# Assignment: Mining User Groups

Sihem Amer-Yahia, Behrooz Omidvra-Tehrani

April 29, 2019

We are given a user dataset **D** where each line has the following schema **<u, i>** which means that user u expressed interest in item **i** (for instance, **u** purchased **i**, **u** rated **i**). Your task is to mine groups from user data. To do that you will use a pattern mining algorithm LCM [1], and a multi-objective optimization algorithm, MOMRI [2]. The notion of "user group" is described in [3,4,5].

**Step 1.** First, the dataset **D** should be transformed. Identifiers for both users and items are mapped to a non-negative integer space. For instance if the movie Titanic (as an item) is mapped to "25" and the user "John" is also mapped to "120", the tuple <120,25> means that John rated the movie Titanic. For this aim, you will need to execute the Python script **pmr.py** (https://github.com/behroozomidvar/PatternMiningReady.git). The script takes as input the dataset **D** as a set of <user,item,rating>. It returns a text file **pmr.txt** (abbr. pattern-mining ready). The separator between users and items in **D** is ",". If another character is used, change the variable **sep** in line 5 of **pmr.py**. Also if the user IDs do not start from zero, change the **offset** variable in line 6.

**Dataset to be used:**
**MovieLens 1M records**: available at http://files.grouplens.org/datasets/movielens/ml-1m.zip
Step 1 is necessary.

**Step 2 (support-based mining).** Use LCM to mine user groups:
http://research.nii.ac.jp/~uno/code/lcm53.zip
The following command will mine close (`CfI`) frequent groups whose description size is between 5 and 100 (optional parameters) and whose minimum support is 3.

```
./lcm CfI -l 5 -u 100 pmr.txt 3 out.txt
```

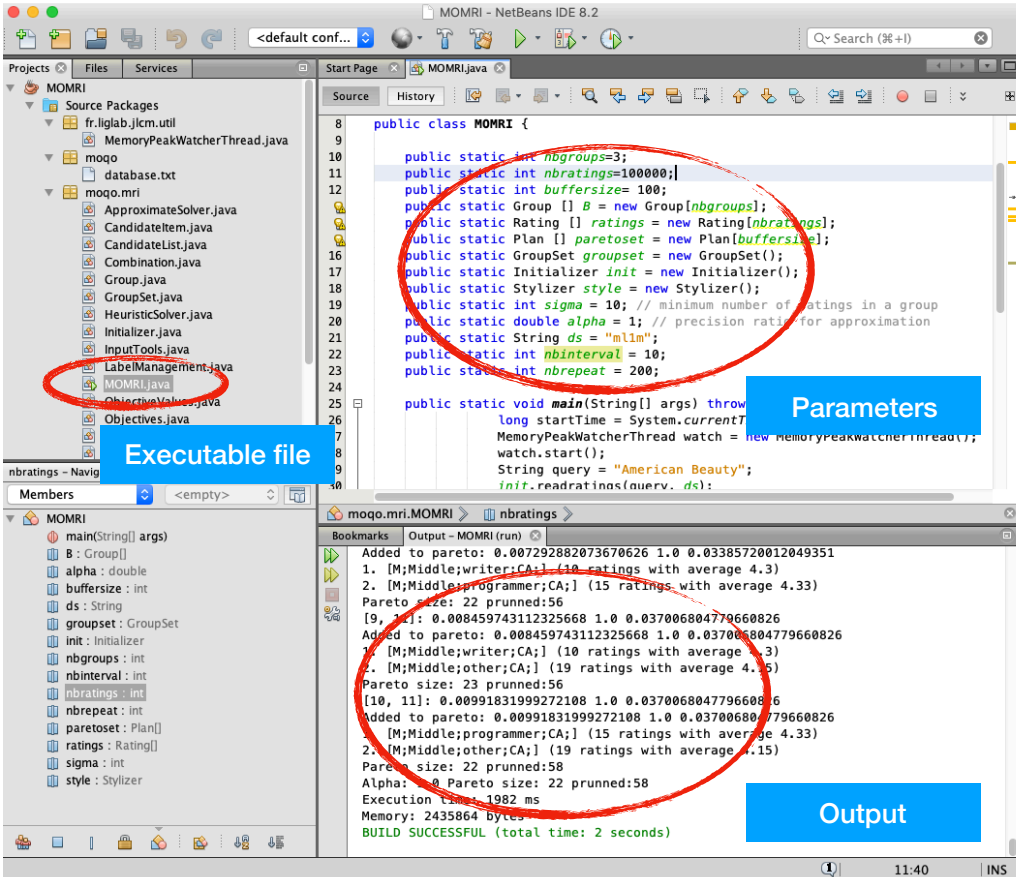
Each line in the output file **out.txt** represents a group:

```
Group description:[set of items] (support) Group content:[set of users]
```

The description of the group is `[set of items]`. The set of group members is `[set of users]`. The size of each group (number of members) is `support`.

**Step 3 (multi-objective mining).** You need to download and unzip the file at
https://www.dropbox.com/s/xl32w5uly9uxsb3/MOQO-MRI.zip?dl=0

It is a Java NetBeans project. It should be opened as a NetBens project (NetBeans File menu → Open project). In the package called "MOQO.MRI", the MOMRI.java is the executable file. Lines 10-23 are the configuration parameters. Refer to [2] for more details. The output of the algorithm reports the progress in finding Pareto plans. Each numbered line represents an alpha-non-dominating group whose description is mentioned in brackets. The set of groups forms a group-set which is added to the Pareto.




**Input data**. The parameter “ds” (line 21 of MOMRI.java) specifies the name of the dataset to use. MovieLens 1M (ds=“ml1m”) is considered as the default dataset. You can also try MovieLens 100K dataset (ds=“ml100k”). The method “read ratings()” in line 30 of MOMRI.java reads ratings from the data file on disk. The data file is hosted in the “data” directory, followed by a sub-directory with the exact same name as the value of the “ds” parameter. In case of MovieLens 1M, there is a file in the directory “data/ml1m” with the name “database.txt”. It is a TSV file (i.e., tab-separated) where each line represents a rating, as follows:

user_id movie_title rating_score user_gender user_age_category user_occupation user_location

For instance, line 23 of “database.txt” describes a rating for the movie “Die Hard” by a user with ID 299, who is a male young doctor living in Missouri state in the US (MO), and rates the movie with the score 4 (out of 5). The “database.txt” is the concatenation of all three files in the original MovieLens 1M dataset for an easier data processing.

**Multi-objective optimizer**. The class “ApproximateSolver” is responsible for finding the alpha-approximated Pareto front. The method “solve()” of the class (line 36 of MOMRI.java) executes the alpha-approximation multi-objective optimizer. It admits as input the following parameters:

- nbgroups (size of group-sets)

- ratings (the data, read from file in line 30 of MOMRI.java)

- nbratings (the number of ratings, i.e., size of the data)

- sigma (minimum size of groups, i.e., groups having less than sigma users won't be considered)

- alpha (approximation precision)

**Objectives**. By default, the alpha-MOMRI algorithm optimizes all three objectives: diversity "d", coverage "c" , and rating diameter "e". One can reduce this set to two objectives (or even to one), or introduce his/her own objective for optimization. The class "Plan" used in line 51 of "ApproximatedSolver.java" builds a plan for a group-set (refer to Definition 2 in our PKDD'16 paper). A plan consists of the values of all objectives ("d", "c", and "e") applied on the group-set (see lines 17-25 in "Plan.java"). Each of these objective computation (lines 21-23 in "Plan.java") can be commented out and the optimizer won't optimize for that objective.

Now suppose we want to add a new objective to our optimizer, e.g., "rating_deviation" which returns a normalized score in the range [0,1] and represents the heterogeneity of the rating scores in the group-set. Given a group-set gs, in case all its rating scores are equal, rating_deviation(gs)=0. First, the "compute" function in line 94 of "Objectives.java" should be extended with a new case, e.g., for "r", where "val = this.rating_deviation(gs);". Then the actual implementation of the "rating_deviation()" should be also written in "Objectives.java". The function "rating_deviation()" should get as input only a group-set, and return a value between 0 and 1.

# References

[1] Uno, Takeaki, Tatsuya Asai, Yuzo Uchida, and Hiroki Arimura. "An efficient algorithm for enumerating closed patterns in transaction databases." In Discovery science, pp. 57-59. Springer Berlin/Heidelberg, 2004

[2] Omidvar-Tehrani, Behrooz, Sihem Amer-Yahia, Pierre-François Dutot, and Denis Trystram. "Multi-Objective Group Discovery on the Social Web." In Proceedings of ECML/PKDD 2016

[3] Mahashweta Das, Sihem Amer-Yahia, Gautam Das, Cong Yu: MRI: Meaningful Interpretations of Collaborative Ratings. PVLDB 4(11): 1063-1074 (2011)

[4] Omidvar-Tehrani, Behrooz, Sihem Amer-Yahia, and Alexandre Termier. "Interactive user group analysis." In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 403-412. ACM, 2015

[5] Sihem Amer-Yahia, Sofia Kleisarchaki, Naresh Kumar Kolloju, Laks V. S. Lakshmanan, Ruben H. Zamar: Exploring Rated Datasets with Rating Maps. WWW 2017: 1411-1419