## Problem1

## (a)

**the movie titles from least popular to most popular are:**

Fifty_Shades_of_Grey
The_Last_Airbender
Magic_Mike
Prometheus
Bridemaids
World_War_Z
Man_of_Steel
Mad_Max:_Fury_Road
Drive
Thor
Pitch_Perfect
The_Hunger_Games
Fast_Five
The_Hateful_Eight
Iron_Man_2
The_Perks_of_Being_a_Wallflower
American_Hustle
The_Help
Avengers:_Age_of_Ultron
21_Jump_Street
Captain_America:_The_First_Avenger
Les_Miserables
Star_Wars:_The_Force_Awakens
Jurassic_World
The_Great_Gatsby
X-Men:_First_Class
The_Revenant
Her
Ex_Machina
Room
Django_Unchained
The_Girls_with_the_Dragon_Tattoo
Frozen
Midnight_in_Paris
The_Avengers
Wolf_of_Wall_Street
Harry_Potter_and_the_Deathly_Hallows:_Part_1
Black_Swan
Toy_Story_3
Harry_Potter_and_the_Deathly_Hallows:_Part_2
Gone_Girl
The_Theory_of_Everything
12_Years_a_Slave

Now_You_See_Me
The_Social_Network
The_Martian
Shutter_Island
Interstellar
The_Dark_Knight_Rises
Inception

## (e)

**The log-likelihood increases at each iteration.**

| iteration | Log-likelihood L |
|---|---|
| 0 | -23.6819 |
| 1 | -14.3421 |
| 2 | -12.9096 |
| 4 | -12.1506 |
| 8 | -11.8679 |
| 16 | -11.6822 |
| 32 | -11.5655 |
| 64 | -11.5401 |

## ( f )

**The posterior probability for the row from my trained model is :**

[ 0.009590536847491178
 0.021922669980367248
 0.001821733166512483
 0.966665060005629 ]

**Since I only watched 7 movies, the expected ratings on the rest 43 movies which I haven't yet seen are(sorted from lowest rating to highest rating) :**

[('The_Help', 0.035118044756033626),
 ('Bridemaids', 0.16461838128767836),
 ('World_War_Z', 0.2431793460734898),
 ('Room', 0.3189341222580677),
 ('Toy_Story_3', 0.34061055618953023),
 ('The_Last_Airbender', 0.3818583396127309),
 ('Pitch_Perfect', 0.430814287672144),
 ('Thor', 0.4644002847626714),
 ('Man_of_Steel', 0.48428348593616),
 ('Prometheus', 0.4864715992429235),
 ('Jurassic_World', 0.5344977507585111),
 ('21_Jump_Street', 0.5600138279015179),
 ('Frozen', 0.5690276479852795),
 ('Magic_Mike', 0.6072052238420473),
 ('Captain_America:_The_First_Avenger', 0.6088987396710249),

('Django_Unchained', 0.6121737544732642),
('Mad_Max:_Fury_Road', 0.6409821885115692),
('The_Revenant', 0.6459081553763055),
('X-Men:_First_Class', 0.6560818586973689),
('Iron_Man_2', 0.6607929066377504),
('The_Hateful_Eight', 0.6708842085908168),
('Star_Wars:_The_Force_Awakens', 0.6802500064636465),
('Gone_Girl', 0.7145707241869514),
('12_Years_a_Slave', 0.7513080985453203),
('Ex_Machina', 0.7715476604023027),
('Avengers:_Age_of_Ultron', 0.7741039664751153),
('The_Girls_with_the_Dragon_Tattoo', 0.7834856996089611),
('Wolf_of_Wall_Street', 0.793237304776406),
('Midnight_in_Paris', 0.8109402224030511),
('Les_Miserables', 0.8474045965203133),
('Drive', 0.8489144426753004),
('American_Hustle', 0.8811870606265134),
('The_Social_Network', 0.8953833593264506),
('The_Martian', 0.8988992821874364),
('The_Avengers', 0.9412437344021858),
('The_Dark_Knight_Rises', 0.9632653923285268),
('The_Perks_of_Being_a_Wallflower', 0.9886518724432136),
('Now_You_See_Me', 0.9900496035708816),
('The_Theory_of_Everything', 0.9916308512727602),
('Her', 0.9923139249037656),
('Shutter_Island', 0.9952370243014849),
('Interstellar', 0.9953275865797924),
('Inception', 0.999551929006019)]

**I think this list reflects my taste better than the list in part(a)**

**Source code:**

```
import numpy
from math import log
```

```
movie_title=[]
with open('hw8_movieTitles.txt') as inputfile:
    for line in inputfile:
        movie_title.append(line.strip())
```

```
studentPID=[]
with open('hw8_studentPID.txt') as inputfile:
    for line in inputfile:
        studentPID.append(line.strip())
```

```
rating = []
with open('hw8_ratings.txt') as inputfile:
    for line in inputfile:
        rating.append(line.strip().split(' '))
```

```
#8.1 (a)
rating_movie_row = [[row[i] for row in rating] for i in range(50)]
popularity = []
for i in range(len(rating_movie_row)):
    popularity.append(rating_movie_row[i].count('1')/(rating_movie_row[i].count('1')+rating_movie_row[i].count('0')))
movie_popularity = {}
for i in range(50):
    movie = movie_title[i]
    popular = popularity[i]
    movie_popularity[movie] = popular
import operator
sorted_m_p =sorted(movie_popularity.items(),key=operator.itemgetter(1))
print('the movie titles from least popular to most popular are: ')
for data in sorted_m_p:
    print(data[0])
```

```
#8.1(e)
probZ = []
with open('hw8_probZ_init.txt') as inputfile:
    for line in inputfile:
        probZ.append(line.strip())
for i in range(len(probZ)):
    probZ[i] = float(probZ[i])
probRgivenZ = []
with open('hw8_probRgivenZ_init.txt') as inputfile:
    for line in inputfile:
        probRgivenZ.append(line.strip().split('   '))
for i in range(len(probRgivenZ)):
    for j in range(4):
        probRgivenZ[i][j] = float(probRgivenZ[i][j])
```

```
for times in range(65):
    likelihood = 0
    for t in range(len(studentPID)): #user t
        sum_mul_probRgivenZ = 0
        for i in range(4):
            mul_probRgivenZ = 1    #calculate for every user, j belongs to Omega t, the multiplication of probRgivenZ.
            for j in range(len(probRgivenZ)):
                if rating[t][j] =='?':
                    continue
                elif rating[t][j] =='0':
                    mul_probRgivenZ *= 1-probRgivenZ[j][i]
                else:
                    mul_probRgivenZ *= probRgivenZ[j][i]
            mul_probRgivenZ = mul_probRgivenZ*probZ[i]
            sum_mul_probRgivenZ += mul_probRgivenZ
        likelihood += log(sum_mul_probRgivenZ)
    likelihood = likelihood /(len(studentPID))
    print('iteration',times,' likelihood = ',likelihood)
    ############### E-step ######################
    rou = []
    for t in range(len(studentPID)):
        rou_user = []
        sum_mul_probRgivenZ = 0
        for i in range(4):
            mul_probRgivenZ = 1    #calculate for every user, j belongs to Omega t, the multiplication of probRgivenZ.
            for j in range(len(probRgivenZ)):
                if rating[t][j] =='?':
                    continue
                elif rating[t][j] =='0':
                    mul_probRgivenZ *= 1-probRgivenZ[j][i]
                else:
                    mul_probRgivenZ *= probRgivenZ[j][i]
            mul_probRgivenZ = mul_probRgivenZ*probZ[i]
            rou_user.append(mul_probRgivenZ)
            sum_mul_probRgivenZ += mul_probRgivenZ
        for i in range(4):
            rou_user[i] = rou_user[i]/sum_mul_probRgivenZ
        rou.append(rou_user)
    ################ M-step ######################
    update_probZ = []
    for i in range(4):
        prob_z_i = 0
        for t in range(len(studentPID)):
            prob_z_i += rou[t][i]
        prob_z_i = prob_z_i/len(studentPID)
        update_probZ.append(prob_z_i)
    update_probRgivenZ = []
    for j in range(50):
        per_movie = []
        for i in range(4):
            num1 = 0
            num2 = 0
            denominator = 0
            for t in range(len(studentPID)):
                denominator += rou[t][i]
                if rating[t][j] == '1':
                    num1 += rou[t][i]
                elif rating[t][j] =='?':
                    num2 += rou[t][i]*probRgivenZ[j][i]
                else:
                    continue
            per_movie.append((num1+num2)/denominator)
        update_probRgivenZ.append(per_movie)
    probZ = update_probZ
    probRgivenZ = update_probRgivenZ
```

```
#8.1 (f)
my_index = studentPID.index('A53213765')
my_data = rating[my_index]
my_rou = rou[my_index]
print('The posterior probability for the row from my trained model is : ')
print('[',my_rou[0])
print(' ',my_rou[1])
print(' ',my_rou[2])
print(' ',my_rou[3],'  ]')
```

```
not_seen = {}
for movie_index in range(len(my_data)):
    if my_data[movie_index] == '?':
        expected_rating = 0
        for i in range(4):
            expected_rating += my_rou[i]*probRgivenZ[movie_index][i]
        not_seen[movie_title[movie_index]] = expected_rating
```

```
import operator
sorted_not_seen = sorted(not_seen.items(),key=operator.itemgetter(1))
sorted_not_seen
```