
Foundations of Natural Language Processing

Lecture 9

Text Classification / Naive Bayes

Ivan Titov

(Slides from Alex Lascarides and Sharon Goldwater)

2 February 2024



Text classification: example

Dear Prof. [ZZ]:

My name is [XX]. I am an ambitious applicant for the Ph.D program of Electrical Engineering and Computer Science at your university. Especially being greatly attracted by your research projects and admiring for your achievements via the school website, I cannot wait to write a letter to express my aspiration to undertake the Ph.D program under your supervision.

I have completed the M.S. program in Information and Communication Engineering with a high GPA of 3.95/4.0 at [YY] University. In addition to throwing myself into the specialized courses in [...] I took part in the research projects, such as [...]. I really enjoyed taking the challenges in the process of the researches and tests, and I spent two years on the research project [...]. We proved the effectiveness of the new method for [...] and published the result in [...].

Having read your biography, I found my academic background and research experiences indicated some possibility of my qualification to join your team. It is my conviction that the enlightening instruction, cutting-edge research projects and state-of-the-art facilities offered by your team will direct me to make breakthroughs in my career development in the arena of electrical engineering and computer science. Thus, I shall be deeply grateful if you could give me the opportunity to become your student. Please do not hesitate to contact me, should you need any further information about my scholastic and research experiences.

Yours sincerely, [XX].

Text classification

We might want to categorize the *content* of the text:

- Spam detection (binary classification: spam/not spam)
- Sentiment analysis (binary or multiway)
 - movie, restaurant, product reviews (pos/neg, or 1-5 stars)
 - political argument (pro/con, or pro/con/neutral)
- Topic classification (multiway: sport/finance/travel/etc)

Text classification

Or we might want to categorize the *author* of the text (**authorship attribution**):

- Native language identification (e.g., to tailor language tutoring)
- Diagnosis of disease (psychiatric or cognitive impairments)
- Identification of gender, dialect, educational background (e.g., in forensics [legal matters], advertising/marketing).

N-gram models for classification?

N-gram models can sometimes be used for classification. But

- For many tasks, sequential relationships between words are largely irrelevant: we can just consider the document as a **bag of words**.
- On the other hand, we may want to include other kinds of features (e.g., part of speech tags) that N-gram models don't include.

In this and the next lecture, we consider two alternative models for classification:

- **Naive Bayes** (this should be review)
- **Maximum Entropy** (aka **multinomial logistic regression**).

Bag of words

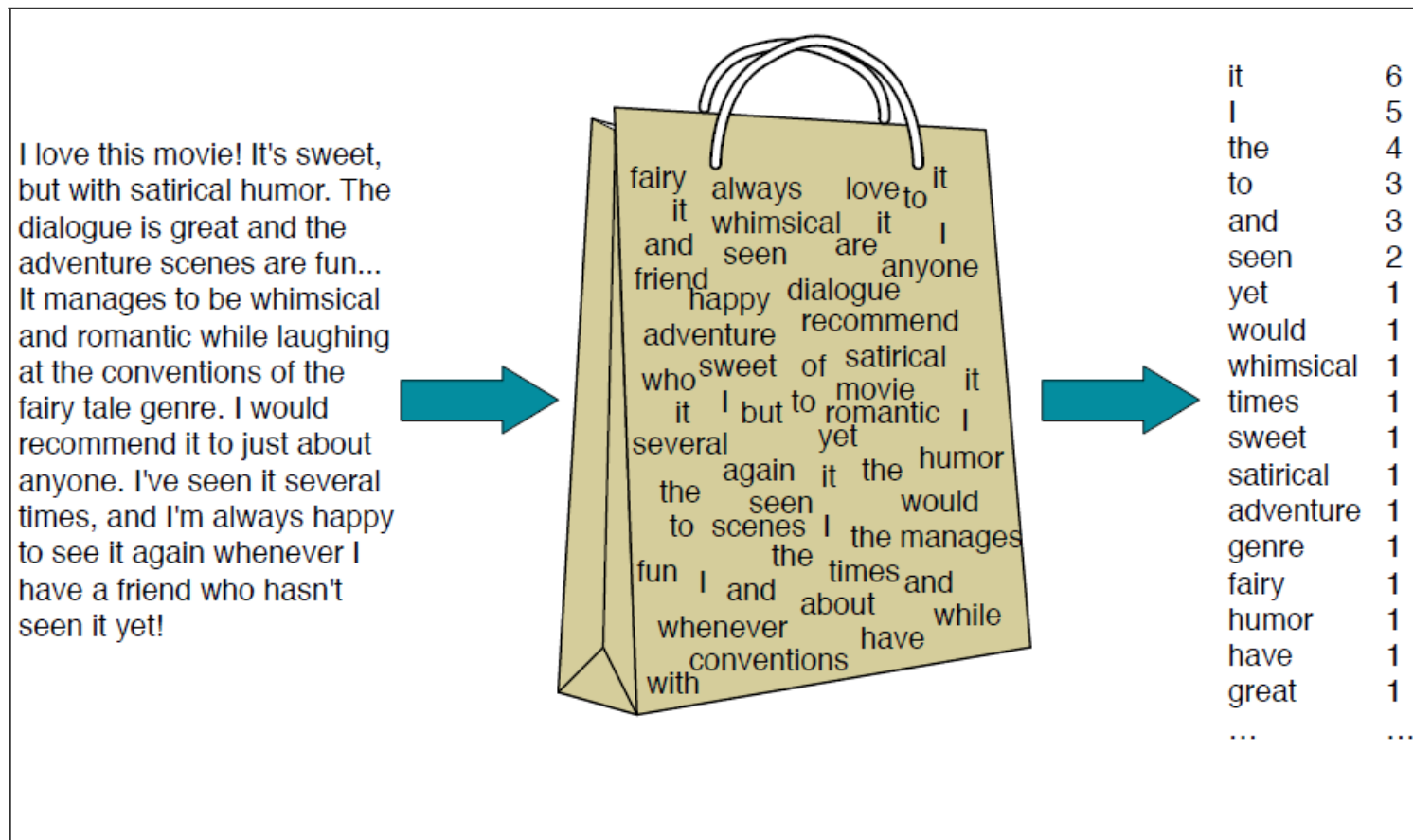


Figure 7.1 Intuition of the multinomial naive Bayes classifier applied to a movie review. The position of the words is ignored (the *bag of words* assumption) and we make use of the frequency of each word.

Figure from J&M 3rd ed. draft, sec 7.1

Naive Bayes: high-level formulation

- Given document d and set of categories C (say, spam/not-spam), we want to assign d to the most probable category \hat{c} .

$$\hat{c} = \operatorname{argmax}_{c \in C} P(c|d)$$

Naive Bayes: high-level formulation

- Given document d and set of categories C (say, spam/not-spam), we want to assign d to the most probable category \hat{c} .

$$\begin{aligned}\hat{c} &= \operatorname{argmax}_{c \in C} P(c|d) \\ &= \operatorname{argmax}_{c \in C} \frac{P(d|c)P(c)}{P(d)} \\ &= \operatorname{argmax}_{c \in C} P(d|c)P(c)\end{aligned}$$

- Just as in spelling correction, we need to define $P(d|c)$ and $P(c)$.

How to model $P(d|c)$?

- First, define a set of **features** that might help classify docs.
 - Here we'll assume these are all the words in the vocabulary.
 - But, we could just use **some** words (more on this later...).
 - Or, use other info, like parts of speech, if available.
- We then represent each document d as the set of features (words) it contains: f_1, f_2, \dots, f_n . So

$$P(d|c) = P(f_1, f_2, \dots, f_n|c)$$

Naive Bayes assumption

- As in LMs, we can't accurately estimate $P(f_1, f_2, \dots, f_n|c)$ due to sparse data.
- So, make a **naive Bayes assumption**: features are conditionally independent given the class.

$$P(f_1, f_2, \dots, f_n|c) \approx P(f_1|c)P(f_2|c)\dots P(f_n|c)$$

- That is, the prob. of a word occurring depends **only** on the class.
 - Not on which words occurred before or after (as in N-grams)
 - Or even which other words occurred at all

Naive Bayes assumption

- Effectively, we only care about the **count** of each feature in each document.
- For example, in spam detection:

	the	your	model	cash	Viagra	class	account	orderz
doc 1	12	3	1	0	0	2	0	0
doc 2	10	4	0	4	0	0	2	0
doc 3	25	4	0	0	0	1	1	0
doc 4	14	2	0	1	3	0	1	1
doc 5	17	5	0	2	0	0	1	1

Naive Bayes classifier

Putting together the pieces, our complete classifier definition:

- Given a document with features f_1, f_2, \dots, f_n and set of categories C , choose the class \hat{c} where

$$\hat{c} = \operatorname{argmax}_{c \in C} P(c) \prod_{i=1}^n P(f_i|c)$$

- $P(c)$ is the **prior probability** of class c before observing any data.
- $P(f_i|c)$ is the probability of seeing feature f_i in class c .

Estimating the class priors

- $P(c)$ normally estimated with MLE:

$$\hat{P}(c) = \frac{N_c}{N}$$

- N_c = the number of training documents in class c
 - N = the total number of training documents
- So, $\hat{P}(c)$ is simply the proportion of training documents belonging to class c .

Learning the class priors: example

- Given training documents with correct labels:

	the	your	model	cash	Viagra	class	account	orderz	spam?
doc 1	12	3	1	0	0	2	0	0	-
doc 2	10	4	0	4	0	0	2	0	+
doc 3	25	4	0	0	0	1	1	0	-
doc 4	14	2	0	1	3	0	1	1	+
doc 5	17	5	0	2	0	0	1	1	+

- $\hat{P}(\text{spam}) = 3/5$

Learning the feature probabilities

- $P(f_i|c)$ normally estimated with simple smoothing:

$$\hat{P}(f_i|c) = \frac{\text{count}(f_i, c) + \alpha}{\sum_{f \in F} (\text{count}(f, c) + \alpha)}$$

- $\text{count}(f_i, c)$ = the number of times f_i occurs in class c
- F = the set of possible features
- α : the smoothing parameter, optimized on held-out data

Learning the feature probabilities: example

	the	your	model	cash	Viagra	class	account	orderz	spam?
doc 1	12	3	1	0	0	2	0	0	-
doc 2	10	4	0	4	0	0	2	0	+
doc 3	25	4	0	0	0	1	1	0	-
doc 4	14	2	0	1	3	0	1	1	+
doc 5	17	5	0	2	0	0	1	1	+

Learning the feature probabilities: example

	the	your	model	cash	Viagra	class	account	orderz	spam?
doc 1	12	3	1	0	0	2	0	0	-
doc 2	10	4	0	4	0	0	2	0	+
doc 3	25	4	0	0	0	1	1	0	-
doc 4	14	2	0	1	3	0	1	1	+
doc 5	17	5	0	2	0	0	1	1	+

$$\hat{P}(\text{your}|+) = \frac{(4+2+5+\alpha)}{(\text{all words in + class})+\alpha F} = (11 + \alpha)/(68 + \alpha F)$$

Learning the feature probabilities: example

	the	your	model	cash	Viagra	class	account	orderz	spam?
doc 1	12	3	1	0	0	2	0	0	-
doc 2	10	4	0	4	0	0	2	0	+
doc 3	25	4	0	0	0	1	1	0	-
doc 4	14	2	0	1	3	0	1	1	+
doc 5	17	5	0	2	0	0	1	1	+

$$\hat{P}(\text{your}|+) = \frac{(4+2+5+\alpha)}{(\text{all words in + class})+\alpha F} = (11 + \alpha)/(68 + \alpha F)$$

$$\hat{P}(\text{your}|-) = \frac{(3+4+\alpha)}{(\text{all words in - class})+\alpha F} = (7 + \alpha)/(49 + \alpha F)$$

$$\hat{P}(\text{orderz}|+) = \frac{(2+\alpha)}{(\text{all words in + class})+\alpha F} = (2 + \alpha)/(68 + \alpha F)$$

Classifying a test document: example

- Test document d :

get your cash and your orderz

- Suppose all features not shown earlier have $\hat{P}(f_i|+) = \frac{\alpha}{(68+\alpha F)}$

$$\begin{aligned} P(+|d) &\propto P(+)\prod_{i=1}^n P(f_i|+) \\ &= P(+)\cdot\frac{\alpha}{(68+\alpha F)}\cdot\frac{11+\alpha}{(68+\alpha F)}\cdot\frac{7+\alpha}{(68+\alpha F)} \\ &\quad\cdot\frac{\alpha}{(68+\alpha F)}\cdot\frac{11+\alpha}{(68+\alpha F)}\cdot\frac{2+\alpha}{(68+\alpha F)} \end{aligned}$$

Classifying a test document: example

- Test document d :

get your cash and your orderz

- Do the same for $P(-|d)$
- Choose the one with the larger value

Very small numbers...

Multiplying large numbers of small probabilities together is problematic in practice

- Even in our toy example $P(-|class)P(-|account)P(-|Viagra)$ with $\alpha = 0.01$ is 5×10^{-5}
- So it would only take two dozen similar words to get down to 10^{-44} , which cannot be represented with as single-precision floating point number
- Even double precision fails once we get to around 175 words with total probability around 10^{-310}

So most actual implementations of Naive Bayes use **costs**

- Costs are negative log probabilities
- So we can sum them, thereby avoiding underflow
- And look for the *lowest* cost overall

Costs and linearity

Using costs, our Naive Bayes equation looks like this

$$\hat{c} = \operatorname{argmin}_{c \in C} (-\log P(c) + \sum_{i=1}^n -\log P(f_i|c))$$

We're finding the *lowest* cost classification.

This amounts to classification using a linear function (in log space) of the input features.

- So Naive Bayes is called a **linear classifier**
- As is Logistic Regression (to come)

Alternative feature values and feature sets

- Use only **binary** values for f_i : did this word occur in d or not?
- Use only a subset of the vocabulary for F
 - Ignore **stopwords** (function words and others with little content)
 - Choose a small task-relevant set (e.g., using a sentiment lexicon)
- Use more complex features (bigrams, syntactic features, morphological features, ...)

Task-specific features

Example words from a **sentiment lexicon**:

Positive:

absolutely	beaming	calm
adorable	beautiful	celebrated
accepted	believe	certain
acclaimed	beneficial	champ
accomplish	bliss	champion
achieve	bountiful	charming
action	bounty	cheery
active	brave	choice
admire	bravo	classic
adventure	brilliant	classical
affirm	bubbly	clean
...		...

Negative:

abysmal	bad	callous
adverse	banal	can't
alarming	barbed	clumsy
angry	belligerent	coarse
annoy	bemoan	cold
anxious	beneath	collapse
apathy	boring	confused
appalling	broken	contradictory
atrocious		contrary
awful		corrosive
		corrupt
		...

From <http://www.enchantedlearning.com/wordlist/>

Choosing features can be tricky

- For example, sentiment analysis might need domain-specific non-sentiment words
 - Such as *quiet*, *memory* for computer product reviews.
- And for other tasks, stopwords might be very useful features
 - E.g., People with schizophrenia use more 2nd-person pronouns (Watson et al, 2012), those with depression use more 1st-person (Rude, 2004).
- Probably better to use too many irrelevant features than not enough relevant ones.

Annotated data is often scarce

- In practice, **annotated texts are often hard to obtain**
 - e.g., one would need someone to label emails as spam vs not-spam
- Thus, we often do not have enough of them
- However, **unannotated texts are generally plentiful**
 - e.g., any email messages

How do we incorporate unannotated texts when estimating the NB model?

“semi-supervised learning”


Semi-supervised learning with NB

		Bayes	your	model	cash	Viagra	class	orderz	spam?
labeled data	lab doc 1	0	1	3	0	0	2	0	-
	lab doc 2	0	2	0	4	0	0	0	+
	lab doc 3	0	2	2	0	0	3	0	-
	lab doc 4	0	3	2	1	3	0	1	+
	lab doc 5	0	1	0	2	0	0	1	+
unlabeled data	unlab doc 1	1	1	1	0	0	2	1	Labels missing
	unlab doc 2	2	2	0	0	0	0	0	
	unlab doc 3	0	1	0	0	1	0	1	

In practice, we would use many more unlabeled examples than labelled ones

Idea 1: 'self-training'

		Bayes	your	model	cash	Viagra	class	orderz	spam?
labeled data	lab doc 1	0	1	3	0	0	2	0	-
	lab doc 2	0	2	0	4	0	0	0	+
	lab doc 3	0	2	2	0	0	3	0	-
	lab doc 4	0	3	2	1	3	0	1	+
	lab doc 5	0	1	0	2	0	0	1	+
unlabeled data	unlab doc 1	1	1	1	0	0	2	1	Labels missing
	unlab doc 2	2	2	0	0	0	0	0	
	unlab doc 3	0	1	0	0	1	0	1	

1. Train NB on labeled data alone
 2. Predict labels on on unlabelled data
 3. Re-estimate NB (in the usual way), but now using also self-labelled data
- 

Estimate NB probabilities on labeled data

		Bayes	your	model	cash	Viagra	class	orderz	spam?
labeled data	lab doc 1	0	1	3	0	0	2	0	-
	lab doc 2	0	2	0	4	0	0	0	+
	lab doc 3	0	2	2	0	0	3	0	-
	lab doc 4	0	3	2	1	3	0	1	+
	lab doc 5	0	1	0	2	0	0	1	+
unlabeled data	unlab doc 1	1	1	1	0	0	2	1	Labels missing
	unlab doc 2	2	2	0	0	0	0	0	
	unlab doc 3	0	1	0	0	1	0	1	

$$\hat{P}(\text{your}|+) = (6 + \alpha)/(20 + \alpha * F)$$

$$\hat{P}(\text{Bayes}|+) = \alpha/(20 + \alpha * F)$$

$$\hat{P}(\text{your}|-) = (3 + \alpha)/(13 + \alpha * F)$$

$$\hat{P}(\text{Bayes}|-) = \alpha/(13 + \alpha * F)$$

$$\hat{P}(\text{spam}) = 3/5$$

$$F = 7, \alpha = 0.1$$

Using that model to predict unknown labels

		Bayes	your	model	cash	Viagra	class	orderz	spam?
labeled data	lab doc 1	0	1	3	0	0	2	0	-
	lab doc 2	0	2	0	4	0	0	0	+
	lab doc 3	0	2	2	0	0	3	0	-
	lab doc 4	0	3	2	1	3	0	1	+
	lab doc 5	0	1	0	2	0	0	1	+
unlabeled data	unlab doc 1	1	1	1	0	0	2	1	-
	unlab doc 2	2	2	0	0	0	0	0	+
	unlab doc 3	0	1	0	0	1	0	1	+

mistake

$$\hat{P}(\text{your}|+) = (6 + \alpha)/(20 + \alpha * F)$$

$$\hat{P}(\text{Bayes}|+) = \alpha/(20 + \alpha * F)$$

$$\hat{P}(\text{your}|-) = (3 + \alpha)/(13 + \alpha * F)$$

$$\hat{P}(\text{Bayes}|-) = \alpha/(13 + \alpha * F)$$

$$\hat{P}(\text{spam}) = 3/5$$

Re-estimate NB parameters

		Bayes	your	model	cash	Viagra	class	orderz	spam?
labeled data	lab doc 1	0	1	3	0	0	2	0	-
	lab doc 2	0	2	0	4	0	0	0	+
	lab doc 3	0	2	2	0	0	3	0	-
	lab doc 4	0	3	2	1	3	0	1	+
	lab doc 5	0	1	0	2	0	0	1	+
unlabeled data	unlab doc 1	1	1	1	0	0	2	1	-
	unlab doc 2	2	2	0	0	0	0	0	+
	unlab doc 3	0	1	0	0	1	0	1	+

mistake

$$\hat{P}(\text{your}|+) = (6 + 3 + \alpha)/(20 + 7 + \alpha * F) \quad \hat{P}(\text{Bayes}|+) = (2 + \alpha)/(20 + 7 + \alpha * F)$$

$$\hat{P}(\text{your}|-) = (3 + 1 + \alpha)/(13 + 6 + \alpha * F) \quad \hat{P}(\text{Bayes}|-) = (1 + \alpha)/(13 + 6 + \alpha * F)$$

$$\hat{P}(\text{spam}) = (3 + 2)/(5 + 3)$$

Re-estimate NB parameters

		Bayes	your	model	cash	Viagra	class	orderz	spam?
labeled data	lab doc 1	0	1	3	0	0	2	0	-
	lab doc 2	0	2	0	4	0	0	0	+
	lab doc 3	0	2	2	0	0	3	0	-
	lab doc 4	0	3	2	1	3	0	1	+
	lab doc 5	0	1	0	2	0	0	1	+
unlabeled data	unlab doc 1	1	1	1	0	0	2	1	-
	unlab doc 2	2	2	0	0	0	0	0	+
	unlab doc 3	0	1	0	0	1	0	1	+

mistake

Learned incorrect association between token “Bayes” and label Spam

$$\hat{P}(\text{Bayes}|+) = (2 + \alpha) / (20 + 7 + \alpha * F)$$

$$\hat{P}(\text{Bayes}|-) = (1 + \alpha) / (13 + 6 + \alpha * F)$$

$$\hat{P}(\text{Bayes}|+) > \hat{P}(\text{Bayes}|-)$$

What caused the issue?

		Bayes	your	model	cash	Viagra	class	orderz	spam?
labeled data	lab doc 1	0	1	3	0	0	2	0	-
	lab doc 2	0	2	0	4	0	0	0	+
	lab doc 3	0	2	2	0	0	3	0	-
	lab doc 4	0	3	2	1	3	0	1	+
	lab doc 5	0	1	0	2	0	0	1	+
unlabeled data	unlab doc 1	1	1	1	0	0	2	1	-
	unlab doc 2	2	2	0	0	0	0	0	+
	unlab doc 3	0	1	0	0	1	0	1	-

mistake

Unlab doc 2:

$$\hat{P}(\text{spam}|d) \approx 0.53$$

The model was not confident in its prediction, while self-training this label as equivalent to 'gold standard'

Self-training

- Advantages:
 - Simplicity and applicable to any classifier (not only NB)
- Disadvantages:
 - Does not account for uncertainty of a classifier
 - No theoretical motivation (kind of...)
- To make it work, well requires
 - discarding low-confidence predictions
 - curriculum (start with examples similar to labeled data)
 - ...

Expectation Maximization

		Bayes	your	model	cash	Viagra	class	orderz	spam?
labeled data	lab doc 1	0	1	3	0	0	2	0	-
	lab doc 2	0	2	0	4	0	0	0	+
	lab doc 3	0	2	2	0	0	3	0	-
	lab doc 4	0	3	2	1	3	0	1	+
	lab doc 5	0	1	0	2	0	0	1	+
unlabeled data	unl doc 2	2	2	0	0	0	0	0	

Unlab doc 2:

$$\hat{P}(\text{spam}|d) \approx 0.53$$

Use soft label: 0.53 of the data point is labelled as “+”, 0.47 as “-”

Expectation Maximization

		Bayes	your	model	cash	Viagra	class	orderz	spam?
labeled data	lab doc 1	0	1	3	0	0	2	0	-
	lab doc 2	0	2	0	4	0	0	0	+
	lab doc 3	0	2	2	0	0	3	0	-
	lab doc 4	0	3	2	1	3	0	1	+
	lab doc 5	0	1	0	2	0	0	1	+
unlabeled data	unl doc 2	2 x 0.53	2 x 0.53	0	0	0	0	0	+ (.53)
		2 x 0.47	2 x 0.47	0	0	0	0	0	- (.47)

Unlab doc 2:

$$\hat{P}(\text{spam}|d) \approx 0.53$$

Use soft label: 0.53 of the data point is labelled as “+”, 0.47 as “-”

Expectation Maximization

		Bayes	your	model	cash	Viagra	class	orderz	spam?
labeled data	lab doc 1	0	1	3	0	0	2	0	-
	lab doc 2	0	2	0	4	0	0	0	+
	lab doc 3	0	2	2	0	0	3	0	-
	lab doc 4	0	3	2	1	3	0	1	+
	lab doc 5	0	1	0	2	0	0	1	+
unlabeled data		2×0.53	2×0.53	0	0	0	0	0	+ (.53)
	unl doc 2	2×0.47	2×0.47	0	0	0	0	0	- (.47)

$$\hat{P}(\text{your}|+) = (6 + 2 \times 0.53 + \alpha) / (20 + 4 \times 0.53 + \alpha * F)$$

$$\hat{P}(\text{your}|-) = (3 + 2 \times 0.47 + \alpha) / (13 + 3 \times 0.47 + \alpha * F)$$

$$\hat{P}(\text{Bayes}|+) = (2 \times 0.53 + \alpha) / (20 + 4 \times 0.53 + \alpha * F)$$

$$\hat{P}(\text{Bayes}|-) = (2 \times 0.47 + \alpha) / (13 + 4 \times 0.47 + \alpha * F)$$

$$\hat{P}(\text{spam}) = \frac{3 + 0.53}{5 + 1}$$

This is just for one data point

EM for Semi-supervised Learning

1. Train NB on labeled data alone
2. Make soft prediction on on unlabelled data ("E-step")
3. Recompute NB parameters using the soft counts

We defined the method algorithmically, but it can be shown to optimize the likelihood of observed data (i.e. a combination labelled and unlabeled portions)

- EM is very general, and some of its generalizations (e.g., Variational Autoencoders / VAE) are standard tools in Deep Learning
- Self-training for NB is known as "hard EM"

justifying the name, "Expectation maximization"

Advantages of Naive Bayes

- Very easy to implement
- Very fast to train, and to classify new documents (good for huge datasets).
- Doesn't require as much training data as some other methods (good for small datasets).
- Usually works reasonably well
- This should be your baseline method for any classification task

Problems with Naive Bayes

- Naive Bayes assumption is naive!
- Consider categories TRAVEL, FINANCE, SPORT.
- Are the following features independent given the category?

beach, sun, ski, snow, pitch, palm, football, relax, ocean

Problems with Naive Bayes

- Naive Bayes assumption is naive!
- Consider categories TRAVEL, FINANCE, SPORT.
- Are the following features independent given the category?

beach, sun, ski, snow, pitch, palm, football, relax, ocean

- No! Ex: Given TRAVEL, seeing beach makes sun more likely, but ski less likely.
- Defining finer-grained categories might help (beach travel vs ski travel), but we don't usually want to.

Non-independent features

- Features are not usually independent given the class
- Adding multiple feature types (e.g., words and morphemes) often leads to even stronger correlations between features
- Accuracy of classifier can sometimes still be ok, but it will be highly **overconfident** in its decisions.
 - Ex: NB sees 5 features that all point to class 1, treats them as five independent sources of evidence.
 - Like asking 5 friends for an opinion when some got theirs from each other.

How to evaluate performance? (recap)

- As discussed before, if classes are fairly well-balanced, **accuracy** is a sensible measure.
 - Simply report the percentage of correct classification decisions.
- However, if (say) 95% of documents belong to class A, it's easy (but not useful) to get 95% accuracy by always guessing A.

A less naive approach

- Although Naive Bayes is a good starting point, often we have enough training data for a better model (and not so much that slower performance is a problem).
- We may be able to get better performance using loads of features and a model that doesn't assume features are conditionally independent.
- Namely, a **Maximum Entropy model**. We will talk about it next time.