Homework #1 STA410H1F/2102H1F

due Friday October 9, 2020 at 11:59pm

Instructions: Solutions to problems 1 and 2 are to be submitted via Quercus and only PDF files will be accepted. You are strongly encouraged to do problems 3 and 4 but these are not to be submitted for grading.

1. Suppose that Z is an $m \times n$ pixel image where $m = 2^k$ and $n = 2^\ell$. If H_m and H_n are $m \times m$ and $n \times n$ Hadamard matrices then we can define the Walsh-Hadamard tranform of Z by

$$\widehat{Z} = H_m Z H_n.$$

In this problem, we will explore using this Walsh-Hadamard transform to denoise an image.

(a) Show that $Z = H_m \widehat{Z} H_n/(mn)$. (This gives the inverse Walsh-Hadamard transform.)

(b) To denoise an image using the Walsh-Hadamard transform, we first compute $\widehat{Z}$ and then perform hard- or soft-thresholding to obtain a thresholded (or shrunken) transform $\widehat{Z}^*$ (which will typically be "sparser" than $\widehat{Z}$). Our hope is that the components of $\widehat{Z}$ corresponding to noise in the image are eliminated and so the denoised image can be obtained by applying the inverse transform to $\widehat{Z}^*$.

Using an R function `fwht2d` (available on Quercus) to compute the Walsh-Hadamard transform, write R functions to perform both hard- and soft-thresholding (dependent on a parameter λ). This function can be used to compute the W-H transform and its inverse as follows:

```
> xhat <- fwht2d(x) # x is a 2^k x 2^k matrix
> xx <- fwht2d(xhat)/ncol(xhat)^2 # inverse transform
```

In the code above, the matrices `x` and `xx` will be identical (up to round-off error).

(c) The file `design.txt` contains a noisy 256×256 pixel grayscale image of a design . Its entries are numbers between 0 and 1 where 0 represents black and 1 represents white. The data can be read into R and displayed as an image as follows:

```
> design <- matrix(scan("design.txt"),ncol=256,byrow=T)
> colours <- grey(seq(0,1,length=256))
> image(design, axes=F, col=colours)
```

Using your functions from part (b), try to denoise the image as best as possible. Note that this process is quite subjective – what you should see is that as you reduce the noise in the image, you will inevitably introduce bias, which is manifested by a reduction in contrast in the denoised image. The challenge is to find an "optimal" balance between the two.

2. Suppose that $X_1, X_2, \cdots$ is an infinite sequence of independent identically distributed random variables with some distribution function F and N has a Geometric distribution with

$$P(N = n) = (1-\theta)\theta^n \quad \text{for } n = 0, 1, 2, \cdots$$

(for some $0 < \theta < 1$) where N is independent of the sequence $\{X_i\}$. Then we can define a *compound Geometric* random variable Y by

$$S = \sum_{i=1}^{N} X_i$$

where $S = 0$ if $N = 0$. Compound distributions arise naturally in risk theory and insurance – for example, if N represents the number of claims and $X_1, X_2, \cdots$ the amounts paid for each claim then S is the total sum paid. For the purposes of risk management, it is useful to know the distribution of S, particularly its tail.

(a) Suppose that $\{X_i\}$ are discrete integer-valued random variables with probability generating function $\phi(t) = E(t^{X_i})$. Show that the probability generating function of S is $g(\phi(t))$ where $g(t)$ is the probability generating function of N, which is given by

$$g(t) = \frac{1-\theta}{1-\theta t}$$

provided that $\theta t < 1$ or $t < 1/\theta$. (Hint: Write

$$E(t^S) = \sum_{n=0}^{\infty} E(t^S | N = n) P(N = n)$$

and note that given $N = n$, $S = X_1 + \cdots + X_n$.)

(b) The distribution of S can be approximated using the Discrete Fourier Transform. Assume that the random variables $\{X_i\}$ have a distribution $p(x)$ on the integers $0, 1, \cdots, \ell$. The complication is that, unlike the distribution of $X_1 + \cdots + X_n$, the distribution of S is not concentrated on a finite set of integers. Therefore, we need to find an integer M such that $P(S \geq M)$ is smaller than some pre-determined threshold ϵ; M will depend on the parameter θ as well as the integer ℓ (or more precisely, the distribution $p(x)$). (Also to optimize the FFT algorithm, M should be a power of 2 or a product of small prime numbers although this isn't absolutely necessary unless M is very large.)

Show that if $P(N \geq m) \leq \epsilon$ then $P(Y \geq m\ell) \leq \epsilon$ and so we can take $M \geq m\ell$.

(c) The bound M determined in part (b) is typically very conservative (i.e. too large) and can be decreased substantially. One approach to determining a better bound is based on the probability generating function of S derived in part (a) and Markov's inequality. Specifically, if $\theta^{-1} > \phi(t) > 1$ (in which case $t > 1$), we have

$$P(S \geq M) = P(t^S \geq t^M) \leq \frac{E(t^S)}{t^M} = \frac{1}{t^M}\left(\frac{1-\theta}{1-\theta\phi(t)}\right).$$

Use this fact to show that for $P(Y \geq M) < \epsilon$, we can take

$$M = \inf_{1<\phi(t)<\theta^{-1}} \frac{\ln(1-\theta) - \ln(1-\theta\phi(t)) - \ln(\epsilon)}{\ln(t)}.$$

(d) Given M (which depends on ϵ), the algorithm for determining the distribution of S goes as follows:

1. Evaluate the DFT of $\{p(x) : x = 0, \cdots, M-1\}$:

$$\widehat{p}(j) = \sum_{x=0}^{M-1} \exp\left(-2\pi\iota \frac{j}{M} x\right) p(x)$$

   where $p(\ell+1) = \cdots = p(M-1) = 0$.

2. Evaluate $g(\widehat{p}(j)) = (1-\theta)/(1-\theta\widehat{p}(j))$ for $j = 0, \cdots, M-1$.

3. Evaluate $P(S = s)$ by computing the inverse FFT:

$$P(S = s) = \frac{1}{M} \sum_{j=0}^{M-1} \exp\left(2\pi\iota \frac{s}{M} j\right) g(\widehat{p}(j))$$

Write an R function to implement this algorithm where M is determined using the method in part (c) with $\epsilon = 10^{-5}$. Use this function to evaluate the distribution of S in the case where $\theta = 0.9$ and the distribution of $\{X_i\}$ is

$$p(x) = P(X_i = x) = \binom{10}{x}\left(\frac{1}{2}\right)^{10} \quad \text{for } x = 0, \cdots, 10$$

and the probability generating function for each X_i is $\phi(t) = (1+t)^{10}/2^{10}$.
(You do not need to evaluate the bound M from part (c) with great precision; for example, a simple approach is to take a discrete set of points $\mathcal{T} = \{1 < t_1 < t_2 < \cdots < t_k\}$ (with $\phi(t_k) < 1/\theta$) and define

$$M = \min_{t \in \mathcal{T}} \frac{\ln(1-\theta) - \ln(1-\theta\phi(t)) - \ln(\epsilon)}{\ln(t)}$$

where $\delta = t_{i+1} - t_i$ and t_k are determined graphically (that is, by plotting the appropriate function) so that you are convinced that the value of M is close to the actual infimum.)

Supplemental problems (not to hand in):

3. As noted in lecture, catastrophic cancellation in the subtraction $x - y$ can occur when x and y are subject to round-off error. Specifically, if $\mathrm{fl}(x) = x(1+u)$ and $\mathrm{fl}(y) = y(1+v)$ then

$$\mathrm{fl}(x) - \mathrm{fl}(y) = x - y + (xu - yv)$$

where the absolute error $|xu - yv|$ can be very large if both x and y are large; in some cases, this error may swamp the object we are trying to compute, namely $x - y$, particularly if $|x - y|$ is relatively small compared to $|x|$ and $|y|$. For example, if we compute the sample variance using the right hand side of the identity

$$\sum_{i=1}^{n}(x_i - \bar{x})^2 = \sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2, \tag{1}$$

a combination of round-off errors from the summations and catastrophic cancellation in the subtraction may result in the computation of a negative sample variance! (In older versions of Microsoft Excel, certain statistical calculations were prone to this unpleasant phenomenon.) In this problem, we will consider two algorithms for computing the sample variances that avoid this catastrophic cancellation. Both are "one pass" algorithms, in the sense that we only need to cycle once through the data (as is the case if we use the right hand side of (1)); to use the left hand side of (1), we need two passes since we need to first compute $\bar{x}$ before computing the sum on the left hand side of (1). In parts (a) and (b) below, define $\bar{x}_k$ be the sample mean of $x_1, \cdots, x_k$ and note that

$$\bar{x}_{k+1} = \frac{k}{k+1}\bar{x}_k + \frac{1}{k+1}x_{k+1}$$

with $\bar{x} = \bar{x}_n$.

(a) Show that $\sum_{i=1}^{n}(x_i - \bar{x})^2$ can be computed using the recursion

$$\sum_{i=1}^{k+1}(x_i - \bar{x}_{k+1})^2 = \sum_{i=1}^{k}(x_i - \bar{x}_k)^2 + \frac{k}{k+1}(x_{k+1} - \bar{x}_k)^2$$

for $k = 1, \cdots, n-1$. (This is known as West's algorithm.)

(b) A somewhat simpler one-pass method replaces $\bar{x}$ by some estimate x_0 and then corrects for the error in estimation. Specifically, if x_0 is an arbitrary number, show that

$$\sum_{i=1}^{n}(x_i - \bar{x})^2 = \sum_{i=1}^{n}(x_i - x_0)^2 - n(x_0 - \bar{x})^2.$$

(c) The key in using the formula in part (b) is to choose x_0 to avoid catastrophic cancellation, that is, x_0 should be close to $\bar{x}$. How might you choose x_0 (without first computing $\bar{x}$) to minimize the possibility of catastrophic cancellation? Ideally, x_0 should be calculated using $o(n)$ operations.

(An interesting paper on computational algorithms for computing the variance is "Algorithms for computing the sample variance: analysis and recommendations" by Chan, Golub, and LeVeque; this paper is available on Quercus.)

4. Consider the Geometric distribution given in Problem 2 with $P(N = n) = (1 - \theta)\theta^n$ for $n = 0, 1, \cdots$.

(a) Show that the distribution function of N is

$$F(x) = P(N \le x) = 1 - \theta^{n+1} \quad \text{for } n \le x < n + 1$$

for $x \ge 0$ and integers $n = 0, 1, 2, \cdots$ with $F(x) = 0$ for $x < 0$.

(b) Show that the inverse of F is given by

$$F^{-1}(t) = n \quad \text{if } 1 - \theta^n < t \le 1 - \theta^{n+1}$$

for $0 < t < 1$.

(c) Suppose that $U \sim \text{Unif}(0, 1)$ and define $N = n$ if

$$\theta^{n+1} \le 1 - U < \theta^n$$

(where n is an integer). Show that N has a Geometric distribution.

(d) Suppose that X has an Exponential distribution with mean $-1/\ln(\theta) > 0$ and define

$$N = \lfloor X \rfloor$$

where $\lfloor x \rfloor$ denotes the integer part of x. Show that N has a Geometric distribution.