Multiagent Systems Semester Project 2021-22

Social Choice Mechanisms for Recommender Systems

Perakis Georgios 2016030188

Maria Eleni Tzortzi 2016030140

Kyriakos Siaterlis 2016030---

**TASKS**

1. On the first task, the users’ and items’ D-dimensional distributions were used in order to find a preference list for each user. First the Kullback-Leibler Divergence Criterion was used in order to compute the similarity of the D-Dimensional Gaussians. Using the Kullback-Leibler score of a user for an item,

We computed each user’s rating on every item using the following formula:

Based on these ratings the 10 best items for the first 50 users were found and printed.

The corresponding code for the first task can be found in the files ***MA\_Task1.m*** and ***findPrefTable.m***

The executional file is ***MA\_Task1.m*** where the two xls files are read. In order for the code to work, the file paths have to be modified to the location of the files in your computer.

The results for the first 10 users are shown in the table bellow.

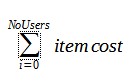
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Item\User* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* |
| *1st* | 44 | 287 | 15 | 429 | 163 | 68 | 307 | 147 | 214 | 53 |
| *2nd* | 432 | 131 | 253 | 31 | 37 | 496 | 229 | 256 | 284 | 212 |
| *3rd* | 336 | 450 | 454 | 384 | 448 | 76 | 161 | 370 | 162 | 202 |
| *4th* | 99 | 122 | 332 | 67 | 93 | 452 | 295 | 55 | 396 | 456 |
| *5th* | 55 | 289 | 424 | 180 | 166 | 40 | 141 | 397 | 482 | 165 |
| *6th* | 480 | 202 | 339 | 69 | 356 | 427 | 120 | 94 | 209 | 486 |
| *7th* | 464 | 43 | 415 | 280 | 281 | 474 | 160 | 361 | 318 | 199 |
| *8th* | 151 | 182 | 174 | 1 | 403 | 166 | 372 | 302 | 184 | 167 |
| *9th* | 239 | 165 | 438 | 70 | 351 | 210 | 268 | 282 | 94 | 219 |
| *10th* | 427 | 322 | 429 | 236 | 90 | 9 | 1 | 240 | 338 | 405 |

1. To compute the borda Count, we assign a score to each item in their list based on their rank in this list.

Then we choose the item with the maximum score.

The executional file is ***MA\_Task3.m*** where the first time you run it you save all the matrices in an excel file in order to not compute them every time. In order to accomplice this you must set the **compute\_cached** =1.

We designed a mechanism that fairly allocates the total cost of an item among the agents in a group, based solely on their budget and their rating in this item. The sum of the share, each player pays, is equal to the item\_cost.

 The r(***rating***) each user has for an item is a real number ϵ [0,10].

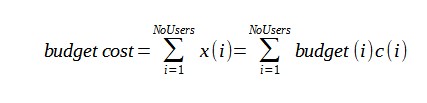
The sum of the players budget is bigger or equal to the ***item\_cost***.

The first step of this procedure is to open the xls files and find the items that can be bought, based on the group budget, and insert them in the list ***feasible\_items***.

The second step is an initial approximation and is to compute the case where each user pays an amount of money weighed by his rating of this item (**user\_budget**).

Each user pays x(i)=budget(i)\*c(i), where c is a normalized rating which is a positive number between [0,1]

The maximum amount of money all players can(!) pay(based on their preferences) is

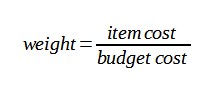
 If the ***budget\_cost*** is less than the ***item\_cost*** of the selected feasible item then each player that pays less than his budget should cover the deficit of money, until this deficit becomes zero.

More specifically, algorithmically an initial approximation of payment takes place where all players that pay more than their budget, pay their budget (not weighed).This deficit of each player adds up and then the more satisfied players pay more in order to cover this deficit

If the ***budget\_cost*** is bigger or equal than the***item\_cost*** then each user should pay

where:

for ***budget\_cost*** the ***weight =1***

for ***item\_cost*** the ***weight***=?

The corresponding code for the 4th task can be found in the files ***MA\_Task4.m*** and ***findPrefTable.m***

The executional file is ***MA\_Task4.m***.

TASK 5

BONUS