

Homework Assignment #1

Instructor: Xie, Yao

Name: Chen-Yang(Jim), Liu GTID: 903450732

Problem 1: Algorithms

(10+20+5+5+20=60 points)

(a) Simple questions (10 pts/2.5 pts each question.)

- What does algorithm efficiency mean? What are two types of algorithm efficiency measures?

1. Algorithm Efficiency:

A usual method to appraise the quality of algorithm efficiency is speed. That is, we would like to know how long an algorithm runs to produce its result.

2. Two Types:

- (a) Space Efficiency
the memory we need to store an algorithm
- (b) Time Efficiency
the time required to go through a series of steps of an algorithm
- What does algorithm robustness mean? Given one example of robust algorithm.
A robust algorithm can be applied to a wide range of data. An example of this is robust optimization.

$$\min_x c^T x : a_i \in \mu_i, a_i^T \leq b_i, i = 1, \dots, m$$

This is a Robust Linear Programming which can be used when data is uncertain and a solution still can be derived.

- What does algorithm stability mean? What's the difference of algorithm stability and robustness?
Stable algorithm means that a little perturbation does not affect big difference in deriving solutions. The key difference between robustness and stability is that robust program will detect inappropriate data input for both the algorithm and the implementation of the program. On the other hand, stable algorithms run appropriate data input and then a small difference in data input does not create largely different solutions.
- Given two commonly seen definition of algorithm accuracy. Why do we sometimes prefer approximate algorithms?
Although some algorithms are exact, other algorithms are usually approximate since some of computed results does not have specific closed-form solution. For example, Taylor expansion says a function could be approximated by multiple derivatives of functions. That is an approximate algorithm.

(b) Bisection

```
from scipy.stats import t
import matplotlib.pyplot as plt
def bisection(f,a,b):
    '''Approximate solution of f(x)=0 on interval [a,b] by bisection method.
    -----
    f : function
        The function for which we are trying to approximate a solution f(x)=0.
    a,b : numbers
        The interval in which to search for a solution. The function returns
        None if f(a)*f(b) >= 0 since a solution is not guaranteed.
```

Returns

x_N : number

The midpoint of the Nth interval computed by the bisection method. The initial interval [a₀, b₀] is given by [a, b]. If f(m_n) == 0 for some midpoint m_n = (a_n + b_n)/2, then the function returns this solution. If all signs of values f(a_n), f(b_n) and f(m_n) are the same at any iteration, the bisection method fails and return None.

'''

```

if f(a)*f(b) >= 0:
    print("Bisection method fails.")
    return None
a_n = a
b_n = b
iteration = 0
while (b - a > (10 ** -4)):
    m_n = (a_n + b_n)/2
    f_m_n = f(m_n)
    if f(a_n)*f_m_n < 0:
        a_n = a_n
        b_n = m_n
    elif f(b_n)*f_m_n < 0:
        a_n = m_n
        b_n = b_n
    elif f_m_n == 0:
        print("Found exact solution.")
        print("Iteration times: ", iteration)
        print("Interval is: ", a_n, b_n)
        return 'Solution point is: ', m_n
    else:
        print("Bisection method fails.")
        print("Iteration times: ", iteration)
        return None
    iteration += 1
return (a_n + b_n)/2

```

```

f = lambda x: t.cdf(x, df = 5, loc=0, scale=1) - 0.95
bisection(f, 1.291, 2.582)

```

```

Found exact solution.
('Iteration times: ', 48)
('Interval is: ', 2.0150483733330207, 2.0150483733330256)

```

(c) Worst-case complexity of quicksort

Worst case means: Given a strictly decreasing sequence with n numbers, i.e. $e_i < e_j$ for $i < j$ where $i, j = 1, 2, \dots, n$. Now pick e_n as a pivot and from the following calculation, $O(n^2)$ is time complexity of the worst case.

$$\begin{aligned}
 e_n &= n - 1 + \sum_{i=1}^{n-1} e_i \\
 &= n - 1 + n - 2 + \sum_{i=2}^{n-1} e_i \\
 &= n - 1 + n - 2 + \dots + 1 \\
 &= \frac{n(n-1)}{2} = \frac{n^2 - n}{2}
 \end{aligned}$$

(d) Fourier transform of a delayed signal

$$\begin{aligned}
F(x(t - \tau)) &= F(x(t - \tau) * h) \text{ where } h = 1 \\
&= \iint x(t - \tau) e^{-i2\pi f t} d\tau dt \\
&= \int \left[\int x(t - \tau) e^{-i2\pi f t} dt \right] d\tau \\
&= X(f) \int e^{-i2\pi f \tau} * 1 d\tau \\
&= X(f) e^{-i2\pi f \tau} \text{ where } \int e^{-i2\pi f \tau} * 1 d\tau = 1
\end{aligned}$$

(d) Fourier transform of a delayed signal

$$\begin{aligned}
F(x(t - \tau)) &= F(x(t - \tau) * h) \text{ where } h = 1 \\
&= \iint x(t - \tau) e^{-i2\pi f t} d\tau dt \\
&= \int \left[\int x(t - \tau) e^{-i2\pi f t} dt \right] d\tau \\
&= X(f) \int e^{-i2\pi f \tau} * 1 d\tau \\
&= X(f) e^{-i2\pi f \tau} \text{ where } \int e^{-i2\pi f \tau} * 1 d\tau = 1
\end{aligned}$$

(e) Steps for deriving FFT

- (i) e_n : x_{2n} is even-indexed samples and o_n : x_{2n+1} is odd-indexed samples Given $x_n = 0$ if $n \notin [0, N-1]$ and N is even Shift coverage, i.e. $n \notin [1, N]$ Since $e_n + o_n = x_{2n} + x_{2n+1} = 0$ i.e. x_{2n} where $N/2$ samples are not equal to 0, $x_{2n} = 0$ if $n \notin [1, N/2]$ Then, we shifted coverage again. We get $n \notin [0, N/2 - 1]$
- (ii) Goal: $\tilde{x} = \sum_{j=0}^{N-1} x_j e^{-i\frac{2\pi}{N}jk} = \sum_{j=0}^{N-1} x_j W_N^k$

$$\begin{aligned}
\tilde{x} &= \sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi}{N}nk} \\
&= \sum_{n=0, \text{even}}^{N/2-1} x_n e^{-i\frac{2\pi}{N}nk} + \sum_{n=0, \text{odd}}^{N/2-1} x_n e^{-i\frac{2\pi}{N}nk} \\
&= \sum_{n=0}^{N/2-1} x_{2n} e^{-i\frac{2\pi}{N}2nk} + \sum_{n=0}^{N/2-1} x_{2n+1} e^{-i\frac{2\pi}{N}(2n+1)k} \\
&= \sum_{n=0}^{N/2-1} e_n e^{-i\frac{2\pi}{N/2}nk} + \sum_{n=0}^{N/2-1} o(n) e^{-i\frac{2\pi}{N/2}nk} e^{-i\frac{2\pi}{N}k} \\
&= \frac{1}{2} \sum_{n=0}^{N/2-1} 2e_n (W_{N/2})^{kn} + \frac{1}{2} W_N \sum_{n=0}^{N/2-1} 2o_n (W_{N/2})^{kn} \\
&= \frac{1}{2} \tilde{E}_k + \frac{1}{2} W_N^k \tilde{O}_k
\end{aligned}$$

(iii)

$$\begin{aligned}
\tilde{E}_{k+N/2} &= 2 \sum_{n=0}^{N/2-1} e_n W_{N/2}^{n(k+N/2)} \\
&= 2 \sum_{n=0}^{N/2-1} e_n W_{N/2}^{nk} W_{N/2}^{\frac{nN}{2}} \\
&= 2 \sum_{n=0}^{\frac{N}{2}-1} e_n e^{-i\frac{2\pi}{N}nk} e^{-i\frac{2\pi}{N}\frac{nN}{2}}
\end{aligned}$$

Now consider $e^{-i\frac{2\pi}{N}\frac{nN}{2}}$ and by Euler formula,

$$\begin{aligned}
e^{-i\frac{2\pi}{N}\frac{nN}{2}} &= e^{-i2n\pi} \\
&= \cos(2n\pi) - i\sin(2n\pi) \\
&= 1 - 0 = 1
\end{aligned} \tag{1}$$

By (1),

$$2 \sum_{n=0}^{\frac{N}{2}-1} e_n e^{-i\frac{2\pi}{N}nk} e^{-i\frac{2\pi}{N}\frac{nN}{2}} = 2 \sum_{n=0}^{\frac{N}{2}-1} e_n e^{-i\frac{2\pi}{N}nk} = \tilde{E}_k$$

Similarly,

$$\begin{aligned}
\tilde{O}_{k+N/2} &= 2 \sum_{n=0}^{\frac{N}{2}-1} o_n e^{-i\frac{2\pi}{N}nk} \\
&= \tilde{O}_k
\end{aligned}$$

Problem 2: Basic linear algebra and statistical inference

(5+20+10+5=40 points)

(a) Rank of a product

$$A_{4 \times 3} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix}, B_{3 \times 5} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} \end{bmatrix} AB_{4 \times 5} = [Ab_1 \quad Ab_2 \quad Ab_3 \quad Ab_4 \quad Ab_5]$$

Now we could investigate some cases about AB matrix from the given assumption,

- B is singular B is singular $\Rightarrow \text{rank}(AB) \leq \text{rank}(B)$
- A is singular A is singular $\Rightarrow \text{rank}(AB) \leq \text{rank}(A)$

Therefore, $\text{rank}(AB)$ depends on whether A or B is singular. i.e. $\text{rank}(AB) \leq \min(\text{rank}(A), \text{rank}(B))$ Back to question, value r is 2.

(b) Simple Bayesian inference

1. Prior: $p(\mu) \sim N(\theta, \tau^2)$, Observed data $x \sim N(\mu, \sigma^2)$

Those two distributions are i.i.d. And posterior distribution is:

$$\begin{aligned} f(\mu | x) &= \frac{f(x | \mu)p(\mu | \tau)}{f(\mu)} \\ f(\mu | x) &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}} \times \frac{1}{\sqrt{2\pi}\tau} e^{-\frac{(\mu - \theta)^2}{2\tau^2}} \\ &\propto e^{-\frac{1}{2} \sum_{i=1}^n \frac{(x_i - \mu)^2}{\sigma^2} + \frac{(\mu - \theta)^2}{\tau^2}} \\ &= e^{-\frac{1}{2} \left(\frac{\sum_{i=1}^n x_i^2 - 2\mu \sum_{i=1}^n x_i + n\mu^2}{\sigma^2} + \frac{\mu^2 - 2\mu\theta + \theta^2}{\tau^2} \right)} \\ &= e^{-\frac{1}{2} \left[\left(\frac{n}{\sigma^2} + \frac{1}{\tau^2} \right) \mu^2 - 2 \left(\frac{n\bar{x}}{\sigma^2} + \frac{\theta}{\tau^2} \right) \mu + \frac{\sum_{i=1}^n x_i^2}{\sigma^2} + \frac{\theta^2}{\tau^2} \right]} \\ &\propto e^{-\frac{1}{2} \frac{(\mu - \mu_{\text{posterior}})^2}{\tau_{\text{posterior}}^2}} \\ &= e^{-\frac{1}{2} \frac{\mu^2 - 2\mu\mu_{\text{posterior}} + \mu_{\text{posterior}}^2}{\tau_{\text{posterior}}^2}} \quad \text{where } n = 1 \text{ i.e. } x = x_1 \\ \frac{1}{\tau_{\text{posterior}}^2} &= \frac{1}{\sigma^2} + \frac{1}{\tau^2} \implies \tau_{\text{posterior}} = \frac{\tau^2 \sigma^2}{\tau^2 + \sigma^2} \end{aligned}$$

$$\begin{aligned} \frac{\mu_{\text{posterior}}}{\tau_{\text{posterior}}^2} &= \frac{x}{\sigma^2} + \frac{\theta}{\tau^2} \implies \mu_{\text{posterior}} = \frac{\tau^2}{\tau^2 + \sigma^2} x + \frac{\sigma^2}{\tau^2 + \sigma^2} \theta \\ &\sim N\left(\frac{\tau^2}{\tau^2 + \sigma^2} x + \frac{\sigma^2}{\tau^2 + \sigma^2} \theta, \frac{\tau^2 \sigma^2}{\tau^2 + \sigma^2}\right) \end{aligned}$$

2. Prior: $p(\mu) \sim \text{Uniform}(0, 1)$, observed data $x \sim N(\mu, \sigma^2)$

Those two distributions are i.i.d. And posterior distribution is:

$$\begin{aligned} f(\mu | x) &= \frac{f(x | \mu)p(\mu | \tau)}{f(\mu)} \\ f(\mu | x) &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}} \times 1 \\ &\sim N(\mu, \sigma^2) \end{aligned}$$

(c) Maximum likelihood estimator

(i) Find MLE

$$\begin{aligned}
L(a, b | x) &\triangleq \prod_{i=1}^n f(x) = \left(\frac{1}{b-a}\right)^n I(a, b)^n \\
&= \left(\frac{1}{b-a}\right)^n I(a \leq x_i) I(x_i \leq b)
\end{aligned}$$

According to MLE, our goal is to find a parameter that most likely occurs.
Now, we have some cases to consider:

$$\hat{a} = \arg \max_a L(a, b | x) \implies \hat{a} = X_{(1)} \text{ [order statistics, } b \text{ fixed]}$$

$$\hat{b} = \arg \max_b L(a, b | x) \implies \hat{b} = X_{(n)} \text{ [order statistics, } a \text{ fixed]}$$

(ii) Are \hat{a}, \hat{b} unbiased estimators?**(d) Hypothesis test of the mean**

Hypotheses Testing:

$$H_0 : \mu = \mu_0 = 75 \text{ [Null Hypothesis]} \quad H_1 : \mu = \mu_1 < \mu_0 \text{ [Alternative Hypothesis]}$$

In this case, we try to know additive is effective or not. H_0 means new additive does not have any effect.
Likelihood Ratio Test:

$$\begin{aligned}
\lambda(\mu_0, \mu_1) &= \prod_{i=1}^{25} \frac{1}{\sqrt{2\pi}9} e^{\frac{-(x_i - \mu_1)^2}{2 \times 9^2}} / \prod_{i=1}^{25} \frac{1}{\sqrt{2\pi}9} e^{\frac{-(x_i - 75)^2}{2 \times 9^2}} \\
&= e^{(-\sum_{i=1}^{25} (x_i - \mu_1) + \sum_{i=1}^{25} (x_i - 75)^2) / 2 \times 9^2} > b \\
\ell(\lambda(\mu_0, \mu_1)) &= -\sum_{i=1}^{25} (x_i - \mu_1) + \sum_{i=1}^{25} (x_i - 75)^2 > b \times 2 \times 9^2 = b_1 \\
&\implies \sum_{i=1}^{25} -2\mu_1 x_i + \sum_{i=1}^{25} -150x_i + 75^2 - \mu_1^2 > b_1 \\
&\implies \sum_{i=1}^{25} x_i (-2\mu_i - 150) > b_1 - 75^2 + \mu_1^2 = b_2 \text{ where } -2\mu_i - 150 < 0 \\
&\implies (\sum_{i=1}^{25} x_i) / 25 < b_2 (-2\mu_i - 150) / 25 = b_3 \\
&\text{s.t. } P(\bar{x} < b_3 | \mu = \mu_0) = \alpha = 0.05 \\
&\text{i.e. } P(\bar{x} - \mu_0 / \sqrt{18/25} < -Z_{0.05, n=25} | \mu = \mu_0) = 0.05 \\
&\text{i.e. } P(\bar{x} - \mu_0 / \sqrt{18/25} < -Z_{0.05, n=25} | \mu = \mu_0) = 0.05 \\
&\text{i.e. } P(\bar{x} - \mu_0 / \sqrt{18/25} < -1.645 | \mu = \mu_0) = 0.05
\end{aligned}$$

From the above calculation, the threshold is -1.645.