

# MovieRecommendation

November 27, 2017

```
In [1]: import numpy as np
```

```
movieTitles = open("hw8_movieTitles.txt").read().split() # 50x1 mat of movie titles
movieCount = len(movieTitles)
studentPids = open("hw8_studentPIDs.txt").read().split() # 279x1 mat of student PIDs
studentCount = len(studentPids)
ratings = [] #279x50 mat of userxmovie ratings

for l in open("hw8_ratings.txt"):
    temp = l.rstrip('\n').split()
    ratings.append(temp)

ratings = np.array(ratings)
probZinit = np.loadtxt("hw8_probZ_init.txt") #1x4
groupCount = len(probZinit) # the number of groups we are considering
probRgivenZ = np.loadtxt("hw8_probRgivenZ_init.txt") #50x4
```

```
In [2]: from collections import Counter
moviePopularityRating = [Counter(ratings[:,t])['1']/(Counter(ratings[:,t])['1']+Counter(ratings[:,t])['2']) for t in range(movieCount)]
for index in np.argsort(moviePopularityRating)[::-1]:
    print(movieTitles[index])
```

```
Inception
The_Dark_Knight_Rises
Interstellar
Shutter_Island
The_Martian
The_Social_Network
Now_You_See_Me
12_Years_a_Slave
The_Theory_of_Everything
Gone_Girl
Harry_Potter_and_the_Deathly_Hallows:_Part_2
Toy_Story_3
Black_Swan
Harry_Potter_and_the_Deathly_Hallows:_Part_1
Wolf_of_Wall_Street
The_Avengers
```

Midnight\_in\_Paris  
 Frozen  
 The\_Girls\_with\_the\_Dragon\_Tattoo  
 Django\_Unchained  
 Room  
 Ex\_Machina  
 Her  
 The\_Revenant  
 X-Men:\_First\_Class  
 The\_Great\_Gatsby  
 Jurassic\_World  
 Star\_Wars:\_The\_Force\_Awakens  
 Les\_Miserables  
 Captain\_America:\_The\_First\_Avenger  
 21\_Jump\_Street  
 Avengers:\_Age\_of\_Ultron  
 The\_Help  
 American\_Hustle  
 The\_Perks\_of\_Being\_a\_Wallflower  
 Iron\_Man\_2  
 The\_Hateful\_Eight  
 Fast\_Five  
 The\_Hunger\_Games  
 Pitch\_Perfect  
 Thor  
 Drive  
 Mad\_Max:\_Fury\_Road  
 Man\_of\_Steel  
 World\_War\_Z  
 Bridemaids  
 Prometheus  
 Magic\_Mike  
 The\_Last\_Airbender  
 Fifty\_Shades\_of\_Grey

```
In [3]: probZgivenR = np.zeros(shape=(groupCount, studentCount))
```

```

def Estep():
    global probRgivenZ, probZgivenR, probZinit
    for s in range(studentCount):
        sRatings = ratings[s]
        ratedIndices = [i for i, x in enumerate(sRatings) if x == '1' or x == '0']
        probZgivenRt = np.ones(shape=(groupCount))
        denom = 0.0;
        for g in range(groupCount):
            probZgivenRt[g] = probZinit[g];
            for index in ratedIndices:

```

```

        if(sRatings[index]=='1'):
            probZgivenRt[g] *= probRgivenZ[index][g]
        else:
            probZgivenRt[g] *= (1-probRgivenZ[index][g])
        denom += probZgivenRt[g]
    for g in range(groupCount):
        probZgivenR[g][s] = probZgivenRt[g]/denom

In [4]: def Mstep():
    global probRgivenZ, probZgivenR, probZinit
    #updating Zinit
    for g in range(groupCount):
        updatedVal = 0.0
        for s in range(studentCount):
            updatedVal += probZgivenR[g][s]
        updatedVal = updatedVal/studentCount
        probZinit[g] = updatedVal
    #updating probRgivenZ
    for g in range(groupCount):
        for m in range(movieCount):
            updatedVal = 0.0
            denom = 0.0
            for s in range(studentCount):
                if(ratings[s][m]=='1'):
                    updatedVal += probZgivenR[g][s]
                elif(ratings[s][m]=='?'):
                    updatedVal += (probZgivenR[g][s]*probRgivenZ[m][g])
                denom += probZgivenR[g][s]
            probRgivenZ[m][g] = updatedVal/denom;

In [5]: import math
    def logLikelihood():
        global probRgivenZ, probZgivenR, probZinit
        denom = studentCount
        ll = 0;
        for s in range(studentCount):
            llisum = 0
            for g in range(groupCount):
                update = probZinit[g]
                ratedIndices = [i for i,x in enumerate(ratings[s]) if x=='1' or x=='0']
                for index in ratedIndices:
                    if(ratings[s][index] == '1'):
                        update *= probRgivenZ[index][g]
                    else:
                        update *= (1-probRgivenZ[index][g])
                llisum += update
            ll += math.log(llisum)
        ll = ll/denom
        print(ll)

```

```
In [6]: print(logLikelihood()) #At 0th step
```

```
-23.68194303920656
```

```
None
```

```
In [7]: for i in range(64):  
        Estep()  
        Mstep()  
        print("Iteration: "+str(i+1))  
        print(logLikelihood())
```

```
Iteration: 1
```

```
-14.342138732059837
```

```
None
```

```
Iteration: 2
```

```
-12.909592229693583
```

```
None
```

```
Iteration: 3
```

```
-12.390960695569433
```

```
None
```

```
Iteration: 4
```

```
-12.15061981794007
```

```
None
```

```
Iteration: 5
```

```
-12.031760625835124
```

```
None
```

```
Iteration: 6
```

```
-11.960644019670974
```

```
None
```

```
Iteration: 7
```

```
-11.90884865715703
```

```
None
```

```
Iteration: 8
```

```
-11.867860888836633
```

```
None
```

```
Iteration: 9
```

```
-11.833882674009132
```

```
None
```

```
Iteration: 10
```

```
-11.804593081398394
```

```
None
```

```
Iteration: 11
```

```
-11.7787057143901
```

```
None
```

```
Iteration: 12
```

```
-11.755106316683584
```

```
None
```

Iteration: 13  
-11.733413153894418  
None  
Iteration: 14  
-11.713950522388867  
None  
Iteration: 15  
-11.696950121777396  
None  
Iteration: 16  
-11.682204313006867  
None  
Iteration: 17  
-11.668285628739406  
None  
Iteration: 18  
-11.654498444787581  
None  
Iteration: 19  
-11.641607935197339  
None  
Iteration: 20  
-11.630481743512357  
None  
Iteration: 21  
-11.620730278967113  
None  
Iteration: 22  
-11.60984936605054  
None  
Iteration: 23  
-11.59658086131925  
None  
Iteration: 24  
-11.586600888666458  
None  
Iteration: 25  
-11.58018210070515  
None  
Iteration: 26  
-11.576016178631011  
None  
Iteration: 27  
-11.573148132356842  
None  
Iteration: 28  
-11.571006520139381  
None

Iteration: 29  
-11.569276690641953  
None  
Iteration: 30  
-11.567808285296941  
None  
Iteration: 31  
-11.56654639843747  
None  
Iteration: 32  
-11.565450274671582  
None  
Iteration: 33  
-11.564437191089352  
None  
Iteration: 34  
-11.563374638237683  
None  
Iteration: 35  
-11.56208288653384  
None  
Iteration: 36  
-11.560352430154955  
None  
Iteration: 37  
-11.558050271014766  
None  
Iteration: 38  
-11.555327781667074  
None  
Iteration: 39  
-11.552676860808665  
None  
Iteration: 40  
-11.550510152897992  
None  
Iteration: 41  
-11.548895801784521  
None  
Iteration: 42  
-11.547730013045145  
None  
Iteration: 43  
-11.546876076291396  
None  
Iteration: 44  
-11.546220589690792  
None

Iteration: 45  
-11.545685665067175  
None  
Iteration: 46  
-11.545220737606659  
None  
Iteration: 47  
-11.54479221700967  
None  
Iteration: 48  
-11.544377277250769  
None  
Iteration: 49  
-11.543962252746242  
None  
Iteration: 50  
-11.543543808467176  
None  
Iteration: 51  
-11.543129117511485  
None  
Iteration: 52  
-11.5427316200594  
None  
Iteration: 53  
-11.54236385022987  
None  
Iteration: 54  
-11.542032674943902  
None  
Iteration: 55  
-11.541739096002724  
None  
Iteration: 56  
-11.541480267416459  
None  
Iteration: 57  
-11.541251420402844  
None  
Iteration: 58  
-11.541047196162305  
None  
Iteration: 59  
-11.540862567650946  
None  
Iteration: 60  
-11.540693437794108  
None

```
Iteration: 61
-11.540536900968107
None
Iteration: 62
-11.540391188558587
None
Iteration: 63
-11.540255396796082
None
Iteration: 64
-11.540129139078093
None
```

```
In [9]: myIndex = studentPids.index('A53244405') #204 index
        unseenIndexes = [i for i, x in enumerate(ratings[myIndex]) if x=='?']
        probMyUnseen = np.ones(movieCount)
        for u in unseenIndexes:
            print(movieTitles[u])
            unseenProb = 0;
            for g in range(groupCount):
                probTemp = probRgivenZ[u][g]
                probTemp *= probZgivenR[g][myIndex]
                unseenProb += probTemp
            probMyUnseen[u] = unseenProb
            print(unseenProb)
```

```
Black_Swan
0.777160769578
Shutter_Island
0.916768459654
The_Last_Airbender
0.41936899168
Bridemaids
0.40730924267
The_Girls_with_the_Dragon_Tattoo
0.614695520663
Drive
0.415460260347
Midnight_in_Paris
0.813838407899
Prometheus
0.481687423616
Django_Unchained
0.684712426299
Les_Miserables
0.565015875928
21_Jump_Street
```



0.651450715988  
Magic\_Mike  
0.46759972124  
Her  
0.561161578191  
12\_Years\_a\_Slave  
0.751319306016  
World\_War\_Z  
0.54134111622  
American\_Hustle  
0.707790157046  
Ex\_Machina  
0.689212495903  
Room  
0.560915743242  
The\_Martian  
0.832948696225  
The\_Hateful\_Eight  
0.404439382717  
The\_Revenant  
0.695193108419