical interaction with each other

For example, the roll of a dice today and the weather tomorrow

At other times, independence is not obvious so we need to check if they satisfy the independence condition

In the die roll example, we had
$P(A) = \frac{|\{1,3,5\}|}{|S|} = \frac{3}{6} = \frac{1}{2}$ and
$P(B) = \frac{|\{1,2,3\}|}{|S|} = \frac{3}{6} = \frac{1}{2}$ and
$P(A \cap B) = \frac{1}{3}$

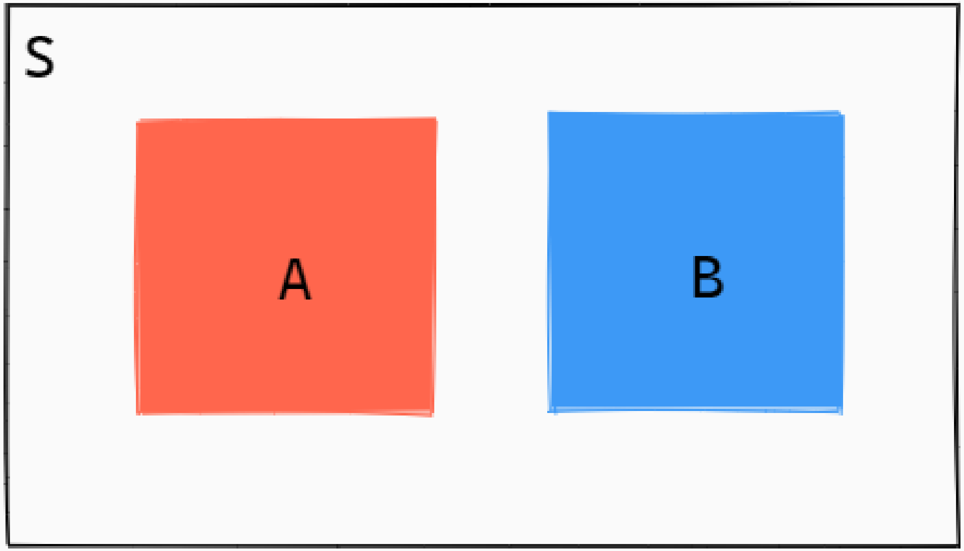
A and B are not independent since

$$P(A)P(B) = \left(\frac{1}{2}\right)^2 = \frac{1}{4} \qquad \neq \qquad P(A \cap B) = \frac{1}{3}$$

Special cases (I)

If A and B are disjoint they cannot occur together, so $A \cap B = \varnothing$.

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(\varnothing)}{P(B)} = 0$$

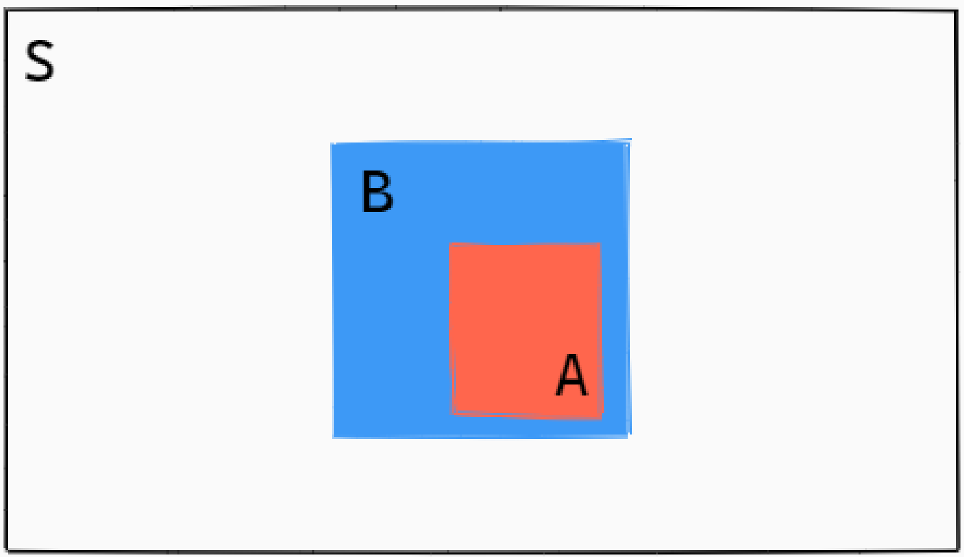


Special cases (II)

If A is a subset of B, then whenever A happens B happens as well.

In this case, $A \cap B = A$.

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

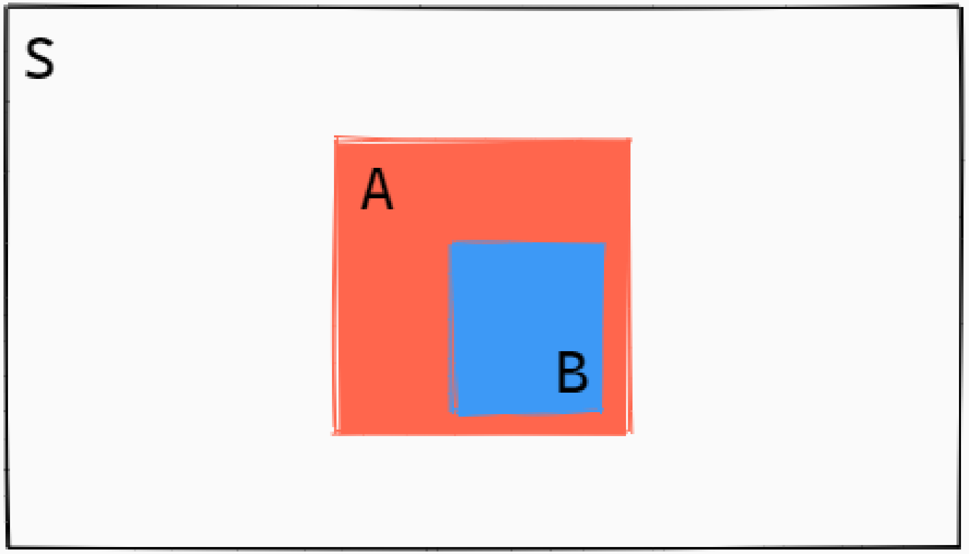


Special cases (III)

If B is a subset of A, then whenever B happens A also happens.

In this case, $A \cap B = B$.

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B)}{P(B)} = 1$$



Bayes theorem

Bayes theorem

Rearranging the definition of conditional probability gives the probability of the intersection:

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

An alternative way of expressing the probability of the intersection is based on $P(B|A)$.

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

Setting both equations equal to another and rearranging gives the **Bayes theorem**:

$$P(A)P(B|A) = P(B)P(A|B)$$

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

Applying Bayes theorem

The rate of spam mails for a certain email address is 2%. A spam filter identifies a spam mail with a probability of 95%. At the same time, 10% of non-spam messages are classified as spam.

(a) What is the probability that a mail that was marked as spam is truly a spam mail?

(b) What is the probability that a mail that was not identified as spam is spam?

Naive Bayes

Spam classification

```
# 4 texts with known and 3 texts with unknown category
txt <- c(k1 = "$ Win $",
         k2 = "$ Prize $",
         k3 = "Earn $ Easily",
         k4 = "Paypal 100 $",
         u1 = "$",
         u2 = "$ $",
         u3 = "Paypal 100 $ $")
x <- dfm(txt)
y <- factor(c("Spam","Spam","Spam","No Spam", NA, NA, NA), ordered = T)
```

training dfm with known categories

	$	win	prize	earn	easily	paypal	100
k1	2	1	0	0	0	0	0
k2	2	0	1	0	0	0	0
k3	1	0	0	1	1	0	0
k4	1	0	0	0	0	1	1

training vector with known categories

y
Spam
Spam
Spam
No Spam

P(word | category)

```r
dfm(txt, groups = y)                # compute word frequency given category
```

```
##          features
## docs      $ win prize earn easily paypal 100
##   No Spam 1   0     0    0      0      1   1
##   Spam    5   1     1    1      1      0   0
```

```r
dfm(txt, groups = y) + 1            # but we need to apply Laplacian smoothing
```

```
##          features
## docs      $ win prize earn easily paypal 100
##   No Spam 2   1     1    1      1      2   2
##   Spam    6   2     2    2      2      1   1
```

```r
# get probability of word given category
# divide feature frequency by category frequency (10 and 16 respectively)
(dfm(txt, groups = y) + 1) / rowSums(dfm(txt, groups = y) + 1)
```

```
##          features
## docs          $   win prize  earn easily paypal    100
##   No Spam 0.200 0.100 0.100 0.100  0.100 0.2000 0.2000
##   Spam    0.375 0.125 0.125 0.125  0.125 0.0625 0.0625
```

nb <- textmodel_nb(x,y,prior="docfreq")

```r
( Pc <- nb$priors )                                   # extract prior for class
```

```
## No Spam    Spam
##    0.25    0.75
```

```r
( PwGc <- nb$param )                                  # get p of word given category
```

```
##              $   win prize  earn easily paypal    100
## No Spam 0.200 0.100 0.100 0.100  0.100 0.2000 0.2000
## Spam    0.375 0.125 0.125 0.125  0.125 0.0625 0.0625
```

```r
# get p of class given word(s)
( PcGw <- predict(nb, type="prob") )
```

```
##    No Spam Spam
## k1    0.07 0.93
## k2    0.07 0.93
## k3    0.10 0.90
## k4    0.65 0.35
## u1    0.15 0.85
## u2    0.09 0.91
## u3    0.49 0.51
```

```r
# categorize based on highest p
( cats <- predict(nb, type="class") )
```

```
## Spam
## Spam
## Spam
## No Spam
## Spam
## Spam
## Spam
```

Manually computing P(spam | words)

Recall that we have the prior P(spam)=0.75 and estimated these conditional probabilities:

	$	**win**	**prize**	**earn**	**easily**	**paypal**	**100**
No Spam	0.200	0.100	0.100	0.100	0.100	0.2000	0.2000
Spam	0.375	0.125	0.125	0.125	0.125	0.0625	0.0625

Notice how our belief of the message being spam changes as we consider various features

P(spam | $)

```
(.75*.375) / (.75*.375 + .25*.2)
```

```
## [1] 0.85
```

P(no spam | $)

```
(.25*.2) / (.25*.2 + .75*.375)
```

```
## [1] 0.15
```

P(spam | $ ∩ $)

```
(.75 * .375^2) /
 (.75 * .375^2 + .25 * .2^2)
```

```
## [1] 0.91
```

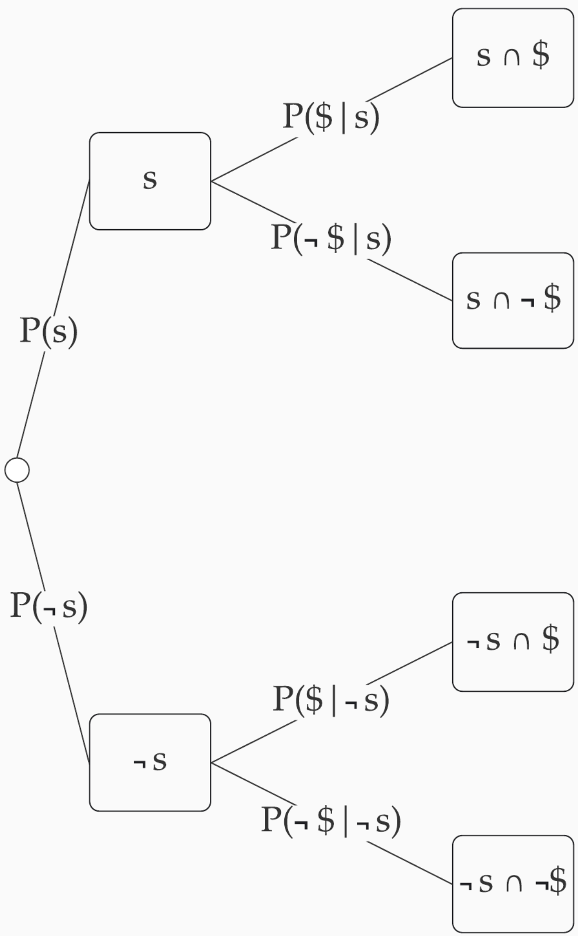
P(spam | $ ∩ $ 100 ∩ Paypal)

```
(.75 * .375^2 * .0625^2) /
 (.75 * .375^2 *.0625^2 + .25 * .2^4)
```

```
## [1] 0.51
```

P(spam | $)

What is the probability of spam given a dollar sign in the message?



$$
\begin{aligned}
P(\mathrm{s}|\$) =& \frac{P(\mathrm{s})P(\$|\mathrm{s})}{P(\mathrm{s})P(\$|\mathrm{s}) + P(\neg\,\mathrm{s})P(\$|\neg\,\mathrm{s})} \\
=& \frac{0.75 \times 0.375}{0.75 \times 0.375 + 0.25 \times 0.2} \\
=& 0.85
\end{aligned}
$$

Numerator

Look at the branches where the evidence ($) occurs *and* the hypothesis (spam) is true; compute the joint probability P(s ∩ $)

Denominator

Consider all possibilities where the evidence occurs: numerator *added* to the joint probability of the evidence occurring given the hypothesis is not true

Considering multiple features

The Naive Bayes model (wrongly) assumes **conditional independence of word counts given the class**.

- This is why the model is called "naive": it assumes that seeing a word does not change the probability of observing other words in a document.
- However, the word "weather" is more likely to be followed by related words like "forecast" and "report" rather than unrelated words such as "guitar".

This assumption has practical advantages as we can *multiply* the conditional probabilities of a word given a class $P(w_j|c_k)$ to get the *joint* probability of the words occurring in a document.

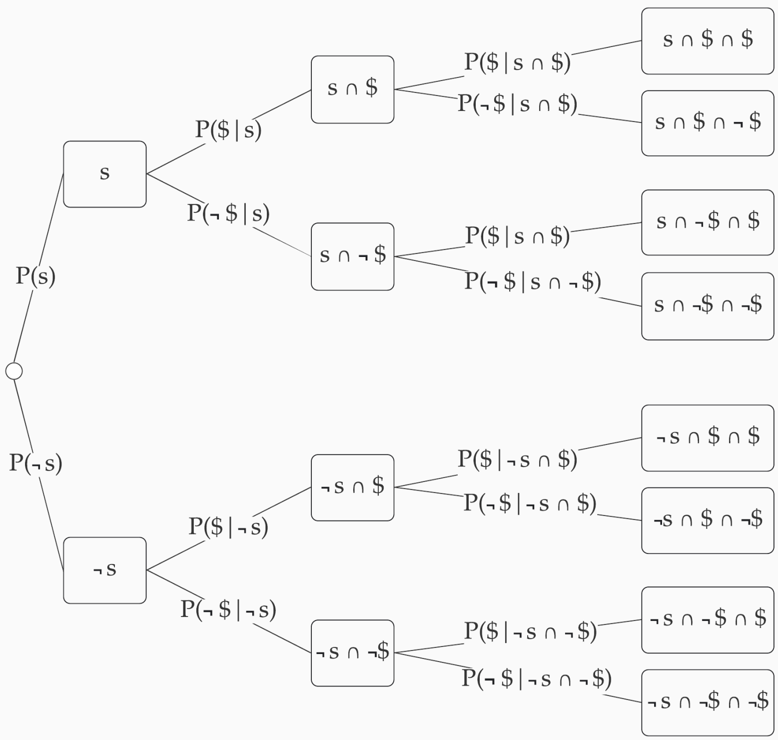
Then, the probability that a document belongs to a class can be calculated with

$$P(c_k|d) = P(c_k|w_1 \cap w_2 \cap \dots w_J) = P(c_k) \prod_j^J \frac{P(w_j|c_k)}{P(w_j)}$$

where $\prod_j^J$ means to multiply the J word probabilities.

P(spam | $ ∩ $)

What is the probability of spam given **two** dollar signs in the message?



Assuming conditional independence of features given the class allows us to easily calculate the probability of spam given **multiple** features.

Independence simplifies the calculation, for example P($ | s ∩ $) = P($| s)

Substantively, we assume that observing a $ does not affect the probability of observing further $ (or any other feature)

$$\begin{aligned} &P(\mathrm{s}|\$ \cap \$) \\ =&\frac{P(\mathrm{s})P(\$|\mathrm{s})^2}{P(\mathrm{s})P(\$|\mathrm{s})^2 + P(\neg\,\mathrm{s})P(\$|\neg\,\mathrm{s})^2} \\ =&\frac{0.75 \times 0.375^2}{0.75 \times 0.375^2 + 0.25 \times 0.2^2} \\ =&0.91 \end{aligned}$$

P(spam | $ ∩ $ ∩ Paypal ∩ 100)

What is the probability of spam given the message includes $, $, Paypal, and 100?

Recall that we have the prior of P(spam)=0.75 and estimated the following conditional probabilities:

	$	win	prize	earn	easily	paypal	100
No Spam	0.200	0.100	0.100	0.100	0.100	0.2000	0.2000
Spam	0.375	0.125	0.125	0.125	0.125	0.0625	0.0625

$$
\begin{aligned}
&P(\mathrm{s}|\$ \cap \$ \cap \mathrm{Paypal} \cap 100) \\
=&\frac{P(\mathrm{s})P(\$|\mathrm{s})^2 P(\mathrm{Paypal}|s)P(100|s)}{P(\mathrm{s})P(\$|\mathrm{s})^2 P(\mathrm{Paypal}|s)P(100|s) + P(\neg\,\mathrm{s})P(\$|\neg\,\mathrm{s})^2 P(\mathrm{Paypal}|\neg s)P(100|\neg s)} \\
=&\frac{0.75 \times 0.375^2 \times 0.0625^2}{0.75 \times 0.375^2 \times 0.0625^2 + 0.25 \times 0.2^4} \\
=&0.51
\end{aligned}
$$

Prior probabilities

Document frequency

A prior probability is the baseline expectation of observing a category without considering any evidence.

`textmodel_nb()` allows specifying different priors for training a classifier.

Relative document frequency

Throughout this lecture we assumed that the relative frequency of a category occurring in the corpus is a reasonable baseline expectation of receiving a spam message

y
Spam
Spam
Spam
No Spam

$$P(\text{ s }) = \frac{3}{4}$$

$$P(\neg s) = \frac{1}{4}$$

However, there may be nothing informative in the relative numbers of documents.

Term frequency

Using term frequency makes the priors equal to the fraction of total feature counts found in the grouped documents in each training class

Therefore, classes with the largest number of features are assigned the largest priors

Coincidentally, in our example the fraction of total features is the same as the relative document frequency

```
dfm(txt, groups = y)
```

```
##          features
## docs      $ win prize earn easily paypal 100
##   No Spam 1   0     0    0      0      1   1
##   Spam    5   1     1    1      1      0   0
```

```
rs <-rowSums(dfm(txt,groups=y))
```

```
##
##
## 3
## 9
```

```
rs / sum(rs)
```

```
## No Spam    Spam
##    0.25    0.75
```

Uniform priors

Using uniform priors means to set the unconditional probability of observing one class to be the same as observing any other class

To illustrate, let's recalculate the probability of P(spam | $) with uniform priors

$$P(\mathrm{s}|\$) = \frac{P(\mathrm{s})P(\$|\mathrm{s})}{P(\mathrm{s})P(\$|\mathrm{s}) + P(\neg\,\mathrm{s})P(\$|\neg\,\mathrm{s})}$$

Equal prior probabilities simplify the calculation since $P(s) = P(\neg s)$

$$\begin{aligned} P(\mathrm{s}|\$) &= \frac{P(\mathrm{s})P(\$|\mathrm{s})}{P(\mathrm{s})P(\$|\mathrm{s}) + P(\mathrm{s})P(\$|\neg\,\mathrm{s})} \\ &= \frac{P(\$|\mathrm{s})}{P(\$|\mathrm{s}) + P(\$|\neg\,\mathrm{s})} \\ &= \frac{0.375}{0.375 + 0.2} = 0.65 \end{aligned}$$

Therefore, assuming uniform priors implies calculating the probability of the category given the evidence with **no priors**!

Uniform priors

Assuming that categories have the same probability of occurring can be an explicit *decision* of the analyst

This is appropriate if there is no reason to expect the occurrence of one category to be more likely than others.

Uniform priors can also result from the available data in the following scenarios

1. Setting `prior="docfreq"` and having the *same number of documents* in each training class

   For example, there are 500 spam messages and 500 non-spam messages in the dataset

2. Setting `prior="termfreq"` and having the *same the total count of features* in each training class

   For example, all spam messages taken together contain 100,000 words and all non-spam messages also contain 100,000 words

Coding exercise