

## Assignment 3: Sequential Recommendations

Group members: Tuomas Porkamaa, Mueed Irfan

We Implemented solutions for all subtasks.

## How to run the code:

The submission consist a single python file:

- *Assignment3.py*

Program can be executed in terminal by a command `<python Assignment3.py>`. However, the suggested environment for running the program is the editor environment of Spyder IDE. There you can run parts of the program (implemented as cells) separately, and the program outputs make more sense and you can also modify user-specified hyperparameters more easily.

### NOTE BEFORE RUNNING THE CODE:

Modify the root variable to point to the directory (explicitly) where the program data is located (ratings.csv and movies.csv). Current setting is

```
root = '/home/tuomas/Python/DATA.ML.360/Assignment2/ml-latest-small/'
```

This variable can be found in following line:

- *Assignment3.py*, line 10.

Also, if you want to change the current user group under examination (current group initialized to [1,2,3]), change the following line:

- *Assignment3.py*, line 125.

We also implemented two group aggregation methods. You can select which one to use by setting the variable *weighted\_avg* to True or False. Current setting is *weighted\_avg=True*. You can modify it in following line:

- *Assignment3.py*, line 230.

## Assumptions

Same assumptions hold that were noted in the manual of assingment 1 (Assignment1\_manual.pdf). Also, in the group recommendation part, we considered only the common unrated movies between group members.

## Summary of implementation details (group aggregation)

The task given in this assignment was to implement a new method for producing sequential group recommendations. We came up with two alternative methods, that are quite different from each other. In this chapter we give a summary of both methods.

### Method 1. Weighted average

The first group aggregation method uses the satisfaction scores of group members as a weights of individual user ratings, and then calculates a weighted average, that is used as an overall score for a specific item. Individual satisfaction scores are implemented as they were defined in the lecture notes and research paper. Mathematical details of this method are the following:

$$\begin{aligned} sat(u_i, Gr_j) &= \frac{GroupListSat(u_i, Gr_j)}{UserListSat(u_i, A_{u_i,j})} \\ GroupListSat(u_i, Gr_j) &= \sum_{dz \in Gr_j} p_j(u_i, dz) \\ UserListSat(u_i, A_{u_i,j}) &= \sum_{dz \in A_{u_i,j}} p_j(u_i, dz) \end{aligned}$$

Problem: How to calculate score for item  $d_z$  for group  $G$  in iteration  $j$ , a.k.a  $score(G, d_z, j)$ .

Basic approach is to use the group average only (Naive solution):

$$score(G, d_z, j) = \frac{1}{|G|} \sum_{u_i \in G} p_j(u_i, d_z)$$

Instead of traditional mean, consider a weighted average:

$$score(G, d_z, j) = \sum_{u_i \in G} w_{u_i,j} * p_j(u_i, d_z)$$

where  $\sum_{u_i \in G} w_{u_i,j} = 1$ .

We want to give more weight (eg. more decision-making power to rank the item) for those users, that were dissatisfied in previous iteration. Basically this can be done by setting  $w_{u_i,j}$  to a larger number, if user  $u_i$  was dissatisfied in iteration  $j - 1$ . We implemented two alternative methods that keep track of this behaviour:

$$w_{u_i,j} = \frac{sat(u_i, Gr_{j-1})^{-1}}{\sum_{u_k \in G} sat(u_k, Gr_{j-1})^{-1}} \quad (\text{Equation 1})$$

$$w_{u_i,j} = \frac{1 - sat(u_i, Gr_{j-1})}{\sum_{u_k \in G} (1 - sat(u_k, Gr_{j-1}))} \quad (\text{Equation 2})$$

In both cases (Eq1 & Eq2) the weight gets larger if the user  $u_i$  was dissatisfied in previous iteration. In practice, Equation 1 is more “aggressive”, since it gives more easily a large weight value for those users who were extremely dissatisfied and other users were mainly satisfied. In other case, Equation 2 is more “uniform” in same kind of cases, but it gives more weight for the users that were only a bit of dissatisfied compared to others. We used Equation 1 in our implementation.

## Method 2. Combination of group average and Least misery

This method includes a combination of the average method and the least misery method. The average method ensures that all users are treated equally and the preference is the average for the whole group. The least misery method ensures that the user with the lowest rating in the group acts as a veto for the group. Our goal here is to make all the users somewhat equally satisfied with the recommendations we produce.

$$score(G, d_z, j) = \frac{avgScore(G, d_z, j) + LeastScore(G, d_z, j)}{2}$$

If we use least misery it would ensure that the user with the lowest score gets satisfied. If we use average method it would ensure that the average for the group's recommendations are processed.

The method we introduce treats all the users equally giving fairly little advantage to the least satisfied user in the group. This method brings the overall group satisfaction to an average by satisfaction score by increasing the least satisfied score and decreasing the most satisfied users.

# Example outputs of our program

## Method 1. Weighted average

Sequential group recommendations using Weighted group average.  
List of 20 most relevant movies in 5 iterations for user group [1 2 3]:

-----

### Iteration 1

(Id : 132333), Seve (2014)  
(Id : 5490), The Big Bus (1976)  
(Id : 25947), Unfaithfully Yours (1948)

.  
. (14 between)

.  
(Id : 26326), Holy Mountain, The (Montaña sagrada, La) (1973)  
(Id : 5416), Cherish (2002)  
(Id : 3951), Two Family House (2000)

### Iteration 2

(Id : 5328), Rain (2001)  
(Id : 4495), Crossing Delancey (1988)  
(Id : 6201), Lady Jane (1986)

.  
. (14 between)

.  
(Id : 6442), Belle époque (1992)  
(Id : 3792), Duel in the Sun (1946)  
(Id : 3303), Black Tar Heroin: The Dark End of the Street (2000)

### Iteration 3

(Id : 495), In the Realm of the Senses (Ai no corrida) (1976)  
(Id : 1349), Vampire in Venice (Nosferatu a Venezia) (Nosferatu in Venice) (1986)  
(Id : 1759), Four Days in September (O Que É Isso, Companheiro?) (1997)

.  
. (14 between)

.  
(Id : 3473), Jonah Who Will Be 25 in the Year 2000 (Jonas qui aura 25 ans en l'an 2000) (1976)  
(Id : 3056), Oxygen (1999)  
(Id : 2037), Candleshoe (1977)

#### Iteration 4

(Id : 94810), Eva (2011)

(Id : 88448), Paper Birds (Pájaros de papel) (2010)

(Id : 92494), Dylan Moran: Monster (2004)

.

.(14 between)

.

(Id : 65740), Revenge of the Nerds IV: Nerds in Love (1994)

(Id : 65738), Revenge of the Nerds III: The Next Generation (1992)

(Id : 52867), Ex, The (2007)

#### Iteration 5

(Id : 55020), Ten, The (2007)

(Id : 82744), Faster (2010)

(Id : 95149), Superman/Batman: Public Enemies (2009)

.

.

.

(Id : 26169), Branded to Kill (Koroshi no rakuin) (1967)

(Id : 3096), My Man Godfrey (1957)

(Id : 4055), Panic (2000)

Didn't have time to give example outputs about method 2. (Deadline closes in 2 minutes)