Probability

Lecture 6

Centre for Data Science, ITER
Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha, India.

Contents

Introduction

- Probability is a way of quantifying the uncertainty associated with events chosen from some universe of events.
- Notationally, we write P(E) to mean “the probability of the event E.”

Dependence and independence of events

- Mathematically, we say that two events E and F are independent if the probability that they both happen is the product of the probabilities that each one happens:
  $P(E,F) = P(E)P(F)$
- For instance, if we flip a fair coin twice, knowing whether the first flip is heads gives us no information about whether the second flip is heads. These events are independent.
- On the other hand, knowing whether the first flip is heads certainly gives us information about whether both flips are tails. (If the first flip is heads, then definitely it's not the case that both flips are tails.) These two events are dependent.

Conditional Probability

- If two events E and F are not necessarily independent (and if the probability of F is not zero), then we define the probability of E "conditional on F" as:
  $P(E \mid F) = P(E, F)/P(F)$
- We can say that this is the probability that E happens, given that we know that F happens.
- We often rewrite this as:
  $P(E, F) = P(E \mid F)P(F)$

Conditional Probability (Contd.)

- When E and F are independent, you can check that this gives:
  $P(E \mid F) = P(E)$
  which is the mathematical way of expressing that knowing F occurred gives us no additional information about whether E occurred.

Bayes's Theorem

- Bayes's theorem is a way of "reversing" conditional probabilities.
- Let's say we need to know the probability of some event E conditional on some other event F occurring. But we only have information about the probability of F conditional on E occurring.
- Using the definition of conditional probability twice tells us that: $P(E \mid F) = P(E, F)/P(F) = P(F \mid E)P(E)/P(F)$

Bayes's Theorem

- The event F can be split into the two mutually exclusive events "F and E" and "F and not E." If we write -E for "not E" (i.e., "E doesn't happen"), then:
  $P(F) = P(F, E) + P(F, -E)$
  so that:
  P(E| $F) = P(F \mid E)P(E)/[P(F \mid E)P(E) + P(F \mid -E)P(-E)]$
  which is how Bayes's theorem is often stated.

Random Variables

- A random variable is a variable whose possible values have an associated probability distribution.
- Eg: A very simple random variable equals 1 if a coin flip turns up heads and 0 if the flip turns up tails.
- The **expected value** of a random variable, which is the average of its values weighted by their probabilities.
- Eg: The coin flip variable has an expected value of
  1/2 (= 0 * 1/2 + 1 * 1/2)
  and the range(10) variable has an expected value of 4.5.

Continuous Distributions

- A coin flip corresponds to a **discrete distribution**—one that associates positive probability with discrete outcomes.
- A **continuous distribution** describes the probabilities of the possible values of a continuous random variable i.e. a random variable which has infinite and uncountable set of possible values as number of outcomes.
- Eg: The **uniform distribution** puts equal weight on all the numbers between 0 and 1.

Probability Density Function

- Because there are infinitely many numbers between 0 and 1, this means that the weight it assigns to individual points must necessarily be zero.
- For this reason, we represent a continuous distribution with a probability density function (PDF) such that the probability of seeing a value in a certain interval equals the integral of the density function over the interval.
- The density function for the *uniform distribution* is just:

```
def uniform_pdf(x: float) -> float:
    return 1 if 0 <= x < 1 else 0
```

Cumulative Distribution Function

- We will often be more interested in the **cumulative distribution function** (CDF), which gives the probability that a random variable is less than or equal to a certain value.
- CDF for the *uniform distribution* will be:

```
def uniform_cdf(x: float) -> float:
    if x < 0: return 0
    elif x < 1: return x
    else: return 1
```

The Normal Distribution

- The normal distribution is the classic bell curve–shaped distribution and is completely determined by two parameters: its mean μ (mu) and its standard deviation σ (sigma).
- The mean indicates where the bell is centered, and the standard deviation how "wide" it is.
- It has the PDF: $f(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

Normal Distribution (Contd.)

- It can be implemented as:

```
import math
SQRT_TWO_PI = math.sqrt(2 * math.pi)
def normal_pdf(x: float, mu: float = 0, sigma: float = 1) -> float:
        return (math.exp(-(x-mu) ** 2 / 2 / sigma ** 2) / (SQRT_TWO_PI * sigma))
```

- When $\mu = 0$ and $\sigma = 1$, it's called the standard normal distribution.
- If Z is a standard normal random variable, then it turns out that: $X = \sigma Z + \mu$ is also normal but with mean μ and standard deviation σ.
- Conversely, if X is a normal random variable with mean μ and standard deviation σ,
  $Z = (X - \mu)/\sigma$ is a standard normal variable.

Normal Distribution (Contd.)

- The CDF for the normal distribution cannot be written in an “elementary” manner, but we can write it using Python’s `math.erf` error function:

```
def normal_cdf(x: float, mu: float = 0, sigma: float = 1) -> float:
        return (1 + math.erf((x - mu) / math.sqrt(2) / sigma)) / 2
```

The Central Limit Theorem

- If x1, ..., xn are random variables with mean μ and standard deviation σ, and if n is large, then:

  $$\frac{1}{n}(x_1 + x_2 + \ldots + x_n)$$

  is approximately normally distributed with mean μ and standard deviation $\frac{\sigma}{\sqrt{n}}$.
- Equivalently (but often more usefully),

  $$\frac{(x_1+x_2+\ldots+x_n)-\mu n}{\sigma\sqrt{n}}$$

  is approximately normally distributed with mean 0 and standard deviation 1.

Central Limit Theorem (Contd.)

- A Binomial(n,p) random variable is simply the sum of n independent Bernoulli(p) random variables, each of which equals 1 with probability p and 0 with probability 1 – p:

```
def bernoulli_trial(p:  float) -> int:
     return 1 if random.random() < p else 0

def binomial(n:  int, p:  float) -> int:
     return sum(bernoulli_trial(p) for _ in range(n))
```

Central Limit Theorem (Contd.)

- The mean of a Bernoulli(p) variable is p, and its standard deviation is $\sqrt{p(1-p)}$.
- The central limit theorem says that as n gets large, a Binomial(n,p) variable is approximately a normal random variable with mean $\mu = np$ and standard deviation $\sigma = \sqrt{np(1-p)}$.

References

[1] Data Science from Scratch: First Principles with Python by Joel Grus

Thank You
Any Questions?