ICT for Health
Laboratory # 5
ROC

Monica Visintin

Politecnico di Torino

2017/18

Warning

No report for this laboratory

Problem 1 [1]

You know (Maxwell's devil told you) that the value of bilirubin in the blood of a healthy person is a Gaussian random variable with mean $\mu_0 > 0$ and variance σ_0^2; a person with pancreas tumor has the value of bilirubin in the blood which is again Gaussian with mean $\mu_1 > \mu_0$ and variance σ_1^2. You decide that the bilirubin test is positive if the measured bilirubin value is above threshold T.
Write the **formulas** of your test specificity $P(T_n|D_n)$ and sensitivity $P(T_p|D_y)$ (pen and paper). Remember that

$$\int_T^\infty \frac{1}{\sqrt{2\pi\sigma^2}} e^{-u^2/(2\sigma^2)}\, du = \frac{1}{2}\text{erfc}\left(\frac{T}{\sqrt{2\sigma^2}}\right)$$

Let us say that the incidence of pancreas tumor in the population is p. Write the **formulas** for the probabilities $P(D_y|T_p)$ that a person is really affected by pancreas tumor given that the test is positive and $P(D_n|T_n)$ that a person really is healthy given that the test is negative.

Problem 1 [2]

Write a Python script that plots:

1. in the same figure, the pdf of bilirubin of a healthy person and of a person affected by pancreas tumor
2. test sensitivity and specificity as functions of the threshold T
3. the ROC curve (i.e. sensitivity $P(T_p|D_y)$ versus probability of false alarm $1 - P(T_n|D_n)$)
4. $P(D_y|T_p)$ as function of T and p (a surface)
5. $P(D_n|T_n)$ as function of T and p (a surface)
6. the contour plots of $P(D_y|T_p)$ and $P(D_n|T_n)$ at probability values $0.5, 0.9, 0.99, 0.999, 0.9999$
7. $P(e) = P(T_p|D_n)(1-p) + P(T_n|T_p)p$ as function of T and p (a surface)

Use initially

- $\mu_0 = 10, \sigma_0^2 = 4$
- $\mu_1 = 15, \sigma_1^2 = 8$

Problem 1 [3]

Once you can plot the results for these means and variances, play with the values as you like, and see the effects.
Note that in Python the erfc function is the SciPy library
`scipy.special.erfc(x)`.
For 3D plotting with `matplotlib`, visit
`https://matplotlib.org/mpl_toolkits/mplot3d/tutorial.html`

Problem 2 [1]

(D. MacKay problem, discussed in Chapt. 3) Unstable particles are emitted from a source and decay at a distance x, a real number that has an exponential probability distribution with characteristic length λ. Decay events can be observed only if they occur in a window extending from $x = 1$ cm to $x = 20$ cm. N decays are observed at locations $\mathbf{x} = \{x_1, ..., x_N\}$.
What is λ?
The probability density function of x (assuming λ known) is

$$f_x(u|\lambda) = \begin{cases} \frac{c(\lambda)}{\lambda} e^{-u/\lambda} & 1 < u < 20 \\ 0 & \text{otherwise} \end{cases}$$

where $c(\lambda)$ is such that $\int f_x(u|\lambda)du = 1$:

$$\frac{c(\lambda)}{\lambda} \int_1^{20} e^{-u/\lambda} = \frac{c(\lambda)}{\lambda} \frac{e^{-1/\lambda} - e^{-20/\lambda}}{1/\lambda} = 1, \quad \Longrightarrow \quad c(\lambda) = \frac{1}{e^{-1/\lambda} - e^{-20/\lambda}}$$

We assume that the emissions are **statistically independent**.

Problem 2 [2]

Assume $N = 6$ and

$$\mathbf{x} = [1.5, 2, 3, 4, 5, 12]$$

1. Maximum likelihood approach: write the probability density function of $\mathbf{x}$ given λ (the likelihood function) and estimate $\hat{\lambda}_{ML}$ by finding the value of λ that maximizes $f(\mathbf{x}|\lambda)$. Plot
   - the likelihood function versus λ
   - the log-likelihood function $\log[f(\mathbf{x}|\lambda)]$ versus λ.
2. Frequentist approach: the theoretical mean of the random variable x for a given λ is

$$\mu_x(\lambda) = \int_{-\infty}^{\infty} u f_x(u|\lambda) du = \lambda + \frac{e^{-1/\lambda} - 20e^{-20/\lambda}}{e^{-1/\lambda} - e^{-20/\lambda}};$$

   measure the mean value m_x of $\mathbf{x}$ and find $\hat{\lambda}_f$ by solving $\mu_x(\lambda) = m_x$.

Problem 2 [3]

3. Bayesian approach: write the probability density function of λ given $\mathbf{x}$ (note that λ is a parameter but bayesians consider it a random variable). What is the missing piece of information? as you see, it is not possible to find the bayesian estimation.

Problem 2 [4]

Afterwards, generate a vector $\mathbf{x}$ of $N = 2$, $N = 100$, $N = 1000$, $N = 10000$ values with the given probability $f_x(u|\lambda)$ and with $\lambda = 5$ and repeat the exercise with these data.
To generate the vector of random variables, you can use the NumPy class `random.exponential`, but notice that it draws values according to the true exponential pdf (possible values from 0 to ∞) and not the truncated pdf in thye range $[1, 20]$ that is needed in this exercise. What can you do to solve this problem? Once you've found the solution, to check that the data are generated according to the correct pdf, plot the theoretical **cumulative distribution function (CDF)**:

$$F_x(u|\lambda) = P(x \leq u|\lambda) = \begin{cases} 0 & u < 1 \\ \frac{\exp(-1/\lambda) - \exp(-u/\lambda)}{\exp(-1/\lambda) - \exp(-20/\lambda)} & 1 \leq u \leq 20 \\ 1 & u > 20 \end{cases}$$

and the measured one. Note that, to get the measured CDF it is sufficient to use sorting. Any idea?