STAT 672: Homework #1

Tom Wallace

February 9, 2018

Problem 1

A

See attached code `hw1_A.py` for an implementation of rejection sampling from the unit cube.

The acceptance probability is the volume of an n-dimensional ball divided by the volume of an n-dimensional cube:

$$P_{X\sim B^d_\infty}(||X||_2 \leq 1) = \frac{\mathrm{Vol}(B^d_2)}{\mathrm{Vol}(B^2_\infty)} = \frac{\frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)}}{2^d} \tag{1}$$

We know that $\Gamma(n+1) = n!$ Thus, $\Gamma(\frac{d}{2}+1) = (\frac{d}{2})!$

Stirling's approximation for factorials is useful here.

$$n! \approx \sqrt{2\pi n} n^n e^{-n} \tag{2}$$

Let us consider the case of $2d$. In this case:

$$\mathrm{Vol}(B^{2d}_2) = \frac{\pi^d}{\Gamma(d+1)} = \frac{\pi^d}{d!} \approx \frac{\pi^d}{\sqrt{2\pi d} d^d e^{-d}} \tag{3}$$

Rearranging terms, we have:

$$= \frac{1}{\sqrt{2\pi d}}\left(\frac{\pi e}{d}\right)^d \tag{4}$$

Referring to $\left(\frac{\pi e}{d}\right)^d$, the denominator grows (much) faster with d than does the numerator. As a consequence, if we extend d infinitely, the limit of the ratio is 0.

$$\lim_{d\to\infty}\left(\frac{\pi e}{d}\right)^d = 0 \tag{5}$$

Which implies:

$$\lim_{d\to\infty}\frac{1}{\sqrt{2\pi d}}\left(\frac{\pi e}{d}\right)^d = 0 \tag{6}$$

Returning to (1), we compute the the volume of the unit cube as $d \to \infty$.

$$\lim_{d\to\infty} 2^{2d} = \infty \tag{7}$$

So, the ratio of the unit sphere to the unit cube is:

$$\lim_{d\to\infty}\frac{\mathrm{Vol}(B^d_2)}{\mathrm{Vol}(B^2_\infty)} = \frac{0}{\infty} = 0 \tag{8}$$

Since this ratio also is the probability of accepting a sample in a rejection sampling scheme, our expected runtime to generate a single random vector from B^d_2 with rejection sampling grows asymptotically with d. For example, for $d = 10000$, our program would run for a very, very, *very* long time.

B

We can generate Θ as $\Theta = Z/||Z||_2$, where $Z \sim N(0, I_d)$.
The square root of sums of squares of independent standard normal random variables is the chi distribution.

$$Z_1^2 + Z_2^2 + ...Z_n^2 = \chi_n^2 \tag{9}$$

C

The two methods implemented in **A** and **B**—rejection and polar sampling, respectively— can be empirically compared. Each program was used to generate 100 samples with $d = 10$. The execution runtime was measured with the Linux `time` command, e.g. `time python3 hw1_1A.py`. Method B is significantly faster than Method A, as shown in Table 1.

Table 1: Runtime comparison (seconds)

	A	B
real	2.131	0.744
user	2.150	0.744
sys	0.172	0.192

Problem 2

A

Below are the results of my simulation, the code of which is attached as `hw1_B.py`. As d grows larger, the frequency of the event $\{||X_+ - Z||_2 \geq ||X_- - Z||_2\}$ increases. This is somewhat counter-intuitive. Because Z is drawn from the same distribution as X_+, we would expect their difference $X_+ - Z$ to result in something close to a zero-vector, which should have a smaller Euclidean norm than that of $X_- - Z$ (since these two have different means), resulting in *low* frequency of the event. Yet, as dimensionality grows, this is not the case: the event occurs with *high* frequency. This has three explanations. One, because the mean is $\pm 5/\sqrt{d}$, as d increases, the means increasingly converge towards 0. Two, X_+ and Z have larger variance than X_-, and this variance does not decrease with d. Thus, as d increases, the larger variance of X_+ and Z predominates over their increasingly similar means, resulting in them more often being further apart (their difference has a larger Euclidean norm) than X_- and Z. Three, the curse of dimensionality means the probability that any two vectors are far apart increases with dimensionality. This magnifies the effect of variance, which as previously discussed is greater for X_+ and Z.

Figure 1: Frequency of event