Assignment 1, due February 12 or February 19

Corrections: Feb. 12: Exercise 2, $f_n = \frac{t^n e^{-t}}{n!}$ corrected to $f_{n+1} = \frac{t^n e^{-t}}{n!}$. Exercise 3, $f(x) = \frac{2}{\pi}\sin(\pi x)$ corrected to $f(x) = \frac{\pi}{2}\sin(\pi x)$.

1. (*This exercise has two purposes. One is to understand why a sampler might not work well in high dimensions. Another is to understand why some functions have good Gaussian approximations.*) Suppose C_n is the side length 2 cube in n dimensions centered at the origin. This may be written $C_n = [-1,1]^n$. For $x \in \mathbb{R}^n$, we have $x \in C_n$ if $|x_k| \leq 1$ for all k. You can generate a "random" point $X \in C_n$ by taking $X_k = 2U_k - 1$, where the U_k are independent standard uniforms. The n dimensional volume of C_n is 2^n, so the probability density of X is 2^{-n} if $x \in C_n$ and zero otherwise. Let B_n be the unit ball in n dimensions. We have $x \in B_n$ if $(x_1^2 + \cdots + x_n^2)^{1/2} \leq 1$. Clearly $B_n \subset C_n$. We can generate X uniformly distributed in B_n by generating X uniformly distributed in C_n and accepting X if $X \in B_n$. The efficiency of this algorithm is the ratio of the volumes
$$Z_n = \frac{\mathrm{vol}(B_n)}{\mathrm{vol}(C_n)} \; .$$
This exercise derives an approximate formula for Z_n that shows that $Z_n \to 0$ as $n \to \infty$ exponentially. This makes the sampling method impractical for large n. An exercise from the Week 1 notes suggests a different sampler that is practical for large n. (It is possible to find the large n behavior of $I(n)$ in (1) using a change of variables $r^2/2 = s$ to express it in terms of the Γ function, whose asymptotics are available on wikepedia – *Stirling's formula*. Please don't do it this way. The asymptotics of Γ are found using the method of this problem, so that approach is not actually easier.)

   (a) The *unit sphere* in n dimensions is $S_{n-1} = \{|x| = 1\}$. The "surface area" (or $n-1$ dimensional volume) of S_{n-1} is ω_{n-1}. Show that
   $$\mathrm{vol}(B_n) = \frac{\omega_{n-1}}{n} \; .$$
   You can do this by
   $$\mathrm{vol}(B_n) = \int_{x \in B_n} dx$$
   using polar coordinates, which involves $\omega_{n-1} r^{n-1} dr$.

(b) Show that

$$\omega_{n-1} = \frac{(2\pi)^{n/2}}{I(n)} ,$$

where

$$I(n) = \int_0^\infty r^{n-1} e^{-r^2/2}\, dr . \tag{1}$$

Hint: integrate

$$\int_{x\in\mathbb{R}^n} e^{-|x|^2/2}\, dx$$

in polar coordinates.

(c) Write $I(n) = \int e^{-\phi(r)}\, dr$ and identify ϕ. Show that ϕ has a unique maximum value achieved at r_*. Calculate $\phi''(r_*)$, $\phi'''(r_*)$, and possibly one more. Let $q(r)$ be the quadratic Taylor approximation to $\phi(r)$ about r_*, which is

$$q(r) = \phi(r_*) + \tfrac{1}{2}\phi''(r_*)(r - r_*)^2 . \tag{2}$$

Write the formula for

$$J(n) = \int_{-\infty}^{\infty} e^{-q(r)}\, dr .$$

(d) $J(n)$ is an approximation of $I(n)$. The error is written $K(n) = I(n) - J(n)$. Show that

$$\frac{K(n)}{I(n)} \to 0 \quad \text{as } n \to \infty .$$

Hint: there are two kinds of r values: those where the quadratic approximation (2) is accurate, and those where ϕ and q are much smaller than values that matter. For this exercise, you can take the "values that don't matter" set to be $|r - r_*| > n^p$ with $0 < p < \frac{1}{6}$. When $|r - r_*| = n^p$, then e^{-q} does not matter, and $q(r)$ is still relatively close to $\phi(r)$ (use ϕ''' to verify this).

(e) Write the large n asymptotic approximation of Z_n that shows that sampling uniformly in the ball by rejection from the cube is an exponentially bad idea.

2. (*Direct samplers often are for probability distributions that depend on a parameter. It is not enough that the sampler "works" for each parameter value. It must be efficient uniformly over the parameter. This Exercise is an example of such a sampler.*) Suppose T_n is the $n-th$ arrival time in a Poisson process with rate constant $\lambda = 1$. The goal is to find a sampler that samples T_n using an amount of work that is bounded as $n \to \infty$. A direct simulation $T_n = S_1 + \cdots + S_n$ (the S_k are i.i.d. exponentials) takes order n work.

(a) Show that the probability density for T_{n+1} is $f_{n+1}(t) = \frac{t^n}{n!}e^{-t}$ if $t \geq 0$. Hint: $T_{n+1} = T_n + S_{n+1}$.

(b) Determine the behavior of $f_n(t)$ for typical T_n values using the method of Exercise 1. Find the most likely value of T_n by maximizing f_n, then make a Gaussian approximation of f_n about this value, t_{n*}.

(c) Explain why it is not a good idea to use the Gaussian approximation as a proposal distribution for rejection sampling of f_n.

(d) Explore using a *double exponential* as a proposal distribution. That is $g_n(t) = \frac{1}{Z}e^{-\alpha_n|t-t_{n*}|}$. Calculate the normalization constant Z. Do not worry about negative T values. Those are rare for large n, and can be rejected for any n.

(e) What formula for α_n is suggested by the Gaussian approximation (same power of n in typical $T_n - t_{n*}$, same power of n in the variance)?

(f) Determine wether this α_n leads to a sampler whose efficiency does not go to zero as $n \to \infty$. If so, you are done. If not, can you adjust α_n to make the sampler uniformly efficient?

3. (*Programming exercise. Please read the material on the class web site on programming conventions. When you modify and re-use posted code, please keep the automation features, such as making plots automatically with computational parameters and legends. If you add a computational parameter, figure out how to make it appear in the plot. If you remove a parameter, make it disappear from the plot. Update the* `makefile` *to keep everything automated.*) Download the file `Week1.tar`. This is an archive with several files for the assignment. Save it in some directory. In the UNIX command line, `cd` to that directory and unpack using the command `tar -xvf Week1.tar`. (Type `man tar` to see what `x`, `v`, and `f` mean, or google "unix tar".) The individual files will appear. Then type `make fTest`. A lot of things should happen, but eventually a picture should pop up that looks like `Week1.pdf` on the assignment page. You may have to surf the web to see how the Unix command `make` works. You will need to know at least a little because you will be adding another C++ procedure, which will not be compiled unless you add it to the `CPP_SOURCES` list.

Modify the code to sample the density $f(x) = \frac{\pi}{2}\sin(\pi x)$ for $x \in [0,1]$, and $f(x) = 0$ otherwise. Use rejection sampling with proposal distribution $g(x) = 6x(1-x)$ as described in the notes. Here is a suggested sequence of steps. If you are not used to the Unix command line, there may be lots of "learning curve" involved in some of them.

(a) The proposal distribution is sampled using procedures presently in the file `f.cpp`. You need to copy this to `g.cpp` and change the names of the routines to be g instead of f. It should be clear how to do this. You also need to adjust `header.h`. If you do this correctly and run the code again, you should get the same plot, except that it will be called g.

(b) Now modify `f.cpp` to do the rejection sampling using g as a trial. Change everything that needs changing, including the string that describes the distribution. Test it using the histogram procedure. If you can, put both the f and g target distribution curves in the plot, so you can see that you have changed from f to g. Use a sample size that makes it clear that the empirical histogram represents f, not g.