Lecture 24 – More Naive Bayes

DSC 40A, Spring 2023

Announcements

- Midterm 2 review session is tonight from 7-9pm in FAH 1101.
  - No groupwork, no attendance.
  - Come to ask questions about the mock exam posted on the course website.
  - You should do the exam on your own beforehand.
- Homework 7 is due **tomorrow at 11:59pm**. This is the **last homework!**

Midterm 2 is Monday during lecture

- You may use an unlimited number of handwritten note sheets for Midterm 2 (and Final Part 2). Start working on this now as you study!
- No calculators.
- Leave all answers **unsimplified** in terms of permutations, combinations, factorials, exponents, etc.
- Assigned seats will be posted on Campuswire.
- We will not answer questions during the exam. State your assumptions if anything is unclear.

Midterm 2 is Monday during lecture

- The exam will definitely include short-answer questions such as multiple choice or filling in the numerical answer to a probability or combinatorics question. Short-answer questions will be graded on correctness only, so you don't need to show your work or provide explanation for these questions.

- The exam may also include long-answer homework-style questions, which would require explanation and be graded with partial credit.

- Midterm 2 covers all material that was not covered on Midterm 1. Clustering is in scope, but the vast majority will be probability and combinatorics. This week's lectures are also in scope.

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

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

- 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?

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

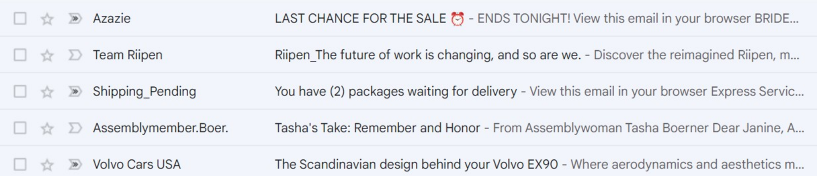
You have a soft green-black Hass avocado. Using Naive Bayes, **with smoothing**, would you predict that your avocado is ripe or unripe?

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** $\vec{x}$:

$$\vec{x} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \ldots \\ x^{(d)} \end{bmatrix}$$

  where
  - $x^{(i)} = 1$ if word i is present in the email, and
  - $x^{(i)} = 0$ otherwise.

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.”**

Naive Bayes for spam classification

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

- 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

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

- 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

$$\begin{aligned} &P(x^{(1)} = 0, x^{(2)} = 1, \ldots, x^{(d)} = 0 \mid \text{spam}) \\ =&P(x^{(1)} = 0 \mid \text{spam}) \cdot P(x^{(2)} = 1 \mid \text{spam}) \cdot \ldots \cdot P(x^{(d)} = 0 \mid \text{spam}) \end{aligned}$$

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.”‘

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:

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

- 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"?

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 time

- 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 time:** measuring performance of classifiers using precision and recall.