1. If R represents user-item rating matrix then matrix T= R\*RT , shows the user-user similarity between user i and j.

Consider for example R =

|  |  |  |
| --- | --- | --- |
| 1 | 0 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |
| 0 | 0 | 1 |

Here in R, rows represents users and columns represents items. So , Rij is 1 if user i likes item j and 0 otherwise.

Now if we compute matrix T= R\*RT,

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 1 | 2 | 1 |
| 1 | 1 | 1 | 0 |
| 2 | 1 | 3 | 1 |
| 1 | 0 | 1 | 1 |

Here, T matrix represents similarity between users, so Tii represents number of items liked by user i and Tij represents number items that are commonly liked by both user i and user j.

1. P diagonal matrix in problem represents number of items liked by user i. Q diagonal matrix in problem represents number of users liked that item i.

P matrix Q matrix

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 3 | 0 |
| 0 | 0 | 0 | 1 |

|  |  |  |
| --- | --- | --- |
| 3 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 3 |

Item similarity matrix SI is defined as = Q-1/2RTRQ-1/2

The inverse square root of diagonal matrix Q is nothing but reciprocal of each element’s square root in diagonal.

So Q-1/2 looks like

|  |  |  |
| --- | --- | --- |
| 1/√3 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1/√3 |

And RTR product evaluates to

|  |  |  |
| --- | --- | --- |
| 3 | 1 | 2 |
| 1 | 1 | 1 |
| 2 | 1 | 3 |

In this similarity matrix value at ij represents number of users commonly liked item i and j together.

And total product Q-1/2RTRQ-1/2 will gives

|  |  |  |
| --- | --- | --- |
| 3/√3.√3=1 | 1/√3.√1 | 2/√3.√3 |
| 1/√1.√3 | 1√1.√1=1 | 1/√3.√1 |
| 2/√3.√3 | 1/√1.√3 | 3/√3.√3=1 |

We can see that entry in ij represents **cosine similarity** between item i and j. For example look at item 1 column (1,1,1,0) and item 3 column (1,0,1,1), if we compute cosine similarity between these two, we’ll get 2/√3.√3 which is the value in entry (1,3) and (3,1) cells. Cosine similarity for a vector to itself is always 1, which is why all the diagonal values are 1.

Similarly, for user similarity matrix SU = P-1/2RRTP-1/2, we’ll get **cosine similarity** between each user pair.

Same holds truth even if ratings are in scale from 1-5. Then product RRT gives numerator part of cosine similarity which is nothing but **dot product** between vectors. P,Q matrices will have **squared sums** of ratings for corresponding users and items respectively. Finally when we compute similarity matrix with the above formulas we’ll get cosine similarity between users for SU and items for SI.

1. For user-user collaborative recommendation

𝑟𝑖𝑗= Σ*𝑐𝑜𝑠𝑆im*(𝑥,𝑖)∗𝑅𝑥𝑗

Here for user i to recommend item j we are taking the cosine similarity between user i and other users and multiplying (dot product) that with ratings for that item j. Consider for example, for user 1 to recommend item 1, we take the cosine similarity between that user and other users i.e nothing but first row of SU and multiply with ratings given to that item 1 by those users.

This makes sense to build recommendation matrix for all users by multiplying SU with R to say the affinity between user-item pairs.

So, for user-user filtering Γ = SU \* R matrix explains the affinity between all users and items.

In same way, item-item filtering Γ = R \* SI matrix explains the affinity between all users and items by taking dot product between cosine similarity of items for an item and rating of that item by different users.

1. Program to compute the top-5 shows for user 20 is written in “Task4.R” file. As it is given in problem we cleared first 100 entries for 20th user and predicted Top-5 recommendations based on user-user and item-item algorithms which are written in “Task4Top5Shows.xlsx”.

As part of next task, true positive rate plot is generated for k – {1,19} in “Task4\_truepositive.pdf”.

1. ItemKNN and WRMF algorithms are run using mono program. To run MediaLite input file is formatted first with one python helper script (Task5Helper.py) to [user,tv-show] column entries. And predictions are written into output file in descending order. Based on these scores top 10 shows were taken for both ItemKNN and WRMF algorithms and they are used to calculate Kendal-tau distance between all four algorithms predictions [ItemKNN, WRMF, item-item and user-user]

Python script to calculate item-item and user-user top 10 shows is in Task5.py file.

Final report of top 10 shows for each algorithm and Kendal-Tau distance is reported in Task5Report.xmlx

Below are the commands run to get ItemKNN and WRMF from MediaLite library.

*C:\Program Files\Mono\bin>mono "C:\Users\nrgurram\Desktop\EMDS\Courses\Analytic Databases\Assignment1\MyMediaLite-3.11\lib\mymedialite\item\_recommendation.exe" --training-file="C:\Users\nrgurram\Desktop\EMDS\Courses\Analytic Databases\Assignment1\dataset\sample\_out.txt" --recommender="WRMF" --prediction-file="C:\Users\nrgurram\Desktop\EMDS\Courses\Analytic Databases\Assignment1\dataset\predictions\_wrmf.txt"*

loading\_time 0.49

memory 13

training data: 9985 users, 563 items, 758878 events, sparsity 86.50057

WRMF num\_factors=10 regularization=0.015 alpha=1 num\_iter=15

training\_time 00:00:12.5462589 Wrote predictions to file C:\Users\nrgurram\Desktop\EMDS\Courses\Analytic Databases\Assignment1\dataset\predictions\_wrmf.txt. prediction\_time 00:00:06.0560414

*C:\Program Files\Mono\bin>mono "C:\Users\nrgurram\Desktop\EMDS\Courses\Analytic Databases\Assignment1\MyMediaLite-3.11\lib\mymedialite\item\_recommendation.exe" --training-file="C:\Users\nrgurram\Desktop\EMDS\Courses\Analytic Databases\Assignment1\dataset\sample\_out.txt" --recommender="ItemKNN" --prediction-file="C:\Users\nrgurram\Desktop\EMDS\Courses\Analytic Databases\Assignment1\dataset\predictions.txt"*

loading\_time 0.61

memory 13

training data: 9985 users, 563 items, 758878 events, sparsity 86.50057

ItemKNN k=80 correlation=Cosine q=1 weighted=False alpha=0.5 (only for BidirectionalConditionalProbability)

training\_time 00:00:11.9196976 Wrote predictions to file C:\Users\nrgurram\Desktop\EMDS\Courses\Analytic Databases\Assignment1\dataset\predictions.txt. prediction\_time 00:05:20.2192107