DSC 40A

Theoretical Foundations of Data Science I

Last Time

- Classification is the problem of predicting a categorical variable.
- The Naive Bayes algorithm gives a strategy for classifying data according to its features.
- It is naive because it relies on an assumption of conditional independence of the features, for a given class.

Naive Bayes Algorithm

- Bayes' Theorem shows how to calculate P(class | features).

$$P(\text{class}|\text{features}) = \frac{P(\text{features}|\text{class}) * P(\text{class})}{P(\text{features})}$$

- Rewrite the numerator, using the naive assumption of conditional independence of features given the class.
- Estimate each term in the numerator based on the training data.
- Select class based on whichever has the larger numerator.

In This Video

- How can we use naive Bayes to classify text?

Bayes' Theorem for Text Classification

Text classification problems include:

- sentiment analysis
    - positive and negative customer reviews
- determining genre
    - news articles, blog posts, etc.
- email foldering
    - promotions tab in Gmail
- **spam filtering**
    - **separating spam from ham (good, non-spam email)**

Features

Represent an email as a vector or array of features

$$(x_1, x_2, x_3, \dots, x_n)$$

where *i* is an index into a dictionary of *n* possible words, and

x_i = 1 if word *i* is present in the email
x_i = 0 otherwise

Features

Called the “bag of words” model:

Ignores location of words within the email, and the frequency of words

Example:

usually n = 10,000 to 50,000 words in practice

Naive Bayes Spam Classifier

$$\boxed{P(\text{class}|\text{features})} = \frac{P(\text{features}|\text{class}) * P(\text{class})}{P(\text{features})}$$

To classify an email, we will use Bayes' Theorem to calculate the probability of it belonging to each class:

P(spam | features) and P(ham | features)

Then choose the class according to the **larger** of these two probabilities.

Naive Bayes Spam Classifier

$$P(\text{class}|\text{features}) = \frac{\boxed{P(\text{features}|\text{class})} * \boxed{P(\text{class})}}{P(\text{features})}$$

Observe: the formulas for P(spam | features) and P(ham | features) have the same denominator, P(features).

We can find the larger of the two probabilities by just comparing numerators.

P(features | spam)*P(spam) vs. P(features | ham)*P(ham)

Naive Bayes Spam Classifier

$$P(\text{class}|\text{features}) = \frac{\boxed{P(\text{features}|\text{class})} * \boxed{P(\text{class})}}{P(\text{features})}$$

To use Bayes' Theorem, need to determine four quantities:

i. P(features | spam)
ii. P(features | ham)
iii. P(spam)
iv. P(ham)

Which of these probabilities should add to 1?

A. i, ii
B. iii, iv
C. both A and B
D. neither A nor B

Estimating Parameters with Training Data

parameter: P(spam) estimate: $$\frac{\text{\# spam emails in training set}}{\text{size of training set}}$$

parameter: P(ham) estimate: $$\frac{\text{\# ham emails in training set}}{\text{size of training set}}$$

P(features | spam)
P(features | ham) ⟵ harder to estimate

Assumption of Conditional Independence

To estimate P(features | spam) and P(features | ham), we assume that the probability of a word appearing in an email of a given class is not affected by other words in the email.

$$P(x_1=0 \text{ and } x_2=1 \text{ and } x_3=1 \text{ and } \ldots \mid \text{spam}) = \xleftarrow{} \textbf{assumed equal}$$

$$P(x_1=0 \mid \text{spam})*P(x_2=1 \mid \text{spam})*P(x_3=1 \mid \text{spam})*\ldots$$

Is this a reasonable assumption?

Estimating Parameters with Training Data

$P(x_1=0 \text{ and } x_2=1 \text{ and } x_3=1 \text{ and } \ldots \mid \text{spam}) =$ $\longleftarrow$ **assumed equal**

$$\boxed{P(x_1=0 \mid \text{spam})} * P(x_2=1 \mid \text{spam}) * P(x_3=1 \mid \text{spam}) * \ldots$$

parameter: $P(x_1=0 \mid \text{spam})$

estimate:
$$\frac{\text{\# spam emails in training set not containing the first word in the dictionary}}{\text{\# spam emails in training set}}$$

Estimating Parameters with Training Data

$P(x_1=0 \text{ and } x_2=1 \text{ and } x_3=1 \text{ and } \ldots \mid \text{spam}) =$ $\longleftarrow$ **assumed equal**

$$P(x_1=0 \mid \text{spam}) * \boxed{P(x_2=1 \mid \text{spam})} * P(x_3=1 \mid \text{spam}) * \ldots$$

parameter: $P(x_2=1 \mid \text{spam})$

estimate: $\dfrac{\text{\# spam emails in training set containing the second word in the dictionary}}{\text{\# spam emails in training set}}$

Estimating Parameters with Training Data

$P(x_1=0 \text{ and } x_2=1 \text{ and } x_3=1 \text{ and } \ldots \mid \text{spam})$ **=** $\longleftarrow$ **assumed equal**

$P(x_1=0 \mid \text{spam}) * P(x_2=1 \mid \text{spam}) * \boxed{P(x_3=1 \mid \text{spam})} * \ldots$

parameter: $P(x_3=1 \mid \text{spam})$

estimate: $\dfrac{\text{\# spam emails in training set containing the third word in the dictionary}}{\text{\# spam emails in training set}}$

Can term-by-term estimate P(features|class)

Naive Bayes Spam Classifier: Recap

Bayes' Theorem shows how to calculate P(spam | features) and P(ham | features).

$$P(\text{class}|\text{features}) = \frac{P(\text{features}|\text{class}) * P(\text{class})}{P(\text{features})}$$

Rewrite the numerator, using the naive assumption of conditional independence of words given the class.

Estimate each term in the numerator based on the training data.

Select class based on whichever has the larger numerator.

Modifications and Extensions

- features are pairs (or longer sequences) of words rather than individual words
    - better captures dependencies between words
    - less naive
    - much bigger feature space
        - n words → n^2 pairs of words

- features are the number of occurrences of each word
    - captures low-frequency vs. high-frequency words

- smoothing
    - better handling of previously unseen words

Smoothing

$$P(x_1\mathit{=0} \text{ and } x_2\mathit{=1} \text{ and } x_3\mathit{=1} \text{ and } \ldots \mid \text{spam}) = P(x_1\mathit{=0} \mid \text{spam})*P(x_2\mathit{=1} \mid \text{spam})*P(x_3\mathit{=1} \mid \text{spam})*\ldots$$

<u>Dictionary</u>

1. a
2. aardvark
3. abacus
4. abandon
5. abate

...

Suppose you are classifying an email containing the word "abacus," which does not appear in any emails in your training data. For this new email's features, what is **P(features | spam)** according to a naive Bayes classifier?

A. P(features | spam) = undefined
B. P(features | spam) = 0
C. P(features | spam) = 1/n
D. P(features | spam) = 1

Smoothing

$$P(x_1=0 \text{ and } x_2=1 \text{ and } x_3=1 \text{ and } \dots \mid \text{ham}) = P(x_1=0 \mid \text{ham}) * P(x_2=1 \mid \text{ham}) * P(x_3=1 \mid \text{ham}) * \dots$$

Dictionary

1. a
2. aardvark
3. abacus
4. abandon
5. abate

...

Suppose you are classifying an email containing the word "abacus," which does not appear in any emails in your training data. For this new email's features, what is **P(features | ham)** according to a naive Bayes classifier?

A. P(features | ham) = undefined
B. P(features | ham) = 0
C. P(features | ham) = 1/n
D. P(features | ham) = 1

Smoothing

P(features | spam) = 0 and P(features | ham) = 0

Tiebreaker: randomly select one of the classes?

Smoothing

P(features | spam) = 0 and P(features | ham) = 0

Tiebreaker: randomly select one of the classes?

Better solution: make sure probabilities can’t be zero

Key idea: just because you’ve never seen something happen doesn’t mean it’s impossible

Estimating Parameters with Training Data

			Without Smoothing	With Smoothing
parameter:	P(spam)	estimate:	$\frac{\#spam}{\#spam + \#ham}$	$\frac{\#spam + 1}{\#spam + 1 + \#ham + 1}$
parameter:	P(ham)	estimate:	$\frac{\#ham}{\#spam + \#ham}$	$\frac{\#ham + 1}{\#spam + 1 + \#ham + 1}$

Estimating Parameters with Training Data

parameter: $P(x_i=1 \mid \text{spam})$

estimate:

$$\frac{\text{\#spam containing word i}}{\text{\#spam containing word i + \#spam not containing word i}}$$

Without Smoothing

$$\frac{\text{(\#spam containing word i) + 1}}{\text{(\#spam containing word i) + 1 + (\#spam not containing word i) + 1}}$$

With Smoothing

Similarly for other parameters $P(x_i=0 \mid \text{spam})$, $P(x_i=1 \mid \text{ham})$, $P(x_i=1 \mid \text{ham})$.

Smoothing

$P(x_1=0 \text{ and } x_2=1 \text{ and } x_3=1 \text{ and } \dots \mid \text{ham}) = P(x_1=0 \mid \text{ham}) * P(x_2=1 \mid \text{ham}) * P(x_3=1 \mid \text{ham}) * \dots$

Dictionary

1. a
2. aardvark
3. abacus
4. abandon
5. abate

...

Suppose you are classifying an email containing the word "abacus," which does not appear in any emails in your training data. What is **$P(x_3= 1 \mid \text{ham})$** according to a naive Bayes classifier ***with smoothing***?

A. $P(x_3= 1 \mid \text{ham}) = 0$
B. $P(x_3= 1 \mid \text{ham}) = 1/2$
C. $P(x_3= 1 \mid \text{ham}) = 1/(\text{total \#ham} +1)$
D. $P(x_3= 1 \mid \text{ham}) = 1/(\text{total \#ham} + 2)$
E. $P(x_3= 1 \mid \text{ham}) = 1/(\text{total \#ham} + \text{total \#spam}+ 2)$

Smoothing

Suppose your training set includes only six emails, all of which are ham. For a new email, how will we estimate P(ham)

	without smoothing?	**with smoothing?**
A.	P(ham) = 0	P(ham) = 1/8
B.	P(ham) = 0	P(ham) = 6/7
C.	P(ham) = 1	P(ham) = 1/8
D.	P(ham) = 1	P(ham) = 6/7
E.	P(ham) = 1	P(ham) = 7/8

Summary

- The Naive Bayes algorithm is useful for text classification.
- The bag of words model treats each word in a large dictionary as a feature.
- Smoothing is one modification that allows for better predictions when there are words that have never been seen before.