

# Recommendation System – Description of Models

This document provides a brief description of the models implemented in the BlueLibs Recommender system.

## **A. What are recommender systems ?**

In simple terms, recommender systems are algorithms aimed at suggesting relevant items to users (items can be movies to watch, books to read or other products, depending on industry).

More generally, recommender systems are often used to find similarities for items or users, and can be applied in diverse fields such as e-commerce, e-learning or social media.

## **B. Description of the models implemented in the BlueLibs Recommender system:**

At this time, there are three types of recommendation models implemented:

- a) A non-personalized (or popularity based) recommender
- b) Two collaborative, also called memory based recommenders:
  - b1) User based recommender
  - b2) Item based recommender

Input-Output of the models:

The current BlueLibs recommendation systems takes as input a .csv file of available ratings (identified by a pair (user,item)), a limit (number of desired ratings), and a queried user for which the ratings are desired (for the collaborative models) The output is a dictionary with items of the form {item:rating} (of size limit).

### **Description of the models:**

#### **a) Non-personalized recommender :**

This type of recommender is not „personalized” to any specific user. In a sense, this type of recommender system is the simplest type of recommender. It provides the same recommendations to all users. This model can be useful in the „cold start” scenarios, in which we do not have any data regarding the personal preferences of a certain user.

In order to make this non-personalized recommendation, we calculate a generic rating for each item, using the formula:

$$(1) \quad R_j = \frac{\sum_{u=1}^{N_j} r_{uj} + \lambda \mu}{N_j + \lambda} \quad \text{where } j = 1..n\_items, N_j \text{ is the number of users that rated item } j,$$

$r_{uj}$  is the rating of item  $j$  by user  $u$ ,  $\mu$  is the overall mean (of all ratings),

$\lambda$  is a parameter that controls how much  $R_j$  is affected by the overall mean (if  $\lambda$  is large ( $\geq 5$ ) then  $R_j$  is strongly affected by the global mean  $\mu$ , if  $\lambda$  is small ( $\ll 1$ ) then  $R_j$  is only slightly affected by the global mean  $\mu$ . Currently we take  $\lambda = 1$ ).

After this rating is calculated, we sort the items descendingly, and provide the top (limit) recommendations.

We use formula (1) instead of a simpler (say mean of each item formula) because such a simpler formula would be un-reliable if the number of ratings for that particular item is small.

A more in depth description of formula (1) can be found in [1].

#### **b1) Collaborative user based recommender:**

An user based (personalized) collaborative recommender is a „true-and-tried” classical recommender system, still in use (although as part of more sophisticated systems) by sites such as Amazon, Netflix and Youtube.

This method has two steps:

-- first, find a similarity measure  $S = (s_{uv})$  between users (a similarity matrix)

-- second, use this similarity matrix to predict the generic rating  $R_{u,i}$  for user u and item i.

The user similarity measure we use is a Pearson based similarity given by:

$$(2) \quad s_{u,v} = \frac{\sum_{j \in \Psi_{u,v}} (r_{u,j} - \bar{r}_u)(r_{v,j} - \bar{r}_v)}{\sqrt{\sum_{j \in \Psi_{u,v}} (r_{u,j} - \bar{r}_u)^2} \sqrt{\sum_{j \in \Psi_{u,v}} (r_{v,j} - \bar{r}_v)^2}},$$

where

$\Psi_{u,v}$  = set of items rated by both users u and v

$\bar{r}_u$  and  $\bar{r}_v$  = average of all ratings for user u and user v, respectively

$r_{u,j}$  = rating of item j by user u

$r_{v,j}$  = rating of item j by user v

Basically the similarity  $s_{u,v}$  measures the cosine of the angle between the „deviations“  $r_{u,j} - \bar{r}_u$  and  $r_{v,j} - \bar{r}_v$  over the set of rated items (which is also the correlation between these two datasets).

The prediction  $R_{u,i}$  is then calculated using the formula:

$$(3) \quad R_{u,i} = \bar{r}_u + \frac{\sum_{v \in \Psi_i} s_{u,v} (r_{v,i} - \bar{r}_v)}{\sum_{v \in \Psi_i} |s_{u,v}|} \quad (\text{user based prediction formula}).$$

$\bar{r}_u$  and  $\bar{r}_v$  = average of all ratings for user u and user v, respectively

where  $\Psi_i$  = set of all users that have rated item i (except user u)

$r_{v,i}$  = rating of item i by user v

We make a further refinement of formula (2) by considering only the top (K) most similar users to the target user u. In our runs we considered K=25.

After the predicted ratings  $R_{u,i}$  are calculated, we sort descendingly the ratings for each user, and provide the top (limit) recommendations of those items not already rated by user u.

## b2) Collaborative item based recommender:

The item based collaborative filtering is theoretically very similar to the user-based collaborative filtering method b1. In practice, however, this method is preferable because it is less computationally intensive and it tends to produce lower RMSE's.

It comprises the same steps as b1, but it views the items as „main objects” now, therefore:

-- first, we find a similarity measure  $S = (s_{ij})$  between items (a similarity matrix) using the formula:

$$(4) \quad s_{ij} = \frac{\sum_{u \in \Psi_{ij}} (r_{ui} - \bar{r}_i)(r_{uj} - \bar{r}_j)}{\sqrt{\sum_{u \in \Psi_{ij}} (r_{ui} - \bar{r}_i)^2} \sqrt{\sum_{u \in \Psi_{ij}} (r_{uj} - \bar{r}_j)^2}},$$

where

$\Psi_{ij}$  = set of users that rated both items i and j

$\bar{r}_i$  and  $\bar{r}_j$  = average of all ratings for items i and j, respectively

$r_{ui}$  = rating of item i by user u

$r_{uj}$  = rating of item j by user u

-- based on this similarity matrix we calculate  $R_{ui}$  = a predicted rating for user u and item i, using the formula:

$$(5) \quad R_{ui} = \bar{r}_i + \frac{\sum_{j \in \Psi_u} s_{ji} (r_{uj} - \bar{r}_j)}{\sum_{j \in \Psi_u} |s_{ji}|} \quad (\text{item based prediction formula}),$$

where  $\Psi_u$  = set of all items that user u has rated (except item i) and

the meanings of the other quantities are similar with those from formula (3).

Once again, we make a further refinement of formula (3) by considering only the top (K) most similar items with the target item i. In our runs we considered K=25.

Similar with the user-based algorithm, after the predicted ratings are calculated, we sort descendingly the ratings for each user, and provide the top (limit) recommendations of those items not already rated by user u.

### C. Performance of the collaborative based algorithms. Computational results:

It has typically been reported in literature (see, for example 4) that the user-based and item-based algorithms perform similarly in terms of RMSE, where:

$$(6) \quad RMSE = \sqrt{\frac{1}{n} \sum_{u,j} (R_{u,j} - r_{u,j})^2}, \text{ where } r_{u,j} \text{ is the exact (known) rating of user } u \text{ for item } j \text{ and } R_{u,j} \text{ is the predicted rating of user } u \text{ for item } j$$

(see, for example [4]).

From a practical point of view, however, item-based recommenders are often preferred in systems where the number of users surpasses the number of items. In addition, in a typical real world scenario, the user base needs to update more often than the item base, which would require an increased computational load on a user based system.

Our own computational results show that the RMSE for the item based recommender is slightly smaller than the RMSE for the user based recommender in all test cases.

Specifically, we performed runs on a PC with 2.4 GHz nominal (4.8 Ghz max in turbo mode) Intel 8 cores I7 processor, 32 GB RAM computer. We have tested the algorithms for standard Movie Lens (<https://grouplens.org/datasets/movielens/>) datasets with 100000 ratings , 1 million and 10 millions ratings. The results are shown in the tables below (we denote nu = number of users and ni =number of items for each case):

#### c1) MovieLens 100K (dataframe with 100 000 rows) (nu=943 , ni=165

Algorithm	Train RMSE	Training time	RAM space needed
User based	0.89	0.26 secs	[45 MB,100MB]
Item based	0.87	0.39 secs	[45 MB,100MB]

c2) Movielens 1M (dataframe with 1 000 000 rows): (nu=6040, ni=3706):

Algorithm	Train RMSE	Training time	RAM space needed
User based	0.88	21 secs	~728 MB
Item based	0.86	19 secs	~ 728 MB

c3) Movielens 10M (dataframe with 10 000 000 rows) (nu=69878, ni= 10677 ):

Algorithm	Train RMSE	Training time	RAM space needed	Observations
User based			~35 GB	Out of memory: message: unable to allocate 36.4 GB is space for the similarity matrix
Item based	0.85	~ 5mins	~15 GB	

## References:

[1] „Comparative analysis of recommender systems and its enhancements” by Kai Pradeep in International Journal of Engineering and Technology, January 2018 at:

[https://www.researchgate.net/publication/328231954\\_Comparative\\_analysis\\_of\\_recommender\\_systems\\_and\\_its\\_enhancements](https://www.researchgate.net/publication/328231954_Comparative_analysis_of_recommender_systems_and_its_enhancements)

[2] “Item-Based Collaborative Filtering Recommendation Algorithms” by B. Sarwar et. al at: [http://files.grouplens.org/papers/www10\\_sarwar.pdf](http://files.grouplens.org/papers/www10_sarwar.pdf)

[3] “Recommender Systems tutorial at International Joint Conference on Artificial Intelligence”, Barcelona, July, 2011, by D. Jannach et. al. at :

[https://ucilnica.fri.uni-lj.si/pluginfile.php/97649/mod\\_resource/content/1/Recommender%20Systems%20Tutorial\\_IJC\\_AI\\_2011.pdf](https://ucilnica.fri.uni-lj.si/pluginfile.php/97649/mod_resource/content/1/Recommender%20Systems%20Tutorial_IJC_AI_2011.pdf)

[4] “Comparison of user based and item based collaborative filtering recommendation systems” by P. Bostrom and M. Flipsson , KTH, Stockholm, Sweden, 2-17 at : <http://www.diva-portal.org/smash/get/diva2:1111865/FULLTEXT01.pdf>