Homework 2

AI 539 - Machine Learning for Non-AI Majors
Instructor: Alireza Aghasi
Due date: See Canvas

April 19, 2024

Please revise the homework guidelines reviewed in the first lecture. Specifically note that:

- Start working on the homework early
- Late homework **is not accepted** and will receive zero credit.
- Each student must write up and turn in their own solutions
- **(IMPORTANT)** If you solve a question together with another colleague, each need to write up your own solution and **need to list the name of people who you discussed the problem with on the first page of the material turned in**
- The homework should be manageable given the material lectured in the class. The long questions are to help clarifying the problem.

Q1. (a) We take n independent samples $x_1, x_2, \ldots, x_n$ from the distribution:

$$f(x) = \frac{2x}{\lambda^2} \exp\left(-\left(\frac{x}{\lambda}\right)^2\right).$$

The goal is estimating the unknown parameterλ.

(a) Derive the maximum likelihood estimates of λ in terms of the samples $x_1, x_2, \ldots, x_n$. Notice that $\lambda \neq 0$, and you would need this assumption to estimate λ without too much work.

(b) Suppose that $n = 3$, and $x_1 = 1.35, x_2 = 2.76, x_3 = 3.12$. Evaluate your ML estimate $\hat{\lambda}$.

Q2. We take n independent samples $x_1, x_2, \ldots, x_n$ from the distribution

$$f(x) = \frac{\exp\left(-\frac{x-\alpha}{\beta}\right)}{\beta\left(1 + \exp\left(-\frac{x-\alpha}{\beta}\right)\right)^2}.$$

The goal is estimating the parameters α and β based on the observed samples $x_1, x_2, \ldots, x_n$. We would not be able to find closed-form expressions for the ML estimates of α and β, and need to consider a numerical minimization technique. Let's go through this task step by step.

(a) Formulate the likelihood function

$$f(x_1, \ldots, x_n | \alpha, \beta) = \ldots$$

(b) Formulate the negative log-likelihood function

$$L(\alpha, \beta) = -\log\left(f(x_1, \ldots, x_n | \alpha, \beta)\right).$$

To estimate the ML estimates $\hat{\alpha}$ and $\hat{\beta}$ we can minimize the negative log likelihood function $L(\lambda, \alpha)$. For this purpose we decide to use a gradient descent scheme. Derive the expression for the gradient components below:

$$\frac{\partial L}{\partial \alpha} = \ldots, \qquad \frac{\partial L}{\partial \beta} = \ldots. \tag{1}$$

(c) Suppose that $n = 5$, and $x_1 = 0.30753, x_2 = 0.56678, x_3 = -0.25177, x_4 = 0.37243, x_5 = 0.26375$. Use the Matlab or Python 3D surface tools to plot $L(\alpha, \beta)$ as a function of α and β, in the following region:

$$0.2 \leq \alpha \leq 0.4, \qquad 0.1 \leq \beta \leq 0.2.$$

(d) Again, suppose that $n = 5$, and $x_1 = 0.30753, x_2 = 0.56678, x_3 = -0.25177, x_4 = 0.37243, x_5 = 0.26375$. Pick a programming language of your choice and write up a gradient descent (GD) scheme using the gradient components you derived in (1). For your scheme set the gradient learning rate to $\eta = 0.001$, and use the initial values $\alpha_0 = 2$ and $\beta_0 = 2$. Run the GD scheme for 1000 iterates, and report the final estimates of α and β. Also provide a plot showing the iterative values of α for the iterates from 1 to 1000. Provide a similar plot showing the iterative values of β for the iterates from 1 to 1000. Also attach your code.

(e) Use an exact same setup as part (d), but this time in your gradient descent scheme use a momentum term of $\gamma = 0.9$. Report the final estimates of α and β. Also provide a plot showing the iterative values of α for the iterates from 1 to 1000. Provide a similar plot showing the iterative values of β for the iterates from 1 to 1000. Comparing these plots with those in part (d), which scheme seems to have converged faster? Attach your code.