question 2 views

Daily Challenge 21.2

(**Due: Wednesday 2/13 at 12:00 noon Eastern**)

And we're back!

(1) Monte Carlo integration.

In this chapter, we will extend the integral to get data about lengths and volumes in addition to areas. In particular, we will obtain a formula for the volume of an n-dimensional ball.

One way to perform integrals of complicated regions on a computer is Monte Carlo integration. You have already used this in your AP Computer Science course to estimate π. The idea of Monte Carlo integration is as follows:

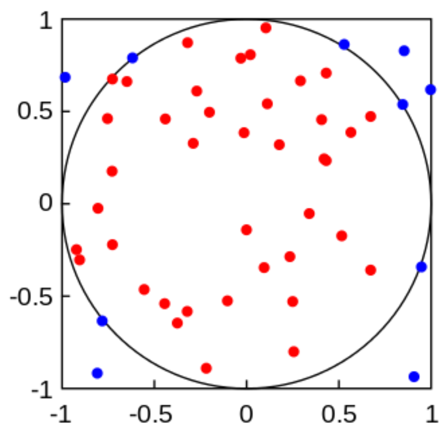
1. Suppose you wish you compute the area or volume of a region R.
2. Choose a larger region S that contains R, i.e. $R \subset S$, and such that $\mathrm{vol}(S)$ is known.
3. Generate a huge number N_{total} of random points that lie in the big region S, and count how many of these points lie in the small region R. Call this number n_{inside}.
4. The volume of R is approximately

$$\mathrm{vol}(R) = \frac{n_{\text{inside}}}{N_{\text{total}}} \cdot \mathrm{vol}(S).$$

For instance, you've used this to find the area of a circle. In this case, the regions were

$$R = \left\{(x, y) \mid x^2 + y^2 \le 1\right\},$$
$$S = \{(x, y) \mid -1 \le x, y \le 1\}.$$

In Python, you generated random numbers (x, y) lying between -1 and 1, then counted how many of them satisfied $x^2 + y^2 \le 1$.



Now let's generalize this to hyperspheres.

(2) Balls and volumes.

Define the *unit n-ball* B^n as follows:

$$B^n = \left\{(x_1, x_2, \cdots, x_n) \mid x_1^2 + x_2^2 + \cdots + x_n^2 \le 1\right\}.$$

For instance, the unit 1-ball is the closed interval $[-1, 1]$ (do you see why?). The length of this interval is 2, so the unit 1-ball has a volume of 2.

(When we use the word "volume" in an n-dimensional space, we always mean the appropriate n-dimensional volume. A one-dimensional volume is a length; a two-dimensional volume is an area; a three-dimensional volume is the usual notion of "volume" in everyday speech.)

The unit 2-ball is the circle of radius 1 in the plane, $B^2 = \left\{(x, y) \mid x^2 + y^2 \le 1\right\}$. Thus the 2-ball has a volume of πr^2 with $r = 1$, or simply $\mathrm{vol}(B^2) = \pi$.

The unit 3-ball is a solid sphere of radius 1 in three-dimensional space, which has a volume of $\frac{4}{3}\pi$.

Similarly, define the *unit n-cube C^n* as:

$$C^n = \{(x_1, \cdots, x_n) \mid -1 \leq x_i \leq 1 \text{ for all } i\}.$$

That is, the unit n-cube consists of all points $(x_1, \cdots, x_n)$ such that $-1 \leq x_1 \leq 1$, and $-1 \leq x_2 \leq 1$, and so on, up to $-1 \leq x_n \leq 1$.

The unit 1-cube is the closed interval $[-1, 1]$, with a volume of 2. The unit 2-cube is the square with volume $(1-(-1)) \cdot (1-(-1)) = 4$ The unit 3-cube is a solid cube with volume 8.

Can you see why the unit n-cube will always have volume 2^n?

(3) Problem: n-ball Monte Carlo integration.

You will compute the volume of a unit n-ball for any n in Python using Monte Carlo integration. Proceed as follows:

1. Write a function `n_ball_volume` which takes an integer `n` as input:

```
def n_ball_volume(n):
    '''
    Takes in an integer n and returns the approximate volume of a unit n-ball
    '''

    ## Your awesome code goes here!

    return volume
```

2. Pick a large number `n_samples` of sample points. For each sample point, generate n random numbers between -1 and 1, then put then into a list or array $(x_1, \cdots, x_n)$. Store your sample points in a list.

3. Count how many of the points in your list of samples satisfy $x_1^2 + \cdots + x_n^2 \leq 1$. For instance, you could create a counter, loop through the list, and increment the counter each time a point satisfies the condition. Call this final count something like `n_inside`.

4. Return the approximate volume $\frac{n_{\text{inside}}}{n_{\text{samples}}} \cdot 2^n$, since we recall from above that 2^n is the volume of the unit n-cube.

When you have implemented your code, do the following:

1. Output the volumes of the unit n-balls for $n = 1$ up to $n = 12$. Compare to the numbers in this table on Wikipedia (set $R = 1$ in their formulas). Debug if they disagree.

2. For which n is the volume of the unit n-ball **largest**? You should find it is maximized for $n = 5$. All higher-dimensional beings for $n > 5$ have tiny balls.

3. Upload your code and any supporting documents (e.g. a Jupyter notebook where you perform the check against Wikipedia) to Github.

daily_challenge

Updated 2 months ago by Christian Ferko

the students' answer, *where students collectively construct a single answer*

Pushing to github shortly. :GWqlabsKek:

Updated 2 months ago by Logan Pachulski

the instructors' answer, *where instructors collectively construct a single answer*

Click to start off the wiki answer

followup discussions *for lingering questions and comments*