

Statistical Signal Processing Project

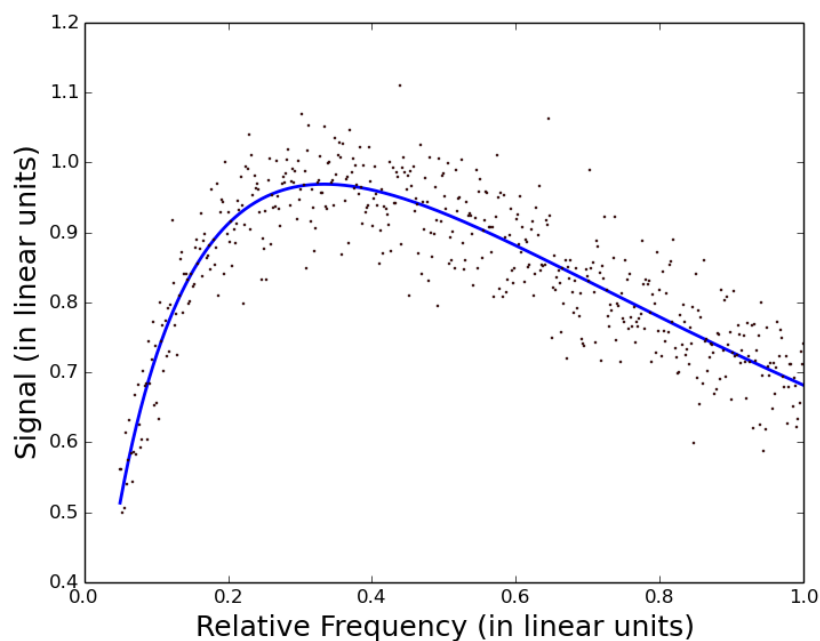
Submission deadline: 4th, March, 2016

Note: You are not allowed to use any module/package for the optimization problems. Email your project report as a single pdf to `gehlot@astro.rug.nl` strictly by the deadline, later submissions will not be accepted. The reports must contain enough information for the reader to understand and reproduce your results. Your reports will be graded based on your understanding of the problem, implementation and interpretation of results, and clarity of information in your text and figures.

Load the MATLAB file `project_data.mat`. This file has three variables:

- (i) `nu`, which the frequency in Hz.
- (ii) `signal`, which is the power in Watts measured at each frequency.
- (iii) `signal_plus_noise`, which is the signal with WGN ($\mu = 0$, $\sigma = 0.05$) added to it.

We'll use `nu` as the independent variable, and `sigma_plus_noise` as the observed data. The Figure shows the plot of the `signal` (blue curve) and `signal_plus_noise` (points) vs. `nu`.



The observed data can be written in the following form:

$$\mathbf{x} = \mathbf{s} + \mathbf{w}$$

Where $\mathbf{x} = [x_1, x_2, \dots, x_N]$ are the observed data points and $\mathbf{w} = [w_1, w_2, \dots, w_N]$ is WGN with $\mathcal{N}(\mu = 0, \sigma^2 = 0.0025)$. We would like to fit the signal with the model of the following form:

$$s = A \times \left(\frac{\nu}{\nu_o}\right)^\alpha \left(1 + \frac{\nu}{\nu_o}\right)^{-4\alpha}$$

Where, ν is the frequency (independent variable) and A, ν_o and α are the parameters to be estimated. Using the above data model, we can write each observed data point as

$$x_i = A \times \left(\frac{\nu_i}{\nu_o}\right)^\alpha \left(1 + \frac{\nu_i}{\nu_o}\right)^{-4\alpha} + w_i$$

1. How does the model behave in the limits $\nu \gg \nu_o$ and $\nu \ll \nu_o$. Plot the model function using log-axes. What can you infer about the probable values of the parameters from the plot ?
2. Write down the Maximum Likelihood Estimator (MLE) for the three parameters. Numerically evaluate the ML estimate of the parameters using the Newton Raphson's optimization method using $(A, \nu_o, \alpha) = (6.0, 2.0, 1.0)$ as the initial guess parameters (you are encouraged to use the other optimization methods such as gradient descent etc.). Plot the residuals of the fit and their PDF.
3. Generate 10000 different Monte-Carlo realizations of \mathbf{x} by adding different realizations of noise ($\sigma = 0.05, \mu = 0$) to the `signal` from the data-file and repeat the same process as in problem 2 for these realizations. Use these estimates from the Monte-Carlo simulations to compute the probability density functions for the parameters. Find the mean and variance of these distributions.
4. Analytically compute the Fisher information matrix, and numerically compute the CRLB matrix at the ML estimate values. How does this CRLB compare with the variance estimated from the Monte-Carlo simulations? Explain.
5. Compute the likelihood function $\mathcal{L}(A, \nu_o, \alpha)$ on a three dimensional grid of values for A, ν_o , and α . The likelihood function is a probability density function, and hence must integrate to unity. Marginalize this function along each of the three axes to compute the marginalized probability densities: $P(A, \nu_o)$, $P(A, \alpha)$, and $P(\nu_o, \alpha)$. Plot these three distributions functions as images, and overlay contours of 1σ , 2σ , and 3σ uncertainties on them. Interpret your plots.

Hints:

1. To read a '.mat' file in python:

```
import scipy.io
contents = scipy.io.loadmat('project_data.mat', squeeze_me=True)
freq = np.asarray(contents['nu']) #reading freq. column in an array
Signal = np.asarray(contents['signal']) #reading Signal column
SpN = np.asarray(contents['signal_plus_noise']) # Signal + Noise
```

2. Newton Raphson method for multiple variables:

Let

$$\mathbf{F}(A, \nu_o, \alpha) = \begin{pmatrix} \frac{\partial \ln \mathcal{L}}{\partial A} \\ \frac{\partial \ln \mathcal{L}}{\partial \nu_o} \\ \frac{\partial \ln \mathcal{L}}{\partial \alpha} \end{pmatrix} = \begin{pmatrix} f_1(A, \nu_o, \alpha) \\ f_2(A, \nu_o, \alpha) \\ f_3(A, \nu_o, \alpha) \end{pmatrix}$$

Therefore, Newton Raphson iterative step can be written as

$$A_f = A_i - \mathbf{J}^{-1} \mathbf{F}(A_i, \nu_{oi}, \alpha_i)$$

$$\nu_{of} = \nu_{oi} - \mathbf{J}^{-1} \mathbf{F}(A_i, \nu_{oi}, \alpha_i)$$

$$\alpha_f = \alpha_i - \mathbf{J}^{-1} \mathbf{F}(A_i, \nu_{oi}, \alpha_i)$$

Where \mathbf{J} is the Jacobian matrix and is given by

$$\mathbf{J} = \begin{pmatrix} \frac{\partial f_1}{\partial A} & \frac{\partial f_1}{\partial \nu_o} & \frac{\partial f_1}{\partial \alpha} \\ \frac{\partial f_2}{\partial A} & \frac{\partial f_2}{\partial \nu_o} & \frac{\partial f_2}{\partial \alpha} \\ \frac{\partial f_3}{\partial A} & \frac{\partial f_3}{\partial \nu_o} & \frac{\partial f_3}{\partial \alpha} \end{pmatrix}$$