STAT 672: Homework #1

Tom Wallace

February 11, 2018

Problem 1

A

See submitted code `hw1_1A.py`.

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

$$P_{X\sim B_\infty^d}(||X||_2 \le 1) = \frac{\text{Vol}(B_2^d)}{\text{Vol}(B_\infty^2)} = \frac{\frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)}}{2^d} \tag{1}$$

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:

$$\text{Vol}(B_2^{2d}) = \frac{\pi^d}{\Gamma(d+1)} = \frac{\pi^d}{d!} \approx \frac{\pi^d}{\sqrt{2\pi d} d^d e^{-d}} \tag{3}$$

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 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:

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

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, i.e. the probability of acceptance, is:

$$\lim_{d\to\infty} \frac{\text{Vol}(B_2^d)}{\text{Vol}(B_\infty^2)} = \frac{0}{\infty} = 0 \tag{8}$$

This finding has severe consequences for our expected run time, which will greatly increase with d. For example, with $d = 10,000$, our program would run for a very long time.

B

See submitted code `hw1_1B.py`.

We can generate Θ using the formula $Z/||Z||_2$, where $Z \sim \mathcal{N}(0, \mathbf{I}_d)$. Let us examine why. The distribution of Z is invariant to rotations about the origin. That is, for any orthogonal matrix $\mathbf{Q}$, $\mathbf{Q}Z \sim \mathcal{N}(0, \mathbf{I}_d)$. Proof: for any $Y \sim \mathcal{N}(\mu, \mathbf{\Sigma})$, and matrix $\mathbf{C}_{p\times n}$ with rank p, $\mathbf{C}Y \sim \mathcal{N}_p(\mathbf{C}\mu, \mathbf{C\Sigma C'})$. So, for any orthogonal matrix $\mathbf{Q}$, $\mathbf{Q}Y \sim \mathcal{N}(\mathbf{Q}\mu, \mathbf{Q\Sigma Q'})$. In the case of Z, $\mu = 0$ and $\mathbf{\Sigma} = \mathbf{I}$, and so $\mathbf{QZ} \sim \mathcal{N}(0\mathbf{Q}, \mathbf{QIQ'})$. Obviously, $0\mathbf{Q} = 0$, and by the respective definitions of identity and orthogonal matrices, $\mathbf{QIQ'} = \mathbf{QQ'} = \mathbf{I}$. So, $\mathbf{Q}Z \sim \mathcal{N}(0, \mathbf{I})$. That Z is invariant to rotation about the origin tells us that its distribution is d-spherical, which is the shape we want to sample from. Our remaining task then is to ensure that the norm of our sample is 1. By the definition of the norm, $||aZ||_2 = |a| \times ||Z||_2$. So, $||\frac{Z}{||Z||}|| = \frac{1}{||Z||} \times ||Z|| = 1$. In conclusion, because the distribution of Z is spherical, and $||Z/||Z||_2||_2 = 1$, we are confident that $Z/||Z||_2$ is a sample from the surface of the d-dimensional unit sphere.

We can generate $||X||_2 = R \sim u^{1/d}$, where u is uniform on [0,1], using inversion sampling. Let us examine why. We know the CDF of R: $F = P(R \leq t) = t^d$ for $0 \leq t \leq 1$. We find the inverse CDF F^{-1} by solving $F(F^{-1}(u)) = u$, leading to $F^{-1}(u) = u^{1/d}$. $F^{-1}(u)$ has F as its CDF. Proof: $P(F^{-1}(u) \leq t) = P(u \leq F(t)) = F(t)$, since for random uniform [0,1] u, $P(u \leq x) = x$. So, sampling via $u^{1/d}$ is equivalent to sampling from the desired distribution with CDF t^d.

C

The methods implemented in **A** and **B** 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. Method B is faster than Method A, as shown in Table 1.

Table 1: Runtime comparison (seconds)

	A	**B**
real	2.131	0.343
user	2.150	0.329
sys	0.172	0.170

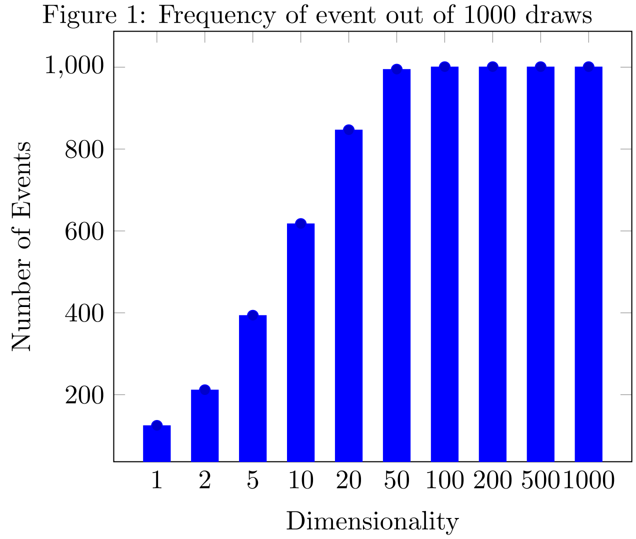
Problem 2

A

See submitted code `hw1_2A.py`.

As d grows larger, the frequency of the event $\{||X_+ - Z||_2^2 \geq ||X_- - Z||_2^2\}$ increases, as depicted in Figure 1. This result 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. This is true in low dimensions, but as dimensionality grows, the event occurs with *high* frequency. Why this counter-intuitive result occurs is explained in **B**.

Figure 1: Frequency of event out of 1000 draws



B

$$E[||X_+ - Z||_2^2]$$
$$= E[||X_+||_2^2] - 2E[X_+^T Z] + E[||Z||_2^2]$$
$$E[||X_+||_2^2] = ||\mu_+||_2^2 + tr(\Sigma_+) = 25 + 4d$$
$$E[||Z||_2^2] = E[||X_+||_2^2] = 25 + 4d$$
$$E[X_+^T Z] = E[X_+^T]E[Z] = ||\mu_+||^2 = 25$$
$$E[||X_+ - Z||_2^2] = 2(25 + 4d) - 2(25) = \boxed{8d}$$

$$E[||X_- - Z||_2^2]$$
$$= E[||X_-||_2^2] - 2E[X_-^T Z] + E[||Z||_2^2]$$
$$E[||X_-||_2^2] = ||\mu_-||_2^2 + tr(\Sigma_-) = 25 + d$$
$$E[||Z||_2^2] = 25 + 4d$$
$$E[X_-^T Z] = E[X_-^T]E[Z] = -25$$
$$E[||X_- - Z||_2^2] = (25 + d) - 2(-25) + (25 + 4d) = \boxed{100 + 5d}$$

This theoretical result explains our empirical observations in **A**. With low dimensionality, $8d < 100 + 5d$, and so our event occurs with low frequency. However, $8d$ grows faster with d than does $100 + 5d$, and so in high dimensions, our event occurs with high frequency. In essence, concentration of measure / curse of dimensionality magnify the effect of X_+'s higher variance as dimensionality grows higher. The different rate of growth of expected value with d is visualized in Figure 2.

Figure 2: Change in expected value with d