PAP334 – Exercises 7 – Model answers

Problem 1

Reminder. *The probability distribution function (PDF), expectation value, and variance of the Poisson distribution are respectively:*

$$f(n|\nu) = \frac{\nu^n}{n!}e^{-\nu}, \quad E[n] = \nu, \quad V[n] = \nu.$$

One first computes the likelihood function L for m observation as:

$$L(\nu) = \prod_{i=1}^{m} f(n|\nu)$$

i) For a single observation, i.e. $m = 1$, the likelihood function is strictly identical to the PDF: $L(\nu) = f(n|\nu)$. As already introduced, a good estimator $\hat{\nu}$ can be derived using the log-likelihood function:

$$\ln L(\nu) = \ln\left[\frac{\nu^n}{n!}e^{-\nu}\right] = n\ln\nu - \ln n! - \nu.$$

Finding extrema of this (log-)likelihood function can be done through a simple minimisation/maximisation problem, i.e. finding values of $\hat{\nu}$ such that $\mathrm{d}\ln L/\mathrm{d}\nu|_{\nu=\hat{\nu}} = 0$. In our case,

$$\left.\frac{\mathrm{d}\ln L}{\mathrm{d}\nu}\right|_{\nu=\hat{\nu}} = \frac{n}{\hat{\nu}} - 1 = 0 \quad \Longrightarrow \quad \hat{\nu} = n.$$

To figure out if this extremum is a minimum or a maximum, one may compute the second order derivative to the log-likelihood function:

$$\left.\frac{\mathrm{d}^2\ln L}{\mathrm{d}\nu^2}\right|_{\nu=\hat{\nu}} = \frac{-n}{\hat{\nu}^2} < 0.$$

Therefore $\hat{\nu} = n$ is the maximum likelihood (ML) estimator for a Poisson distribution with one single observation.

ii)

Reminder. *The bias of an estimator $\hat{\theta}$ of any θ random variable is defined as:*

$$b = E[\hat{\theta}] - \theta.$$

Therefore, an estimator may be considered unbiased if equal to its expectation value.

For the Poisson distribution one may compute

$$E[\hat{\nu}] = E[n] = \nu,$$

which shows that its ML estimator is unbiased. The variance of the $\hat{\nu}$ estimator is obtained from the Poisson distribution variance:

$$V[\hat{\nu}] = V[n] = \nu = n.$$

iii) The Rao-Cramer-Fréchet (RFC) bound is defined as:

$$V[\hat{\nu}] \geq \left[1 + \frac{\partial b}{\partial \nu}\right]^2 \Big/ E\left[-\frac{\partial^2 \ln L}{\partial \nu^2}\right].$$

As $\hat{\nu}$ was shown to be unbiased, the numerator is equal to 1 and one may concentrate on the denominator:

$$E\left[-\frac{\partial^2 \ln L}{\partial \nu^2}\right] = E\left[\frac{n}{\nu^2}\right] = \frac{E[n]}{\nu^2} = \frac{1}{\nu}.$$

Therefore the RCF bound becomes

$$V[\hat{\nu}] \geq \nu.$$

The equality corresponds to the Poisson variance calculated earlier. Therefore the estimator $\hat{\nu}$ is efficient.

iv) For the generalised m observation, one may compute again the ML using the log-likelihood function introduced earlier:

$$\ln L_m = \sum_{i=1}^{m} \ln f(n_i|\nu),$$

which can be solved for $m = 2$:

$$\ln L_2 = (n_1 \ln \nu - \ln n_1! - \nu) + (n_2 \ln \nu - \ln n_2! - \nu) = (n_1 + n_2) \ln \nu - \ln n_1! - \ln n_2! - 2\nu,$$

and extended to any m:

$$\ln L_m = \left(\sum_{i=1}^{m} n_i\right) \ln \nu - \sum_{i=1}^{m} \ln n_i! - m\nu.$$

The ML estimator can again be derived as an extrema finding problem:

$$\frac{\partial}{\partial \nu} \ln L_m \bigg|_{\nu=\hat{\nu}} = \left(\sum_{i=1}^{m} n_i\right) \frac{1}{\hat{\nu}} - m = 0 \quad \Longrightarrow \quad \hat{\nu} = \frac{1}{m} \sum_{i=1}^{m} n_i,$$

which is the well known expression for the sample mean.

Problem 2

In this exercise we will use the following PDF of the time t_i treated as a random variable:

$$f(t_i|f) = |\sin(2\pi f t_i)|, \text{ with } 0 < t_i < 1 \text{ the normalised time.}$$

One can again use the log-likelihood function defined as:

$$\ln L = \sum_{i}^{N} \ln f(t_i|f),$$

and shown in Figure 1 for the two samples with respectively 20 and 100 observations. As the range of f to be scanned is covering a broad range (2 orders of magnitude) we will use a logarithmic distribution of frequencies. This function will again be maximised (numerically this time) to obtain the ML estimator for the frequency, $\hat{f}$.

i) For $m = 20$, the ML estimator for $\hat{f} = 1.4879$ Hz (with $\max(\ln L_{20}) = -5.388$).

ii) The graphical method allows to determine the 1σ uncertainty on the estimator $\hat{f}$ through the values of f such that

$$\ln L = \max(\ln L) - 0.5.$$

For $m = 20$, this corresponds to $f_{\text{low}} = 1.4407$ and $f_{\text{high}} = 1.5403$, hence the ML estimator for the frequency is

$$\hat{f}_{m=20} = 1.488^{+0.052}_{-0.047}.$$

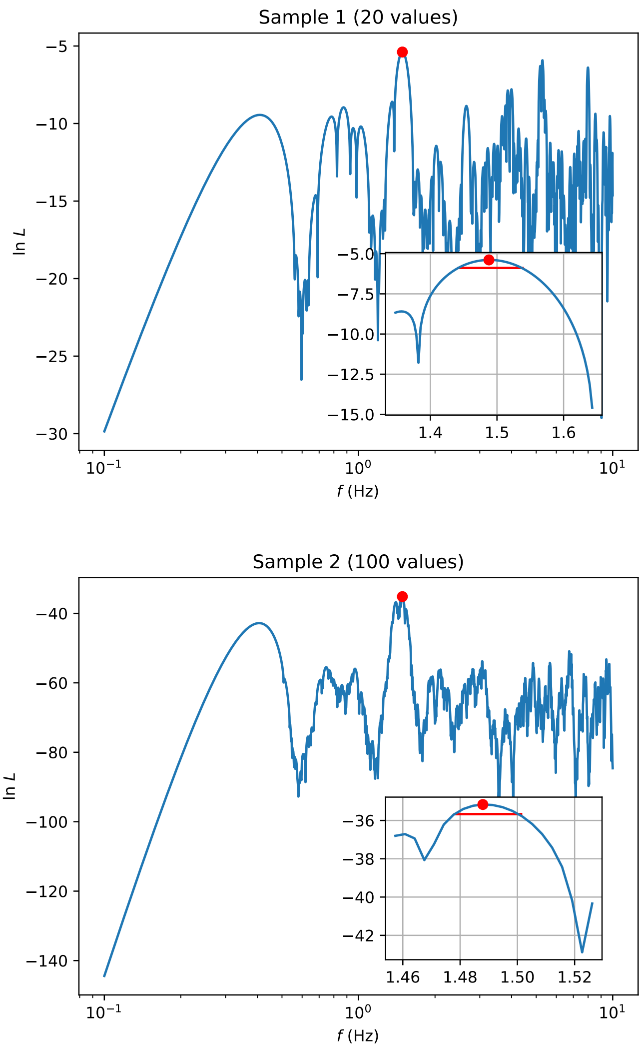

Figure 1: Log-likelihood distribution for (resp.) $m = 20$ and 100 observations of the time t_i.

iii) For $m = 100$, the ML estimator for $\hat{f} = 1.4879$ Hz (with $\max(\ln L_{100}) = -35.168$).

iv) For $m = 100$, the ML estimator for the frequency is

$$\hat{f}_{m=100} = 1.488^{+0.014}_{-0.010},$$

corresponding to a ratio of uncertainties of 3.8 and 4.6 respectively, much better than the expected improvement $\sqrt{10020} \simeq 2.24$ from the size difference of both data samples.

Code listing

Problem 2

```
import matplotlib.pyplot as plt
import numpy as np

smp1 = np.loadtxt('ml_sample_1.txt')
smp2 = np.loadtxt('ml_sample_2.txt')

# logarithmic x scale
npoints = 2000
expf = np.linspace(-1., 1., npoints)
f = 10**expf
lnL_smp1 = np.zeros(npoints)
lnL_smp2 = np.zeros(npoints)
for i in range(len(f)):
    lnL_smp1[i] = sum(np.log(abs(np.sin(2*np.pi*f[i]*smp1))))
    lnL_smp2[i] = sum(np.log(abs(np.sin(2*np.pi*f[i]*smp2))))

print('=== Exercise i)')

def find_max(xarr, yarr):
    idx_max = np.argmax(yarr)
    return (xarr[idx_max], yarr[idx_max])

max_f_smp1, max_lnL_smp1 = find_max(f, lnL_smp1)
print('ln L max={} at f={} Hz'.format(max_lnL_smp1, max_f_smp1))

print('=== Exercise ii)')

def find_uncertainties(xarr, yarr):
    idx_max = np.argmax(yarr)
    max_val = yarr[idx_max]
    for i in range(idx_max, 0, -1):
        x_low = xarr[i]
        if yarr[i] <= max_val-0.5:
            break
    for i in range(idx_max, len(xarr)):
        x_high = xarr[i]
        if yarr[i] <= max_val-0.5:
            break
    return (x_low, x_high)

low, high = find_uncertainties(f, lnL_smp1)
unc_f_low_smp1 = max_f_smp1-low
unc_f_high_smp1 = high-max_f_smp1
print('f={} and {} for ln L(f)=max(ln L)-0.5'.format(low, high))
print('uncertainty: -{} +{}'.format(unc_f_low_smp1, unc_f_high_smp1))

print('=== Exercise iii)')

max_f_smp2, max_lnL_smp2 = find_max(f, lnL_smp2)
print('ln L max={} at f={} Hz'.format(max_lnL_smp2, max_f_smp2))

print('=== Exercise iv)')

low, high = find_uncertainties(f, lnL_smp2)
unc_f_low_smp2 = max_f_smp2-low
unc_f_high_smp2 = high-max_f_smp2
print('uncertainty: -{} +{}'.format(unc_f_low_smp2, unc_f_high_smp2))

print('=== Exercise v)')
print('ratio of lower uncertainties:', unc_f_low_smp1/unc_f_low_smp2)
print('ratio of higher uncertainties:', unc_f_high_smp1/unc_f_high_smp2)

fig = plt.figure(1)
plt.plot(f, lnL_smp1)
plt.plot([max_f_smp1], [max_lnL_smp1], marker='o', color='r')
plt.xscale('log')
plt.xlabel('$f$ (Hz)')
plt.ylabel('$\ln~L$')
plt.title('Sample 1 (20 values)')
# zoom on max
ax = plt.axes([0.55, 0.175, 0.3, 0.3])
f_zoom = []
lnL_smp1_zoom = []
for i in range(len(f)):
    if f[i] >= max_f_smp1-3*unc_f_low_smp1 and f[i] <= max_f_smp1+3*unc_f_high_smp1:
        f_zoom.append(f[i])
        lnL_smp1_zoom.append(lnL_smp1[i])
```

```
ax.plot(f_zoom, lnL_smp1_zoom)
ax.plot([max_f_smp1], [max_lnL_smp1], marker='o', color='r')
ax.hlines(max_lnL_smp1-0.5, max_f_smp1-unc_f_low_smp1, max_f_smp1+unc_f_high_smp1, colors='r')
ax.grid()
plt.show()

fig = plt.figure(2)
plt.plot(f, lnL_smp2)
plt.plot([max_f_smp2], [max_lnL_smp2], marker='o', color='r')
plt.xscale('log')
plt.xlabel('$f$ (Hz)')
plt.ylabel('$\ln~L$')
plt.title('Sample 2 (100 values)')
# zoom on max
ax = plt.axes([0.55, 0.175, 0.3, 0.3])
f_zoom = []
lnL_smp2_zoom = []
for i in range(len(f)):
    if f[i] >= max_f_smp2-3*unc_f_low_smp2 and f[i] <= max_f_smp2+3*unc_f_high_smp2:
        f_zoom.append(f[i])
        lnL_smp2_zoom.append(lnL_smp2[i])
ax.plot(f_zoom, lnL_smp2_zoom)
ax.plot([max_f_smp2], [max_lnL_smp2], marker='o', color='r')
ax.hlines(max_lnL_smp2-0.5, max_f_smp2-unc_f_low_smp2, max_f_smp2+unc_f_high_smp2, colors='r')
ax.grid()
plt.show()
```