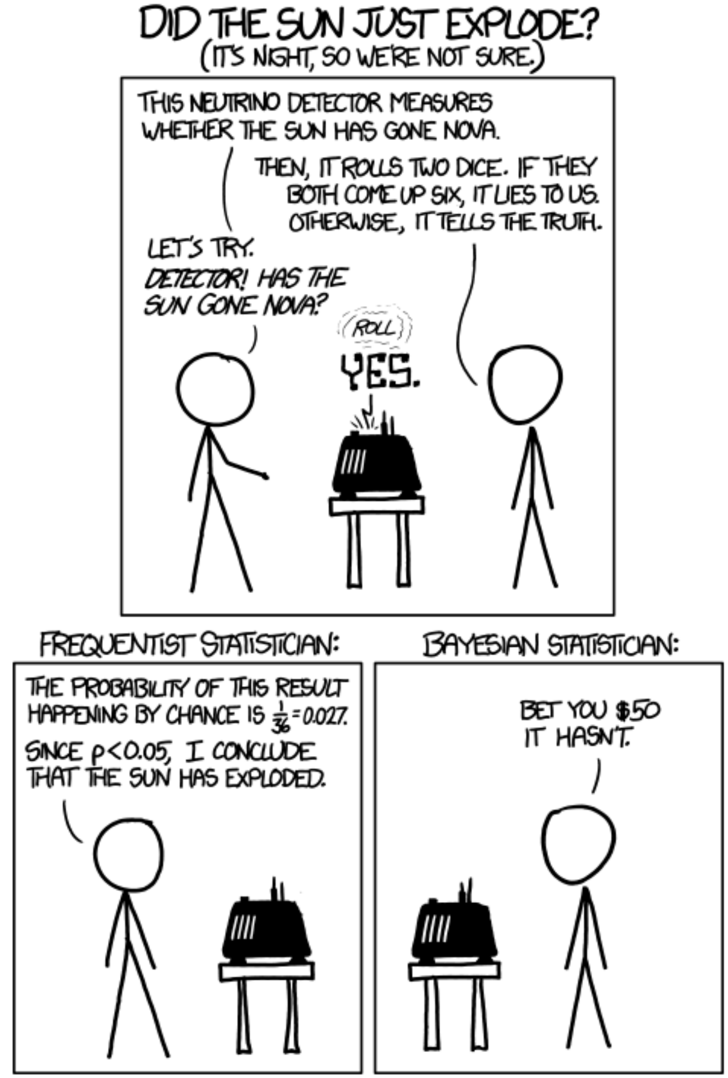
CS440/ECE448 Lecture 4: Naïve Bayes

Mark Hasegawa-Johnson



Naïve Bayes

- naïve Bayes
- unigrams and bigrams
- parameter estimation

MPE classifier using Bayes' rule

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

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

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

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"$|Y = \text{spam})$?

Unigram 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***:

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

The unigram Naïve Bayes classifier

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

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.

Pros and Cons of Naïve Bayes

- What kind of classification is unigram naïve Bayes good for?
- What kind of classification would unigram naïve Bayes probably fail?

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."
- 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

- naïve Bayes
- unigrams and bigrams
- parameter estimation

Reducing the naivety of naïve Bayes

Unigram naïve Bayes is unable to represent this fact:

<u>**True Statement:**</u>

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

We can modify naïve Bayes model to give it this power, using bigrams.

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:
  - Slightly more accurate
- Disadvantage:
  - Far more parameters to store
  - Slightly more computational complexity
  - Far more possibility that we learn useless quirks of the training data, instead of true facts about spam and ham emails. This is called “overtraining the model.”

Naïve Bayes

- naïve Bayes
- unigrams and bigrams
- parameter estimation

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
...	

Maximum likelihood parameter estimation

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

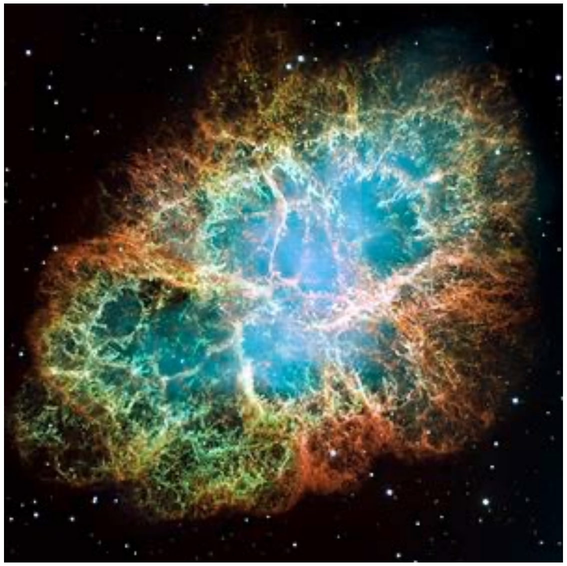
The “maximum likelihood estimate of the parameter” is the most intuitively obvious estimate:

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

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.

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 with out-of-vocabulary events

- If the domain $\mathcal{W}$ is known in advance, the Laplace smoothed estimates are:

$$P(W = w|Y = y) = \frac{k + \text{Count}(w, y)}{\sum_{v \in \mathcal{W}}(k + \text{Count}(v, y))}$$

...which satisfies $\sum_{w \in \mathcal{W}} \mathrm{P}(W = w|Y = y) = 1$.

- ...but suppose we don't know all the possible words, $\mathcal{W}$. We only know some of the words that might be used, $\mathcal{V}$, where $\mathcal{V} \subset \mathcal{W}$. In that case we can allocate some of the probabilities to the "out of vocabulary words" $w \notin \mathcal{V}$:

$$P(W \notin \mathcal{V}|Y = y) = \frac{k}{k + \sum_{v \in \mathcal{V}}(k + \text{Count}(v, y))}$$

$$P(W = w, w \in \mathcal{V}|Y = y) = \frac{k + \text{Count}(w, y)}{k + \sum_{v \in \mathcal{V}}(k + \text{Count}(v, y))}$$

...which still satisfies $\sum_{w \in \mathcal{W}} \mathrm{P}(W = w|Y = y) = 1$.

Quiz!

- Go to the PrairieLearn, try the quiz!

Conclusions

- Naïve Bayes classifier:

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

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

- maximum likelihood parameter estimation:

$$P(W = w_i|Y = y) = \frac{\text{Count}(w_i, y)}{\sum_{v \in V} \text{Count}(v, y)}$$

- Laplace Smoothing:

$$P(W = w|Y = y) = \frac{k + \text{Count}(w, y)}{\sum_{v \in \mathcal{W}} (k + \text{Count}(v, y))}$$