Text Classification with Naive Bayes

CS 485, Fall 2024
Applications of Natural Language Processing

Brendan O'Connor
College of Information and Computer Sciences
University of Massachusetts Amherst

[including slides from SLP3 and Mohit Iyyer]

- Canvas announcement/email just now?

- Roadmap
  - Introduce text classification, as an NLP task
  - Method #1: Manually-defined rules and keywords
  - Method #2: Supervised learning
    - Naive Bayes language model

text classification

From European Union <info@eu.org>
Subject
Reply to [redacted]

Please confirm to us that you are the owner of this very email address
with your copy of identity card as proof.

YOU EMAIL ID HAS WON $10,000,000.00 ON THE ONGOING EUROPEAN UNION
COMPENSATION FOR SCAM VICTIMS, CONTACT OUR EMAIL:
CONTACT US NOW VIA EMAIL: [redacted] NOW TO CLAIM YOUR COMPENSATION

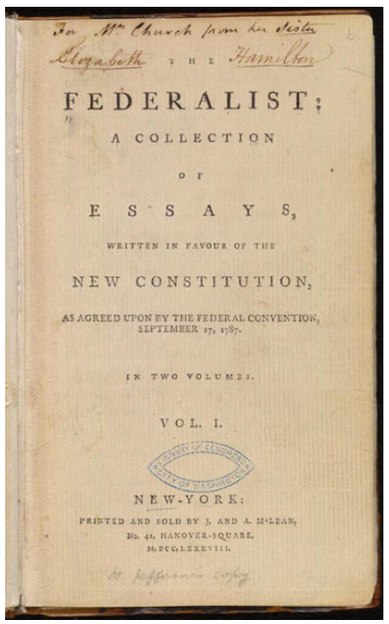
is this **spam**, or **not_spam**?

text classification

Who wrote which *Federalist Papers*?

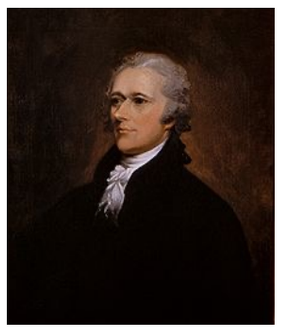
1787-8: essays anonymously written by:

Alexander Hamilton, James Madison, and John Jay

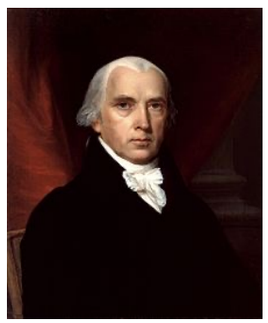


to convince New York to ratify U.S Constitution

Authorship of 12 of the letters unclear between:



Alexander Hamilton



James Madison

1963: solved by Mosteller and Wallace using Bayesian methods

text classification

Positive or negative movie review?

unbelievably disappointing

Full of zany characters and richly applied satire, and some great plot twists

this is the greatest screwball comedy ever filmed

It was pathetic. The worst part about it was the boxing scenes.

text classification

What is the subject of this article?

MEDLINE Article

Syntactic frame and verb bias in aphasia: Plausibility judgments of undergoer-subject sentences

Susanne Gahl,* Lisa Marsh,† Gail Rosenberger,† Daniel S. Jurafsky,† Elizabeth Elder,† Molly Reynga,* and L. Holland Audley*

?

MeSH Subject Category Hierarchy

Antogonists and Inhibitors

Blood Supply

Chemistry

Drug Therapy

Embryology

Epidemiology

…

other examples?

Assigning subject categories, topics, or genres

Spam detection

Authorship identification (who wrote this?)

Language Identification (is this Portuguese?)

Sentiment analysis

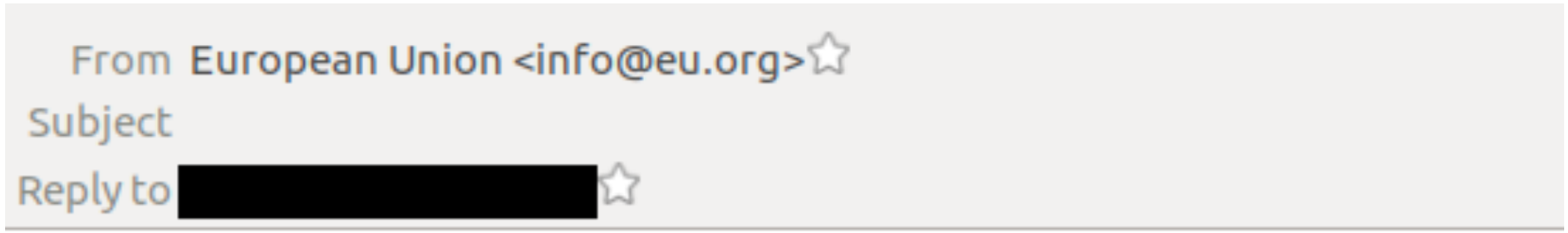
…

text classification

- input: some text **x** (e.g., sentence, document)
- output: a label **y** *(*from a finite label set)
- goal: create a mapping function *f* from **x** to **y**

fyi: basically every NLP problem reduces to learning a mapping function with various definitions of **x** and **y**!

input **x**:



label **y**: **spam** or **not spam**

we'd like to create a mapping *f* such that
f(**x**) = **spam**

Demo: Keyword count classifier

- Let's consider this task:
  binary sentiment classification of movie reviews
  - $y \in$ {POS, NEG}

- Can *manually defined* keyword lists be a useful indicator of text sentiment?
  - For each category, define set of words
    (a.k.a. a sentiment **lexicon** or **dictionary**)
  - For a text: predict the category with highest word frequency
- Let's try manually defined keywords!
  - Sending link (email/canvas)

- Design decisions already....
  - Which class should be predicted? Based on the higher count?
  - How to normalize words to match the lexicon?

f can be hand-designed rules

- if “won $10,000,000” in **x**, **y** = **spam**
- if “CS485” in **x**, **y** = **not spam**

any disadvantages of this method?

f can be **learned** from data

- given **training data** (already-labeled **x,y** pairs for the **task**) we'll create *f* to predict for future instances f(**x**)=y
  - this is known as supervised learning

text classification approaches

- 1. Hand-designed rules (no training)
- 2. Prompting an LLM (needs pretraining)
- 3. Supervised learning
  - Many methods: logistic regression, k-NN, neural networks... today, Naive Bayes

training data:

x (email text)	**y** (spam or not spam)
learn how to fly in 2 minutes	spam
send me your bank info	spam
CS585 Gradescope consent poll	not spam
click here for trillions of $$$	spam

… ideally many more examples!

heldout data:

x (email text)	**y** (spam or not spam)
CS485 important update	not spam
ancient unicorns speaking english!!!	spam

training data:

x (email text)	**y** (spam or not spam)
learn how to fly in 2 minutes	spam
send me your bank info	spam
CS585 Gradescope consent poll	not spam
click here for trillions of $$$	spam

… ideally many more examples!

heldout data:

x (email text)	**y** (spam or not spam)
CS485 important update	not spam
ancient unicorns speaking english!!!	spam

learn mapping function on training data,
measure its accuracy on heldout data

f can be **learned** from data

- given **training data** (already-labeled **x,y** pairs for the **task**) we'll create *f* to predict for future instances f(**x**)=y
  - this is known as supervised learning

- rest of today: instead of standard supervised learning, we'll apply Bayes Rule to extremely simple **language models** ("unigram LM," "Naive Bayes"), that are trained only based on counting words!

probability review

- random variable X takes value x with probability $p(X = x)$; shorthand $p(x)$
- joint probability: $p(X = x, Y = y)$
- conditional probability: $p(X = x \mid Y = y)$

$$= \frac{p(X = x, Y = y)}{p(Y = y)}$$

- when does $p(X = x, Y = y) = p(X = x) \cdot p(Y = y)$?

Naive Bayes

models that estimate *p*(text) are called **language models**. Naive Bayes uses a unigram LM.

Generative story: each document category has its own LM. A text is generated one word at a time, from its associated LM.

To predict on a real document: choose which LM is more likely to have produced the text!

The Bag of Words Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

x

(the doc's text representation)

bag-of-words representation

(token sequence)

x = i hate the actor i love the movie

(BoW vector)

x =

word	count
i	2
hate	1
love	1
the	2
movie	1
actor	1

equivalent representation to:

actor i i the the love movie hate

Naive Bayes

- **w**: a word in the vocabulary V
  x: document text (words), x = (w_1, w_2, w_3, ...)
  y: document's category or label
  p(y): prior probability of the categories
  p(w | y): per-category word probabilities

- Prediction: for new document, choose the y that has highest probability:
  p(y | x)

- Need assumption: each word is independent of all other words, conditional on document label

- Training: given labeled data, we estimate per-class LMs, e.g.
  p(w | y=POS), p(w | y=NEG)

example dataset

- vocabulary V: {i, hate, love, the, movie, actor}
- training data (movie reviews):
  - **y**=NEG **x**=i hate the movie
  - **y**=POS **x**=i love the movie
  - **y**=NEG **x**=i hate the actor
  - **y**=POS **x**=the movie i love
  - **y**=POS **x**=i love love love love love the movie
  - **y**=NEG **x**=hate movie
  - **y**=NEG **x**=i hate the actor i love the movie

labels:
POS
NEG

Training: computing the prior…

- **y**=NEG **x**=i hate the movie
- **y**=POS **x**=i love the movie
- **y**=NEG **x**=i hate the actor
- **y**=POS **x**=the movie i love
- **y**=POS **x**=i love love love love love the movie
- **y**=NEG **x**=hate movie
- **y**=NEG **x**=i hate the actor i love the movie

$p(y)$: estimated simply by counting

label y	count	p(Y=y)	log(p(Y=y))
POS	3	0.43	-0.84
NEG	4	0.57	-0.56

Training: computing the LMs…

for each per-category subcorpus, count words and normalize!

p(w | y=POS)

word	count	p(w l y)
i	3	0.19
hate	0	0.00
love	7	0.44
the	3	0.19
movie	3	0.19
actor	0	0.00
total	**16**	

p(w | y=NEG)

word	count	p(w l y)
i	4	0.22
hate	4	0.22
love	1	0.06
the	4	0.22
movie	3	0.17
actor	2	0.11
total	**18**	

vocabulary (**V**): use set of all words seen in training corpus

probabilistic prediction

Bayes rule (ex: x = text, y = label in {POS, NEG})

$$p(y \mid x) = \frac{p(y) \cdot P(x \mid y)}{p(x)}$$

our predicted label is the one with the highest posterior probability, i.e.,

probabilistic prediction

Bayes rule (ex: x = text, y = label in {POS, NEG})

$$p(y \mid x) = \frac{\overbrace{p(y)}^{\text{prior}} \cdot \overbrace{P(x \mid y)}^{\text{likelihood}}}{p(x)}$$

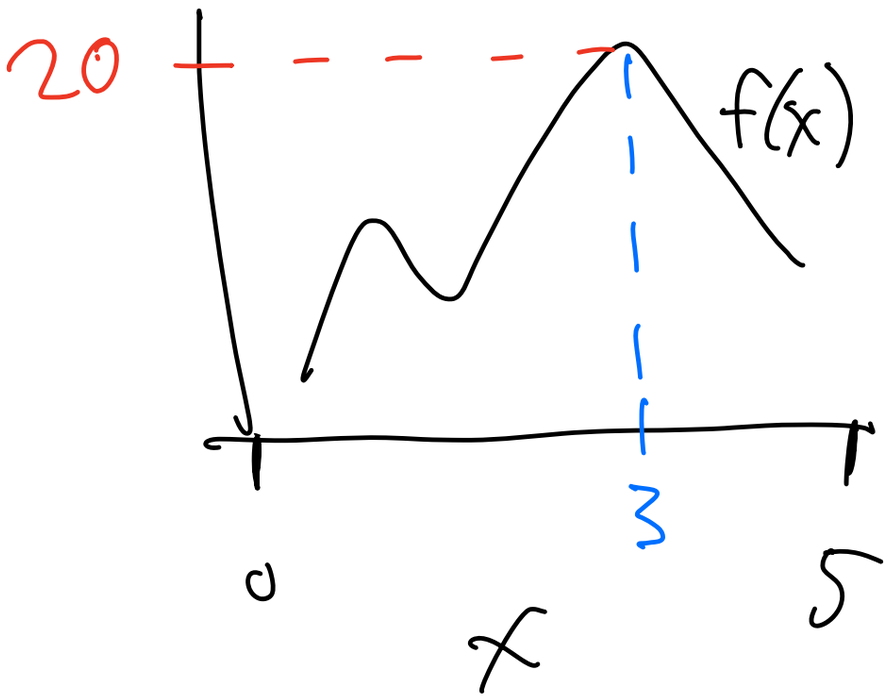
$$\overbrace{p(y \mid x)}^{\text{posterior}}$$

our predicted label is the one with the highest posterior probability, i.e.,

$$\hat{y} = \arg\max_{y \in Y} \left(p(y) \cdot P(x \mid y) \right)$$

what happened to the denominator???

argmax notation



$$\left[\max_{x \in [0,5]} f(x)\right] = 20$$

$$\left[\arg\max_{x \in [0,5]} f(x)\right] = 3$$

apply the independence assumption!

maximum a posteriori (MAP) class

$$\hat{y} = \arg\max_{y \in Y} p(y) \cdot P(x \mid y)$$

$$= \arg\max_{y \in Y} p(y) \cdot \prod_{w \in x} P(w \mid y)$$

$$= \arg\max_{y \in Y} \log p(y) + \sum_{w \in x} \log P(w \mid y)$$

logs to avoid underflow

$p(w_1) \cdot p(w_2) \cdot p(w3) \ \ldots \cdot p(w_n)$

can get really small esp. with large *n*

$$\log \prod p(w_i) = \sum \log p(w_i)$$

$$p(\text{i}) \cdot p(\text{love})^5 \cdot p(\text{the}) \cdot p(\text{movie}) \ = 5.95374181\text{e-}7$$

$$\log p(\text{i}) + 5 \log p(\text{love}) + \log p(\text{the}) + \log p(\text{movie})$$

$$= -14.3340757538$$

Q: Possible range of log p(words)?

[This implementation trick is very common in ML and NLP]

p(X | y=POS)

word	count	p(w I y)
i	3	0.19
hate	0	0.00
love	7	0.44
the	3	0.19
movie	3	0.19
actor	0	0.00
total	**16**	

p(X | y=NEG)

word	count	p(w I y)
i	4	0.22
hate	4	0.22
love	1	0.06
the	4	0.22
movie	3	0.17
actor	2	0.11
total	**18**	

new review X_{new}: love love the movie

$$\log p(X_{\text{new}} | \text{POS}) = \sum_{w \in X_{\text{new}}} \log p(w | \text{POS}) = -4.96$$

$$\log p(X_{\text{new}} | \text{NEG}) = -8.91$$

posterior probs for X_{new}

$$\log p(\text{POS} \mid X_{\text{new}}) \propto \log P(\text{POS}) + \log p(X_{\text{new}} \mid \text{POS})$$
$$= -0.84 - 4.96 = -5.80$$

$$\log p(\text{NEG} \mid X_{\text{new}}) \propto -0.56 - 8.91 = -9.47$$

What does NB predict?

Naive Bayes

- **w**: a word in the vocabulary V
  x: document text (words), x = (w_1, w_2, w_3, ...)
  y: document's category or label
  p(y): prior probability of the categories
  p(w | y): per-category word probabilities

- Steps to use
  - 1. Training: learn p(y) and p(w|y) parameters for all classes and words, based on their counts in labeled training data
  - 2. Prediction: given learned parameters, for new doc, use Bayes Rule to predict posterior probability of class labels

LM parameter estimation

$$\text{unsmoothed } P(w_i \mid y) = \frac{\text{count}(w_i, y)}{\sum_{w \in V} \text{count}(w, y)}$$

LM parameter estimation

$$\text{unsmoothed } P(w_i \mid y) = \frac{\text{count}(w_i, y)}{\sum_{w \in V} \text{count}(w, y)}$$

what if we see no positive training documents containing the word “awesome”?

$$p(\text{awesome} \mid \text{POS}) = 0$$

Add-α (pseudocount) smoothing

$$\text{unsmoothed } P(w_i \mid y) = \frac{\text{count}(w_i, y)}{\sum_{w \in V} \text{count}(w, y)}$$

$$\text{smoothed } P(w_i \mid y) = \frac{\text{count}(w_i, y) + \alpha}{\sum_{w \in V} \text{count}(w, y) + \alpha \mid V \mid}$$

what happens if we do
add-**α** smoothing as **α** increases?

- stopped here 9/16

Other details

- Binarization
  - Issue: overcounting word repetitions
  - Solution: cap word repetitions to 1

- Negation handling
  - Issue:
  - Solution: heuristic

Evaluation

- Must assess accuracy on held-out data.
  - Train/test split
  - (Alternative: cross-validation)
- Must tune hyperparameters (e.g. pseudocount) on a "development" or "tuning" set.
  - Train/dev/test split