Homework #1

ECE 461/661: Introduction to Machine Learning for Engineers
Prof. Gauri Joshi
Due: Thursday, January 24, 2019 at 11:59:59pm ET

Please remember to show your work for all problems and to write down the names of any students that you collaborate with. The full collaboration and grading policies are available on the course website: https://andrew.cmu.edu/course/18-661/.

Your solutions should be uploaded to Gradescope (https://www.gradescope.com/) in PDF format by the deadline. We will not accept hardcopies. If you choose to hand-write your solutions, please make sure the uploaded copies are legible. Gradescope will ask you to identify which page(s) contain your solutions to which problems, so make sure you leave enough time to finish this before the deadline. We will give you a 30-minute grace period to upload your solutions in case of technical problems.

1 Warmup [20 points]

a. Multivariable Calculus: Consider $y = x \cos(z)e^{-3x+z}$. What is the partial derivative of $y$ with respect to $x$?

b. Mean and Variance: If the variance of a zero-mean random variable $X$ is $\sigma^2$, what is the variance of $2X$? What about the variance of $X + 2$?

c. Probability: Consider the following joint distribution between $X$ and $Y$. What is $P(X = T|Y = b)$?

<table>
<thead>
<tr>
<th>$P(X,Y)$</th>
<th>$a$</th>
<th>$b$</th>
<th>$c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(X = T</td>
<td>Y = b)$</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>$P(X = F</td>
<td>Y = b)$</td>
<td>0.05</td>
<td>0.15</td>
</tr>
</tbody>
</table>

d. Divide and Conquer: Given an array with $n$ elements with all entries equal either to 0 or +1 such that all 0 entries appear before +1 entries, i.e. is of the form \{0,0,\ldots,0,1,1\ldots,1\}. We want to find the index where the transition happens, i.e., the index with the last occurrence of 0.

Give an algorithm that runs in time $O(\log n)$. Explain your algorithm in words, describe why the algorithm is correct, and justify its running time.

2 Convex Functions and Information Theory [10 points]

a. Show that the function $f(x) = |x| + \exp(x)$ is convex.

b. Suppose the random variable $X$ is distributed according to a $k$-class multi-nominal distributions with class probabilities $p_1, p_2, \ldots, p_k$, such that $\sum_{i=1}^{k} p_i = 1$. Find the values of $p_i, i = 1, \ldots, k$ such that the entropy of $X$ is maximized. Note entropy is defined as $H(X) = -\sum_{i=1}^{k} p_i \log p_i$
3 Linear algebra [10 points]

a. The covariance matrix $\Sigma$ of a random vector $X$ is defined as $\Sigma = \mathbb{E}[(X - \mathbb{E}X)(X - \mathbb{E}X)^\top]$, where $\mathbb{E}X$ is the expectation of $X$. Is $\Sigma$ positive-semidefinite?

b. Let $A$ and $B$ be two $\mathbb{R}^{D\times D}$ symmetric matrices. Suppose $A$ and $B$ have the exact same set of eigenvectors $u_1, u_2, \ldots, u_D$ with the corresponding eigenvalues $\alpha_1, \alpha_2, \ldots, \alpha_D$ for $A$, and $\beta_1, \beta_2, \ldots, \beta_D$ for $B$. Write down the eigenvectors and their corresponding eigenvalues for the following matrices:

- $C = A + B$
- $D = A - B$
- $E = AB$
- $F = A^{-1}B$ (assume $A$ is invertible)

4 Vector-valued functions [15 points]

Compute the first and second derivatives of the following functions:

a. $f(x) = c^\top x$, where $x, c \in \mathbb{R}^m$.

b. $f(x) = \frac{1}{2}M^\top Mx$, where $x \in \mathbb{R}^m$ and $M \in \mathbb{R}^{m \times m}$. What happens if $M = M^\top$?

c. $f(X) = \text{tr}(MX)$, where $M \in \mathbb{R}^{m \times n}$ and $X \in \mathbb{R}^{n \times m}$.

5 Python [20 points]

Follow the instructions below and use the included Python code along with your own code to solve the Lighthouse problem.

When uploading to Gradescope, you will need to produce a PDF version of your solutions and code. One way to do this is to use a notebook (https://jupyter.org); if you wish to use this, we have provided a Jupyter version of the problem where you can fill in your solutions in Homework1.ipynb.
A lighthouse is somewhere off a piece of straight coastline at a position $\alpha$ along the shore and a distance $\beta$ out at sea. It emits a series of short highly collimated flashes at random intervals and hence at random azimuths. These pulses are intercepted on the coast by photo-detectors that record only the fact that a flash has occurred, but not the angle from which it came. $N$ flashes have been recorded so far at positions $\{x_k\}$.

Suppose $\beta$ is given. Where is the lighthouse?

**Guided solution**

We need to estimate the parameter $\alpha$. Let us start by writing the likelihood for this problem; since the flashes are thrown at random azimuths, we know that:

$$P(\theta_k | \alpha, \beta) = \frac{1}{\pi}.$$ 

Moreover,

$$\beta \tanh(\theta_k) = x_k - \alpha,$$

and by changing variables we get

$$P(x_k | \alpha, \beta) = \frac{\beta}{\pi \left[ \beta^2 + (x_k - \alpha)^2 \right]}.$$
# Scientific computing and plotting packages
import numpy as np
import matplotlib.pyplot as plt

# Likelihood definition
def likelihood(x, alpha, beta):
    return beta / (np.pi * (beta ** 2 + (x - alpha) ** 2))

# Parameters
alpha = 30.0  # alpha appears here, only for simulations purposes, we want to find the value of this parameter
beta = 10.0  # beta is given

# Compute and plot the likelihood
x = np.linspace(-90, 90, 1001)
plt.plot(x, likelihood(x, alpha, beta))

Out[1]:
[<matplotlib.lines.Line2D at 0x11bf18320>]

The above likelihood is the a Cauchy or Lorentz distribution. We will sample from it so that we can have some synthetic data to work with.

**Generate some synthetic data**

In [2]:

from scipy.stats import cauchy
samples = cauchy.rvs(loc = alpha, scale = beta, size = 1000)

In order to write the posterior we use Bayes theorem

\[ P(\alpha|\{x_k\}, \beta) \propto \prod_{k=1}^{N} P(\{x_k\}|\alpha, \beta) \]
In [3]:

```python
# Computes the (unnormalized) posterior for a given set of samples

def posterior(x, alpha, beta):
    post = np.ones(len(alpha))
    for x_k in x:
        post *= likelihood(x_k, alpha, beta)
        post /= np.sum(post)
    return post

def plot_posterior(n_samples):
    alphas = np.linspace(0, 60, 1001)
    plt.plot(alphas, posterior(samples[:n_samples], alphas, beta))
    plt.axvline(np.mean(samples[:n_samples]), c = "r", lw = 2)

plot_posterior(10)
```

Exercise 1: Create 4 subplots for different values of $N = 2, 5, 20, 100$. 
In [4]:
```python
fig, axs = plt.subplots(2, 2)
fig.tight_layout()
alphas = np.linspace(0, 60, 1001)
# Your solution goes here
```

Note the mean does not coincide with the mode of the posterior!

Why is that? Will they coincide in the $N \to \infty$ limit?

Now compute the value of $\alpha$ that maximizes the posterior (and the likelihood, since our prior here is uniform). The log-likelihood reads:

$$\mathcal{L}(\alpha) = \sum_k \log P(x_k | \alpha, \beta) = -\sum_k \log[\beta^2 + (x_k - \alpha)^2] + c,$$

where $c$ is a constant.

Hence the maximum is obtained at

$$2 \sum_k \frac{x_k - \alpha^*}{\beta^2 + (x_k - \alpha^*)^2} = 0.$$

Now let’s solve this numerically for different values of $N$.

**Exercise 2:** Plot the ML estimate of $\alpha$ for $N$ between 10 and 1000.

In [5]:
```python
# Use a off the shelf method to find a root of a function on an interval - ex: bisect, brentq, brent, ridder
from scipy.optimize import bisect # Bisection method is probably the simpler to understand
# Your solution goes here
```