HW6_P3

November 12, 2018

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
# Problem 3.5
```

In [3]:
```
d = 1
n = 500
k = 2
pi = np.array([0.7, 0.3])
w = np.array([-2, 1])
b = np.array([0.5, -0.5])
sigma = np.array([0.4, 0.3])
X = np.random.uniform(0, 1, n)
mu = 0.5#np.mean(X)
#mu = mu * w + b
```

In [4]:
```
y = np.zeros(n)
for l in range(k):
    for i in range(n):
        y[i] += pi[l] * (np.random.normal(w[l]*X[i] + b[l], sigma[l], 1))
```

In [5]:
```
def E_step(n, k, mu_hat, y, pi_hat, w_hat, b_hat, sigma_square_hat):
    W_temp = np.zeros((n, k))
    W = np.zeros((n, k))
    for l in range(k):
        for i in range(n):
            #W_temp[i][l] = pi_hat[l] * (1/(2*np.pi*(sigma_square_hat[l]))**0.5) * np.
            W_temp[i][l] = pi_hat[l] * (1/(2*np.pi*(sigma_square_hat[l]))**0.5) * np.ex

    for i in range(n):
        W[i] = W_temp[i] / sum(W_temp[i])

    return W
```

In [59]:
```
def M_step(n, k, mu, y, pi_hat, w_hat, b_hat, sigma_hat, W):

    # calculate pi
    for l in range(k):
```

```
            pi_hat[l] = sum(W[:,l]) / n

        '''# calculate w
        w_hat_new = np.zeros(k)
        w_hat_temp = np.zeros((n, k))
        for l in range(k):
            w_hat_temp[:,l] = W[:,l] * (X - b_hat[l])
            w_hat_new[l] = sum(w_hat_temp[:,l]) / (mu * sum(W[:,l]))

        # calculate b
        b_hat_new = np.zeros(k)
        b_hat_temp = np.zeros((n, k))
        for l in range(k):
            b_hat_temp[:,l] = W[:,l] * (X - w_hat[l]*mu)
            b_hat_new[l] = sum(b_hat_temp[:,l]) / sum(W[:,l])'''


        mu_hat = np.zeros(k)
        mu_hat_temp = np.zeros((n, k))
        for l in range(k):
            mu_hat_temp[:,l] = W[:,l] * y
            mu_hat[l] = sum(mu_hat_temp[:,l]) / sum(W[:,l])

        w_hat = (mu_hat - b_hat)/mu
        b_hat = mu_hat - w_hat*mu

        # calculate sigma square!
        sigma_square_hat_temp = np.zeros((n, k))
        for l in range(k):
            #sigma_square_hat_temp[:,l] = W[:,l] * ((X - w_hat_new[l]*mu - b_hat_new[l])*
            #sigma_square_hat_temp[:,l] = W[:,l] * ((y - w_hat[l]*mu - b_hat[l])**2)
            sigma_square_hat_temp[:,l] = W[:,l] * ((y - mu_hat[l])**2)
            sigma_square_hat[l] = sum(sigma_square_hat_temp[:,l]) / sum(W[:,l])
        #return pi_hat, w_hat_new, b_hat_new, sigma_square_hat, mu_hat
        return pi_hat, w_hat, b_hat, sigma_square_hat, mu_hat
```

In [60]:
```
# main function

# initializing parameters
pi_hat = np.array([0.5, 0.5])
w_hat = np.array([1, -1])
b_hat = np.array([0, 0])
sigma_square_hat = np.array([np.std(y), np.std(y)])**2
log_like = []

log_like_prev = -1000
increase = 1
mu_hat = w_hat * mu + b_hat
```

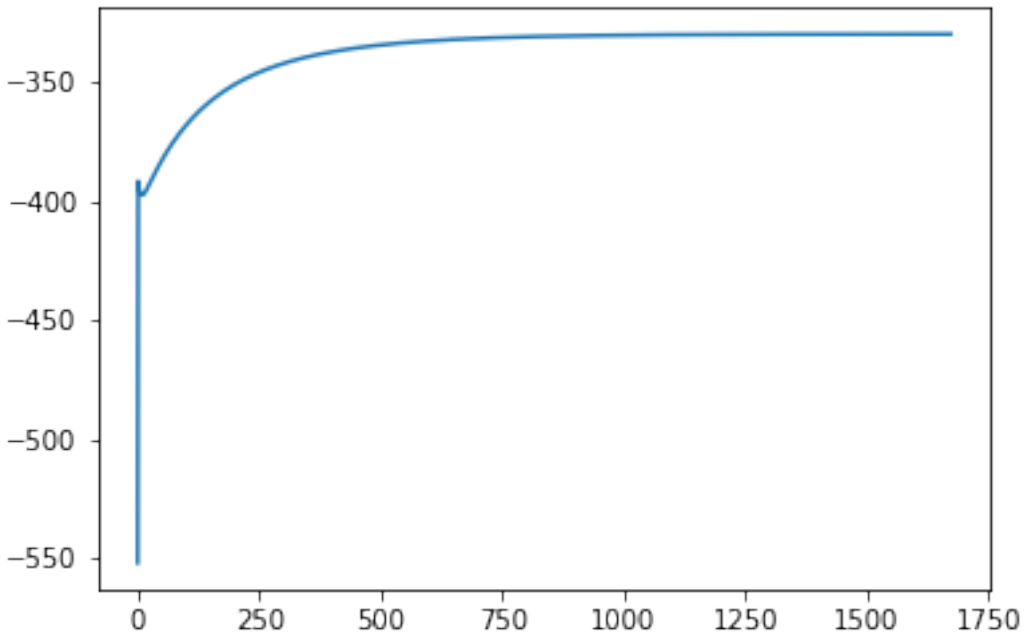
```
while abs(increase) > 1e-04:
    W = E_step(n, k, mu_hat, y, pi_hat, w_hat, b_hat, sigma_square_hat)

    J = 1
    for i in range(n):
        J_temp = 0
        for l in range(k):
            #J += W[i][l] * pi[l] * (1/(2*np.pi*(sigma_square_hat[l]))**0.5) * np.exp
            J_temp += pi[l] * (1/(2*np.pi*(sigma_square_hat[l]))**0.5) * np.exp(-(y[i
        J *= J_temp

    log_like.append(np.log(J))
    increase = np.log(J) - log_like_prev
    log_like_prev = np.log(J)

    [pi_hat, w_hat, b_hat, sigma_square_hat, mu_hat] = M_step(n, k, mu, y, pi_hat, w_
```

In [61]:
```
plt.plot(log_like)
plt.show()
```



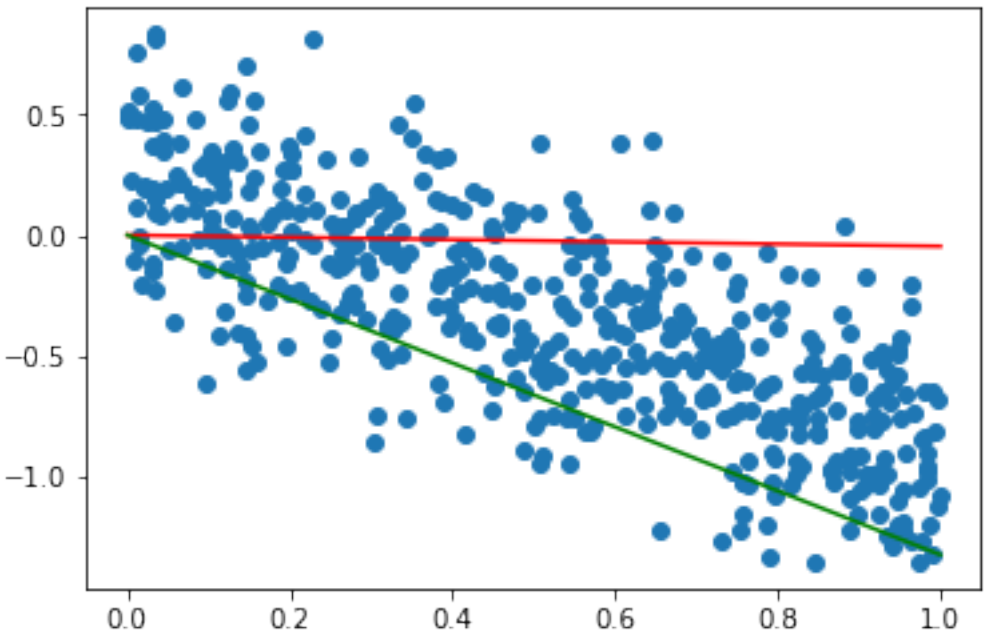
In [62]:
```
[pi, w, b, sigma]
```

Out[62]:
```
[array([0.7, 0.3]), array([-2,  1]), array([ 0.5, -0.5]), array([0.4, 0.3])]
```

In [63]:
```
[pi_hat, w_hat, b_hat, sigma_square_hat**0.5, mu_hat]
```

Out[63]:
```
[array([0.51030693, 0.48969307]),
 array([-0.04623321, -1.32331744]),
 array([0., 0.]),
 array([0.32728379, 0.31641133]),
 array([-0.0231166 , -0.66165872])]
```

In [64]:
```
plt.plot(X, y, 'o')
xaxis = np.linspace(0,1,9)
plt.plot(xaxis, w_hat[0]*xaxis + b_hat[0], color='r')
plt.plot(xaxis, w_hat[1]*xaxis + b_hat[1], color='g')
plt.show()
```



In [ ]:

In [ ]: