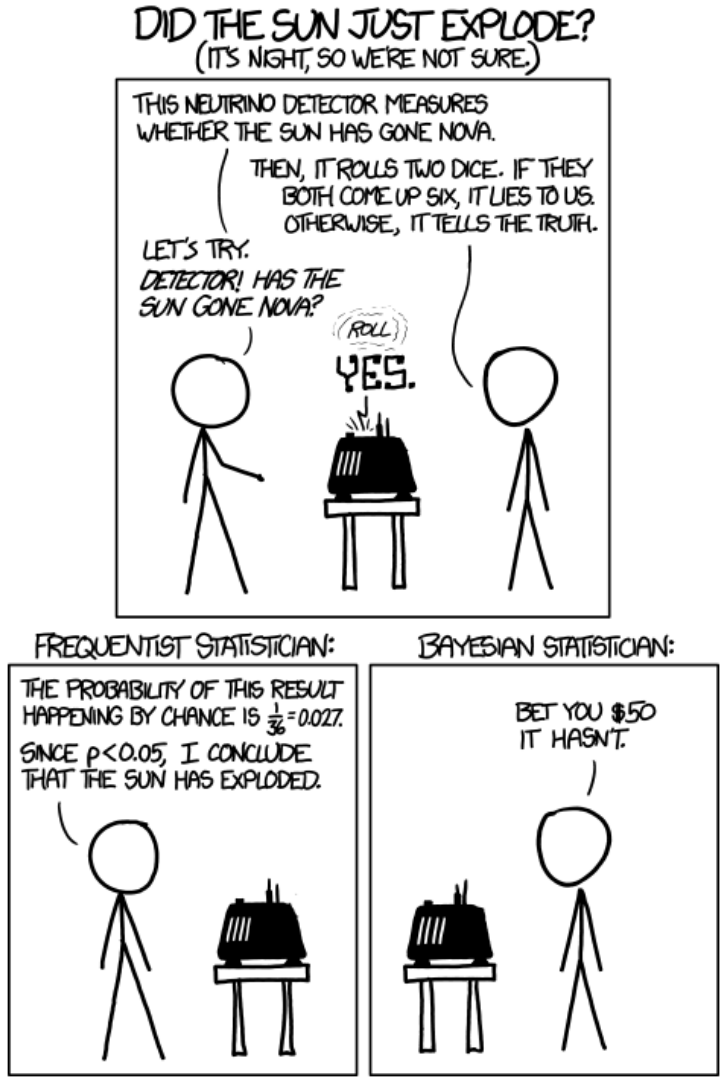
CS440/ECE448 Lecture 4: Naïve Bayes

Mark Hasegawa-Johnson, 1/2024



Naïve Bayes

- minimum probability of error using Bayes' rule
- naïve Bayes
- unigrams and bigrams
- estimating the likelihood: maximum likelihood parameter estimation
- Laplace smoothing

MPE = MAP using Bayes' rule

$$f(x) = \operatorname*{argmax}_y P(Y = y | X = x)$$

$$= \operatorname*{argmax}_y \frac{P(Y = y)P(X = x | Y = y)}{P(X = x)}$$

$$= \operatorname*{argmax}_y P(Y = y)P(X = x | Y = y)$$

Naïve Bayes

- minimum probability of error using Bayes' rule
- naïve Bayes
- unigrams and bigrams
- estimating the likelihood: maximum likelihood parameter estimation
- Laplace smoothing

The problem with likelihood: Too many words

What does it mean to say that the words, x, have a particular probability?

Suppose our training corpus contains two sample emails:

Email1: $Y = \text{spam}, X =$"Hi there man – feel the vitality! Nice meeting you…"

Email2: $Y = \text{ham}, X =$"This needs to be in production by early afternoon…"

Our test corpus is just one email:

Email1: *X*="Hi! You can receive within days an approved prescription for increased vitality and stamina"

How can we estimate $P(X =$ "Hi! You can receive within days an approved prescription for increased vitality and stamina"$|Y = \text{spam})$?

Naïve Bayes: the "Bag-of-words" model

We can estimate the likelihood of an e-mail by pretending that the e-mail is just a bag of words (order doesn't matter).

With only a few thousand spam e-mails, we can get a pretty good estimate of these things:

- $P(W = \text{“hi”}|Y = \text{spam})$, $P(W = \text{“hi”}|Y = \text{ham})$
- $P(W = \text{“vitality”}|Y = \text{spam})$, $P(W = \text{“vitality”}|Y = \text{ham})$
- $P(W = \text{“production”}|Y = \text{spam})$, $P(W = \text{“production”}|Y = \text{ham})$

Then we can approximate $P(X|Y)$ by assuming that the words, W, are ***conditionally independent of one another given the category label***:

$$P(X = x|Y = y) \approx \prod_{i=1}^{n} P(W = w_i|Y = y)$$

Naïve Bayes Representation

- Goal: estimate likelihoods P(Document | Class) and priors P(Class)
- Likelihood: ***bag of words*** representation
  - The document is a sequence of words $[w_1, w_2, \ldots, w_n]$
  - The order of the words in the document is not important
  - Each word is conditionally independent of the others given document class

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial and top secret. …

TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

99 MILLION EMAIL ADDRESSES FOR ONLY $99

Ok, Iknow this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

Bag of words illustration

US Presidential Speeches Tag Cloud
http://chir.ag/projects/preztags/

Bag of words illustration

US Presidential Speeches Tag Cloud
http://chir.ag/projects/preztags/

Bag of words illustration

US Presidential Speeches Tag Cloud
http://chir.ag/projects/preztags/

Why naïve Bayes is "naïve"

We call this model "naïve Bayes" because the words aren't ***really*** conditionally independent given the label. For example, the sequence "for you" is more common in spam emails than it would be if the words "for" and "you" were conditionally independent.

True Statement:

$$P(X = \text{for you} | Y = \text{Spam}) > P(W = \text{for} | Y = \text{Spam}) P(W = \text{you} | Y = \text{Spam})$$

The naïve Bayes approximation simply says: estimating the likelihood of every word sequence is too hard, so for computational reasons, we'll pretend that sequence probability doesn't matter.

Naïve Bayes Approximation:

$$P(X = \text{for you} | Y = \text{Spam}) \approx P(W = \text{for} | Y = \text{Spam}) P(W = \text{you} | Y = \text{Spam})$$

We use naïve Bayes a lot because, even though we know it's wrong, it gives us computationally efficient algorithms that work remarkably well in practice.

MPE = MAP using naïve Bayes

Using naïve Bayes, the MPE decision rule is:

$$f(x) = \underset{y}{\operatorname{argmax}}\, P(Y = y) \prod_{i=1}^{n} P(W = w_i | Y = y)$$

Quiz!

- Go to the course web page, click on “24-Jan” to access the 24-Jan quiz on PrairieLearn

Floating-point underflow

$$f(x) = \underset{y}{\operatorname{argmax}} P(Y = y) \prod_{i=1}^{n} P(W = w_i | Y = y)$$

- That equation has a computational issue. Suppose that the probability of any given word is roughly $P(W = w_i | Y = y) \approx 10^{-3}$, and suppose that there are 103 words in an email. Then $\prod_{i=1}^{n} P(W = w_i | Y = y) = 10^{-309}$, which gets rounded off to zero. This phenomenon is called "floating-point underflow."
- In order to avoid floating-point underflow, we can take the logarithm of the equation above:

$$f(x) = \underset{y}{\operatorname{argmax}} \left(\ln P(Y = y) + \sum_{i=1}^{n} \ln P(W = w_i | Y = y) \right)$$

Naïve Bayes

- minimum probability of error using Bayes' rule
- naïve Bayes
- unigrams and bigrams
- estimating the likelihood: maximum likelihood parameter estimation
- Laplace smoothing

Reducing the naivety of naïve Bayes

Remember that the bag-of-words model is unable to represent this fact:

True Statement:

$$P(X = \text{for you} | Y = \text{Spam}) > P(W = \text{for} | Y = \text{Spam}) P(W = \text{you} | Y = \text{Spam})$$

Though the bag-of-words model can't represent that fact, we can represent it using a slightly more sophisticated naïve Bayes model, called a "bigram" model.

N-Grams

Claude Shannon, in his 1948 book *A Mathematical Theory of Communication*, proposed that the probability of a sequence of words could be modeled using N-grams: sequences of N consecutive words.

- **Unigram**: a unigram (1-gram) is an isolated word, e.g., “you”
- **Bigram**: a bigram (2-gram) is a pair of words, e.g., “for you”
- **Trigram**: a trigram (3-gram) is a triplet of words, e.g., “prescription for you”
- **4-gram**: a 4-gram is a 4-tuple of words, e.g., “approved prescription for you”

Bigram naïve Bayes

A bigram naïve Bayes model approximates the bigrams as conditionally independent, instead of the unigrams. For example,

$$P(X = \text{“approved prescription for you”} | Y = \text{Spam}) \approx$$

$$P(B = \text{“approved prescription”} | Y = \text{Spam}) \times$$
$$P(B = \text{“prescription for”} | Y = \text{Spam}) \times$$
$$P(B = \text{“for you”} | Y = \text{Spam})$$

Advantages and disadvantages of bigram models relative to unigram models

- Advantage: the bigram model can tell you if a particular bigram is much more frequent in spam than in ham emails.
- Disadvantage: over-training. Even if probabilities of individual words in the training and test corpora are similar, probabilities of bigrams might be different.

Naïve Bayes

- minimum probability of error using Bayes' rule
- naïve Bayes
- unigrams and bigrams
- estimating the likelihood: maximum likelihood parameter estimation
- Laplace smoothing

What are “parameters”?

- Oxford English dictionary: parameter (noun): a numerical or other measurable factor forming one of a set that defines a system or sets the conditions of its operation.
- The naïve Bayes model has two types of parameters:
  - The *a priori* parameters: $P(Y = y)$
  - The *likelihood* parameters: $P(W = w_i|Y = y)$

- In order to create a naïve Bayes classifiers, we must somehow estimate the numerical values of those parameters.

Parameter estimation

Model parameters: feature likelihoods P(Word | Class) and priors P(Class)

- How do we obtain the values of these parameters?

prior

spam:	0.33
¬spam:	0.67

P(word | spam)

the :	0.0156
to :	0.0153
and :	0.0115
of :	0.0095
you :	0.0093
a :	0.0086
with:	0.0080
from:	0.0075
...	

P(word | ham)

the :	0.0210
to :	0.0133
of :	0.0119
2002:	0.0110
with:	0.0108
from:	0.0107
and :	0.0105
a :	0.0100
...	

Parameter estimation: Prior

The prior, $P(Y)$, is usually estimated in one of two ways.

- If we believe that the test corpus is like the training corpus, then we just use frequencies in the training corpus:

$$P(Y = \text{Spam}) = \frac{\text{Docs}(Y = \text{Spam})}{\text{Docs}(Y = \text{Spam}) + \text{Docs}(Y \neq \text{Spam})}$$

where "Docs(Y=Spam)" means the number of documents in the training corpus that have the label Y=Spam.

- If we believe that the test corpus is different from the training corpus, then we set $P(Y = \text{Spam})$ = the frequency with which we believe spam will occur in the test corpus.

Parameter estimation: Likelihood

The likelihood, $P(W = w_i | Y = y)$, is also estimated by counting.

The "maximum likelihood estimate of the likelihood parameter" is the most intuitively obvious estimate:

$$P(W = w_i | Y = \text{Spam}) = \frac{\text{Count}(W = w_i, Y = \text{Spam})}{\text{Count}(Y = \text{Spam})}$$

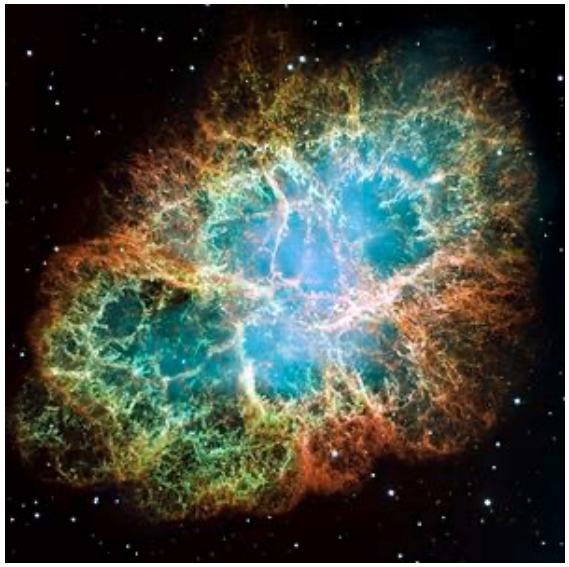
where "$\text{Count}(W = w_i, Y = \text{Spam})$" means the number of times that the word w_i occurs in the Spam portion of the training corpus, and "$\text{Count}(Y = \text{Spam})$" is the total number of words in the Spam portion.

Naïve Bayes

- minimum probability of error using Bayes' rule
- naïve Bayes
- unigrams and bigrams
- estimating the likelihood: maximum likelihood parameter estimation
- Laplace smoothing

What is the probability that the sun will fail to rise tomorrow?

- # times we have observed the sun to rise = 1,825,000
- # times we have observed the sun not to rise = 0
- Estimated probability the sun will not rise = $\frac{0}{0+1,825,000} = 0$



Oops….

Laplace Smoothing

- The basic idea: add k "unobserved observations" to every possible event
- # times the sun has risen or might have ever risen = 1,825,000+k
- # times the sun has failed to rise or might have ever failed to rise = 0+k
- Estimated probability the sun will rise tomorrow = $\frac{1{,}825{,}000+k}{1{,}825{,}000+2k}$
- Estimated probability the sun will not rise = $\frac{k}{1{,}825{,}000+2k}$
- Notice that, if you add these two probabilities together, you get 1.0.

Laplace Smoothing for Naïve Bayes

- The basic idea: add k "unobserved observations" to the count of every unigram
  - If a word occurs 2000 times in the training data, Count = 2000+k
  - If a word occur once in training data, Count = 1+k
  - If a word never occurs in the training data, then it gets a pseudo-Count of k
- Estimated probability of a word that occurred Count(w) times in the training data: =

$$P(W = w) = \frac{k + \text{Count}(W = w)}{k + \sum_v (k + \text{Count}(W = v))}$$

- Estimated probability of a word that never occurred in the training data (an "out of vocabulary" or OOV word):

$$P(W = OOV) = \frac{k}{k + \sum_v (k + \text{Count}(W = v))}$$

- Notice that

$$P(W = OOV) + \sum_w \text{P}(W = w) = 1$$

Conclusions

- MPE = MAP with Bayes' rule:

$$f(x) = \text{argmax}(\log P(Y = y) + \log P(X = x|Y = y))$$

- naïve Bayes:

$$\log P(X = x|Y = y) \approx \sum_{i=1}^{n} \log P(W = w_i|Y = y)$$

- maximum likelihood parameter estimation:

$$P(W = w_i) = \frac{\text{Count}(W = w_i)}{\sum_v \text{Count}(W = v)}$$

- Laplace Smoothing:

$$P(W = w_i) = \frac{k + \text{Count}(W = w_i)}{k + \sum_v (k + \text{Count}(W = v))}$$