

Homework 7

1. Matched filtering for sonar

A sonar system operates by emitting a short pulse of sound, and listening for the reflection. The time between the emission and the reflection then determines the distance to the object. We will model the sonar in discrete-time, and use a sample period of 1 in appropriate units.

A commonly used signal shape for sonar is a *chirp*, which has the form of a sound with slowly rising frequency, given by

$$u(t) = \sin(c_1 t^2) \quad \text{for } t \in [0, p-1]$$

Different frequency ranges are used depending on the desired range of distances at which objects are to be detected.

The emitted signal is then

$$u_{\text{emitted}} = u(t+p) \quad \text{for } t \in [-p, -1]$$

For simplicity we assume that the received signal has the same shape as the emitted chirp. Of course, it is delayed by an amount proportional to the distance to the object, and corrupted by noise.

$$y(t) = \begin{cases} u(t-d) + w(t) & \text{if } d \leq t \leq p+d-1 \\ w(t) & \text{otherwise} \end{cases}$$

We listen for this reflection on the interval $[0, m-1]$. The noise $w(0), w(1), w(2), \dots$ is IID and Gaussian, with

$$w(t) \sim \mathcal{N}(0, \sigma^2) \quad \text{for all } t$$

We would now like to measure the signal $y(0), y(1), \dots, y(m-1)$ and determine d . Our prior information regarding the random variable d is that it is uniformly distributed on the discrete interval $[0, m-p]$. To simplify notation, define

$$y = \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(m-1) \end{bmatrix} \quad u = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(p-1) \end{bmatrix} \quad w = \begin{bmatrix} w(0) \\ w(1) \\ \vdots \\ w(m-1) \end{bmatrix}$$

Also define the shift matrices $S(k) \in \mathbb{R}^{m \times p}$ for $k = 0, 1, \dots, m-p$ by

$$S(k) = \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix} = \begin{bmatrix} 0_{k \times p} \\ I_p \\ 0_{(m-p-k) \times p} \end{bmatrix}$$

where $O_{n \times m}$ denotes the $n \times m$ matrix of zeros, and I_p denotes the $p \times p$ identity matrix. Then we have

$$y = S(d)u + w$$

Then $\Sigma = \sigma^2 I_m$ is the covariance of w .

(a) Show that the conditional pdf of y given $d = k$ is

$$y | (d = k) \sim \mathcal{N}(S(k)u, \Sigma)$$

(b) We would like to construct an estimator of d given a measurement of y . Show that the estimator $f : \mathbb{R}^m \rightarrow \{0, 1, \dots, m-p\}$ which minimizes the probability of error is

$$f_{\text{est}}(y_{\text{meas}}) = \arg \max_k g_k(y_{\text{meas}})$$

where the functions g_k are defined by

$$g_k(y) = u^T S(k)^T y$$

- (c) In fact there is a very efficient method of computing this. Let h be the vector u flipped upside down; that is

$$h(t) = u(p + 1 - t) \quad \text{for } t = 0, \dots, p - 1$$

Show that

$$g_k(y) = r(p + k - 1)$$

where $r = h * y$ is the *convolution* of h with y , i.e.,

$$r(j) = \sum_{i=0}^m h(j - i)y(i)$$

with the usual convention that $h(i) = 0$ if i is too large or too small.

This means that, to find the estimate of d , all we do is construct the convolution, and look at the *peak* of the signal r . If this peak occurs at r_k , then the estimate for d is $k - p$. The nice thing about convolution is that it is simply a *filtering* operation, and the filter r is called the *matched* filter.

- (d) We will use chirp parameters $c_1 = 0.003$ and $p = 100$, and listen for signals up to length $m = 500$.

Suppose the noise covariance is $\sigma^2 = 0.01$. Simulate the generation of y , and construct the signal r . Plot the signals r , y and u . It's worth using the same horizontal scale for all the plots, so you can see what the signal is you are looking for. Note that you can use the MATLAB commands `flipud` and `conv` to compute r .

- (e) Now run the simulation a large number of times, say 50,000. Use this to estimate the conditional probability of error

$$\text{Prob}(f_{\text{est}}(y) \neq k \mid d = k)$$

and also the conditional probability of an error of magnitude greater than 1

$$\text{Prob}(|f_{\text{est}}(y) - d| > 1 \mid d = k)$$

- (f) Repeat parts (??) through (??) above with a noise covariance of $\sigma^2 = 1$. Notice how difficult it is to locate the signal simply by looking at the plot of y .

2. Navigation

Suppose a ship is located at position $x \in \mathbb{R}^2$. We measure distances to beacons located at

$$q_1 = \begin{bmatrix} 500 \\ 0 \end{bmatrix} \quad q_2 = \begin{bmatrix} 0 \\ 500 \end{bmatrix} \quad q_3 = \begin{bmatrix} 500 \\ -200 \end{bmatrix}$$

As usual we model this as $y = Ax + w$, where $y \in \mathbb{R}^3$ is a vector of measurements (with appropriate constants to make the relationship linear) and $w \in \mathbb{R}^3$ is Gaussian noise, $w \sim \mathcal{N}(0, \Sigma_w)$. We also have prior information $x \sim \mathcal{N}(\mu_x, \Sigma_x)$. Here

$$\mu_x = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \quad \Sigma_x = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \Sigma_w = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- (a) The MMSE estimate has the form

$$\phi_{\text{mmse}}(y) = \mu_x + L_{\text{mmse}}(y - A\mu_x)$$

Find the estimator gain L_{mmse} .

- (b) Perform the experiment in Matlab; i.e., simulate x and w with the above distributions and compute y . Using this y , plot the 90% confidence ellipsoid for x conditioned on your measurement, centered on your estimate.
- (c) Now collect data, consisting of 500 samples of $y = Ax + w$. For each measurement y , compute the corresponding estimate $\phi_{\text{mmse}}(y)$, and plot the error in \mathbb{R}^2 given by $x - \phi_{\text{mmse}}(y)$. Also plot the 90% confidence ellipsoid for the error (centered at the origin.)
- (d) Suppose instead we were to use the least squares estimator

$$\phi_{\text{ls}}(y) = A^\dagger y$$

What is the mean square error

$$\mathbb{E}\|\phi_{\text{ls}}(y) - x\|^2$$

achieved by this estimator?

- (e) Using the same data as above, plot the error in \mathbb{R}^2 given by $x - \phi_{\text{ls}}(y)$. Also plot the 90% confidence ellipsoid for the error, centered at the origin.
- (f) Is it true that the ellipsoid of part (??) is always contained in the ellipsoid of part (??)? Prove or give a counterexample.