STAT 339: HOMEWORK 4 (MAXIMUM LIKELIHOOD ESTIMATION)

DUE VIA GITHUB SUNDAY NOVEMBER 14TH

Instructions. Create a directory called `hw4` in your `stat339` GitHub repo. Your main writeup should be called `hw4.pdf`.

You may also use any typesetting software to prepare your writeup, but the final document should be a PDF. LaTeXis highly encouraged.

This assignment requires more coding than the last one, but still much less than HW1 and HW2. Most of it is mathematical derivation, similar to HW3.

I will access your work by cloning your repository; make sure that any file path information is written relative to your repo – don't use absolute paths on your machine, or the code won't run for me!

Date: Last Revised: November 10, 2021.

1. **Coding Probability Computations.** Suppose we have a random variable, Y that has a $\mathsf{Poisson}(\lambda)$ distribution. Recall from the previous homework that this means that Y takes non-negative integers as its values, and its PMF is

$$p_Y(y) = \frac{e^{-\lambda}\lambda^y}{y!}$$

   (a) Write a `poisson_pmf()` function that takes y and λ as inputs and returns $p_Y(y)$, the value of the PMF of Y evaluated at y.

   (b) Write a `poisson_cdf()` function that takes k and λ as inputs and $F_Y(k)$, the value of the CDF of Y evaluated at k.

   (c) Using only a random number generator that generates values continuously uniformly distributed between 0 and 1, write a `poisson_sample()` function that takes inputs λ and n and generates n values sampled independently from a $\mathsf{Poisson}(\lambda)$ distribution. **Hint:** For each draw, generate a random number, u, from a continuous uniform distribution between 0 and 1. Then, check whether

$$u \leq F_Y(k)$$

   starting with $k = 0$. If so, return $Y = k$. If not, increment k (without changing u) and check again. Eventually, $F_Y(k)$ has to get bigger than u (why?), and the algorithm will terminate.

   (d) Show that the probability that this algorithm returns any particular value $Y = k$ is equal to $p_Y(y)$. (**Hint:** find the range of values of u that cause the algorithm to return $Y = k$, and then find the probability that u falls in that range)

   (e) Use your `poisson_sample()` function to generate N independent observations, and compute the mean. Start with $N = 1$, and increment N, recording and then plotting the sequence of means as a function of N. Try this for a few different values of λ. Use your plots to make an educated guess about the relationship between $\mathbb{E}[Y]$ and λ (don't look up the Poisson distribution!).

   (f) Do the same thing with the variance.

2. **Monte Carlo Integration.** Probability can be useful to get numerical solutions to calculus problems that may have nothing to do with probability. Consider the integral

$$\int_{-\infty}^{\infty} \cos(x)e^{-x^2}\ dx$$

You will have a hard time finding an analytic solution to the above without some fairly hefty real analysis tools; however, we can get an approximate numerical solution using random variables.

   (a) Find a function $g(x)$ and parameters μ and σ^2 such that if X has a $\mathcal{N}(\mu, \sigma^2)$ density, then $\mathbb{E}[g(X)]$ is given by the integral above.

   (b) Simulate N independent values of X from this Normal distribution for a reasonably large value of N (say, 10000), and this simulated data to estimate $\mathbb{E}[g(X)]$ by taking advantage of the law of large numbers. **You do not have to write the sampling code yourself**; use a `numpy` function that samples from a Normal distribution. Check that your numerical answer approaches the solution given by an integral solver such as Wolfram Alpha, and report the approximation error (the difference between the true value and the simulated value) as a percentage of the true value of the integral.

3. **Bernoulli MLE.** (Adapted from FCML Ex. 2.9): Assume that a dataset of N binary values, $x_1, ..., x_N$, was produced by sampling independently from a Bernoulli distribution with parameter $\mu := P(X_1 = 1)$.

   (a) **Write out the likelihood function**, $L(\mu; \mathbf{x})$ for μ assuming that we have the full dataset. Be clear about the domain! Remember that the likelihood has the same formula as the joint PMF conditioned on μ: $p(x_1, \ldots, x_N \mid \mu)$, except that we treat it as a function of μ with $x_1, \ldots, x_N$ as constants.

   (b) **Find a formula for the maximum likelihood estimator (MLE)**, $\hat{\mu}$, in terms of $x_1, \ldots, x_N$ (**Hint:** take the natural log of the likelihood first; remember that since log is a strictly increasing transformation, the value of μ that maximizes the log likelihood also maximizes the likelihood.)

4. **Univariate Normal (Gaussian) MLE.** (Adapted from FCML Ex. 2.8) Assume that a dataset $y_1, \ldots, y_N$ consists of N independent draws from a $\mathcal{N}(\mu, \sigma^2)$ distribution.
   (a) Write down the likelihood function for μ and σ^2, $L(\mu, \sigma^2; \mathbf{x})$, based on the full sample of all N observations.
   (b) Find formulas for the pair $(\hat{\mu}, \hat{\sigma}^2)$ that maximize the likelihood L. (**Hint 1:** Remember that the product of exponentials is also the exponential of a sum. **Hint 2:** The parameters that maximize the log likelihood also maximize the likelihood. **Hint 3:** You will need to differentiate the log likelihood separately with respect to μ and σ^2, and set both derivatives to zero simultaneously. You may need to find the MLE of one parameter first in terms of the other and then substitute.)