EE-411, HomeWork 1 : Maximum Likelihood & Probability

This homework involves some coding, with the language of your choice. Present your results with graphs, plots, and data. Jupyter notebooks are a good option, and we recommend you to send your work as a notebook on Google colab.

1 Statistical inference and Maximum Likelihood

We consider again the problem introduced in the lecture : We have a radioactive source at $x = 0$ emitting particles in the positive x-direction, and an "imperfect" detector, which is able to detect a particle's decay only in the range $x \in [1, 20]$. As we have seen, the density probability associated to this problem, for a fixed value of λ, is given by

$$p_\lambda(x_i) = \begin{cases} \frac{1}{Z(\lambda)} e^{-x_i/\lambda} & \text{if } 1 \leq x_i \leq 20 \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

In the following, we shall assume that we know that the true λ (denoted λ^*) is in the range $\lambda^* \in \ [0.05, 50]$, and starting from the observation of a set of decays $\{x_i\}$ we aim to estimate the decay rate λ^*.

1. Compute $Z(\lambda)$ such that the probability in eq.(1) is normalized. Show that $\mathbb{E}[X] = \lambda^2 \partial_\lambda \log Z(\lambda)$ and $\text{var}[X] = \lambda^2 \partial_\lambda \mathbb{E}[X]$. Compute and plot the mean and variance as a function of λ.
2. Write explicitly the probability $p_\lambda(\{x_i\}_{i=1}^n)$ to observe a set of n *independent* events at positions $\{x_i\}_{i=1}^n$ and the log-likelihood function $\mathcal{L}_\lambda(\{x_i\}_{i=1}^n) = \frac{1}{n} \log(p_\lambda(\{x_i\}_{i=1}^n))$.
3. Write a program that simulates n such observations sampled from the probability distribution (1) for a fixed $\lambda = 10$. This can be done, for instance, with the numpy.random.exponential command in python, keeping only events x that are in $[1, 20]$. Start with $n = 10$ observations and plot the (log)likelihood as a function of λ. Repeat for $n = 20, 100$, discuss and comment what you see.
4. We now assume that we are given a set of n observations $\{x_i\}_{i=1}^n$, without being told the true value λ^*. We consider the maximum likelihood estimator

   $$\hat{\lambda}_{\text{ML}}(\{x_i\}_{i=1}^n) = \text{argmax}_\lambda \log\left(p_\lambda(\{x_i\}_{i=1}^n)\right) \tag{2}$$

   and we shall define the squared error as $\text{SE} = (\hat{\lambda}_{ML}(\{x\}) - \lambda^*)^2$.
   Create some data sets with $n = 10, 100, 1000$ for different values of $\lambda^* \in (0.05, 50)$ and see how the ML estimator performs. Note that finding the maximizer $\hat{\lambda}_{ML}$ can be done numerically in python, for instance using the scipy.minimize method.
   Hint : we know that $\lambda^* \in \ [0.05, 50]$, *use this in the optimization !*
5. One can show (*Bonus 1 : prove it !*) that the Fisher information in our problem is given by

   $$I(\lambda) = \frac{I_n(\lambda)}{n} = \mathbb{E}\left[\left(\frac{\partial}{\partial\lambda} \log p_\lambda(X)\right)^2\right] = \frac{1}{\lambda^4}\mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \frac{\text{var}[X]}{\lambda^4}. \tag{3}$$

   Knowing this, another interesting estimator is given by maximum as posteriori (MAP) with the Jeffreys prior :

   $$\hat{\lambda}_{\text{J}}(\{x_i\}_{i=1}^n) = \text{argmax}_\lambda \log\left(p_\lambda(\{x_i\}_{i=1}^n)\sqrt{I(\lambda)}\right) \tag{4}$$

   Using scipy, implement this estimator and repeat the analysis done in point 4.
6. If we average of many realizations (say about a hundred) we can obtain numerically the averaged mean squared error $MSE(\lambda^*, \hat{\lambda}, n)$ which is thus a function of n, λ^* and of the estimator $\hat{\lambda}$. Compute and plot, for $n = 10, 100, 1000$, the curves $MSE(\lambda^*, \hat{\lambda}_{\text{ML}}, n)$ and $MSE(\lambda^*, \hat{\lambda}_{\text{J}}, n)$ as a function of λ^*.
7. How do the MSE curves at various n compare with the Cramér-Rao bound for unbiased estimator $\text{MSE}(\hat{\lambda}) \geq \frac{1}{nI(\lambda^*)}$ (where $I(\lambda)$ is the Fisher information) ? How does the Jeffrey and ML estimator behave ? Which one would you choose ?

2 Probability bounds and a pooling problem

We are going to follow the steps we took in lecture 1 and prove an interesting inequality : Let $Z_1, \dots, Z_m$ be independent random variables *Bernoulli*-distributed, i.e. such that $Z_i = 1$ with probability p, and 0 with probability $1-p$. Then, for any $\epsilon \geq 0$ we have

$$\mathbb{P}\left(\frac{1}{m}\sum_i Z_i \geq p+\epsilon\right) \leq e^{-2m\epsilon^2} \tag{5}$$

1. Our starting point is to realize that $\mathbb{P}(a \geq b) = \mathbb{P}(e^{\lambda a} \geq e^{\lambda b})$ for any $\lambda \geq 0$. Using Markov inequality show that :

$$\mathbb{P}\left(\frac{1}{m}\sum_i Z_i \geq p+\epsilon\right) \leq \left(\frac{pe^{\lambda}+(1-p)}{e^{\lambda(p+\epsilon)}}\right)^m \tag{6}$$

   Hint : Since the Z_i are independent, we can write $\mathbb{E}_{\mathbf{Z}}[\exp\sum_i Z_i] = \prod_i \mathbb{E}_{Z_i}[\exp Z_i]$. Use the proof strategy discussed in the lectures.

2. Using the value of λ that minimizes the right-hand-side of the former equation, show that

$$\mathbb{P}\left(\frac{1}{m}\sum_i Z_i \geq p+\epsilon\right) \leq e^{-mf(p,\epsilon)}$$

   with

$$f(p,\epsilon) = (p+\epsilon)\log\left(\frac{p+\epsilon}{p}\right) + (1-(p+\epsilon))\log\left(\frac{1-(p+\epsilon)}{1-p}\right)$$

   Hint : Instead of differentiating the right-hand-side of (6), one can differentiate its log*, since the logarithm is monotone and thus its extrema are the same of its argument.*

3. Show that

$$f(p,\epsilon=0)=0, \; \left.\frac{\partial f(p,\epsilon)}{\partial \epsilon}\right|_{\epsilon=0} = 0, \text{ and that } \frac{\partial^2 f(p,\epsilon)}{\partial \epsilon^2} \geq 4 \text{ for any } \epsilon \text{ s.t. } 0 \leq p+\epsilon \leq 1.$$

4. Use Taylor's theorem (that states that $f(p,\epsilon) = f(p,0) + \epsilon f'(p,0) + \epsilon^2 f''(p,\tilde{\epsilon})/2$ for some unknown $\tilde{\epsilon}$, and where the prime stands for derivative with respect to ϵ) to show that $f(p,\epsilon) \geq 2\epsilon^2$, and prove the inequality (5).

Similarly, it is possible to show (*Bonus 2 : prove it*) that

$$\mathbb{P}\left(\frac{1}{m}\sum_i Z_i \leq p-\epsilon\right) \leq e^{-2m\epsilon^2} \qquad \text{so that} \qquad \mathbb{P}\left(\left|\frac{1}{m}\sum_i Z_i - p\right| \geq \epsilon\right) \leq 2e^{-2m\epsilon^2} \tag{7}$$

5. The most important use of such a bound is in terms of pooling problems. Suppose you want to know what fraction of the population in a country approves its current president : how many people should you ask to be confident, with probability at least 95 percent, that the error in estimating the fraction of people who approves the president is correct within one percent (so that $\hat{p}$ is in $[p-0.01, p+0.01]$ with 95% probability) ?

6. Compare the number m^* you find this way with what you observe when performing numerical experiments in python :
   - Define a function that takes the number of people m and the probability p as arguments and returns a random array of m votes. *Hint : You can generate Bernoulli-distributed samples in python using the* scipy.stats.bernoulli *method.*
   - Starting with fixed values of $m = m^*$ and $p \in \{0.2, 0.5, 0.8\}$
     - Use this function evaluated in m^* and p to simulate polls.
     - Just by averaging the generated votes, estimate p.
     - Quantify the probability that $\hat{p}$ is correct within one percent.
   - Which values of p seem to be harder to estimate ? Do you find that the bound is accurate, or does it grossly overestimate the needed number ?
   - For each p, repeat for different values of m to find the value that (more or less) gives an estimate which is correct within one percent with 95% probability.
   - *Bonus 3* : Plot the behaviour of the probability of error $\mathbb{P}\big(\hat{p} \notin [p-0.01, p+0.01]\big)$ as a function of p for values of $m \in [10, 10^4]$, and compare it with the theoretical m^*.