Lecture 17 – More Naive Bayes

DSC 40A, Fall 2021 @ UC San Diego

Suraj Rampure, with help from **many others**

Announcements

- Homework 8 is due **Friday 12/3 at 11:59pm**.
- No groupwork assignment, but we've posted a "Probability Review" worksheet on the course website that we'll take up in discussion section in-person on Wednesday.
  - Consists of past exam problems.
- Lecture on Thursday will be a high-level summary + combinatorics review problems.
- Fill out CAPEs + the End-of-Quarter survey. If 90% of the class does both, everyone gets 0.5% extra credit added to their final course grade.
- The Final Exam is on **Wednesday 12/8 from 11:30AM–2:30PM**.
  - You'll take the exam remotely by downloading a PDF from Gradescope and submitting your answers as a PDF by the deadline.
  - More logistical details to come.

Agenda

- Revisit Naive Bayes with a new application — text classification.
- Practical demo.

Recap: Naive Bayes classifier

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

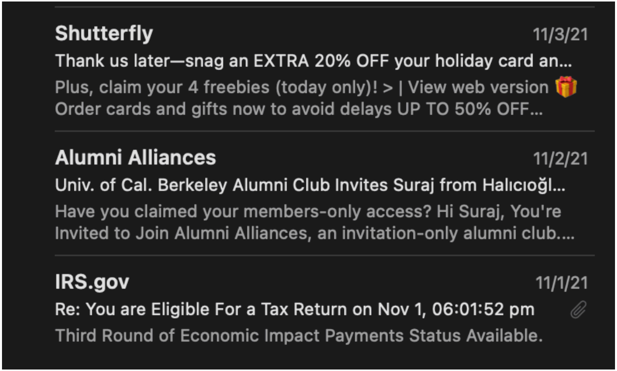
$$P(\text{class}|\text{features}) = \frac{P(\text{class}) \cdot P(\text{features}|\text{class})}{P(\text{features})}$$

- 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!

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.**
- **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, e.g. “prince”, “money”, “free”...

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

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

  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.

Concrete example

- Dictionary: "prince", "money", "free", and "xxx".

- 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 40A: free points if you fill out CAPEs!"**
  - **"Click here to make a tax-free donation to the IRS."**
  - **"Free COVID-19 tests at Prince 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

To answer, go to `menti.com` and enter 7053 7461.

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

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

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

Assumption of conditional independence

- Note that P(features | 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(features | 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 “xxx”.

- 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 40A: free points if you fill out CAPEs!”**
  - **“Click here to make a tax-free donation to the IRS.”**
  - **“Free COVID-19 tests at Prince 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 “xxx 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.
  - **Don't** smooth the estimates of unconditional probabilities (e.g. $P(\text{spam})$).

Concrete example with smoothing

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

Practical demo

Follow along with the demo by clicking the **code** link on the course website next to Lecture 17.

Summary

Summary, next time

- The Naive Bayes classifier can be used for text classification, using the bag-of-words model.
- **Next time:** brief high-level summary of the course + combinatorics practice problems.