Example sheet 2

Working with distributions

Network Performance—DJW—2011/2012

Question 1. I have taken a sample of n values, $X_1, \ldots, X_n$. For each value in the sample I know the value of an associated predictor variable $w_1, \ldots, w_n$. I believe that $X_i \sim \text{Exp}(\lambda w_i)$, where λ is unknown. Calculate the maximum likelihood estimator for λ.

Question 2. I have taken a series of measurements of file sizes, and plotted their empirical distribution function. Based on my plot, I propose to fit the distribution

$$\log \mathbb{P}(X \geq x) = \begin{cases} -\lambda x & \text{if } x \leq 1024 \\ -\lambda x - \mu(x - 1024) & \text{if } x > 1024. \end{cases}$$

Show that this distribution has density function

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \leq 1024 \\ (\lambda + \mu) e^{-(\lambda+\mu)x + 1024\mu} & \text{if } x > 1024. \end{cases}$$

(i) Find formulae for the maximum likelihood estimators of λ and μ.
(ii) Give pseudocode for a random number generator that generates random samples from this distribution.

Question 3. Let X be the sum of three throws of a dice. The possible outcomes are $\Omega = \{3, 4, \ldots, 18\}$. Find the mean, median, and standard deviation of X.

Question 4. (i) Let X be an Exponential random variable with parameter λ. Let $Y = aX$, for some constant $a > 0$. Calculate the distribution function, i.e. find $\mathbb{P}(Y \geq y)$ as a function of y. What is the common name for the distribution of Y?
(ii) Let $X_1, X_2, \ldots, X_n$ be independent Exponential random variables with parameters $\lambda_1, \lambda_2, \ldots, \lambda_n$ respectively. Let $Z = \min(X_1, X_2, \ldots, X_n)$. Calculate the distribution function for Z. Show that $Z \sim \text{Exp}(\lambda_1 + \cdots + \lambda_n)$.
(iii) Let $X_1, \ldots, X_n$ be as above, and let $Z = \max(X_1, X_2, \ldots, X_n)$. Calculate the distribution function for Z. *[Hint. First find $\mathbb{P}(Z < z)$.]*

Question 5. Let X, Y and Z be generated from the following three random number generators respectively:

```
def rexp(λ): return -1.0/λ * math.log(random.random())
def rpareto(α): return math.pow(random.random(), -1.0/α)
def rpareto2(α,m): return m*(1-1.0/α)*rpareto(α)
```

Y is called the Pareto distribution, $Y \sim \text{Pareto}(\alpha)$. Given λ, find α and m such that Z has the same mean and variance as X.

Question 6. Consider the log file of the `www.wischik.com` website, available on Moodle.
(i) The `Size` column contains the size in bytes of the body of each http response. Plot the empirical distribution function (EDF) of `Size`.
(ii) It has been suggested that `Size+1` has the Pareto(α) distribution for some parameter α. Fit this distribution.

(iii) If `Size+1` $\sim$ Pareto(α) then `Size` $\sim$ Pareto(α) -1, i.e. we may generate a random http response size by generating a Pareto(α) random variable and subtracting 1. Generate a random sample in this way, using your fitted value for α, and superimpose its EDF on your plot from part (i).

(iv) It has also been suggested that `Size+1` has a lognormal distribution, i.e. that log(`Size`$+1$) $\sim$ Normal(μ,σ^2) for some parameters μ and σ. Fit this distribution. Generate a random sample of http response sizes based on this fit, and superimpose its EDF on your plot from part (i).

(v) Which of the two distributions looks to be a better fit?

Question 7. This quesion concerns the arrival process of requests to `www.wischik.com`. We wish to know if the arrival process is Poisson, i.e. if interarrival times are independent and exponentially distributed. Since arrival rates vary according to time of day and day of the week, restrict attention to records which apply to weekday afternoons, 2pm–4pm.

(i) Plot the EDF of interarrival time. Fit an exponential distribution, and plot its distribution function on the same graph. Do they agree?

(ii) A better method is to transform the scales of your EDF plot, so that if the interarrival times truly are exponential then the EDF should follow a straight line. Does it?

(iii) Split interarrival times into pairs, and produce a scatter-plot of the first time against the second time. Does it seem that successive interarrival times are independent?

(iv) Another way to visualize independence is as follows. Split the data set of interarrival times into three classes, depending on whether the preceding interarrival time was short, medium or large. (Choose the cutoff points so that the three classes have roughly the same number of data points.) Plot the EDF for each of the three classes. Are they the same?

Question 8. On the next page there are six different generators[1] for sequences of random variables, intended to be used as request interarrival times for a simulator of a web server. The first, `rexp(λ)`, generates a sequence of independent Exp(λ) random variables; the others were submitted by students, and are intended to represent bursty arrivals. Suppose the interarrival times are $X_1, X_2, \ldots$. Then we can calculate the mean arrival rate by finding

$$\lim_{n\to\infty} \frac{n}{\mathbb{E}(X_1+X_2+\cdots+X_n)}.$$

For each of the generators listed below, find a formula for the mean arrival rate. You should validate your formula by using a computer to generate a reasonably long sequence $X_1,\ldots,X_n$ and computing $n/(X_1+\cdots+X_n)$; repeat the computation for large enough values of n to make you confident you have computed an accurate answer.

Example. For generator `bursty1(λ,w)`, the code generates the sequence $X_1 = Y_1$, $X_2 = Y_2+\cdots+Y_w$, $X_3 = Y_{w+1}$, $X_4 = Y_{w+2}+\cdots+Y_{2w}$ and so on, where each Y_i is Exp(λ). Therefore $\mathbb{E}X_1 = 1/\lambda$, $\mathbb{E}X_2 = (w-1)/\lambda$, and so on. Thus

$$\mathbb{E}(X_1+\cdots+X_n) = \begin{cases} \frac{n}{2}(w/\lambda) & \text{if } n \text{ even} \\ \frac{n-1}{2}(w/\lambda)+1/\lambda & \text{if } n \text{ odd.} \end{cases}$$

When n is large, $\mathbb{E}(X_1+\cdots+X_n)/n \to w/(2\lambda)$. Hence the mean arrival rate is $2\lambda/w$. I found close agreement when I validated this formula by running

```
for n in [1000,10000,100000]:
    g = bursty1(1,5)
    x = [g.next() for i in range(n)]
    print 'n={n}, avg.rate={r}, theory={t}'.format(n=n,r=len(x)/sum(x),t=2.0/5)
```

[1] For an explanation of generators in Python, and why they are useful for generating sequences of random variables, see `http://www.cs.ucl.ac.uk/staff/D.Wischik/Teach/NP/Handouts/pythonic.html`.

```
import math,random

def rexp(λ):
    while True: yield -1.0/λ * math.log(random.random())

def bursty1(λ,waittime):
    count = 0
    x = 0
    while True:
        x = x + (-1.0/λ * math.log(random.random()))
        if (count % waittime) in [0,waittime-1]:
            yield x
            x = 0
        count = count+1

def bursty2(λ):
    burst,add,curr = 0,0,0
    while True:
        burst += 1
        if burst==3:
            add,burst = curr*λ,0
        else:
            add = 0
        curr = (-1.0/λ * math.log(random.random()))
        yield curr+add

def bursty3(λ,p=2):
    r1 = λ*(p+1)/2.0
    r2 = r1/p
    while True:
        yield -1.0/r1 * math.log(random.random())
        yield -1.0/r2 * math.log(random.random())

def bursty4(λ):
    a = False
    while True:
        a = not a
        r = λ*3/2.0 if a else λ*3/4.0
        yield -1.0/r * math.log(random.random())

def bursty5(λ,fr,bi,bl=2):
    while True:
        if random.random()<=fr:
            for i in range(bl): yield bi
        else:
            for i in range(bl): yield -1.0/λ * math.log(random.random())
```