Introduction to Web Science/595: Assignment #8

Dr. Nelson

Francis Pruter

Thursday, 13 November, 2014

Contents

Problem 1

```
The goal of this project is to use the basic recommendation principles
we have learned for user-collected data. You will modify the code
given to you which performs movie recommendations from the MovieLense
data sets.

The MovieLense data sets were collected by the GroupLens Research
Project at the University of Minnesota during the seven-month period
from September 19th, 1997 through April 22nd, 1998. It is available
for download from http://www.grouplens.org/node/73

There are three files which we will use:

1.  u.data: 100,000 ratings by 943 users on 1,682 movies. Each
user has rated at least 20 movies. Users and items are numbered
consecutively from 1. The data is randomly ordered. This is a tab
separated list of

user id | item id | rating | timestamp

The time stamps are unix seconds since 1/1/1970 UTC.

Example:

196 242 3 881250949
186 302 3  891717742
22 377 1  878887116
244 51 2  880606923
166 346 1  886397596
298 474 4  884182806
115 265 2 881171488

2.  u.item: Information about the 1,682 movies. This is a tab
separated list of

movie id | movie title | release date | video release date | IMDb URL | unknown | Action |

The last 19 fields are the genres, a 1 indicates the movie is of
that genre, a 0 indicates it is not; movies can be in several genres
at once. The movie ids are the ones used in the u.data data set.

Example:

161|Top Gun (1986)|01-Jan-1986||http://us.imdb.com/M/title-exact?Top%20Gun%20(1986)|0|1|0|0
162|On Golden Pond (1981)|01-Jan-1981||http://us.imdb.com/M/title-exact?On%20Golden%20Pond%
163|Return of the Pink Panther, The (1974)|01-Jan-1974||http://us.imdb.com/M/title-exact?Re

3.  u.user: Demographic information about the users. This is a tab
```

```
separated list of:

user id | age | gender | occupation | zip code

The user ids are the ones used in the u.data data set.

Example:

1|24|M|technician|85711
2|53|F|other|94043
3|23|M|writer|32067
4|24|M|technician|43537
5|33|F|other|15213

The code for reading from the u.data and u.item files and creating
recommendations is described in the book Programming Collective
Intelligence (check email for more details). You are to modify
recommendations.py to answer the following questions. Each question
your program answers correctly will award you 10 points. You must
have the question answered completely correct; partial credit will
only be awarded if your answer is very close to the correct one.

1.  What 5 movies have the highest average ratings? Show the movies
and their ratings sorted by their average ratings.
```

DISCUSSION After downloading recommendation.py from [1], I modified the loadMovieLens function to include the loading of user data (age and gender).

Listing 1: loadMovieLens

```python
def loadMovieLens():
  # Get movie titles

    for line in open('u.item'):
        (id, title) = line.split('|')[0:2]
        movies[id] = title
  # Load data
    prefs = {}
    for line in open('u.data'):
        (user, movieid, rating, ts) = line.split('\t')
        prefs.setdefault(user, {})
        prefs[user][movies[movieid]] = float(rating)
  # Load user data
    userData = {}
    for line in open('u.user'):
        (user, age, gender) = line.split('|')[0:3]
        userData.setdefault(user, {})
        userData[user]["Ratings"] = prefs[user]
        userData[user]["Age"] = age
        userData[user]["Gender"] =  gender
    return userData
```

Additionally, I loaded a all the lists required to answer all the questions (the ones I was able to answer).

Listing 2: main

```
if __name__ == '__main__':
   # get a list of all ratings
    prefs = loadMovieLens()
    movieRatings = {}
    movieRatingsWomen = {}
    movieRatingsWomenGT40 = {}
    movieRatingsWomenLT40 = {}
    movieRatingsMen = {}
    movieRatingsMenGT40 = {}
    movieRatingsMenLT40 = {}


    for usr, usrData in prefs.iteritems():
        for movie, movieRating in usrData["Ratings"].iteritems():
            movieRatings.setdefault(movie, []).append(int(movieRating))
            if (usrData["Gender"] == 'F'):
                movieRatingsWomen.setdefault(movie, []).append(int(movieRating))
                if (int(usrData["Age"]) < 40):
                    movieRatingsWomenLT40.setdefault(movie, []).append(int(movieRating
                        ))
                elif (int(usrData["Age"]) > 40):
                    movieRatingsWomenGT40.setdefault(movie, []).append(int(movieRating
                        ))

            else:
                movieRatingsMen.setdefault(movie, []).append(int(movieRating))
                if (int(usrData["Age"]) < 40):
                    movieRatingsMenLT40.setdefault(movie, []).append(int(movieRating))
                elif (int(usrData["Age"]) > 40):

                    movieRatingsMenGT40.setdefault(movie, []).append(int(movieRating))
```

SOLUTION

When computing the highest 5 average movie ratings, there was actually a 10 way tie. The following is the code used to calulate the highest average:

Listing 3: loadMovieLens

```
    avgRatings = {}

    for movie, ratings in movieRatings.iteritems():
        avgRatings[movie] = np.mean(ratings)

    avgRatingsDescending = OrderedDict(sorted(avgRatings.items(), key = lambda k: k
        [1], reverse = True))

    ctr = 0
    prev = 0
    with open("q1", "w+") as f:
      for x in avgRatingsDescending.items():
        if (ctr < 5 or prev == x[1]):
            f.write(str(x[1]) + ' & ' + str(x[0]) + '\\\\ \n' )
            prev = x[1]
            ctr = ctr + 1
        else:
            break
```

Below is the solution:

Avg Rating	Movie Title
5.0	They Made Me a Criminal (1939)
5.0	Santa with Muscles (1996)
5.0	Someone Else's America (1995)
5.0	Saint of Fort Washington, The (1993)
5.0	Entertaining Angels: The Dorothy Day Story (1996)
5.0	Marlene Dietrich: Shadow and Light (1996)
5.0	Star Kid (1997)
5.0	Aiqing wansui (1994)
5.0	Prefontaine (1997)
5.0	Great Day in Harlem, A (1994)

Problem 2

```
2.  What 5 movies received the most ratings? Show the movies and
the number of ratings sorted by number of ratings.
```

SOLUTION
The following movies had the most ratings:

Num Ratings	Movie Title
583	Star Wars (1977)
509	Contact (1997)
508	Fargo (1996)
507	Return of the Jedi (1983)
485	Liar Liar (1997)

I used the following code segment to calcuate the solution:

Listing 4: loadMovieLens

```
numRatings = {}

for movie, ratings in movieRatings.iteritems():
    numRatings[movie] = len(ratings)


numRatingsDescending = OrderedDict(sorted(numRatings.items(), key = lambda k: k
    [1], reverse = True))

ctr = 0
prev = 0
with open("q2", "w+") as f:
    for x in numRatingsDescending.items():
        if (ctr < 5 or prev == x[1]):
            f.write(str(x[1]) + " & " + str(x[0]) + "\\\\ \n")
            ctr = ctr+1
            prev = x[1]
        else:
            break
```

Problem 3

```
3.  What 5 movies were rated the highest on average by women? Show
the movies and their ratings sorted by ratings.
```

SOLUTION
The following is a list of the highest rated movies by women; again, although the problem only calls for the top 5, there was an 11 way tie.

Avg Ratings	Movie Title
5.0	Stripes (1981)
5.0	Someone Else's America (1995)
5.0	Everest (1998)
5.0	Maya Lin: A Strong Clear Vision (1994)
5.0	Mina Tannenbaum (1994)
5.0	Year of the Horse (1997)
5.0	Faster Pussycat! Kill! Kill! (1965)
5.0	Foreign Correspondent (1940)
5.0	Telling Lies in America (1997)
5.0	Prefontaine (1997)
5.0	Visitors, The (Visiteurs, Les) (1993)

I used the following code segment to calcuate the solution:

Listing 5: loadMovieLens

```
avgRatings = {}

for movie, ratings in movieRatingsWomen.iteritems():
    avgRatings[movie] = np.mean(ratings)

avgRatingsDescending = OrderedDict(sorted(avgRatings.items(), key = lambda k: k
    [1], reverse = True))

ctr = 0
prev = 0
with open("q3", "w+") as f:
  for x in avgRatingsDescending.items():
    if (ctr < 5 or prev == x[1]):
        f.write(str(x[1]) + ' & ' + str(x[0]) + '\\\\ \n' )
        prev = x[1]
        ctr = ctr + 1
    else:
        break
```

Problem 4

```
4.  What 5 movies were rated the highest on average by men? Show
the movies and their ratings sorted by ratings.
```

SOLUTION

The following is a list of the highest rated movies by men; again, although the problem only calls for the top 5, there was an 15 way tie.

Avg Ratings	Movie Title
5.0	They Made Me a Criminal (1939)
5.0	Santa with Muscles (1996)
5.0	Letter From Death Row, A (1998)
5.0	Saint of Fort Washington, The (1993)
5.0	Quiet Room, The (1996)
5.0	Entertaining Angels: The Dorothy Day Story (1996)
5.0	Marlene Dietrich: Shadow and Light (1996)
5.0	Star Kid (1997)
5.0	Little City (1998)
5.0	Aiqing wansui (1994)
5.0	Prefontaine (1997)
5.0	Love Serenade (1996)
5.0	Leading Man, The (1996)
5.0	Great Day in Harlem, A (1994)
5.0	Delta of Venus (1994)

I used the following code segment to calcuate the solution:

Listing 6: loadMovieLens

```
avgRatings = {}

for movie, ratings in movieRatingsMen.iteritems():
    avgRatings[movie] = np.mean(ratings)

avgRatingsDescending = OrderedDict(sorted(avgRatings.items(), key = lambda k: k
    [1], reverse = True))

ctr = 0
prev = 0
with open("q4", "w+") as f:
  for x in avgRatingsDescending.items():
   if (ctr < 5 or prev == x[1]):
       f.write(str(x[1]) + ' & ' + str(x[0]) + '\\\\ \n' )
       prev = x[1]
       ctr = ctr + 1
   else:
       break
```

Problem 5

```
5.  What movie received ratings most like Top Gun? Which movie
received ratings that were least like Top Gun (negative correlation)?
```

Problem 6

```
6.  Which 5 raters rated the most films? Show the raters' IDs and
the number of films each rated.
```

SOLUTION
Below is a list of the top 5 raters:

Num Ratings	Rater's ID
736	405
678	655
632	13
538	450
516	276

I used the following code segment to calcuate the solution:

Listing 7: loadMovieLens

```
numUsrRatings = {}

for usr, usrData in prefs.iteritems():
    numUsrRatings[usr] = len(usrData["Ratings"])


numRatingsDescending = OrderedDict(sorted(numUsrRatings.items(), key = lambda k: k
    [1], reverse = True))

ctr = 0
prev = 0
with open("q6", "w+") as f:
    for x in numRatingsDescending.items():
        if (ctr < 5 or prev == x[1]):
            f.write(str(x[1]) + " & " + str(x[0]) + "\\\\ \n")
            ctr = ctr+1
            prev = x[1]
        else:
            break
```

Problem 7

```
7.  Which 5 raters most agreed with each other? Show the raters'
IDs and Pearson's r, sorted by r.
```

Problem 8

```
8.  Which 5 raters most disagreed with each other (negative
correlation)? Show the raters' IDs and Pearson's r, sorted by r.
```

Problem 9

```
9.  What movie was rated highest on average by men over 40? By men
under 40?
```

SOLUTION
The following movies all received a perfect 5.0 for men over 40:

Avg Ratings	Movie Title
5.0	Indian Summer (1996)
5.0	Leading Man, The (1996)
5.0	Unstrung Heroes (1995)
5.0	Little Princess, The (1939)
5.0	Great Day in Harlem, A (1994)
5.0	They Made Me a Criminal (1939)
5.0	Spice World (1997)
5.0	Boxing Helena (1993)
5.0	Little City (1998)
5.0	Double Happiness (1994)
5.0	Poison Ivy II (1995)
5.0	Two or Three Things I Know About Her (1966)
5.0	Star Kid (1997)
5.0	Ace Ventura: When Nature Calls (1995)
5.0	Grateful Dead (1995)
5.0	Aparajito (1956)
5.0	World of Apu, The (Apur Sansar) (1959)
5.0	Rendezvous in Paris (Rendez-vous de Paris, Les) (1995)
5.0	Prefontaine (1997)
5.0	Solo (1996)
5.0	Late Bloomers (1996)
5.0	Strawberry and Chocolate (Fresa y chocolate) (1993)
5.0	Marlene Dietrich: Shadow and Light (1996)
5.0	Faithful (1996)
5.0	Hearts and Minds (1996)

The following is a list of perfect 5.0 for men under 40

Problem 9 continued on next page. . .

Avg Ratings	Movie Title
5.0	Letter From Death Row, A (1998)
5.0	Perfect Candidate, A (1996)
5.0	Saint of Fort Washington, The (1993)
5.0	Quiet Room, The (1996)
5.0	Magic Hour, The (1998)
5.0	Entertaining Angels: The Dorothy Day Story (1996)
5.0	Maya Lin: A Strong Clear Vision (1994)
5.0	Angel Baby (1995)
5.0	Star Kid (1997)
5.0	Love in the Afternoon (1957)
5.0	Aiqing wansui (1994)
5.0	Prefontaine (1997)
5.0	Love Serenade (1996)
5.0	Leading Man, The (1996)
5.0	Crossfire (1947)
5.0	Santa with Muscles (1996)
5.0	Delta of Venus (1994)

I used the following code segment to calcuate the solution:

Listing 8: loadMovieLens

```
avgRatingsGT40 = {}
avgRatingsLT40 = {}

for movie, ratings in movieRatingsMenGT40.iteritems():
    avgRatingsGT40[movie] = np.mean(ratings)

avgRatingsDescendingGT40 = OrderedDict(sorted(avgRatingsGT40.items(), key = lambda
     k: k[1], reverse = True))

for movie, ratings in movieRatingsMenLT40.iteritems():
    avgRatingsLT40[movie] = np.mean(ratings)

avgRatingsDescendingLT40 = OrderedDict(sorted(avgRatingsLT40.items(), key = lambda
     k: k[1], reverse = True))



with open("q9", "w+") as f:
    prev = -1
    f.write("Older than 40:\n")
    for x in avgRatingsDescendingGT40.items():
        if (prev == -1 or prev == x[1]):
            f.write(str(x[1]) + " & " + str(x[0]) + "\\\\ \n")
            prev = x[1]
        else:
            break

    prev = -1
    f.write("Under 40:\n")
    for x in avgRatingsDescendingLT40.items():
        if (prev == -1 or prev == x[1]):
                f.write(str(x[1]) + " & " + str(x[0]) + "\\\\ \n")
                prev = x[1]
            else:
                break
```

Problem 10

```
10. What movie was rated highest on average by women over 40? By
women under 40?
```

SOLUTION
The following is a list of perfect 5.0 for women over 40:

Avg Ratings	Movie Title
5.0	Safe (1995)
5.0	Pocahontas (1995)
5.0	Bride of Frankenstein (1935)
5.0	Grand Day Out, A (1992)
5.0	Nightmare Before Christmas, The (1993)
5.0	Gold Diggers: The Secret of Bear Mountain (1995)
5.0	Mary Shelley's Frankenstein (1994)
5.0	Mina Tannenbaum (1994)
5.0	Letter From Death Row, A (1998)
5.0	Band Wagon, The (1953)
5.0	Angel Baby (1995)
5.0	Balto (1995)
5.0	Quest, The (1996)
5.0	Top Hat (1935)
5.0	Wrong Trousers, The (1993)
5.0	Tombstone (1993)
5.0	Foreign Correspondent (1940)
5.0	Best Men (1997)
5.0	Swept from the Sea (1997)
5.0	Ma vie en rose (My Life in Pink) (1997)
5.0	Funny Face (1957)
5.0	In the Bleak Midwinter (1995)
5.0	Shall We Dance? (1937)
5.0	Visitors, The (Visiteurs, Les) (1993)
5.0	Great Dictator, The (1940)
5.0	Shallow Grave (1994)

The following is a list of perfect 5.0 for women under 40:

Problem 10 continued on next page. . .

Avg Ratings	Movie Title
5.0	Stripes (1981)
5.0	Don't Be a Menace to South Central While Drinking Your Juice in the Hood (1996)
5.0	Someone Else's America (1995)
5.0	Grace of My Heart (1996)
5.0	Horseman on the Roof, The (Hussard sur le toit, Le) (1995)
5.0	Wedding Gift, The (1994)
5.0	Heaven's Prisoners (1996)
5.0	Everest (1998)
5.0	Umbrellas of Cherbourg, The (Parapluies de Cherbourg, Les) (1964)
5.0	Nico Icon (1995)
5.0	Maya Lin: A Strong Clear Vision (1994)
5.0	Mina Tannenbaum (1994)
5.0	Year of the Horse (1997)
5.0	Faster Pussycat! Kill! Kill! (1965)
5.0	Telling Lies in America (1997)
5.0	Prefontaine (1997)
5.0	Backbeat (1993)

I used the following code segment to calcuate the solution:

Listing 9: loadMovieLens

```
avgRatingsGT40 = {}
avgRatingsLT40 = {}

for movie, ratings in movieRatingsWomenGT40.iteritems():
    avgRatingsGT40[movie] = np.mean(ratings)

avgRatingsDescendingGT40 = OrderedDict(sorted(avgRatingsGT40.items(), key = lambda
     k: k[1], reverse = True))

for movie, ratings in movieRatingsWomenLT40.iteritems():
    avgRatingsLT40[movie] = np.mean(ratings)

avgRatingsDescendingLT40 = OrderedDict(sorted(avgRatingsLT40.items(), key = lambda
     k: k[1], reverse = True))



with open("q10", "w+") as f:
    prev = -1
    f.write("Older than 40:\n")
    for x in avgRatingsDescendingGT40.items():
        if (prev == -1 or prev == x[1]):
            f.write(str(x[1]) + " & " + str(x[0]) + "\\\\ \n")
            prev = x[1]
        else:
            break

    prev = -1
    f.write("Under 40:\n")
    for x in avgRatingsDescendingLT40.items():
```

Problem 10 continued on next page. . .

```
            if (prev == -1 or prev == x[1]):
                f.write(str(x[1]) + " & " + str(x[0]) + "\\\\ \n")
                prev = x[1]
            else:
                break
```

References

[1] K. Arthur Endsley. recommendation.py. https://github.com/arthur-e/Programming-Collective-Intelligence/blob/master/chapter2/recommendations.py, 2012.

[2] SciPy. Tentative NumPy Tutorial. http://wiki.scipy.org/Tentative_NumPy_Tutorial, 2012.