CS 559: Homework Set 2
Due: Nov. 6th, 11:59pm

Collaboration Policy. Homeworks will be done individually: each student must hand in their own answers. Use of partial or entire solutions obtained from others or online is strictly prohibited.

Late Policy. No late submissions will be allowed without consent from the instructor. If urgent or unusual circumstances prohibit you from submitting a homework assignment in time, please e-mail me explaining the situation.

Submission Format. Electronic submission on Canvas is mandatory.

Problem 1. (30 points) Let x have an exponential density

$$p(x|\theta) = \begin{cases} \theta e^{-\theta x} & x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Suppose that n samples $x_1, ..., x_n$ are drawn independently according to $p(x|\theta)$. Show that the maximum-likelihood estimate for θ is given by

$$\hat{\theta} = \frac{1}{\frac{1}{n}\sum_{k=1}^{n} x_k}.$$

Problem 2. (40 points) Download the "Pima Indians Diabetes Database" from Canvas.

(a) Implement a classifier using Maximum-Likelihood Estimation that takes into account features 2 to 4, among the 8 available features.

(b) Train the classifier on the same samples and run them 10 or more times. Record the mean and standard deviation of the accuracy. Use 50% of the data for training and the rest for testing. Make sure that the two sets are disjoint.

(c) Submit code, but not data, taking into account the assumptions made.

Hints:

- The $cov()$ command in Matlab can be used to compute the necessary covariance matrices.
- You can choose any programming language, but you will need to be able to compute the inverse and the determinant of 3×3 matrices. You will also need to randomly split into training and test sets multiple times.

Problem 3. (30 points) Use the "Pima Indians Diabetes Database" and implement a k-Nearest Neighbor classifier. Split the data in half to form the training and test sets and use features 2 to 4 as above. Report mean accuracy for k=1, 5 and 11, as well as its standard deviation, over at least 10 trials for each value of k.

Hints:

- The $knnsearch()$ command in Matlab can be used to find the nearest neighbors.

Submit the code, but no data, printouts or screenshots.