Btry 694 Class Homework 2

9/17/98

1 Problem 2.1

Dennis Carlson

Referring to example 2.1.4 show that:

(a) The *arcsine distribution* with density $f(x) = \frac{1}{\pi}\sqrt{x(1-x)}$ is invariant under the transform $y = 1 - x$, that is, $f(x) = f(y)$.

(b) Ths uniform distribution $\mathcal{U}_{[0,1]}$ is invariant under the "tent" transform.

$$D(x) = \begin{cases} 2x & \text{if } x \leq 1/2 \\ 2(1-x) & \text{if } x > 1/2 \end{cases}$$

(c) Check these two properities via a computer by plotting the histograms of successive iterates.

Solutions:

a) We have that $f(x) = \frac{1}{\pi}\sqrt{x(1-x)}$. Consider $y = h(x) = 1 - x$. This is a strictly decreasing function of x, and we therefore have that:

$f(y) = f(h^{-1}(x)) \cdot |\frac{dx}{dy}| = \frac{1}{\pi}\sqrt{(1-y)(1-(1-y))} \cdot |-1| = \frac{1}{\pi}\sqrt{y(1-y)}$.

That is, the arcsine distribution is invariant under the transform $1 - x$.

We wish to show that D is distributed $\mathcal{U}_{[0,1]}$. Now, for d between 0 and 1,

$$
\begin{aligned}
P(D \leq d) &= P(2x \leq d, x \leq \frac{1}{2}) + P(2(1-x) \leq d, x > \frac{1}{2}) \\
&= P(x \leq \frac{1}{2})P(2x \leq d | x \leq \frac{1}{2}) + P(x > \frac{1}{2})P(2(1-x) \leq d | x > \frac{1}{2}) \\
&= \frac{1}{2}(d + d) \\
&= d
\end{aligned}
$$

¿From this we conclude that D is distributed $\mathcal{U}_{[0,1]}$.

c) On seperate page.

2 Problem 2.15

Haiyue Song

Detail the properties of the algorithm based on the Central Limit Theorem for a sequence of uniform $U_1, U_2, \ldots, U_n$ for $n = 12$, $n = 48$ and $n = 96$. (Consider in particular the range of the generated variable.)

Solution:

The algorithm based one Central Limit Theorem is stated as below. If $U_1, U_2, \ldots, U_n$ are independent and identically distributed,

$$Y = \sqrt{n}\frac{\bar{U} - \mu}{\sigma} = \sqrt{12n}(\bar{U} - \frac{1}{2}) \sim N(0, 1).$$

(a) As $U_i's$ are from Uniform(0,1) the biggest value U_i can take is 1, and the smallest value is 0. Hence, the random variable $Y \in (-\sqrt{3n}, \sqrt{3n})$, however, the N(0,1) has the range $(-\infty, \infty)$. Therefore, first we know the random variable Y generated based on this algorithm has a finite range which depends on n. For example if $n = 12$, $n = 48$ and $n = 96$, then the range will be (-6,6), (-12,12) and $(-12\sqrt{2}, 12\sqrt{2})$, respectively.

(b) Now, we consider the moments of the random variable Y, generated by the algorithm based on Central Limit Theorem. As we know the moment generating function of a Uniform(0,1) Random Variable U is

$$M_U(t) = E(e^{Ut}) = \frac{e^t - 1}{t},$$

hence,

$$\begin{aligned} M_Y(t) &= E(e^{Yt}) = E(e^{\sqrt{12n}(\bar{U}-\frac{1}{2})t}) \\ &= \frac{1}{(2t)^n}(\frac{n}{3})^{\frac{n}{2}} e^{-\sqrt{3n}t}(e^{2\sqrt{\frac{3t}{n}}} - 1)^n. \end{aligned}$$

(c) Using *Mathematica* to calculate $m_i = M_Y^{(i)}(t)|_{t=0}$ and $i = 1, \ldots, 4$, it turns $m_1 = 0$ and $m_2 = 1$, i.e., expectation and variance of Y don't depend on n. And the 3rd central moment (skewness),$m_3 = 0$, so Y is symmetrically distributed. The 4th central moment (kurtosis), $m_4 = 3 - \frac{5}{n}$, which depends on n. Because the 4th central moment of a N(0,1) is 3, which is greater than $3 - \frac{5}{n}$, Y has a lighter tail than Standard Normal Random Variables.

So we conclude that the random variable, Y, generated from the algorithm based on the Central Limit Theorem, is symmetric and approximately normal distributed with mean 0 and variance 1. But the distribution of Y has a lighter tail than standard normal, because Y has a finite range depending on n.

3 Problem 2.19

Matt Tom

This derivation can be found on page 211 of red Resnick. Suppose that we have a sequence of exponential random variables X_i each with distribution $F(x) = 1 - e^{-\lambda x}$. Then the c.d.f. of the sum $X_1 + \ldots + X_n$ is the convolution of n $F(x)$'s. Thus,

$$F_{X_1+\ldots+X_n}(x) = F^{n*}(x) = 1 - e^{-x\lambda} \sum_{k=0}^{n-1} \frac{(x\lambda)^k}{k!}.$$

So, the probability that $X_1 + \ldots + X_n \leq 1 \leq X_1 + \ldots + X_{n+1}$ is

$$\begin{aligned} F^{n*}(1) - F^{(n-1)*}(1) &= 1 - e^{-\lambda} \sum_{k=0}^{n-1} \frac{(\lambda)^k}{k!} - \left(1 - e^{-\lambda} \sum_{k=0}^{n} \frac{(\lambda)^k}{k!}\right) \\ &= e^{-\lambda} \frac{(\lambda)^n}{n!} = P_\lambda(N = k) \end{aligned}$$

which is what we wanted.

Let RND be a function which gives a random number from $Unif(0,1)$. We have the following algorithm for generating a Poisson distribution:

```
output = −1;
sum = 0;
while (sum <= 1) {
    output = output + 1;
```

```
        sum = sum - log(RND)/lambda;
    }
    return(output);
```

Now we can optimize. We're taking sums of logarithms ($\log(u_1) + \log(u_2)$, $\log(u_1) + \log(u_2) + \log(u_3)$, etc.). It would just be easier (as in, it would save computation time) to multiply the u_i together first and then take the log.

The real question is: when do we get out of this loop? We get out when:

$$\begin{aligned}
-\sum \frac{\log(u_i)}{\lambda} &> 1 \\
-\sum \log(u_i) &> \lambda \\
-\lambda &> \sum \log(u_i) \\
e^{-\lambda} &> \prod u_i
\end{aligned}$$

This gives rise to the algorithm suggested by the problem:

```
output = -1;
p = 1;
mark = exp(-1 * lambda);
while (p >= mark) {
    output = output + 1;
    p = p * RND;
}
```

```
return(output);
```

4 Problem 2.29

Bárbara Patricia González

Show that a negative binomial random variable is a mixture of Poisson distributions, that is, if $Y \sim Ga(n,(1-p)/p)$ and $X|y \sim P(y)$, then $X \sim Neg(n,p)$. Use this fact to write an algorithm for simulating negative binomial random variables.

Since $Y \sim Ga(n,(1-p)/p)$ and $X|y \sim P(y)$, then

$$f_Y(t;n,(1-p)/p) = \frac{1}{(n-1)!}\frac{(1-p)^n}{p^n}y^{n-1}e^{-\frac{(1-p)y}{p}}$$

and

$$f_{X|y}(x;y) = e^{-y}\frac{y^x}{x!}.$$

So we have that

$$f_X(x;n,p) = \int_0^\infty f_{X|y}(x;y)f_Y(y;n,(1-p)/p)dy = \int_0^\infty e^{-y}\frac{y^x}{x!}\frac{1}{(n-1)!}\frac{(1-p)^n}{p^n}y^{n-1}e^{-\frac{(1-p)}{p}y}dy =$$

$$= \frac{(x+n-1)!p^{x+n}}{x!(n-1)!}\frac{(1-p)^n}{p^n}\int_0^\infty \frac{1}{p^{x+n}(x+n-1)!}y^{x+n-1}e^{-\frac{y}{p}}dy = \frac{(x+n-1)!}{x!(n-1)!}p^x(1-p)^n =$$
$$= C_{x+n-1,x}p^x(1-p)^n.$$

So we have that $X \sim Neg(n,p)$.

Using this fact we can write the following algorithm to generate $X \sim Neg(n,p)$:

- Generate $Y \sim Ga(n, (1-p)/p)$ (for example with algorithm A.7 page 54)
- Generate $X \sim P(Y)$ (for example with algorithm A.8 page 54)

5 Problem 2.35

Baojun Song

The target density is

$$G_a(\alpha, \beta) = \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} e^{-x/\beta}.$$

where α is not necessarily a integer. But one can use accept-reject algorithem based on the Gamma family $G_a([\alpha], \beta)$, where $[\alpha]$ is the integer part of α. i.e. our instrument density is from this family

$$G_a([\alpha], \beta') = \frac{1}{\Gamma([\alpha])\beta'^{[\alpha]}} x^{[\alpha]-1} e^{-x/\beta'}.$$

We will first derive the best β' from this family. According to the idea of accept-reject algorithem, we have to find out the maximum value of $\frac{G_a(\alpha,\beta)}{G_a([\alpha],\beta')}$.

$$\frac{G_a(\alpha, \beta)}{G_a([\alpha], \beta')} = \frac{\frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} e^{-x/\beta}}{\frac{1}{\Gamma([\alpha])\beta'^\alpha} x^{[\alpha]-1} e^{-x/\beta'}} = \frac{\Gamma([\alpha])}{\Gamma(\alpha)} \frac{\beta'^{[\alpha]}}{\beta^\alpha} x^{\alpha-[\alpha]} e^{-x(1/\beta-1/\beta')}$$

we only need to study the function

$$f(x) = \frac{\beta'^{[\alpha]}}{\beta^\alpha} x^{\alpha-[\alpha]} e^{-x(1/\beta-1/\beta')}$$

If $\beta' < \beta$, $f(x)$ is a monotonically increasing function and $M = sup f(x) = \infty$.Accept-reject algorithem is not avilabe for this situation. Geometrically,if $\beta' < \beta$, $G_a(\alpha, \beta)$ has thicker tail than $G_a([\alpha], \beta)$. Thus β' is necessary greater than β as an instrumental density.

$$f'(x) = \frac{\beta'^{[\alpha]}}{\beta^{\alpha}} x^{\alpha-[\alpha]-1} e^{-x(1/\beta-1/\beta')}(\alpha - [\alpha] - x(1/\beta - 1/\beta'))$$

set $f'(x) = 0$, the critical point is $x^* = \frac{\beta\beta'(\alpha-[\alpha])}{\beta'-\beta}$. In fact, we can check $f(x)$ attends the maximum value at x^*.

$$\begin{aligned} M(\beta') = supf(x) = f(x^*) &= \frac{\beta'^{[\alpha]}}{\beta^{\alpha}} (\frac{\beta\beta'(\alpha - [\alpha])}{\beta' - \beta})^{\alpha-[\alpha]} e^{\alpha-[\alpha]} \\ &= \frac{(e(\alpha - [\alpha]))^{\alpha-[\alpha]}}{\beta^{[\alpha]}} \frac{\beta'^{\alpha}}{(\beta' - \beta)^{\alpha-[\alpha]}} \end{aligned}$$

We must choose such β' that $M(\beta')$ minimize. Consider function $h(\beta') = \frac{\beta'^{\alpha}}{(\beta'-\beta)^{\alpha-[\alpha]}}$ is enough.

$$h'(\beta') = \beta'^{\alpha-1}(\beta'-\beta)^{-\alpha+[\alpha]-1}(\alpha(\beta'-\beta)+([\alpha]-\alpha)\beta') = \beta'^{\alpha-1}(\beta'-\beta)^{-\alpha+[\alpha]-1}(-\alpha\beta+[\alpha]\beta')$$

Therefore $h(\beta')$ reaches the minimum value at $\beta' = \frac{\alpha}{[\alpha]}\beta$. This β' gives the same mean for both target varible and instrumental varible. To compare the efficiency of $G_a(\alpha, \beta')$ and $G_a(\alpha, \beta'')$, we just compare $h(\beta')$ with $h(\beta'')$. If $h(\beta'') = h(\beta')$, there is no difference. If $h(\beta'') < h(\beta')$, $G_a(\alpha, \beta'')$ is more efficiency than $G_a(\alpha, \beta'')$. Conversely, if $h(\beta'') > h(\beta')$, $G_a(\alpha, \beta')$ is more efficiency than $G_a(\alpha, \beta'')$

Based on the property of function $h(\beta')$, we can move on further for the comparison. $h(\beta')$ decreases if $\beta' < \frac{\alpha}{[\alpha]}\beta$ and increses if $\beta' > \frac{\alpha}{[\alpha]}\beta$. So the larger is better if both β' and β are less than $\frac{\alpha}{[\alpha]}\beta$; the smaller one is more efficiency if β' and β are greater than $\frac{\alpha}{[\alpha]}\beta$. If $\frac{\alpha}{[\alpha]}\beta$ is in between β' and β, we have to clculate $h(\beta')$ and $h(\beta)$ to compare them.

6 Problem 2.38 - Accept-Reject Algorithm

Mitch Warachka

Want to generate a normal(0,1) on the positive real axis, density f, from an exponential distribution with parameter λ, density g. The first step is to compute the value of M. In this case, the normal density is not properly normalized and an extra factor of 2 will be used so that this density integrates to 1.

$$\begin{aligned} M = \sup \frac{f}{g} &= \frac{2}{\sqrt{2\pi}} \exp \frac{-x^2}{2} \frac{1}{\lambda} \exp \frac{-x}{\lambda} \\ &= \sqrt{\frac{2}{\pi}} \lambda \exp \frac{-x^2}{2} + \frac{x}{\lambda} \end{aligned}$$

differentiating this function, after taking logs to simplify, with respect to x to find the maximum while ignoring the constant in front gives an x equal to $\frac{1}{\lambda}$.

$$\begin{aligned} \frac{\partial}{\partial x} \left(\frac{-x^2}{2} + \frac{x}{\lambda} \right) = 0 \\ &\Rightarrow \frac{-2x}{2} + \frac{1}{\lambda} = 0 \\ &\Rightarrow x = \frac{1}{\lambda} \end{aligned}$$

Therefore, substituting this value of x into the equation for M gives

$$M = \sqrt{\frac{2}{\pi}} \lambda exp^{\frac{1}{2\lambda^2}} \tag{1}$$

Now, the M is obtained in terms of the parameter λ of the exponential distribution. This M can be minimzed by differentiating with respect to λ to obtain the smallest M. First take logs and ignore the constant part to simplify matters

$$\begin{aligned} \frac{\partial}{\partial \lambda}\left(log(\lambda) + \frac{1}{2\lambda^2}\right) &= \frac{1}{\lambda} - \frac{1}{\lambda^3} = 0 \\ &\Rightarrow 1 - \frac{1}{\lambda^2} = 0 \\ &\Rightarrow \lambda = 1 \end{aligned}$$

inserting 1 for λ into the equation for M gives equal to $\sqrt{\frac{2}{\pi}} exp^{\frac{1}{2}}$ or 1.3155.

The following Matlab routine store the number of observations kept as **count** and the values of each of those observations, **Y**. Simulations were done for n= 1000, 5000, and 25000 with λ =1 and M=1.3155. Results are below and on attached graphs. The mean and standard deviation for the three samples were .8397, .7941, .7902 and .5758, .6071, .5898 respectively. The true mean and standard deviation are .7979 and .6028.

7 Problem 2.39

Daniel Fink

For the generation of a Cauchy random variable compare the inversion method with the quotient of normals (QONS) where normals are generated using the efficient Box-Muller transformation.

7.1 Cauchy from Uniform

$\theta \sim U(-.5\pi, .5\pi)$ and let $y = g(\theta) = tan(\theta)$. This transformation is 1-1: $\theta = g^-1(y) = tan^-1(y)$, so the derivation of the pdf is straightforward.

$$\begin{aligned} f_Y(y) &= f_U(y)|J| \\ &= \frac{1}{\pi}\frac{1}{1+y^2} \end{aligned}$$

where $-\infty < y < \infty$. So, $Y \sim Cauchy(0, 1)$.

7.2 Cauchy from Quotient of Normals

$X_1, X_2 \sim N(0, 1)$ Let $Y_1 = X_1/X_2$ and $Y_2 = X_2$. This transformation is 1-1 with $X_1 = Y_1Y_2$ and $X_2 = Y_2$ and $|J| = |Y_2|$. Our objective it to find the marginal pdf of Y_1.

$$f_{Y_1,Y_2}(y_1, y_2) = f_{X_1,X_2}(y_1y_2, y_2)|J|$$

$$
\begin{aligned}
&= \frac{1}{2\pi} exp(-\frac{y_2^2}{2}(y_1^2+1))|y_2| \\
&= \frac{y_2}{2\pi} exp(-\frac{y_2^2}{2}(y_1^2+1)) \text{ if } y_2 >= 0 \\
&= \frac{-y_2}{2\pi} exp(-\frac{y_2^2}{2}(y_1^2+1)) \text{ if } y_2 < 0
\end{aligned}
$$

Integrating Y_2 out of the joint pdf,

$$f_{Y_1}(y_1) = \int_{-\infty}^{\infty} f_{Y_1,Y_2}(y_1, y_2) dy_2 \tag{2}$$

$$= \int_0^{\infty} \frac{y_2}{2\pi} exp(-\frac{y_2^2}{2}(y_1^2+1)) dy_2 = \frac{1}{2\pi} \frac{1}{1+y_1^2} \tag{3}$$

where $y_2 >= 0$ and

$$= \int_{-\infty}^{0} \frac{-y_2}{2\pi} exp(-\frac{y_2^2}{2}(y_1^2+1)) dy_2 = \frac{1}{2\pi} \frac{1}{1+y_1^2} \tag{4}$$

where $y_2 < 0$. So $f_{Y_1}(y_1)$ is

$$f_{Y_1}(y_1) = \frac{1}{\pi} \frac{1}{1+y_1^2} \tag{5}$$

7.3 Comparison

We assume that the same uniform random generator is used for all purposes discussed here. We assume that Normal variates are generated with the efficient Box-Muller method. Consequently, the generation each normal variate will require the generation of more than 1 uniform deviate on average. This is due to the restriction that $S = U_1^2 + U_2^2 <= 1$. Therefore, the generation of a single Cauchy deviate via QONS (quotient of normals) will require more the computation of more than two uniforms on average. On the other hand, generating Cauchy deviates via inversion requires the computation of exactly one uniform on average. This would seem to favor inversion over QONS. Formally, the comparison of overall efficiency comes down to the question of how much slower the computation of the arctan is on average compared to the natural logarithm, compared to the computation of the uniform deviates. A simulation confirmed the expected- QONS is about half the speed of inversion.

8 Problem 2.40

Søren Hvidkjær, JGSM For the Accept-Reject algoritm of $\mathcal{G}a(n,1)$ (Erlang) distribution, based on the $\mathcal{E}xp(\lambda)$ distribution, determine the optimal value of λ.

First, determine the ratio of the target density to the instrumental density:

$$\frac{f}{g_\lambda} = \frac{1}{\Gamma(n)} x^{n-1} \lambda^{-1} e^{(\lambda-1)x}, \tag{6}$$

where $1/\Gamma(n)$ is a normalising constant, f is the density of the $\mathcal{G}a(n,1)$ distribution, and g_λ is the density of the $\mathcal{E}xp(\lambda)$ distribution. Then, find the bound for the ratio, by differentiating wrt. x

$$\begin{aligned} \frac{\partial}{\partial x} &= (n-1)x_0^{n-2}\lambda^{-1}e^{(\lambda-1)x_0} + x_0^{n-1}\lambda^{-1}e^{(\lambda-1)x_0}(\lambda-1) = 0 \\ x_0 &= \frac{n-1}{1-\lambda}, \end{aligned}$$

for $n > 1$ and $\lambda < 1$. Also, the second order derivative reduces to

$$\frac{\left(\frac{n-1}{1-\lambda}\right)^n (\lambda-1)^3}{(n-1)^2} < 0,$$

confirming the max. Insert the solution for x in (6) to determine the bound M

$$M = \frac{1}{\Gamma(n)} \left(\frac{n-1}{(1-\lambda)e}\right)^{n-1} \lambda^{-1}. \tag{7}$$

Finally, to minimize the bound, we maximize $\lambda(1-\lambda)^{n-1}$, yielding the solution $\lambda = 1/n$. Inserting the solution for λ in (7) yields

$$M = \frac{\left(\frac{n}{e}\right)^{(n-1)} n}{\Gamma(n)},$$

and so we can plot the bound as a function of n:

9 Problem 2.42

Krishanu Maulik

(a) : The density of *natural* exponential family with respect to the measure ν is

$$f_1(x) = \exp\{x \cdot \theta - \psi(\theta)\}.$$

So the logarithm of the density is

$$\log f_1(x) = x \cdot \theta - \psi(\theta).$$

This is linear in x and hence is concave (but not *strictly* concave). Thus the *natural* exponential family has log-concave densities.

(b) : The density of the logistic distribution is

$$f_2(x) = \frac{1}{\beta} \frac{\exp\{-(x-\alpha)/\beta\}}{[1+\exp\{-(x-\alpha)/\beta\}]^2}.$$

So the logarithm of the density is

$$\log f_2(x) = -\log\beta - \frac{x-\alpha}{\beta} - 2\log[1+\exp\{-(x-\alpha)/\beta\}].$$

Then

$$\begin{aligned} \frac{d}{dx}\log f_2(x) &= -\frac{1}{\beta} + \frac{2}{\beta}\frac{\exp\{-(x-\alpha)/\beta\}}{[1+\exp\{-(x-\alpha)/\beta\}]} \\ &= -\frac{1}{\beta} + \frac{2}{\beta}[1 - \frac{1}{1+\exp\{-(x-\alpha)/\beta\}}] \end{aligned}$$

hence

$$\frac{d^2}{dx^2}\log f_2(x) = -\frac{2}{\beta^2}\frac{\exp\{-(x-\alpha)/\beta\}}{[1+\exp\{-(x-\alpha)/\beta\}]^2},$$

which is negative and hence logistic density is log-concave.

(c) : The density of the Gumbel distribution is

$$f_3(x) = \frac{k^k}{(k-1)!}\exp\{-kx - ke^{-x}\},$$

where the parameter k is a natural number. So the logarithm of the density is

$$\log f_3(x) = k\log k - \log\{(k-1)!\} - kx - ke^{-x}.$$

Then

$$\frac{d}{dx}\log f_3(x) = -k + ke^{-x}$$

hence

$$\frac{d^2}{dx^2}\log f_3(x) = -ke^{-x},$$

which is negative and hence logistic density is log-concave.

(d) : The density of the *generalized inverse Gaussian* distribution is

$$f_4(x) \propto x^{\alpha}e^{-\beta x - \alpha/x}.$$

So the logarithm of the density is

$$\log f_4(x) = constant + \alpha\log x - \beta x - \alpha/x.$$

Then

$$\frac{d}{dx}\log f_3(x) = \alpha/x - \beta + \alpha/x^2$$

hence

$$\frac{d^2}{dx^2}\log f_3(x) = -\alpha/x^2 - 2\alpha/x^3,$$

which is negative and hence *generalized inverse Gaussian* density is log-concave.