250A_HW8_A59017531

November 29, 2022

[1]:
```
import numpy as np
from collections import defaultdict
from tqdm.notebook import tqdm
from prettytable import PrettyTable
```

0.1 Q1

0.1.1 Importing data

[2]:
```
users = np.loadtxt("hw8_ids.txt",dtype=str)
items = np.loadtxt("hw8_movies.txt",dtype=str)
tmpRatings = np.loadtxt("hw8_ratings.txt",dtype=str)
```

0.1.2 Preprocessing

[3]:
```
ratings = np.zeros((tmpRatings.shape[0],tmpRatings.shape[1]))
for i in range(len(tmpRatings)):
  for j in range(len(tmpRatings[0])):
    try:
      ratings[i][j] = int(tmpRatings[i,j])
    except:
      ratings[i,j] = -1
```

[4]:
```
ratings
```

[4]:
```
array([[ 1.,  1., -1., …,  1., -1., -1.],
       [ 1., -1.,  0., …,  0., -1., -1.],
       [ 1.,  1., -1., …,  0., -1., -1.],
       …,
       [ 1., -1., -1., …, -1., -1., -1.],
       [-1.,  0., -1., …, -1., -1., -1.],
       [ 1., -1., -1., …, -1.,  1.,  1.]])
```

[5]:
```
np.unique(ratings)
```

[5]:
```
array([-1.,  0.,  1.])
```

[6]:
```
UIR = []
```

[7]:
```
for i in range(len(ratings)):
  user = users[i]
  for j in range(len(ratings[0])):
    item = items[j]
    UIR.append([user,item,ratings[i,j]])
```

[8]:
```
UIR[:10]
```

[8]:
```
[['\ufeffA12614795', 'Inception', 1.0],
 ['\ufeffA12614795', 'The_Social_Network', 1.0],
 ['\ufeffA12614795', 'Black_Swan', -1.0],
 ['\ufeffA12614795', 'Shutter_Island', 1.0],
 ['\ufeffA12614795', 'The_Last_Airbender', 0.0],
 ['\ufeffA12614795', 'Harry_Potter_and_the_Deathly_Hallows:_Part_1', -1.0],
 ['\ufeffA12614795', 'Iron_Man_2', 1.0],
 ['\ufeffA12614795', 'Toy_Story_3', 1.0],
 ['\ufeffA12614795', 'Fast_Five', 1.0],
 ['\ufeffA12614795', 'Thor', 1.0]]
```

[9]:
```
ratingsPerUser = defaultdict(list)
ratingsPerItem = defaultdict(list)
for u,b,r in UIR:
    ratingsPerUser[u].append((b,r))
    ratingsPerItem[b].append((u,r))
```

[10]:
```
usersPerItem = defaultdict(set) # Maps an item to the users who rated it
itemsPerUser = defaultdict(set) # Maps a user to the items that they rated
```

[11]:
```
ratingDict = {} # To retrieve a rating for a specific user/item pair
```

[12]:
```
for i in UIR:
  itemsPerUser[i[0]].add(i[1])

  usersPerItem[i[1]].add(i[0])

  ratingDict[(i[0],i[1])] = i[2]
```

[13]:
```
ratingsPerUser["A59017531"][:10]
```

[13]:
```
[('Inception', 1.0),
 ('The_Social_Network', -1.0),
 ('Black_Swan', -1.0),
 ('Shutter_Island', -1.0),
 ('The_Last_Airbender', 1.0),
 ('Harry_Potter_and_the_Deathly_Hallows:_Part_1', 1.0),
 ('Iron_Man_2', 1.0),
 ('Toy_Story_3', 1.0),
```

```
('Fast_Five', 1.0),
('Thor', 1.0)]
```

0.1.3 Q1A

[ ]:
```
popDict = {}
for item in ratingsPerItem:
  recN = 0
  sawN = 0
  for user in ratingsPerUser:
    rat = ratingDict[(user,item)]
    if rat == -1:
      continue
    if rat == 1:
      recN+=1
    sawN+=1
  if sawN!=0:
    popDict[item] = recN/sawN
  else:
    popDict[item] = -1
```

[ ]:
```
popDict = dict(sorted(popDict.items(),key=lambda item:item[1]))
```

[ ]:
```
popDict.keys()
```

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

```
[ ]: popDict.items()
```

```
[ ]: dict_items([('Chappaquidick', 0.34285714285714286), ('The_Last_Airbender',
     0.460431654676259), ('I_Feel_Pretty', 0.4878048780487805),
     ('Fifty_Shades_of_Grey', 0.4975124378109453), ('Fast_&_Furious:_Hobbs_&_Shaw',
     0.5181347150259067), ('Hustlers', 0.5185185185185185), ('Magic_Mike',
     0.5333333333333333), ('Bridemaids', 0.5384615384615384), ('World_War_Z',
     0.5774647887323944), ('The_Shape_of_Water', 0.5795454545454546), ('Good_Boys',
     0.6), ('Prometheus', 0.6017699115044248), ('Pokemon_Detective_Pikachu',
     0.6020942408376964), ('American_Hustle', 0.6052631578947368),
     ('Terminator:_Dark_Fate', 0.6052631578947368), ('The_Farewell',
     0.6111111111111112), ('Man_of_Steel', 0.625), ('Fast_Five', 0.6256684491978609),
     ('The_Hateful_Eight', 0.6282051282051282), ('Star_Wars:_The_Force_Awakens',
     0.6495726495726496), ('The_Help', 0.6515151515151515), ('Rocketman',
     0.6515151515151515), ('Drive', 0.6623376623376623),
     ('The_Girls_with_the_Dragon_Tattoo', 0.6777777777777778), ('Thor',
     0.6804123711340206), ('Avengers:_Age_of_Ultron', 0.6928104575163399),
     ('Phantom_Thread', 0.6976744186046512), ('Us', 0.6976744186046512),
     ('The_Revenant', 0.7007874015748031), ('X-Men:_First_Class', 0.701195219123506),
     ('Pitch_Perfect', 0.7086614173228346), ('Dunkirk', 0.7135416666666666),
     ('Ready_Player_One', 0.7142857142857143), ('Room', 0.7162162162162162),
     ('Jurassic_World', 0.7171717171717171), ('Mad_Max:_Fury_Road',
     0.7243589743589743), ('Once_Upon_a_Time_in_Hollywood', 0.725925925925926),
     ('Manchester_by_the_Sea', 0.7272727272727273),
     ('The_Perks_of_Being_a_Wallflower', 0.7346938775510204),
     ('Spiderman:_Far_From_Home', 0.7450331125827815), ('Her', 0.75),
     ('Captain_America:_The_First_Avenger', 0.7523510971786834), ('Frozen',
     0.7552447552447552), ('Hidden_Figures', 0.7590361445783133),
     ('The_Hunger_Games', 0.7676767676767676), ('Iron_Man_2', 0.770392749244713),
     ('Les_Miserables', 0.7737226277372263), ('Toy_Story_3', 0.7777777777777778),
     ('Three_Billboards_Outside_Ebbing', 0.7804878048780488), ('Darkest_Hour',
     0.78125), ('Ex_Machina', 0.7863247863247863), ('Gone_Girl', 0.7938931297709924),
     ('Black_Swan', 0.7981651376146789), ('12_Years_a_Slave', 0.811965811965812),
     ('Avengers:_Endgame', 0.8136645962732919), ('The_Avengers', 0.8147058823529412),
     ('Midnight_in_Paris', 0.8170731707317073), ('The_Great_Gatsby',
     0.8178137651821862), ('La_La_Land', 0.8235294117647058),
     ('Avengers:_Infinity_War', 0.826625386996904), ('The_Theory_of_Everything',
     0.8294117647058824), ('Now_You_See_Me', 0.836), ('21_Jump_Street',
     0.837037037037037), ('Django_Unchained', 0.84472049689441), ('The_Martian',
     0.861244019138756), ('Harry_Potter_and_the_Deathly_Hallows:_Part_1',
     0.8628048780487805), ('Joker', 0.8647686832740213), ('Wolf_of_Wall_Street',
     0.8669064748201439), ('The_Lion_King', 0.8717948717948718),
     ('Harry_Potter_and_the_Deathly_Hallows:_Part_2', 0.871875), ('Parasite',
     0.8878923766816144), ('The_Social_Network', 0.8960396039603961),
     ('The_Dark_Knight_Rises', 0.8982456140350877), ('Shutter_Island',
     0.9333333333333333), ('Interstellar', 0.946843853820598), ('Inception',
     0.9605263157894737)])
```

0.1.4 The rankings are somewhat in line with what I would vote, but not all are matching

0.1.5 Q1)E)

[ ]:
```
pz = np.loadtxt("hw8_probZ_init.txt")
prz = np.loadtxt("hw8_probR_init.txt")
```

[ ]:
```
k = 4
N = 256
postProb = np.zeros((k,len(users)))
```

[ ]:
```
def loglikeli(user):
  sumi = 0
  rec, = np.where(ratings[user,:] == 1)
  notRec, = np.where(ratings[user,:] == 0)
  for i in range(k):
    sumi+=(pz[i])*(np.prod(prz[rec,i]))*(np.prod(1-prz[notRec,i]))
  return np.log(sumi)
```

[ ]:
```
def e_step(i,user):
  rec, = np.where(ratings[user,:] == 1)
  notRec, = np.where(ratings[user,:] == 0)
  numi = pz[i]*np.prod(prz[rec,i])*(np.prod(1-prz[notRec,i]))
  deno = 0
  for ind in range(k):
    deno+=pz[ind]*np.prod(prz[rec,ind])*(np.prod(1-prz[notRec,ind]))
  return numi/deno
```

[ ]:
```
def m_step(i,item):
  seen, = np.where(ratings[:,item] == 1)
  nSeen, = np.where(ratings[:,item] == -1)
  sumi = (np.sum(postProb[i,seen])+prz[item,i]*np.sum(postProb[i,nSeen]))/np.
 ↪sum(postProb[i,:])
  return sumi
```

[ ]:
```
logliklii = []
for i in tqdm(range(N+1)):
  logli = 0
  pzTmp = np.zeros(k)
  przTmp = np.zeros((len(items),k))
  for user in range(len(users)):
    logli += loglikeli(user)
    for i in range(k):
      postProb[i,user] = e_step(i,user)

  for i in range(k):
    pzTmp[i] = np.sum(postProb[i,:])/len(users)
```

```
    for item in range(len(items)):
      przTmp[item,i] = m_step(i,item)

  logliklii.append(logli/len(users))
  pz  =pzTmp
  prz = przTmp
```

```
  0%|          | 0/257 [00:00<?, ?it/s]
```

[ ]:
```
indi = [0]
ll = [logliklii[0]]
for i in range(9):
  indi.append(2**i)
  ll.append(logliklii[int(2**i)])
```

[ ]:

[ ]:
```
print("Table:")
x = PrettyTable()
x.add_column("Iteration", indi)
x.add_column("LogLikelihood", ll)
print(x)
```

```
Table:
+-----------+---------------------+
| Iteration |    LogLikelihood    |
+-----------+---------------------+
|     0     | -28.627324487337628 |
|     1     | -19.350314946503318 |
|     2     |  -17.90956481801792 |
|     4     | -17.081155562337017 |
|     8     | -16.629824767528113 |
|     16    | -16.287828721915584 |
|     32    |  -15.80153795397027 |
|     64    | -15.749887678844283 |
|    128    | -15.735940712575664 |
|    256    | -15.728520329683299 |
+-----------+---------------------+
```

0.1.6 Yes the loglikelihood becomes less and less negative, as we expected.

0.1.7 Q1F

[ ]:
```
ratingsPerUser["A59017531"][:10]
```

[ ]:
```
[('Inception', 1.0),
 ('The_Social_Network', -1.0),
 ('Black_Swan', -1.0),
 ('Shutter_Island', -1.0),
```

```
 ('The_Last_Airbender', 1.0),
 ('Harry_Potter_and_the_Deathly_Hallows:_Part_1', 1.0),
 ('Iron_Man_2', 1.0),
 ('Toy_Story_3', 1.0),
 ('Fast_Five', 1.0),
 ('Thor', 1.0)]
```

[ ]:

```
[ ]: (array([9]),)
```

[ ]:
```
user = "A59017531"
userId = np.where(users == user)[0][0]
recommendations = {}
for me in ratingsPerUser[user]:
  if me[1] == -1: #unseen
    sumi = 0
    movieIdx = np.where(items==me[0])[0][0]
    for i in range(k):
      estep = e_step(i,userId)
      mstep = m_step(i,movieIdx)
      sumi += estep*mstep
    recommendations[me[0]] = sumi
```

[ ]:
```
dict(sorted(recommendations.items(),key=lambda item:item[1],reverse=True))
```

```
[ ]: {'The_Theory_of_Everything': 0.9810400408235112,
 'Shutter_Island': 0.9745852668879414,
 'Django_Unchained': 0.9731063931312587,
 '21_Jump_Street': 0.9729138079741417,
 'Midnight_in_Paris': 0.9700705368973399,
 'Parasite': 0.9697855734535975,
 'Hidden_Figures': 0.9648232899695831,
 'The_Hunger_Games': 0.9631952084894266,
 'The_Perks_of_Being_a_Wallflower': 0.9618823785178835,
 'Phantom_Thread': 0.9568020557602742,
 '12_Years_a_Slave': 0.9549226481950793,
 'Gone_Girl': 0.9533566794438127,
 'The_Help': 0.9532991684985187,
 'Her': 0.9487601164156515,
 'The_Social_Network': 0.947674353086089,
 'Prometheus': 0.9431487813732097,
 'Black_Swan': 0.9403008886436867,
 'Drive': 0.9389080698290413,
 'The_Farewell': 0.9364692582778592,
 'Good_Boys': 0.9316805551144837,
 'The_Martian': 0.9243198446314408,
```

```
 'Hustlers': 0.9230392263890211,
 'I_Feel_Pretty': 0.9198683449177839,
 'Darkest_Hour': 0.9188775043021438,
 'Three_Billboards_Outside_Ebbing': 0.9166413870517898,
 'La_La_Land': 0.9075400177503928,
 'Les_Miserables': 0.898955453530586,
 'The_Girls_with_the_Dragon_Tattoo': 0.894674775604871,
 'The_Great_Gatsby': 0.887310662485786,
 'Dunkirk': 0.8799365619470496,
 'The_Revenant': 0.8793113974336506,
 'Ex_Machina': 0.8778101088513737,
 'Manchester_by_the_Sea': 0.8740733188278312,
 'Pitch_Perfect': 0.8623469324170651,
 'World_War_Z': 0.8553481123564523,
 'Mad_Max:_Fury_Road': 0.848810825704276,
 'Room': 0.8367003390062016,
 'Bridemaids': 0.8340954026340719,
 'Once_Upon_a_Time_in_Hollywood': 0.8315862756962407,
 'The_Hateful_Eight': 0.8268013366459704,
 'American_Hustle': 0.8140761941342205,
 'Rocketman': 0.8126156038813724,
 'Us': 0.7974269241726685,
 'The_Shape_of_Water': 0.7931415651194657,
 'Star_Wars:_The_Force_Awakens': 0.7873841266943339,
 'Chappaquidick': 0.763837402223503,
 'Terminator:_Dark_Fate': 0.7487409431505707,
 'Magic_Mike': 0.7146949155178426,
 'Fifty_Shades_of_Grey': 0.6712659458102751}
```

0.1.8 Yup, these recommendations are wayy more accurate than the mean ones.

[ ]: