EE3331 Probability Models in Information Engineering

Project 2

Engineering Applications of Probability Models

Intended Learning Outcomes:

On completion of this project, you should be able to

- Use MATLAB to simulate several probability-related engineering problems
- Implement simple detectors and evaluate their performance
- Implement a simple sinusoidal frequency estimator and evaluate its performance
- Implement a simple text coding method and evaluate its performance

Deliverable:

- Each student is required to submit a **short report** which includes your answers for questions and findings for this project work on or before **12 April 2023.**

Procedure:

1. Consider a radar system and it transmits a sinusoidal pulse of 50 samples:

$$s[k] = 0.8\cos(0.2k + \phi), \quad k = 1, 2, \cdots, 50$$

where ϕ is your student ID number.

(a) Generate $s[k]$ and compute its energy using MATLAB. The energy is defined as:

$$E_s = \sum_{k=1}^{50} s^2[k]$$

(b) Suppose after certain time interval, the radar system receives $x[k]$. This can mean presence ($\mathcal{H}_1$) or absence ($\mathcal{H}_0$) of target:

$$\mathcal{H}_0 : x[k] = n[k]; \qquad \mathcal{H}_1 : x[k] = s[k] + n[k], \quad k = 1, 2, \cdots, 50$$

where $n[k] \sim \mathcal{N}(0, \sigma^2)$ is white Gaussian noise. Given $x[k]$, $k = 1, 2, \cdots, 50$, the task is to decide whether there is a target or not.

A conventional method is to use the energy detector. That is, we compute the energy of the received signal $x[k]$, which corresponds to either $\mathcal{H}_0$ or $\mathcal{H}_1$:

$$E_x = \sum_{k=1}^{50} x^2[k]$$

and then compare E_x with a threshold γ: if $E_x \geq \gamma$, $\mathcal{H}_1$ is picked, and $\mathcal{H}_0$ is chosen for $E_x < \gamma$.

Use `randn` to generate $n[k]$ with $\sigma^2 = 10$, and produce 200000 sets of $x[k]$, that is, 100000 correspond to $\mathcal{H}_0$, and the remaining 100000 for $\mathcal{H}_1$, to produce the receiver operating characteristic (ROC) [1] curve for the energy detector with the use of the command `perfcurve` or by other means. Attach the ROC in your report. The ROC plots the probability of detection (P_{D}) versus the probability of false alarm (P_{FA}). Denoting the event of deciding that the target is present as T, we can write $P_{\mathrm{D}} = P(T|\mathcal{H}_1)$ and $P_{\mathrm{FA}} = P(T|\mathcal{H}_0)$. Note also that P_{D} and $1 - P_{\mathrm{FA}}$ are also known as sensitivity and specificity, respectively.

(c) When the transmitted signal $s[k]$ is known, another popular method is the matched filter, which computes the matching function:

$$M_x = \sum_{k=1}^{50} x[k]s[k]$$

The M_x in fact measures the correlation between $s[k]$ and $x[k]$. As in the energy detector, use 200000 sets of $x[k]$ to plot the ROC curve of the matched filter. Attach the ROC in your report. Compare the results with that of the energy detector. Which detector has better performance? What are your findings?

2. Spectral analysis involves determining the distribution of amplitude or power over frequency, associated with a given signal. A fundamental task is to estimate the frequencies of signals and this problem has many applications such as wireless communications, audio and speech processing, biomedical engineering, power electronics, astronomy, and instrumentation [2]-[3].

Consider the single-tone model:

$$x[n] = \alpha \cos(\omega n + \phi) + q[n], \quad n = 0, 1, \cdots, N-1$$

where α, ω and ϕ, are unknown constants while $q[n] \sim \mathcal{N}(0, \sigma_q^2)$ is a white Gaussian process. To find the sinusoidal frequency ω, the linear prediction property can be utilized:

$$\cos(\omega n + \phi) = \rho \cos(\omega(n-1) + \phi) - \cos(\omega(n-2) + \phi), \quad \rho = 2\cos(\omega)$$

Using least squares, the estimate of ρ, denoted by $\hat{\rho}$, is then given by:

$$\hat{\rho} = \arg\min_{\tilde{\rho}} \sum_{n=2}^{N-1} \left(x[n] + x[n-2] - \tilde{\rho} x[n-1]\right)^2 = \hat{\rho} = \frac{\sum_{n=2}^{N-1} x[n-1](x[n] + x[n-2])}{\sum_{n=2}^{N-1} x^2[n-1]}$$

Then the estimate of ω, denoted by $\hat{\omega}$, is:

$$\hat{\omega} = \cos^{-1}\left(\frac{\hat{\rho}}{2}\right)$$

which is also known as the modified covariance (MC) method [3].

(a) Create a file called “mc.m” with the following MATLAB code:

```
function w = mc(x);
% w = mc(x) is used to estimate frequency
% based on MC method from the vector x
% x is supposed to be a noisy single-tone sequence
% w is the estimated frequency in radian
N=max(size(x));
t1=0;
t2=0;
for n=3:N
     t1=t1+x(n-1)*(x(n)+x(n-2));
     t2=t2+2*(x(n-1))^2;
end
r = t1/t2;
if (r>1)
     r=1;
end
if (r<-1)
     r=-1;
end
w=acos(r);
```

That is, `mc` is the MATLAB function for the MC estimator. What is purpose of setting `r` to 1 and -1 when its value is larger than 1 and smaller than -1, respectively?

(b) Use MATLAB to generate a vector `x`:

$$x[n] = 2\cos(n+\phi), \quad n = 0, 1, \cdots, 49$$

where ϕ is your student ID number. Then employ the command `mc(x)` to compute $\hat{\omega}$. Write down the value of $\hat{\omega}$.

(c) Repeat (b) with `x`:

$$x[n] = 2\cos(n+\phi) + q[n], \quad n = 0, 1, \cdots, 49$$

where $\sigma_q^2 = 0.1$. Use `randn` to generate $q[n]$.

(d) For the signal model in (c), determine the empirical mean, mean square error, and variance of $\hat{\omega}$, that is, $\mathbb{E}\{\hat{\omega}\}$, $\mathbb{E}\{(\hat{\omega}-\omega)^2\}$, and $\mathbb{E}\{(\hat{\omega}-\mathbb{E}\{\hat{\omega}\})^2\}$, respectively, based on 1000 independent trials.

(e) Repeat (d) with $\sigma_q^2 = 1$.

(f) Comparing the results of (d) and (e), what are your findings?

(g) Is MC an unbiased estimator? Briefly explain your answer.

3. Data compression can be achieved by encoding information with fewer bits compared to the original representation. Compression techniques can be classified into two types: lossy and lossless methods. Huffman coding [4] is a classic lossless compression algorithm, whose basic idea is to assign different codelengths to the input symbols according to their frequencies of occurrence, i.e., frequently occurring symbols are coded with fewer bits, and vice versa.

   (a) Download “Huffman.zip” at Canvas. Huffman coding algorithm is implemented using MATLAB as “Huffman.m”. Try to understand the code, particularly the variables `alphabet`, `probability`, and `dict`.
   (b) Run `Huffman`. Observe the produced files encode.txt and decode.txt, and compare them with Dream1.txt. What are your findings?
   (c) Which of the characters in Dream1.txt has the highest probability of occurrence? What is the corresponding probability?
   (d) Compute the information entropy H which is defined as:

$$H = -\sum_{n=1}^{N} p(x_n) \log_2(p(x_n))$$

   where $\{x_n\}_{n=1}^{N}$ denote the N distinct characters in Dream1.txt and $p(x_n)$ is the corresponding probability of occurrence. Then compute the average number of bits per character based on Huffman coding.

   (e) Repeat (d) for Dream2.txt with the use of `Huffman`.

References:

[1] S. Kay, *Fundamentals of Statistical Signal Processing, Volume II: Detection Theory*, Pearson, 1998
[2] S. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, Pearson, 1993
[3] S. M. Kay, *Modern Spectral Estimation: Theory and Application*, Prentice-Hall, 1988
[4] D. A. Huffman, “A method for the construction of minimum-redundancy codes,” *Proceedings of the IRE*, vol.40, no.9, Sep. 1952, pp.1098-1101