Longest word code

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

Relationship between distributions

\[ P(X, Y) = P(Y) \cdot P(X|Y) \]

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

1. How likely is that \( Y \) happened?
2. Given that \( Y \) happened, how likely is it that \( X \) happened?

Relationship between distributions

\[ P(\text{51Pass,EngPass}) = P(\text{EngPass}) \cdot P(\text{51Pass}|\text{EngPass}) \]

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

\[ P(\text{CS51 Pass, Eng Pass}) = P(\text{CS51 Pass}) \times P(\text{Eng Pass}|\text{CS51 Pass}) \]

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

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

- yellow, curved, no leaf, 6oz, banana
- yellow, curved, no leaf, 6oz, apple

Pick the label with the highest probability

- yellow, curved, no leaf, 6oz, banana: 0.004
- yellow, curved, no leaf, 6oz, apple: 0.00002

MAX
Naïve Bayes model

Two parallel ways of breaking down the joint distribution

\[ P(\text{data}, \text{label}) = P(\text{label}) \cdot P(\text{data}|\text{label}) \]

\[ P(\text{data}, \text{label}) = P(\text{data}) \cdot P(\text{label}|\text{data}) \]

\[ P(\text{label}) \cdot P(\text{data}|\text{label}) = P(\text{data}) \cdot P(\text{label}|\text{data}) \]

What is \( P(\text{label}|\text{data}) \)?

Naïve Bayes

\[ P(\text{label}) \cdot P(\text{data}|\text{label}) = P(\text{data}) \cdot P(\text{label}|\text{data}) \]

\[ P(\text{label}|\text{data}) = \frac{P(\text{label}) \cdot P(\text{data}|\text{label})}{P(\text{data})} \]

(This is called Bayes’ rule!)

Naïve Bayes

\[ P(\text{label}|\text{data}) = \frac{P(\text{label}) \cdot P(\text{data}|\text{label})}{P(\text{data})} \]

\[ P(\text{label}|\text{data}) = \frac{P(\text{positive}) \cdot P(\text{data}|\text{positive})}{P(\text{data})} \]

\[ P(\text{label}|\text{data}) = \frac{P(\text{negative}) \cdot P(\text{data}|\text{negative})}{P(\text{data})} \]

For picking the largest \( P(\text{data}) \) doesn’t matter!

One observation

probabilistic model: \( P(\text{label}|\text{data}) \)
One observation

For picking the largest $P(\text{data})$ doesn’t matter!

A simplifying assumption (for this class)

\[
P(\text{positive}) \cdot P(\text{data}|\text{positive}) \quad \text{MAX} \\
(P(\text{negative}) \cdot P(\text{data}|\text{negative})
\]

If we assume $P(\text{positive}) = P(\text{negative})$ then:

\[
P(\text{data}|\text{positive}) \quad \text{MAX} \\
(P(\text{data}|\text{negative})
\]

P(data|label)

\[
P(\text{data}|\text{label}) \approx \left( P(f_1|\text{label}) \cdot P(f_2|\text{label}) \cdots P(f_n|\text{label}) \right)
\]

This is generally not true!

However..., it makes our life easier.

This is why the model is called Naïve Bayes

Naïve Bayes

\[
P(f_1|\text{positive}) \cdot P(f_2|\text{positive}) \cdots P(f_n|\text{positive}) \quad \text{MAX}
\]

\[
P(f_1|\text{negative}) \cdot P(f_2|\text{negative}) \cdots P(f_n|\text{negative})
\]

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.

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

Try it out...

P(feature | label)

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 the positive label?

\[
P(\text{feature} | \text{positive}) = ?
\]
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 the positive label?

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

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

\[ P(\text{feature} | \text{label}) = \frac{\text{number of "example" examples with that feature}}{\text{total number of examples with that label}} \]

Classifying with Naïve Bayes

For each label, calculate the product of \( P(\text{feature} | \text{label}) \) for each label

\[
P(\text{yellow}, \text{curved}, \text{no leaf}, 6\text{oz} | \text{banana}) \times \ldots \times P(6\text{oz} | \text{banana})
\]

\[
P(\text{yellow}, \text{curved}, \text{no leaf}, 6\text{oz} | \text{apple}) \times \ldots \times P(6\text{oz} | \text{apple})
\]

Naïve Bayes Text Classification

Positive

- I loved it
- I loved that movie
- I hated that I loved it

I loved it

Negative

- I hated it
- I hated that movie
- I loved that I hated 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

<table>
<thead>
<tr>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>I loved it</td>
<td>I hated it</td>
</tr>
<tr>
<td>I loved that movie</td>
<td>I hated that movie</td>
</tr>
<tr>
<td>I hated that I loved it</td>
<td>I loved that I hated it</td>
</tr>
</tbody>
</table>

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

### Training the Model

<table>
<thead>
<tr>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>I loved it</td>
<td>I hated it</td>
</tr>
<tr>
<td>I loved that movie</td>
<td>I hated that movie</td>
</tr>
<tr>
<td>I hated that loved it</td>
<td>I loved that hated it</td>
</tr>
</tbody>
</table>

For each word and each label, learn:

\[ p(\text{word} | \text{label}) \]

\[
P(I | \text{positive}) = ?
\]

\[
P(\text{word}|\text{label}) = \frac{\text{number of times word occurred in "label" examples}}{\text{total number of examples with that label}}
\]
Training the model

<table>
<thead>
<tr>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>I loved it</td>
<td>I hated it</td>
</tr>
<tr>
<td>I loved that movie</td>
<td>I hated that movie</td>
</tr>
<tr>
<td>I hated that loved it</td>
<td>I loved that hated it</td>
</tr>
</tbody>
</table>

\[
P(I \mid \text{positive}) = \frac{3}{3} = 1.0
\]

\[
P(I \mid \text{positive}) = \frac{p(\text{word} \mid \text{label})}{\text{total number of examples with that label}}
\]

Training the model

<table>
<thead>
<tr>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>I loved it</td>
<td>I hated it</td>
</tr>
<tr>
<td>I loved that movie</td>
<td>I hated that movie</td>
</tr>
<tr>
<td>I hated that loved it</td>
<td>I loved that hated it</td>
</tr>
</tbody>
</table>

\[
P(I \mid \text{positive}) = 1.0
\]

\[
P(I \mid \text{positive}) = \frac{p(\text{word} \mid \text{label})}{\text{total number of examples with that label}}
\]
## 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

\[
P(I \mid \text{positive}) = 1.0 \quad P(I \mid \text{negative}) = ?
\]
\[
P(\text{loved} \mid \text{positive}) = \frac{2}{3} \quad P(\text{loved} \mid \text{positive}) = \frac{2}{3}
\]
\[
P(\text{hated} \mid \text{positive}) = \frac{1}{3} \quad P(\text{hated} \mid \text{positive}) = \frac{1}{3}
\]

\[
P(\text{word} \mid \text{label}) = \frac{\text{number of times word occurred in } \text{"label" examples}}{\text{total number of examples with that label}}
\]

---

## 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

\[
P(I \mid \text{positive}) = 1.0 \quad P(I \mid \text{negative}) = 1.0
\]
\[
P(\text{loved} \mid \text{positive}) = \frac{2}{3} \quad P(\text{loved} \mid \text{positive}) = \frac{2}{3}
\]
\[
P(\text{hated} \mid \text{positive}) = \frac{1}{3} \quad P(\text{hated} \mid \text{positive}) = \frac{1}{3}
\]

\[
P(\text{word} \mid \text{label}) = \frac{\text{number of times word occurred in } \text{"label" examples}}{\text{total number of examples with that label}}
\]
Classifying

<table>
<thead>
<tr>
<th>Event</th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(I</td>
<td>positive)</td>
<td>1.0</td>
</tr>
<tr>
<td>P(loved</td>
<td>positive)</td>
<td>1.0</td>
</tr>
<tr>
<td>P(it</td>
<td>positive)</td>
<td>2/3</td>
</tr>
<tr>
<td>P(that</td>
<td>positive)</td>
<td>2/3</td>
</tr>
<tr>
<td>P(movie</td>
<td>positive)</td>
<td>1/3</td>
</tr>
<tr>
<td>P(hated</td>
<td>positive)</td>
<td>1/3</td>
</tr>
</tbody>
</table>

Notice that each of this is its own probability distribution

### Trained model

<table>
<thead>
<tr>
<th>Event</th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(I</td>
<td>positive)</td>
<td>1.0</td>
</tr>
<tr>
<td>P(loved</td>
<td>positive)</td>
<td>2/3</td>
</tr>
<tr>
<td>P(it</td>
<td>positive)</td>
<td>2/3</td>
</tr>
<tr>
<td>P(that</td>
<td>positive)</td>
<td>1/3</td>
</tr>
<tr>
<td>P(movie</td>
<td>positive)</td>
<td>1/3</td>
</tr>
<tr>
<td>P(hated</td>
<td>positive)</td>
<td>1/3</td>
</tr>
</tbody>
</table>

How would we classify: “I hated movie”?

\[
P(I \mid positive) \times P(hated \mid positive) \times P(movie \mid positive) = 1.0 \times 1/3 \times 1/3 = 1/9
\]

Trained model

<table>
<thead>
<tr>
<th>Event</th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(I</td>
<td>positive)</td>
<td>1.0</td>
</tr>
<tr>
<td>P(loved</td>
<td>positive)</td>
<td>2/3</td>
</tr>
<tr>
<td>P(it</td>
<td>positive)</td>
<td>2/3</td>
</tr>
<tr>
<td>P(that</td>
<td>positive)</td>
<td>1/3</td>
</tr>
<tr>
<td>P(movie</td>
<td>positive)</td>
<td>1/3</td>
</tr>
<tr>
<td>P(hated</td>
<td>positive)</td>
<td>1/3</td>
</tr>
</tbody>
</table>

How would we classify: “I hated the movie”?

\[
P(I \mid positive) \times P(hated \mid negative) \times P(movie \mid negative) = 1.0 \times 1.0 \times 1/3 = 1/3
\]
The Trained model:

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

\[
P(I | positive) * P(hated | positive) * P(the | positive) * P(movie | positive) = 0
\]

\[
P(I | negative) * P(hated | negative) * P(the | negative) * P(movie | negative) = 0
\]

What are these?

---

Yes. They make the entire product go to 0!
Trained model

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(I</td>
<td>positive)</td>
<td>= 1.0</td>
</tr>
<tr>
<td>P(loved</td>
<td>positive)</td>
<td>= 2/3</td>
</tr>
<tr>
<td>P(it</td>
<td>positive)</td>
<td>= 2/3</td>
</tr>
<tr>
<td>P(that</td>
<td>positive)</td>
<td>= 2/3</td>
</tr>
<tr>
<td>P(movie</td>
<td>positive)</td>
<td>= 2/3</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(I</td>
<td>positive)</td>
<td>= 1.0</td>
</tr>
<tr>
<td>P(loved</td>
<td>positive)</td>
<td>= 2/3</td>
</tr>
<tr>
<td>P(it</td>
<td>positive)</td>
<td>= 2/3</td>
</tr>
<tr>
<td>P(that</td>
<td>positive)</td>
<td>= 2/3</td>
</tr>
<tr>
<td>P(movie</td>
<td>positive)</td>
<td>= 2/3</td>
</tr>
</tbody>
</table>

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