In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
# Load Input files
studentIds = open("hw8_ids.txt").read().splitlines()
movieTitles = open("hw8_movies.txt").read().splitlines()
movieRatings = np.genfromtxt("hw8_ratings.txt", dtype="str")
```

In [3]:

```
print(len(movieTitles))
```

```
76
```

In [4]:

```
movieRatings.shape
```

Out[4]:

```
(258, 76)
```

8.1 (a)

In [5]:

```
movieMeanRatings = []
for i in range(len(movieTitles)):
    movieRatingsCol = movieRatings[:,i]
    numRecommended = (movieRatingsCol == "1").sum()
    numSeen = (movieRatingsCol != "?").sum()
    movieMeanRatings.append((numRecommended/numSeen, movieTitles[i]))
```

In [6]:

```
movieMeanRatings.sort(reverse=True)
```

In [7]:

```python
for meanR, mTitle in movieMeanRatings:
    print(mTitle)
```

```
Inception
Interstellar
Three_Billboards_Outside_Ebbing
Django_Unchained
The_Martian
The_Dark_Knight_Rises
The_Theory_of_Everything
Black_Swan
Shutter_Island
Hidden_Figures
Avengers:_Infinity_War
The_Help
12_Years_a_Slave
The_Avengers
Ready_Player_One
Avengers:_Endgame
Les_Miserables
Parasite
The_Girls_with_the_Dragon_Tattoo
Now_You_See_Me
Joker
The_Lion_King
The_Social_Network
Gone_Girl
Harry_Potter_and_the_Deathly_Hallows:_Part_2
Wolf_of_Wall_Street
Room
Harry_Potter_and_the_Deathly_Hallows:_Part_1
Iron_Man_2
21_Jump_Street
Spiderman:_Far_From_Home
Her
Ex_Machina
La_La_Land
Frozen
Drive
X-Men:_First_Class
Midnight_in_Paris
Captain_America:_The_First_Avenger
Toy_Story_3
Darkest_Hour
Dunkirk
The_Great_Gatsby
The_Hateful_Eight
The_Revenant
The_Perks_of_Being_a_Wallflower
Thor
Terminator:_Dark_Fate
Good_Boys
Chappaquidick
The_Farewell
Bridemaids
Us
Mad_Max:_Fury_Road
Rocketman
Avengers:_Age_of_Ultron
Manchester_by_the_Sea
The_Hunger_Games
Phantom_Thread
Pokemon_Detective_Pikachu
Star_Wars:_The_Force_Awakens
Fast_&_Furious:_Hobbs_&_Shaw
Pitch_Perfect
```

```
Once_Upon_a_Time_in_Hollywood
Jurassic_World
American_Hustle
Fast_Five
Prometheus
Hustlers
World_War_Z
The_Shape_of_Water
Man_of_Steel
Magic_Mike
I_Feel_Pretty
Fifty_Shades_of_Grey
The_Last_Airbender
```

8.1 (e)

In [8]:

```
# Constants
K = 4
T = movieRatings.shape[0]
NUM_MOVIES = movieRatings.shape[1]
NUM_ITERATION = 256
```

In [9]:

```
# Load prob initilization
probZ_init = np.loadtxt('hw8_probZ_init.txt')
probR_givenZ_init = np.loadtxt('hw8_probR_init.txt')
```

In [10]:

```
print(probZ_init.shape)
print(probR_givenZ_init.shape)
```

```
(4,)
(76, 4)
```

In [48]:

```
# Helpers
def estep_numerator(i, t, probZ, probR_givenZ):
    j_rec = np.asarray(movieRatings[t,:] == "1").nonzero()
    j_notrec = np.asarray(movieRatings[t,:] == "0").nonzero()
    return probZ[i] * np.prod(probR_givenZ[j_rec,i]) * np.prod(1-probR_givenZ[j_notrec,i])

def estep_denominator(t, probZ, probR_givenZ):
    denom = 0
    j_rec = np.asarray(movieRatings[t,:] == "1").nonzero()
    j_notrec = np.asarray(movieRatings[t,:] == "0").nonzero()
    for i in range(K):
        denom += probZ[i] * np.prod(probR_givenZ[j_rec,i]) * np.prod(1-probR_givenZ[j_notrec,i])
    return denom

def mstep_probR_givenZ(i, j, posteriors, probR_givenZ):
    # Seen port
    t_seen = np.asarray(movieRatings[:,j] == "1").nonzero()
    sum_seen = np.sum(posteriors[i, t_seen])
    # Unseen part
    t_unseen = np.asarray(movieRatings[:,j] == "?").nonzero()
    sum_unseen = np.sum(posteriors[i, t_unseen]) * probR_givenZ[j, i]
    return sum_seen + sum_unseen

def mstep_prz(i, j, posteriors, priors):
    # sum over students who recommended movie j (I(r_j,1))
    t_seen, = np.where(movieRatings[:,j] == '1')
    numer_seen = np.sum(posteriors[i,t_seen])
    # sum over students who have not seen movie j
    t_unseen, = np.where(movieRatings[:,j] == '?')
    numer_unseen = priors[j,i]*np.sum(posteriors[i,t_unseen])
    return numer_seen+numer_unseen

def logLikelihood(probZ, probR_givenZ):
    logL = 0
    for t in range(T):
        likelihood = 0
        for i in range(K):
            j_rec = np.asarray(movieRatings[t,:] == "1").nonzero()
            j_notrec = np.asarray(movieRatings[t,:] == "0").nonzero()
            likelihood += probZ[i] * np.prod(probR_givenZ[j_rec, i]) * np.prod(1-probR_givenZ[j_notrec,i])
        logL += np.log(likelihood)
    return logL/T

def likelihood(t, pz, priors):
    cumsum = 0
    for i in range(K):
        j_rec, = np.where(movieRatings[t,:] == '1')
        j_notrec, = np.where(movieRatings[t,:] == '0')
        cumsum += pz[i]*np.prod(priors[j_rec,i])*np.prod(1-priors[j_notrec,i])
    return cumsum

def EM():
    # Initialization
    probZ = np.copy(probZ_init)
    probR_givenZ = np.copy(probR_givenZ_init)
    posteriors = np.empty([K,T], dtype='float64')
    probZ_temp = np.empty(K)
    probR_givenZ_temp = np.empty([NUM_MOVIES, K])
    L = [] #log-likelihoods for each iteration
    for iteration in range(NUM_ITERATION+1):
        # Show the log-likelihood
        L.append(logLikelihood(probZ, probR_givenZ))
        if iteration in {0,1,2,4,8,16,32,64,128,256}:
            print("iteration: %d, log-likelihood L: %.4f" % (iteration, L[iteration]))

        # estep - update the posteriors
        for t in range(T):
            e_denom = estep_denominator(t, probZ, probR_givenZ)
            for i in range(K):
                posteriors[i,t] = estep_numerator(i, t, probZ, probR_givenZ)/e_denom
        # mstep - update the CPTs
        for i in range(K):
            sum_posteriors = np.sum(posteriors[i,:])
            probZ_temp[i] = sum_posteriors/T
            for j in range(NUM_MOVIES):
                probR_givenZ_temp[j, i] = mstep_probR_givenZ(i, j, posteriors, probR_giv
enZ)/sum_posteriors
                #probR_givenZ_temp[j, i] = mstep_prz(i, j, posteriors, probR_givenZ)/sum
_posteriors
        # Update CPTs
        probZ = probZ_temp
        probR_givenZ = probR_givenZ_temp

    return L, posteriors, probZ, probR_givenZ
```

In [49]:

```
L, posteriors, probZ, probR_givenZ = EM()
```

```
iteration: 0, log-likelihood L: -29.3276
iteration: 1, log-likelihood L: -18.1393
iteration: 2, log-likelihood L: -16.1713
iteration: 4, log-likelihood L: -14.9416
iteration: 8, log-likelihood L: -14.2107
iteration: 16, log-likelihood L: -13.8581
iteration: 32, log-likelihood L: -13.7640
iteration: 64, log-likelihood L: -13.7398
iteration: 128, log-likelihood L: -13.7377
iteration: 256, log-likelihood L: -13.7375
```

8.1 (f)

In [45]:

```
# Constants
PID = "A53317103"
indexPID = studentIds.index(PID)
```

In [46]:

```
indexPID
```

Out[46]:

```
206
```

In [75]:

```
my_data = movieRatings[indexPID,:]
my_unseen = np.asarray(my_data == '?').nonzero()[0]
expected_ratings = []

for l in my_unseen:
    exp_rating = 0
    for i in range(K):
        estep_term = estep_numerator(i, indexPID, probZ, probR_givenZ)/estep_denominator
(indexPID, probZ, probR_givenZ)
        mstep_term = mstep_probR_givenZ(i,l, posteriors, probR_givenZ)/np.sum(posteriors
[i,:])
        exp_rating += estep_term * mstep_term
    expected_ratings.append((exp_rating, movieTitles[l]))

expected_ratings.sort(reverse=True)
pd.DataFrame(expected_ratings, columns=['Expected rating', 'Movie'])
```

Out[75]:

	Movie	Expected rating
0	0.999637	The_Hateful_Eight
1	0.999290	The_Farewell
2	0.978411	Django_Unchained
3	0.965689	12_Years_a_Slave
4	0.937898	Drive
5	0.937181	The_Help
6	0.932838	Her
7	0.919408	Les_Miserables
8	0.912359	The_Theory_of_Everything
9	0.904949	Harry_Potter_and_the_Deathly_Hallows:_Part_1
10	0.904763	Harry_Potter_and_the_Deathly_Hallows:_Part_2
11	0.898639	Chappaquidick
12	0.896686	Joker
13	0.887107	Three_Billboards_Outside_Ebbing
14	0.886941	Thor
15	0.886367	Wolf_of_Wall_Street
16	0.880042	The_Lion_King
17	0.876021	21_Jump_Street
18	0.871629	The_Social_Network
19	0.870474	The_Great_Gatsby
20	0.845030	Parasite
21	0.844600	The_Perks_of_Being_a_Wallflower
22	0.842671	Rocketman
23	0.840467	Darkest_Hour
24	0.812885	Room
25	0.808614	Mad_Max:_Fury_Road
26	0.802399	Star_Wars:_The_Force_Awakens
27	0.801409	Midnight_in_Paris
28	0.800035	Phantom_Thread
29	0.788319	Dunkirk
30	0.761795	Terminator:_Dark_Fate
31	0.756954	American_Hustle
32	0.752641	Ex_Machina
33	0.733815	Once_Upon_a_Time_in_Hollywood
34	0.721491	Us
35	0.721097	Fast_Five
36	0.697665	Bridemaids
37	0.694684	Fast_&_Furious:_Hobbs_&_Shaw
38	0.636770	Hustlers
39	0.626615	Good_Boys

	Movie	Expected rating
40	0.574680	The_Shape_of_Water
41	0.487208	I_Feel_Pretty
42	0.434327	Magic_Mike
43	0.413620	Fifty_Shades_of_Grey
44	0.195148	The_Last_Airbender

```
In [ ]:
```