Hwk 6,

CSE 250A

6.1

=) so we. get

$P(a, b \mid c, d$

(

d) As discussed

A is a root no

$$P(a) = \frac{1}{T} \sum_{t}^{T}$$

D has B, C as paren

$P(d \mid b, c) = \sum_{t} P$

6.2

a) $\sum_{z} P(Y=1,Z$

b) $\mathcal{L} = \frac{1}{T} \sum_{t} \ell$

$x \in \{0,1\}^{n}$ $\quad y$

$P(Z_i = 1, X_i = 1 \,|$

$$= \begin{cases} 0 \\ \frac{p_i}{1 -} \end{cases}$$

6.3

a) $f(x) = \log$

$f'(x) = \sinh$

range of sech

$\mathrm{sech}^2(x) <$

c). Python.

e) $x_{n+1} = \text{argm}$

$= \text{arg mi}$

$= \text{argmi}$

CSE250Hw6

November 11, 2022

#Q2-d

[ ]:

```
import numpy as np
import warnings
import matplotlib.pyplot as plt
warnings.filterwarnings("ignore")
```

[ ]:

```
X = np.loadtxt("X.txt")
Y = np.loadtxt("Y.txt")
```

[ ]:

```
niter = 257
T = 267
n = 23

def likelihood(p, x, y):
  temp = np.prod((1-p)**x)
  return (1-y)*temp + y*(1-temp)

def conv(p, x, y):
  numi = p*y*x
  deno = 1-np.prod((1-p)**x)
  return numi/deno

Ti = []
for i in range(n):
  Ti.append(np.sum(X[:,i]))
```

[ ]:

```
printed = [2**i for i in range(9) ]
printed.insert(0,0)
def EM(X,Y):
  p = np.array([0.05]*n)
  Nmistakes = []
  LogL = []
  for i in range(niter):
    logl = 0
    nMis = 0
    eSum = 0
    for k in range(267):
      prob = likelihood(p,X[k],Y[k])
      logl += np.log(prob)
      eSum += conv(p,X[k],Y[k])
      if prob <= 0.5:
        nMis += 1
    p = eSum/Ti
    Nmistakes.append(nMis)
    logl /= 267
    LogL.append(logl)

    if i in printed:
      print(i, "\t" ,"\t", nMis, "\t" ,"\t" ,"\t" , logl )

  return Nmistakes,LogL
```

[ ]:

```
print("Iteration No ", "\t" , "Number of Mistakes " , "\t" , "LogLikelihood")
mLis, logLis = EM(X, Y)
```

```
Iteration No     Number of Mistakes      LogLikelihood
0                175                     -0.9580854082157914
1                56                      -0.49591639407753635
2                43                      -0.40822081705839114
4                42                      -0.3646149825001877
8                44                      -0.3475006162087826
16               40                      -0.33461704895854844
32               37                      -0.32258140316749784
64               37                      -0.3148266983628559
128              36                      -0.3111558472151897
256              36                      -0.310161353474076
```

#Q3-c

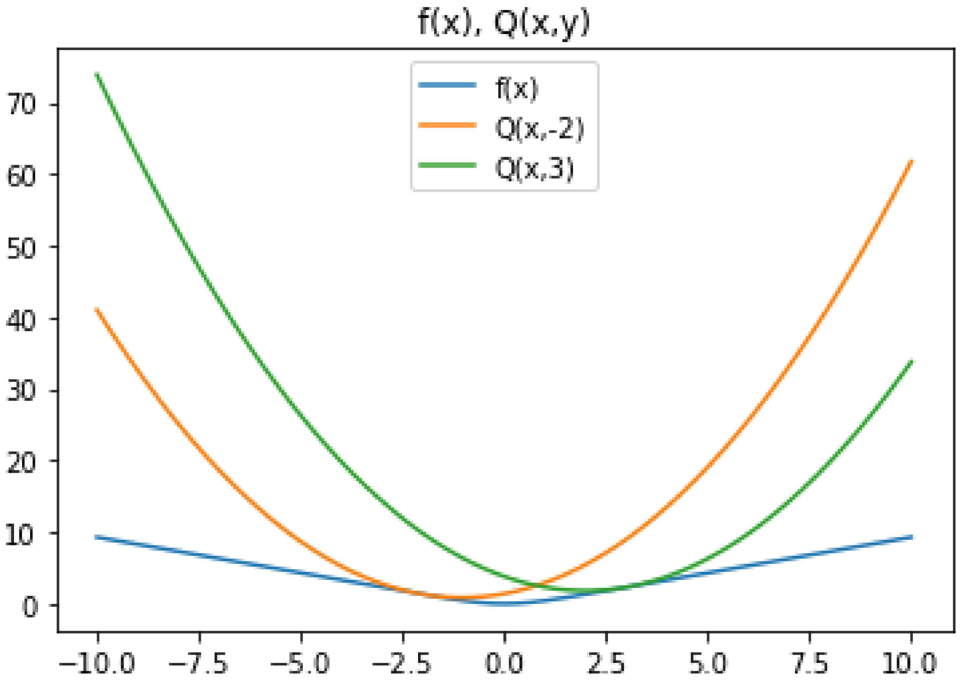
[ ]:

```
def f(x):
    return np.log(np.cosh(x))

def df(x):
    return np.tanh(x)

def Q(x,y):
    return f(y) + df(y)*(x-y) + ((x-y)**2)/2
```

[ ]:

```
x = np.linspace(-10,10,10000)
plt.plot(x, f(x), label="f(x)")
plt.plot(x, Q(x,-2), label="Q(x,-2)")
plt.plot(x, Q(x,3), label="Q(x,3)")
plt.title('f(x), Q(x,y)')
plt.legend()
```

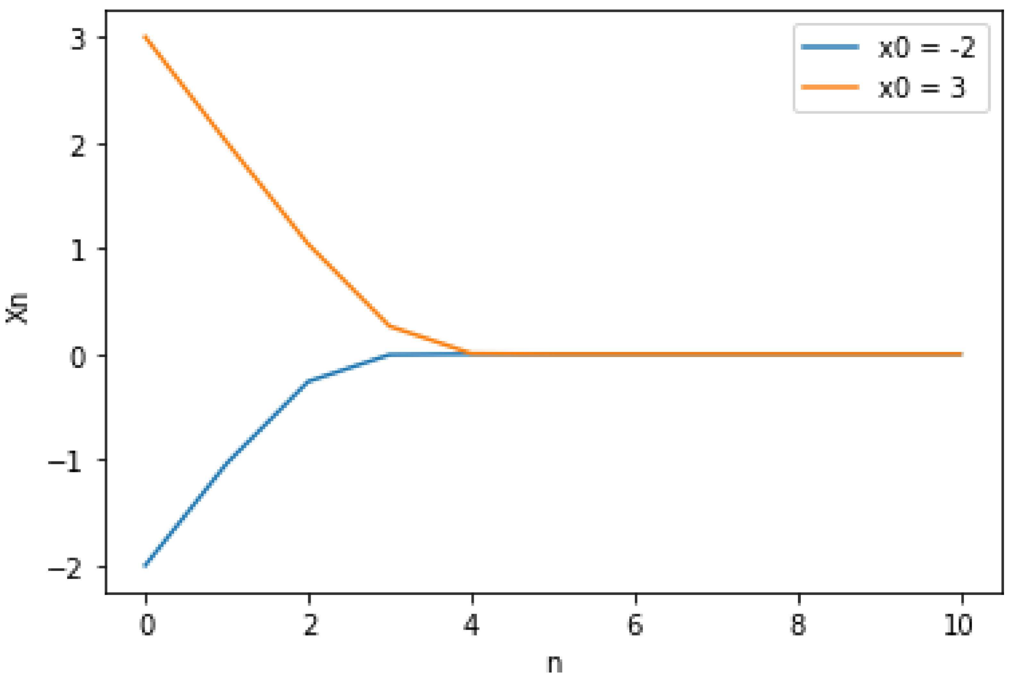
```
[ ]: <matplotlib.legend.Legend at 0x7fe9198eda90>
```



#Q3-f

```
[ ]: def newUpdated(x0, n):
       xn = [x0]
       for i in range(len(n)-1):
           xn.append(xn[-1] - df(xn[-1]))
       return xn

     n = np.arange(0, 11)
     x1 = newUpdated(-2, n)
     x2 = newUpdated(3, n)

     plt.plot(n, x1, label="x0 = -2")
     plt.plot(n, x2, label="x0 = 3")
     plt.ylabel('Xn')
     plt.xlabel('n')
     plt.legend()
```

```
[ ]: <matplotlib.legend.Legend at 0x7fe9195ee750>
```


3
2
1
0
−1
−2
Xn
0
2
4
6
8
10
n
x0 = -2
x0 = 3