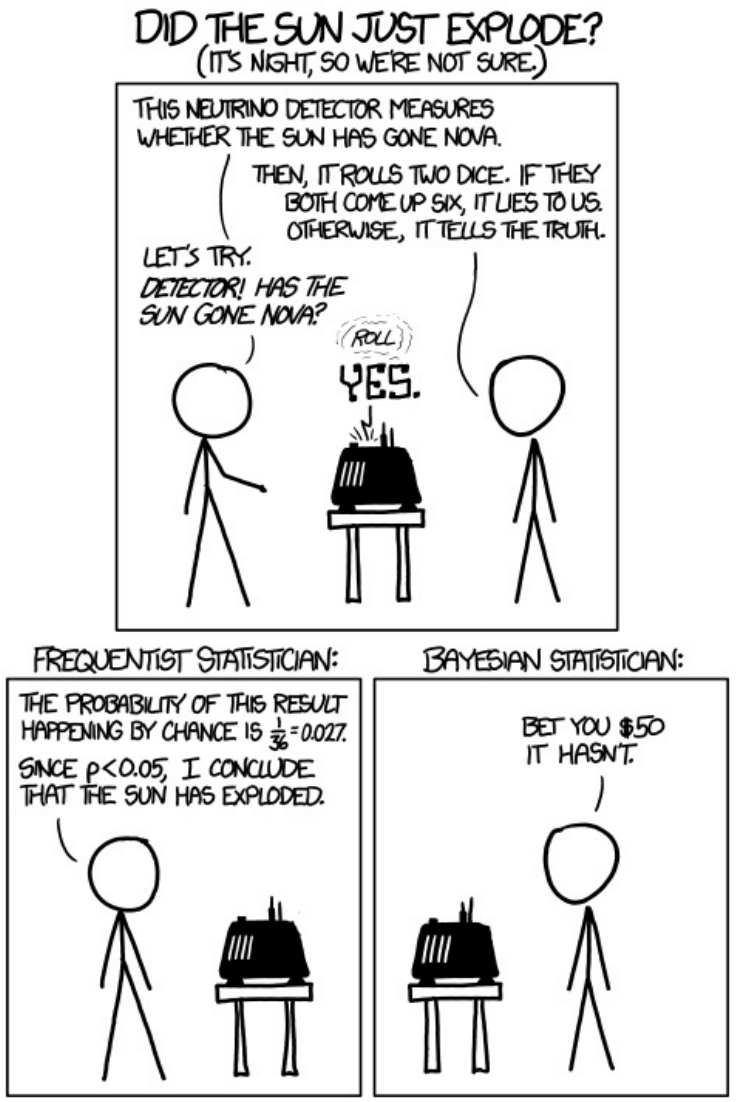
CS440/ECE448 Lecture 3: Naïve Bayes

Mark Hasegawa-Johnson, 1/2022

All content CC-BY 4.0 unless otherwise specified.



© https://www.xkcd.com/1132/

Naïve Bayes

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

Minimum Probability of Error

- Suppose we have an experiment with two random variables, X and Y.
  - X is something we can observe, like the words in an email.
  - Y is something we can't observe, but we want to know. For example, Y=1 means the email is spam (junk mail), Y=0 means it's ham (desirable mail).
- Can we train an AI to read the email, and determine whether it's spam or not?
- Obviously, the AI will sometimes make errors. A useful performance criterion is the probability of an error.

Minimum Probability of Error

- Let's say that f(X) is the decision made by the AI: it sees X, and tries to figure out the value of Y.
- The probability of error is the probability that $f(X) \neq Y$

$$P(\text{Error}) = P(Y = 1, f(X) = 0) + P(Y = 0, f(X) = 1)$$

- Our goal, as system designers, is to design the function f(X) in order to minimize the probability of error.

Minimum Probability of Error

- We can be a bit more specific. Since f(X) is a function of X, we can require that f(X) minimizes the probability of error for every particular value of X:

$$P(\text{Error}|X = x) = \begin{cases} P(Y = 1|X = x) & f(x) = 0 \\ P(Y = 0|X = x) & f(x) = 1 \end{cases}$$

- We don't have control over Y, we don't have control over X. The only thing we can control, in the equation above, is f(x).
- What should f(x) be, in order to minimize P(Error|X)?

Minimum Probability of Error

- We can minimize the probability of error by designing f(x) so that f(x)=1 when Y=1 is more probable, and f(x)=0 when Y=0 is more probable.

$$f(x) = \begin{cases} 1 & P(Y=1|X=x) > P(Y=0|X=x) \\ 0 & P(Y=1|X=x) < P(Y=0|X=x) \end{cases}$$

- This statement is so obvious that we sometimes don't notice how profound it is.

MPE = MAP

- The "minimum probability of error" (MPE) decision rule is the rule that chooses f(X) in order to minimize the probability of error:

$$f(x) = \operatorname{argmin} P(\text{Error}|X = x)$$

- The "maximum *a posteriori*" (MAP) decision rule is the rule that chooses f(X) in order to maximize the *a posteriori* probability:

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

- Those two decision rules are the same. MPE = MAP.

Naïve Bayes

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

The Bayesian Scenario

- Let's use $x \sim X$ to mean that x is an instance of random variable X, and similarly $y \sim Y$.
- In order to minimize the probability of error, we just need to know $P(Y = y | X = x)$ for every pair of values $x \sim X$ and $y \sim Y$. Then we choose $f(x) = \underset{y}{\operatorname{argmax}} P(Y = y | X = x)$.

Example: spam detection

- But how can we estimate $P(Y = y|X = x)$?
- The prior probability of spam might be obvious. If 80% of all email on the internet is spam, that means that

$$P(Y = 1) = 0.8, P(Y = 0) = 0.2$$

- The probability of X given Y is also easy. Suppose we have a database full of sample emails, some known to be spam, some known to be ham. We count how often any word occurs in spam vs. ham emails, and estimate:

$$P(X = x|Y = 1) = \text{frequency of the words } x \text{ in emails known to be spam}$$
$$P(X = x|Y = 0) = \text{frequency of the words } x \text{ in emails known to be ham}$$

- Now we have $P(X = x|Y = y)$ and $P(Y = y)$. How do we get $P(Y = y|X = x)$?

Bayes' Rule

Rev. Thomas Bayes
(1702-1761)

By Unknown - [2][3], Public Domain, https://commons.wikimedia.org/w/index.php?curid=14532025

- The reverend Thomas Bayes solved this problem for us in 1763. His proof has three steps. First, the definition of conditional probability:

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

- Second, he pointed out that there's only two ways X can equal x. Either X=x and Y=0, or X=x and Y=1:

$$P(Y = y|X = x) = \frac{P(X = x, Y = y)}{P(X = x, Y = 0) + P(X = x, Y = 1)}$$

- Finally, apply the definition of conditional probability one more time:

$$P(Y = y|X = x) = \frac{P(X = x|Y = y)P(Y = y)}{P(X = x|Y = 0)P(Y = 0) + P(X = x|Y = 1)P(Y = 1)}$$

Bayes' Rule

Rev. Thomas Bayes
(1702-1761)

By Unknown - [2][3], Public Domain, https://commons.wikimedia.org/w/index.php?curid=14532025

- We can simplify Bayes rule (making it easier to remember) by putting a summation sign in the denominator:

$$P(Y = y|X = x) = \frac{P(X = x|Y = y)P(Y = y)}{\sum_{k\sim Y} P(X = x|Y = k)P(Y = k)}$$

- Most people remember it in an even simpler form: just add together all the terms in the denominator to get:

$$P(Y = y|X = x) = \frac{P(X = x|Y = y)P(Y = y)}{P(X = x)}$$

The four Bayesian probabilities

$$P(Y = y|X = x) = \frac{P(Y = y)P(X = x|Y = y)}{P(X = x)}$$

This equation shows the relationship among four probabilities. This equation has become so world-famous, since 1763, that these four probabilities have standard universally recognized names that you need to know:

- $P(Y = y|X = x)$ is the ***a posteriori*** (after-the-fact) probability, or **posterior**
- $P(Y = y)$ is the ***a priori*** (before-the-fact) probability, or **prior**
- $P(X = x|Y = y)$ is the **likelihood**
- $P(X = x)$ is the **evidence**

Bayes' rule is: the posterior equals the prior times the likelihood over the evidence.

MPE = MAP using Bayes' rule

- MPE = MAP: to minimize the probability of error, design f(X) so that

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

- Bayes' rule:

$$P(Y = y|X = x) = \frac{P(Y = y)P(X = x|Y = y)}{P(X = x)}$$

- Putting the two together:

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

MPE = MAP using Bayes' rule

Suppose your goal is to minimize the probability of error:

$$f(x) = \operatorname{argmin} P(\text{Error}|X = x)$$

…but the only things you know are the prior $P(Y = y)$, and the likelihood $P(X = x|Y = y)$. Well, presto! Using Bayes' rule, we can prove that the MPE decision rule is:

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

Naïve Bayes

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

MPE = MAP using Bayes' rule

Using Bayes' rule, we can prove that the MPE decision rule is:

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

- $P(Y = y)$ is always easy to estimate: we just estimate how many emails on the internet are spam, and how many are not spam.
- $P(X = x|Y = y)$ is a little harder. What does it mean to say that the words, x, have a particular probability?

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: $Y = 1, X =$"hi there man – feel the vitality! Nice meeting you…"

Email2: $Y = 0, X =$"this needs to be in production by early afternoon…"

Our test corpus is just one email:

Email1: *Y=?, X=*"…approved prescription for you…"

How can we estimate $P(X = \text{“ … approved prescription for you … ”} | Y)$?

Naïve Bayes

Naïve Bayes approximates the likelihood. Suppose that x is a list of several consecutive observations (e.g., words), thus

$$x = [w_1, w_2, \ldots, w_n]$$

The naïve Bayes approximation is the assumption that the words are conditionally independent given knowledge of the label:

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

For example,

$$P(X = \text{“approved prescription ...”}|Y = \text{Spam}) \approx$$
$$P(W = \text{approved}|Y = \text{Spam})P(W = \text{prescription}|Y = \text{Spam}) \ldots$$

Naïve Bayes for words = "Bag-of-words"

Naïve Bayes is a general model, applying to any types of observations $x = [w_1, w_2, \ldots, w_n]$. The special case we've been talking about, when w_i are words, is called a "bag of words" model.

We call it "bag of words" because the naïve Bayes approximation notices which words are in the email, but it ignores their order. It's almost like we took all the words in the email, threw them into a bag, and shuffled them up, then asked whether that bag is spam or not.

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

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_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:

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

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
- Bayes' rule
- naïve Bayes
- unigrams and bigrams
- estimating the likelihood: maximum likelihood parameter estimation

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
- Bayes' rule
- naïve Bayes
- unigrams and bigrams
- estimating the likelihood: maximum likelihood parameter estimation

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* probability (prior) 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 | label) and priors p(label)

- 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 | ¬spam)

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, $P(Y = y)$, 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=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. We will refine this estimate in lecture 4, but a useful starting point is what's called the "maximum likelihood estimate of the likelihood parameter:"

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

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.

Likelihood of the training dataset

Consider the following optimization problem. Suppose we want to choose the model parameters, $P(W = w_i|Y = \text{Spam})$, in order to maximize the likelihood of the whole training dataset:

$$\textbf{Maximize}: P(\text{Spam training data}) = \prod_{w_i \in \text{Spam training data}} P(W = w_i|Y = \text{Spam})$$

under the constraint that $P(W = w_i|Y = \text{Spam})$ must be properly normalized probabilities:

$$\textbf{Constraint}: 1 = \sum_{w_i \in \text{dictionary}} P(W = w_i|Y = \text{Spam})$$

If you were to use advanced math (Lagrangians or linear programming) to solve this constrained maximization problem, the result would be:

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

Maximum likelihood estimate of the likelihood parameter

We call this estimate the "maximum likelihood estimate of the likelihood parameter:"

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

… because this is the estimate that maximizes the likelihood of the training dataset, subject to the constraint that

$$1 = \sum_{w_i \in \text{ dictionary}} P(W = w_i | Y = \text{Spam})$$

Conclusions

- MPE = MAP

$$\operatorname{argmin} P(\text{Error}|X = x) = \operatorname{argmax} P(Y = y|X = x)$$

- Bayes' rule

$$P(Y = y|X = x) = \frac{P(Y = y)P(X = x|Y = y)}{P(X = x)}$$

- naïve Bayes

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

- unigrams and bigrams: $w_i =$ "you", $b_i =$ "for you"
- maximum likelihood parameter estimation

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