NAÏVE BAYES

David Kauchak
CS 51A – Spring 2019

Longest word code

http://www.cs.pomona.edu/~dkauchak/classes/cs51a/examples/for_for.txt

Relationship between distributions

$$\underbrace{P(X,Y)}_{\text{joint distribution}} = \underbrace{P(Y)}_{\text{unconditional distribution}} * \underbrace{P(X|Y)}_{\text{conditional distribution}}$$

Can think of it as describing the two events happening in two steps:

The likelihood of X and Y happening:
1. How likely it is that Y happened?
2. Given that Y happened, how likely is it that X happened?

Relationship between distributions

$$P(51Pass, EngPass) = P(EngPass) * P(51Pass|EngPass)$$

The probability of passing CS51 and English is:
1. Probability of passing English *
2. Probability of passing CS51 **given** that you passed English

Relationship between distributions

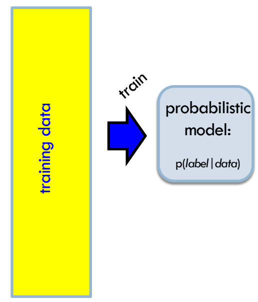
$$P(51Pass, EngPass) = P(51Pass) * P(EngPass|51Pass)$$

The probability of passing CS51 and English is:

1. Probability of passing CS51 *
2. Probability of passing English **given** that you passed CS51

Can also view it with the other event happening first

Back to probabilistic modeling



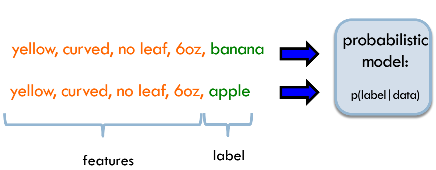
Build a model of the conditional distribution:

P(label | data)

How likely is a label given the data

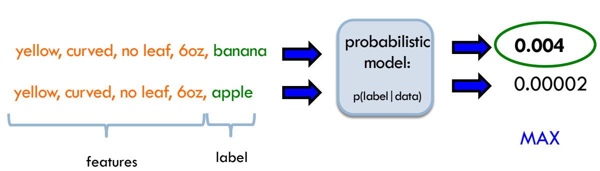
Back to probabilistic models

For each label, calculate the probability of the label given the data



Back to probabilistic models

Pick the label with the highest probability



Naïve Bayes model

Two parallel ways of breaking down the joint distribution

$$P(data, label) = P(label) * P(data|label)$$
$$P(data, label) = P(data) * P(label|data)$$

$$P(label) * P(data|label) = P(data) * P(label|data)$$

What is P(label | data)?

Naïve Bayes

$$P(label) * P(data|label) = P(data) * P(label|data)$$

$$P(label|data) = \frac{P(label) * P(data|label)}{P(data)}$$

(This is called Bayes' rule!)

Naïve Bayes

$$P(label|data) = \frac{P(label) * P(data|label)}{P(data)}$$

probabilistic model:

p(label | data)

$$\frac{P(positive) * P(data|positive)}{P(data)}$$

MAX

$$\frac{P(negative) * P(data|negative)}{P(data)}$$

One observation

$$\frac{P(positive) * P(data|positive)}{P(data)}$$

MAX

$$\frac{P(negative) * P(data|negative)}{P(data)}$$

For picking the largest P(data) doesn't matter!

One observation

$$P(positive) * P(data|positive)$$
$$P(negative) * P(data|negative)$$

MAX

For picking the largest P(data) doesn't matter!

A simplifying assumption (for this class)

$$P(positive) * P(data|positive)$$
$$P(negative) * P(data|negative)$$

MAX

If we assume P(positive) = P(negative) then:

$$P(data|positive)$$
$$P(data|negative)$$

MAX

P(data | label)

$$P(data|label) = P(f_1, f_2, \dots, f_n|label)$$
$$\approx P(f_1|label) * P(f_2|label) * \dots P(f_n|label)$$

This is generally not true!

However…, it makes our life easier.

This is why the model is called **Naïve** Bayes

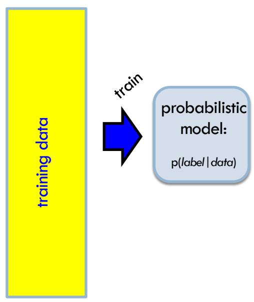
Naïve Bayes

$$P(f_1|positive) * P(f_2|positive) * \dots * P(f_n|positive)$$
$$P(f_1|negative) * P(f_2|negative) * \dots * P(f_n|negative)$$

MAX

Where do these come from?

Training Naïve Bayes



An aside: P(heads)

What is the P(heads) on a fair coin?

0.5

What if you didn't know that, but had a coin to experiment with?

Flip it a bunch of times and count how many times it comes up heads

$$P(heads) = \frac{number\ of\ times\ heads\ came\ up}{total\ number\ of\ coin\ tosses}$$

Try it out…

P(feature | label)

$$P(heads) = \frac{number\ of\ times\ heads\ came\ up}{total\ number\ of\ coin\ tosses}$$

Can we do the same thing here? What is the probability of a feature given positive, i.e. the probability of a feature occurring in in the positive label?

$$P(feature|positive) = ?$$

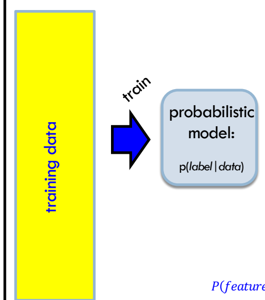
P(feature | label)

$$P(heads) = \frac{number\ of\ times\ heads\ came\ up}{total\ number\ of\ coin\ tosses}$$

Can we do the same thing here? What is the probability of a feature given positive, i.e. the probability of a feature occurring in in the positive label?

$$P(feature|positive) = \frac{number\ of\ positive\ examples\ with\ that\ feature}{total\ number\ of\ positive\ examples}$$

Training Naïve Bayes



1. Count how many examples have each label
2. For all examples with a particular label, count how many times each feature occurs
3. Calculate the conditional probabilities of each feature for all labels:

$$P(feature|label) = \frac{number\ of\ ``label"\ examples\ with\ that\ feature}{total\ number\ of\ examples\ with\ that\ label}$$

Classifying with Naïve Bayes

For each label, calculate the product of p(feature | label) for each label

yellow, curved, no leaf, 6oz

P(yellow | banana)*…*P(6oz | banana)

P(yellow | apple)*…*P(6oz | apple)

MAX

Naïve Bayes Text Classification

Positive	Negative
I loved it	I hated it
I loved that movie	I hated that movie
I hated that I loved it	I loved that I hated it

Given examples of text in different categories, learn to predict the category of new examples

Sentiment classification: given positive/negative examples of text (sentences), learn to predict whether new text is positive/negative

Text classification training

Positive

I loved it

I loved that movie

I hated that I loved it

Negative

I hated it

I hated that movie

I loved that I hated it

We'll assume words just occur once in any given sentence

Text classification training

Positive

I loved it

I loved that movie

I hated that loved it

Negative

I hated it

I hated that movie

I loved that hated it

We'll assume words just occur once in any given sentence

Training the model

Positive

I loved it

I loved that movie

I hated that loved it

Negative

I hated it

I hated that movie

I loved that hated it

For each word and each label, learn:

p(word | label)

Training the model

Positive

I loved it

I loved that movie

I hated that loved it

Negative

I hated it

I hated that movie

I loved that hated it

P(I | positive) = ?

$$P(word|label) = \frac{number\ of\ times\ word\ occured\ in\ \text{"label"}\ examples}{total\ number\ of\ examples\ with\ that\ label}$$

Training the model

Positive	Negative
I loved it	I hated it
I loved that movie	I hated that movie
I hated that loved it	I loved that hated it

P(I | positive) = 3/3 = 1.0

$$P(word|label) = \frac{number\ of\ times\ word\ occured\ in\ \text{"}label\text{"}\ examples}{total\ number\ of\ examples\ with\ that\ label}$$

Training the model

Positive	Negative
I loved it	I hated it
I loved that movie	I hated that movie
I hated that loved it	I loved that hated it

P(I | positive) = 1.0
P(loved | positive) = ?

$$P(word|label) = \frac{number\ of\ times\ word\ occured\ in\ \text{"}label\text{"}\ examples}{total\ number\ of\ examples\ with\ that\ label}$$

Training the model

Positive	Negative
I loved it	I hated it
I loved that movie	I hated that movie
I hated that loved it	I loved that hated it

P(I | positive) = 1.0
P(loved | positive) = 3/3

$$P(word|label) = \frac{number\ of\ times\ word\ occured\ in\ \text{"}label\text{"}\ examples}{total\ number\ of\ examples\ with\ that\ label}$$

Training the model

Positive	Negative
I loved it	I hated it
I loved that movie	I hated that movie
I hated that loved it	I loved that hated it

P(I | positive) = 1.0
P(loved | positive) = 3/3
P(hated | positive) = ?

$$P(word|label) = \frac{number\ of\ times\ word\ occured\ in\ \text{"}label\text{"}\ examples}{total\ number\ of\ examples\ with\ that\ label}$$

Training the model

Positive	Negative
I loved it	I hated it
I loved that movie	I hated that movie
I hated that loved it	I loved that hated it

P(I | positive) = 1.0
P(loved | positive) = 2/3
P(hated | positive) = 1/3
…

P(I | negative) = ?

$$P(word|label) = \frac{number\ of\ times\ word\ occured\ in\ "label"\ examples}{total\ number\ of\ examples\ with\ that\ label}$$

Training the model

Positive	Negative
I loved it	I hated it
I loved that movie	I hated that movie
I hated that loved it	I loved that hated it

P(I | positive) = 1.0
P(loved | positive) = 2/3
P(hated | positive) = 1/3
…

P(I | negative) = 1.0

$$P(word|label) = \frac{number\ of\ times\ word\ occured\ in\ "label"\ examples}{total\ number\ of\ examples\ with\ that\ label}$$

Training the model

Positive	Negative
I loved it	I hated it
I loved that movie	I hated that movie
I hated that loved it	I loved that hated it

P(I | positive) = 1.0
P(loved | positive) = 2/3
P(hated | positive) = 1/3
…

P(I | negative) = 1.0
P(movie | negative) = ?

$$P(word|label) = \frac{number\ of\ times\ word\ occured\ in\ "label"\ examples}{total\ number\ of\ examples\ with\ that\ label}$$

Training the model

Positive	Negative
I loved it	I hated it
I loved that movie	I hated that movie
I hated that loved it	I loved that hated it

P(I | positive) = 1.0
P(loved | positive) = 2/3
P(hated | positive) = 1/3
…

P(I | negative) = 1.0
P(movie | negative) = 1/3
…

$$P(word|label) = \frac{number\ of\ times\ word\ occured\ in\ "label"\ examples}{total\ number\ of\ examples\ with\ that\ label}$$

Classifying

P(I | positive)	= 1.0	P(I | negative)	= 1.0
P(loved | positive)	= 1.0	p(hated | negative)	= 1.0
p(it | positive)	= 2/3	p(that | negative)	= 2/3
p(that | positive)	= 2/3	P(movie | negative)	= 1/3
p(movie | positive)	= 1/3	p(it | negative)	= 2/3
P(hated | positive)	= 1/3	p(loved | negative)	= 1/3

Notice that each of this is it's own probability distribution

P(loved | positive)
P(loved | positive) = 2/3
P(no loved | positive) = 1/3

Trained model

P(I | positive)	= 1.0	P(I | negative)	= 1.0
P(loved | positive)	= 2/3	p(hated | negative)	= 1.0
p(it | positive)	= 2/3	p(that | negative)	= 2/3
p(that | positive)	= 2/3	P(movie | negative)	= 1/3
p(movie | positive)	= 1/3	p(it | negative)	= 2/3
P(hated | positive)	= 1/3	p(loved | negative)	= 1/3

How would we classify: "I hated movie"?

Trained model

P(I | positive)	= 1.0	P(I | negative)	= 1.0
P(loved | positive)	= 2/3	p(hated | negative)	= 1.0
p(it | positive)	= 2/3	p(that | negative)	= 2/3
p(that | positive)	= 2/3	P(movie | negative)	= 1/3
p(movie | positive)	= 1/3	p(it | negative)	= 2/3
P(hated | positive)	= 1/3	p(loved | negative)	= 1/3

P(I | positive) * P(hated | positive) * P(movie | positive) = 1.0 * 1/3 * 1/3 = 1/9

P(I | negative) * P(hated | negative) * P(movie | negative) = 1.0 * 1.0 * 1/3 = 1/3

Trained model

P(I | positive)	= 1.0	P(I | negative)	= 1.0
P(loved | positive)	= 2/3	p(hated | negative)	= 1.0
p(it | positive)	= 2/3	p(that | negative)	= 2/3
p(that | positive)	= 2/3	P(movie | negative)	= 1/3
p(movie | positive)	= 1/3	p(it | negative)	= 2/3
P(hated | positive)	= 1/3	p(loved | negative)	= 1/3

How would we classify: "I hated the movie"?

Trained model

P(I | positive)	= 1.0	P(I | negative)	= 1.0
P(loved | positive)	= 2/3	p(hated | negative)	= 1.0
p(it | positive)	= 2/3	p(that | negative)	= 2/3
p(that | positive)	= 2/3	P(movie | negative)	= 1/3
p(movie | positive)	= 1/3	p(it | negative)	= 2/3
P(hated | positive)	= 1/3	p(loved | negative)	= 1/3

P(I | positive) * P(hated | positive) * P(the | positive) * P(movie | positive) =

P(I | negative) * P(hated | negative) * P(the | negative) * P(movie | negative) =

Trained model

P(I | positive)	= 1.0	P(I | negative)	= 1.0
P(loved | positive)	= 2/3	p(hated | negative)	= 1.0
p(it | positive)	= 2/3	p(that | negative)	= 2/3
p(that | positive)	= 2/3	P(movie | negative)	= 1/3
p(movie | positive)	= 1/3	p(it | negative)	= 2/3
P(hated | positive)	= 1/3	p(loved | negative)	= 1/3

P(I | positive) * P(hated | positive) * P(the | positive) * P(movie | positive) =

P(I | negative) * P(hated | negative) * P(the | negative) * P(movie | negative) =

What are these?

Trained model

P(I | positive)	= 1.0	P(I | negative)	= 1.0
P(loved | positive)	= 2/3	p(hated | negative)	= 1.0
p(it | positive)	= 2/3	p(that | negative)	= 2/3
p(that | positive)	= 2/3	P(movie | negative)	= 1/3
p(movie | positive)	= 1/3	p(it | negative)	= 2/3
P(hated | positive)	= 1/3	p(loved | negative)	= 1/3

P(I | positive) * P(hated | positive) * P(the | positive) * P(movie | positive) =

P(I | negative) * P(hated | negative) * P(the | negative) * P(movie | negative) =

0! Is this a problem?

Trained model

P(I | positive)	= 1.0	P(I | negative)	= 1.0
P(loved | positive)	= 2/3	p(hated | negative)	= 1.0
p(it | positive)	= 2/3	p(that | negative)	= 2/3
p(that | positive)	= 2/3	P(movie | negative)	= 1/3
p(movie | positive)	= 1/3	p(it | negative)	= 2/3
P(hated | positive)	= 1/3	p(loved | negative)	= 1/3

P(I | positive) * P(hated | positive) * P(the | positive) * P(movie | positive) =

P(I | negative) * P(hated | negative) * P(the | negative) * P(movie | negative) =

Yes. They make the entire product go to 0!

Trained model

P(I | positive)	= 1.0	P(I | negative)	= 1.0
P(loved | positive)	= 2/3	p(hated | negative)	= 1.0
p(it | positive)	= 2/3	p(that | negative)	= 2/3
p(that | positive)	= 2/3	P(movie | negative)	= 1/3
p(movie | positive)	= 1/3	p(it | negative)	= 2/3
P(hated | positive)	= 1/3	p(loved | negative)	= 1/3

P(I | positive) * P(hated | positive) * P(the | positive) * P(movie | positive) =

P(I | negative) * P(hated | negative) * P(the | negative) * P(movie | negative) =

Our solution: assume any unseen word has a small, fixed probability, e.g. in this example 1/10

Trained model

P(I | positive)	= 1.0	P(I | negative)	= 1.0
P(loved | positive)	= 2/3	p(hated | negative)	= 1.0
p(it | positive)	= 2/3	p(that | negative)	= 2/3
p(that | positive)	= 2/3	P(movie | negative)	= 1/3
p(movie | positive)	= 1/3	p(it | negative)	= 2/3
P(hated | positive)	= 1/3	p(loved | negative)	= 1/3

P(I | positive) * P(hated | positive) * P(the | positive) * P(movie | positive) = 1/90

P(I | negative) * P(hated | negative) * P(the | negative) * P(movie | negative) = 1/30

Our solution: assume any unseen word has a small, fixed probability, e.g. in this example 1/10

Full disclaimer

I've fudged a few things on the Naïve Bayes model for simplicity

Our approach is very close, but it takes a few liberties that aren't technically correct, but it will work just fine ☺

If you're curious, I'd be happy to talk to you offline