Module 24 – More Naive Bayes

DSC 40A, Summer 2023

Final is Friday from 8:00a-10:59a

- Final will be closed books/notes/electronics/web. You will be allowed to keep with you two A4-sized sheets (four sides) with any content you want.
- Leave all answers **unsimplified** in terms of permutations, combinations, factorials, exponents, etc.
- We will not answer questions during the exam. State your assumptions if anything is unclear.
- Location is slated in regular room, RWAC 0115.

Agenda

- Naive Bayes with smoothing.
- Application — text classification.

Naive Bayes with smoothing

Recap: Naive Bayes classifier

- We want to predict a class, given certain features.
- Using Bayes' theorem, we write

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

A B ...

- For each class, we compute the numerator using the **naive assumption of conditional independence of features given the class**.
- We estimate each term in the numerator based on the training data.
- We predict the class with the largest numerator.
  - Works if we have multiple classes, too!

Example: avocados

color	softness	variety	ripeness
bright green	firm	Zutano	unripe
green-black	medium	Hass	ripe
purple-black	firm	Hass	ripe
green-black	medium	Hass	unripe
purple-black	soft	Hass	ripe
bright green	firm	Zutano	unripe
green-black	soft	Zutano	ripe
purple-black	soft	Hass	ripe
green-black	soft	Zutano	ripe
green-black	firm	Hass	unripe
purple-black	medium	Hass	ripe

You have a soft green-black Hass avocado. Based on this data, would you predict that your avocado is ripe or unripe?

$p(\text{ripe} \mid \text{soft, green-black, hass}) = p(\text{ripe}) \cdot p(\text{soft} \mid \text{ripe}) \cdot p(\text{green-black} \mid \text{ripe}) \cdot p(\text{hass} \mid \text{ripe})$

$7/11 \cdot 4/7 \cdot 3/7 \cdot 5/7$

$p(\text{unripe} \mid \text{soft, green-black, hass}) = p(\text{unripe}) \cdot p(\text{soft} \mid \text{unripe}) \cdot p(\text{green-black} \mid \text{unripe}) \cdot p(\text{hass} \mid \text{unripe})$

$4/11 \cdot 0$

Uh oh...

- There are no soft unripe avocados in the data set.
- The estimate $P(\text{soft}|\text{unripe}) \approx \frac{\text{\# soft unripe avocados}}{\text{\# unripe avocados}}$ is 0.
- The estimated numerator, $P(\text{unripe}) \cdot P(\text{soft, green-black, Hass}|\text{unripe}) = P(\text{unripe}) \cdot P(\text{soft}|\text{unripe}) \cdot P(\text{green-black}|\text{unripe}) \cdot P(\text{Hass}|\text{unripe})$, is also 0.
- But just because there isn't a soft unripe avocado in the data set, doesn't mean that it's impossible for one to exist!
- **Idea:** Adjust the numerators and denominators of our estimate so that they're never 0.

Smoothing

- **Without** smoothing:

$$P(\text{soft}|\text{unripe}) \approx \frac{\#\text{ soft unripe}}{\#\text{ soft unripe} + \#\text{ medium unripe} + \#\text{ firm unripe}}$$

$$P(\text{medium}|\text{unripe}) \approx \frac{\#\text{ medium unripe}}{\#\text{ soft unripe} + \#\text{ medium unripe} + \#\text{ firm unripe}}$$

$$P(\text{firm}|\text{unripe}) \approx \frac{\#\text{ firm unripe}}{\#\text{ soft unripe} + \#\text{ medium unripe} + \#\text{ firm unripe}}$$

- **With** smoothing:

$$P(\text{soft}|\text{unripe}) \approx \frac{\#\text{ soft unripe} + 1}{\#\text{ soft unripe} + 1 + \#\text{ medium unripe} + 1 + \#\text{ firm unripe} + 1}$$

$$P(\text{medium}|\text{unripe}) \approx \frac{\#\text{ medium unripe} + 1}{\#\text{ soft unripe} + 1 + \#\text{ medium unripe} + 1 + \#\text{ firm unripe} + 1}$$

$$P(\text{firm}|\text{unripe}) \approx \frac{\#\text{ firm unripe} + 1}{\#\text{ soft unripe} + 1 + \#\text{ medium unripe} + 1 + \#\text{ firm unripe} + 1}$$

- When smoothing, we add 1 to the count of every group whenever we're estimating a conditional probability.

Example: avocados, with smoothing

color	softness	variety	ripeness
bright green	firm	Zutano	unripe
green-black	medium	Hass	ripe
purple-black	firm	Hass	ripe
green-black	medium	Hass	unripe
purple-black	soft	Hass	ripe
bright green	firm	Zutano	unripe
green-black	soft	Zutano	ripe
purple-black	soft	Hass	ripe
green-black	soft	Zutano	ripe
green-black	firm	Hass	unripe
purple-black	medium	Hass	ripe

You have a soft green-black Hass avocado. Using Naive Bayes, **with smoothing**, would you predict that your avocado is ripe or unripe?

$$p(\text{ripe} \mid \text{soft, green-black, hass}) = p(\text{ripe}) \cdot p(\text{soft} \mid \text{ripe}) \cdot p(\text{green-black} \mid \text{ripe}) \cdot p(\text{hass} \mid \text{ripe})$$

$$7/11 \cdot \frac{4+1}{7+3} \cdot \frac{3+1}{7+3} \cdot \frac{5+1}{7+2}$$

$$p(\text{unripe} \mid \text{soft, green-black, hass}) = p(\text{unripe}) \cdot p(\text{soft} \mid \text{unripe}) \cdot p(\text{green-black} \mid \text{unripe}) \cdot p(\text{hass} \mid \text{unripe})$$

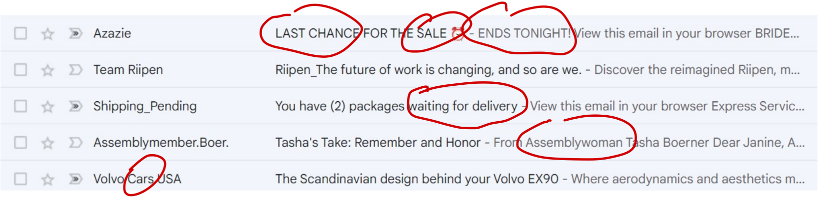
$$4/11 \cdot \frac{0+1}{4+3} \cdot \frac{2+1}{4+3} \cdot \frac{2+1}{4+2}$$

Text classification

Text classification

- Text classification problems include:
  - Sentiment analysis (e.g. positive and negative customer reviews).
  - Determining genre (news articles, blog posts, etc.).
  - Spam filtering.

Spam filtering



- **Our goal:** given the body of an email, determine whether it's **spam** or **ham** (not spam).
- **Question:** How do we come up with features?

Features

Idea:

- Choose a **dictionary** of d words.

- Represent each email with a **feature vector** $\vec{x}$:

$$\vec{x} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \dots \\ x^{(d)} \end{bmatrix} \begin{array}{l} \rightarrow \text{offer} \\ \rightarrow \text{limited} \\ \\ \rightarrow \text{sale} \end{array}$$

  where

  - $x^{(i)} = 1$ if word i is present in the email, and
  - $x^{(i)} = 0$ otherwise. "do I have this word or not?"

This is called the **bag-of-words** model. This model ignores the frequency and meaning of words.

Concrete example

- Dictionary: “prince”, “money”, “free”, and “just”.
- Dataset of 5 emails (red are spam, green are ham):
  - “I am the prince of UCSD and I demand money.”
  - “Tapioca Express: redeem your free Thai Iced Tea!”
  - “DSC 10: free points if you fill out CAPEs!”
  - “Click here to make a tax-free donation to the IRS.”
  - “Free career night at Prince Street Community Center.”

what do our feature vectors look like?

$$
\begin{array}{l}
\text{prince} \\
\text{money} \\
\text{free} \\
\text{just}
\end{array}
\quad
\underbrace{\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}}_{\text{email 1}}
\quad
\underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\text{email 2}}
\quad
\underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\text{email 3}}
\quad
\underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\text{email 4}}
\quad
\underbrace{\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\text{email 5}}
$$

Naive Bayes for spam classification

$$P(\text{class} \mid \text{features}) = \frac{P(\text{class}) \cdot P(\text{features} \mid \text{class})}{P(\text{features})}$$

ham spam presence of 4 words

- To classify an email, we'll use Bayes' theorem to calculate the probability of it belonging to each class:
  - $P(\text{spam} \mid \text{features})$.
  - $P(\text{ham} \mid \text{features})$.
- We'll predict the class with a larger probability.

Naive Bayes for spam classification

$$P(\text{class} \mid \text{features}) = \frac{P(\text{class}) \cdot P(\text{features} \mid \text{class})}{P(\text{features})}$$

- Note that the formulas for $P(\text{spam} \mid \text{features})$ and $P(\text{ham} \mid \text{features})$ have the same denominator, $P(\text{features})$.

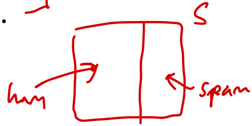
- Thus, we can find the larger probability just by comparing numerators:
  - $P(\text{spam}) \cdot P(\text{features} \mid \text{spam})$.
  - $P(\text{ham}) \cdot P(\text{features} \mid \text{ham})$.

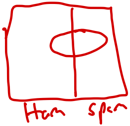
Naive Bayes for spam classification

Discussion Question

We need to determine four quantities:

1. $P(\text{features} \mid \text{spam})$.
2. $P(\text{features} \mid \text{ham})$.
3. $P(\text{spam})$.
4. $P(\text{ham})$.





Which of these probabilities should add to 1?

a) 1, 2

b) 3, 4

c) Both (a) and (b).

d) Neither (a) nor (b).

Estimating probabilities with training data

- To estimate $P(\text{spam})$, we compute

$$P(\text{spam}) \approx \frac{\#\text{ spam emails in training set}}{\#\text{ emails in training set}}$$

- To estimate $P(\text{ham})$, we compute

$$P(\text{ham}) \approx \frac{\#\text{ ham emails in training set}}{\#\text{ emails in training set}}$$

- What about $P(\text{features} \mid \text{spam})$ and $P(\text{features} \mid \text{ham})$?

Assumption of conditional independence

- Note that $P(\text{features} \mid \text{spam})$ looks like

$$\text{Sample} \rightarrow P(\underbrace{x^{(1)} = 0}_{\text{“prince” not included}}, \underbrace{x^{(2)} = 1}_{\text{“money” is included}}, \dots, \underbrace{x^{(d)} = 0}_{\text{“just” not included}} \mid \text{spam})$$

- Recall: the key assumption that the Naive Bayes classifier makes is that **the features are conditionally independent given the class**.

- This means we can estimate $P(\text{features} \mid \text{spam})$ as

$$P(x^{(1)} = 0, x^{(2)} = 1, \dots, x^{(d)} = 0 \mid \text{spam})$$
$$= P(x^{(1)} = 0 \mid \text{spam}) \cdot P(x^{(2)} = 1 \mid \underline{\text{spam}}) \cdot \ldots \cdot P(x^{(d)} = 0 \mid \text{spam})$$

among spam emails how many don't include “prince”

Among spam emails, how many do include “money”

and so on..

Concrete example

- Dictionary: “prince”, “money”, “free”, and “just”.
- Dataset of 5 emails (red are spam, green are ham):
  - **“I am the prince of UCSD and I demand money.”**
  - **“Tapioca Express: redeem your free Thai Iced Tea!”**
  - **“DSC 10: free points if you fill out CAPEs!”**
  - **“Click here to make a tax-free donation to the IRS.”**
  - **“Free career night at Prince Street Community Center.”**

Concrete example

- New email to classify: "Download a free copy of the Prince of Persia."'

$$
\begin{array}{c|ccccc|c}
 & \text{email 1} & \text{email2} & \text{email 3} & \text{email 4} & \text{email 5} & \\
\hline
\text{prince} & 1 & 0 & 0 & 0 & 1 & 1 \\
\text{money} & 1 & 0 & 0 & 0 & 0 & 0 \\
\text{free} & 0 & 1 & 1 & 1 & 1 & 1 \\
\text{just} & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

$$p(\text{spam} \mid \text{features}) = p(\text{spam}) \cdot p(x^{(1)}=1 \mid \text{spam}) \cdot p(x^{(2)}=0 \mid \text{spam}) \cdot p(x^{(3)}=1 \mid \text{spam}) \cdot p(x^{(4)}=0 \mid \text{spam})$$

$$= 2/5 \cdot 1/2 \cdot 1/2 \cdot 1/2 \cdot 2/2$$

$$p(\text{ham} \mid \text{features}) = p(\text{ham}) \cdot p(x^{(1)}=1 \mid \text{ham}) \cdot p(x^{(2)}=0 \mid \text{ham}) \cdot p(x^{(3)}=1 \mid \text{ham}) \cdot p(x^{(4)}=0 \mid \text{ham})$$

$$= 3/5 \cdot 1/3 \cdot 3/3 \cdot 3/3 \cdot 3/3$$

predict ham

Uh oh...

- What happens if we try to classify the email "just what's your price, prince"?

Smoothing

- **Without** smoothing:

$$P(x^{(i)} = 1 \mid \text{spam}) \approx \frac{\text{\# spam containing word } i}{\text{\# spam containing word } i + \text{\# spam not containing word } i}$$

- **With** smoothing:

$$P(x^{(i)} = 1 \mid \text{spam}) \approx \frac{(\text{\# spam containing word } i) + 1}{(\text{\# spam containing word } i) + \underline{1} + (\text{\# spam not containing word } i) + \underline{1}}$$

- When smoothing, we add 1 to the count of every group whenever we're estimating a conditional probability.

Concrete example with smoothing

- What happens if we try to classify the email “just what’s your price, prince”?

$$
\begin{array}{l}
\text{prince} \\
\text{money} \\
\text{free} \\
\text{just}
\end{array}
\underset{\text{email 1}}{\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}}
\underset{\text{email 2}}{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}}
\underset{\text{email 3}}{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}}
\underset{\text{email 4}}{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}}
\underset{\text{email 5}}{\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}}
\qquad
\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}
$$

$$
P(\text{spam} \mid \text{features}) = \underset{2/5}{P(\text{spam})} \cdot \underset{\frac{1+1}{2+2}}{P(x^{(1)}=1 \mid \text{spam})} \cdot \underset{\frac{1+1}{2+2}}{P(x^{(2)}=0 \mid \text{spam})} \cdot \underset{\frac{1+1}{2+2}}{P(x^{(3)}=0 \mid \text{spam})} \cdot \underset{\frac{0+1}{2+2}}{P(x^{(4)}=1 \mid \text{spam})}
$$

$$
P(\text{ham} \mid \text{features}) = \underset{3/5}{P(\text{ham})} \cdot \underset{\frac{1+1}{3+2}}{P(x^{(1)}=1 \mid \text{ham})} \cdot \underset{\frac{3+1}{3+2}}{P(x^{(2)}=0 \mid \text{ham})} \cdot \underset{\frac{0+1}{3+2}}{P(x^{(3)}=0 \mid \text{ham})} \cdot \underset{\frac{0+1}{3+2}}{P(x^{(4)}=1 \mid \text{ham})}
$$

Modifications and extensions

- **Idea:** Use pairs (or longer sequences) of words rather than individual words as features.
  - This better captures the dependencies between words.
  - It also leads to a much larger space of features, increasing the complexity of the algorithm.

Modifications and extensions

- **Idea:** Use pairs (or longer sequences) of words rather than individual words as features.
  - This better captures the dependencies between words.
  - It also leads to a much larger space of features, increasing the complexity of the algorithm.
- **Idea:** Instead of recording whether each word appears, record how many times each word appears.
  - This better captures the importance of repeated words.

Summary

Summary, next module

- Smoothing gives a way to make better predictions when a feature has never been encountered in the training data.
- The Naive Bayes classifier can be used for text classification, using the bag-of-words model.
- **Next module:** measuring performance of classifiers using precision and recall.

Summary, next module

- Smoothing gives a way to make better predictions when a feature has never been encountered in the training data.
- The Naive Bayes classifier can be used for text classification, using the bag-of-words model.
- **Next module:** measuring performance of classifiers using precision and recall.

Summary, next module

- Smoothing gives a way to make better predictions when a feature has never been encountered in the training data.
- The Naive Bayes classifier can be used for text classification, using the bag-of-words model.
- **Next module:** measuring performance of classifiers using precision and recall.