CptS 315 - HW 2

Zach Fechko (011711215)

October 12, 2022

Q1

Consider the following ratings matrix with 3 users and 6 items. Ratings are on a 1-5 star scale. Compute the following from this matrix:

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1	4	5		5	1	
User 2		3	4	3	1	2
User 3	2		1	3		4

a. Treat missing values as 0. Compute the jaccard similarity between each pair of users
b. Treat missing values as 0. Compute the cosine similarity between each pair of users
c. Normalize the matrix by subtracting from each non-zero rating, the average value for its user. Show the normalized matrix.
d. Compute the (centered) cosine similarity between each pair of users using the normalized matrix.

Q1 Solution

a.

$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$

```
import pandas as pd
import numpy as np

# Read in the data
ratings = pd.read_csv('ratings.csv').set_index('User')
print(ratings)

def jaccard(a, b):
    """
    Calculates the Jaccard similarity between two lists.
    """
    a = set(a)
    b = set(b)
    c = a.intersection(b)
```

```
    return float(len(c)) / (len(a) + len(b) - len(c))

print("Users 1 & 2: ", jaccard(list(ratings.loc["User 1"]), list(ratings.loc["User 2"]))) #jaccard simi
print("Users 1 & 3: ", jaccard(list(ratings.loc["User 1"]), list(ratings.loc["User 3"]))) #jaccard simi
print("Users 2 & 3: ", jaccard(list(ratings.loc["User 2"]), list(ratings.loc["User 3"]))) #jaccard simi
```

```
##         Item 1  Item 2  Item 3  Item 4  Item 5  Item 6
## User
## User 1       4       5       0       5       1       0
## User 2       0       3       4       3       1       2
## User 3       2       0       1       3       0       4
## Users 1 & 2:  0.5
## Users 1 & 3:  0.5
## Users 2 & 3:  1.0
```

b.

$C(A, B) = \frac{A \cdot B}{\|A\| \|B\|}$

```
from numpy.linalg import norm

def cosine(a: list, b: list):
    """
    Calculates the cosine similarity between two lists
    """
    A = np.array(a)
    B = np.array(b)
    return np.dot(A, B) / (norm(A) * norm(B))

print("Users 1 & 2: ", cosine(list(ratings.loc["User 1"]), list(ratings.loc["User 2"]))) #cosine simila
print("Users 1 & 3: ", cosine(list(ratings.loc["User 1"]), list(ratings.loc["User 3"]))) #cosine simila
print("Users 2 & 3: ", cosine(list(ratings.loc["User 2"]), list(ratings.loc["User 3"]))) #cosine simila
```

```
## Users 1 & 2:  0.6064457948612227
## Users 1 & 3:  0.5130146972572911
## Users 2 & 3:  0.6139406135149204
```

c.

```
# Normalize the matrix by subtracting from each non-zero rating, the average value for its user. Show t
def normalize_matrix(matrix: pd.DataFrame):
    """
    Normalizes a matrix by subtracting the mean from each value.
    """
    return matrix.sub(matrix.mean(axis=1), axis=0) #subtracts each value by the mean of the row

norm_ratings = normalize_matrix(ratings)
print(norm_ratings)
```

```
##           Item 1    Item 2    Item 3    Item 4    Item 5    Item 6
## User
## User 1  1.500000  2.500000 -2.500000  2.500000 -1.500000 -2.500000
## User 2 -2.166667  0.833333  1.833333  0.833333 -1.166667 -0.166667
## User 3  0.333333 -1.666667 -0.666667  1.333333 -1.666667  2.333333
```

d.

Using the `norm_ratings` df from part c and the `cosine` function from part b, we can calculate the cosine similarity between each pair of users.

```
# Compute the (centered) cosine similarity between each pair of users using the normalized matrix.
print("Users 1 & 2: ", cosine(list(norm_ratings.loc["User 1"]), list(norm_ratings.loc["User 2"]))) #cos
print("Users 1 & 3: ", cosine(list(norm_ratings.loc["User 1"]), list(norm_ratings.loc["User 3"]))) #cos
print("Users 2 & 3: ", cosine(list(norm_ratings.loc["User 2"]), list(norm_ratings.loc["User 3"]))) #cos
```

```
## Users 1 & 2:  -0.08390719402952727
## Users 1 & 3:  -0.10084389681792216
## Users 2 & 3:  -0.055470019622522924
```

Q2

Read the following two papers and write a brief summary of the main points in at most **TWO** pages.
Two Decades of Recommender Systems at Amazon.com
Industry Report: Amazon.com Recommendations: Item-to-Item

Q2 Solution