A Framework for Linear TV Recommendation by Leveraging Implicit Feedback

Abhishek Agarwal[1], Soumita Das[1], Joydeep Das[2] and Subhashis Majumder[1]

[1] Dept. of Computer Sc. & Engg., Heritage Institute of Technology, Kolkata, WB, India
agarwlabhishek@gmail.com
soumitamum210@gmail.com
subhashis.majumder@heritageit.edu

[2] The Heritage Academy, Kolkata, WB, India
joydeep.das@heritageit.edu

Abstract. The problem with recommending shows/programs on linear TV is the absence of explicit ratings from the user. Unlike video-on-demand and other online media streaming services where explicit ratings can be asked from the user, the linear TV does not support any such option. We have to rely only on the data available from the set top box to generate suitable recommendations for the linear TV viewers. The set top box data typically contains the number of views (frequency) of a particular show by a user as well as the duration of that view. In this paper, we try to leverage the feedback implicitly available from linear TV viewership details to generate explicit ratings, which then can be fed to the existing state-of-the-art recommendation algorithms, in order to provide suitable recommendations to the users. In this work, we assign different weightage to both frequency and duration of each user-show interaction pair, unlike the traditional approach in which either the frequency or the duration is considered individually. Finally, we compare the results of the different recommendation algorithms in order to justify the effectiveness of our proposed approach.

Keywords: Recommender Systems, Linear TV, Collaborative Filtering, Implicit Feedback

1 Introduction

Recommender systems (RS) [1] produce recommendations through algorithms like collaborative filtering (CF) or content-based filtering. Content-based filtering [3] predict preferences based on the content of the items and the interests of the users, while CF [13] builds a model from a user's past behavior (items previously consumed or ratings given to those items) as well as decisions made by other similar users. This model is then used to predict items that the user may have an interest in. CF algorithms are of two types: memory based and model based algorithms. Memory based algorithms identify the top-K most similar users (neighbors) to the active user, and then use a weighted sum of the ratings of the neighbors to predict missing ratings for the active user [3]. Model based algorithms [11], in contrast, implement data mining or machine learning algorithms on the training data to estimate or learn a model to make predictions for an active user. Model based algorithms handle the sparsity and scalability problems better than the memory based algorithms. The disadvantages of this technique are in model building and updating that often turn out to be costly.

The large number of TV programs give users many choices, however, it may confuse the users as it is not easy to find a TV program that is interesting, because of the tremendous number of choices available. Thus, TV program recommender systems have become really important. Traditional RS typically use a user-item rating matrix which records the ratings given by different users to different items. However, in case of recommending TV programs we do not get explicit ratings from the users since there is no option for them to rate a particular show on the TV. Therefore we need to leverage the implicit data available (provided by set top boxes) in the system to provide suggestions for TV shows to the existing as well as new users. This implicit feedback focuses mainly on the number of times a particular user has watched a particular show (frequency of user-show interaction) and the corresponding time duration for which the user has watched the show (duration of user-show interaction). Majority of the existing TV recommendation systems [2,7,14] rely only on implicit feedback. However, in order to apply standard CF algorithms, we need to convert the feedback into numeric ratings. In this work, our primary objective is to map the implicit feedback into explicit ratings and then use any of the existing CF based algorithms to generate recommendations.

The previous researches done in the domain of TV recommendation either consider the frequency of user-show interaction [10] or the duration of user-show interaction [14] along with demographic information of the users. Only considering the frequency of views while recommending shows has drawbacks, because it alone cannot indicate whether a user liked or disliked a particular show. For example, if a user switches to a particular show often, it will lead to a higher frequency of views and indicate that the user really likes the show. However, the duration of these views can be very short which might rather indicate that the user does not like the show that much and also does not like to watch it for a long period of time. This kind of observation is very common when a user is surfing through the channels searching for something interesting or during the commercial breaks. Similarly, duration of a view cannot alone indicate whether a user likes or dislikes the show. A higher duration of view may indicate appreciation of one particular episode but a corresponding lower frequency of view may indicate otherwise. Thus, we need to infer the implicit feedback properly by giving appropriate weightage to both the frequency and duration of all such views. In this paper, we propose a recommendation framework where we consider both the frequency and duration of user-item interaction and also assign different weightage to both of them in order to achieve best possible recommendations. The results of the experiments conducted have shown that when both these factors are considered together, the recommendations are more effective than the case when either one of them were considered separately.

The rest of the paper is organized as follows: In section 2, we review some of the past works related to TV recommendation. In section 3, we present the solution framework and our proposed approach. In section 4, we describe our experimental settings and in section 5, we report and interpret our results. We conclude in section 6 discussing our future research directions.

2 Related Work

Recommender systems play a very important role in increasing the popularity of linear TV. Personalized suggestions for TV programs help linear TV compete with the modern video-on-demand services. In one of the early papers on TV recommendation, the authors present the idea of a personalized Electronic Program Guide (EPG) [8]. They identified some important research questions on TV program recommendation including user profiling methods, use of recommendation algorithms and how to use group recommendation to TV users. Another personalized EPG based TV recommendation was proposed [6] where the authors use one hybrid recommendation algorithm to learn users' preferences in terms of different TV channels, genres, etc. to generate new recommendations. Ardissono et al. [2] also proposed a hybrid recommendation approach on the basis of implicit preferences of the users captured in terms of program genres and channels, user classes and viewing history. All these information were gathered from users set top box or were downloaded from satellite stream. The importance of social media in TV shows recommendation is exploited by Chang et al. [5], where the authors propose a user preference learning module that includes user's past viewing experience as well as friendship relations in social networks. Cremonesi et al. [7] proposed a context based TV program recommendation system where the current context of the user along with implicit feedback is explored while making suggestions. There have been some work that intended to compute the top channel, top channel per user and also top channel per user per slot [14]. This is computed based on popularity, which is calculated in terms of total watching minutes accumulated by the channel. They used two important functions namely score aggregation and rank aggregation in order to provide effective recommendations. In this work, we map the implicit preferences of users into explicit ratings and then use standard recommendation algorithms to generate recommendations.

3 Solution Framework

Unlike conventional RS, recommending TV shows is more challenging due to several reasons. Firstly, content of TV programs change over time. Some TV programs are broadcast only once (e.g. movies) and do not repeat over a specific period of time, while some other shows repeat on the same day (e.g. episode of a show). The dynamic content of TV programs become a constraint in providing effective recommendations. Secondly, linear TV programs have a predefined schedule and therefore the set of recommended items is confined to the programs getting broadcasted at the moment when the recommendation is sought. Thirdly, feedback of the users about different TV shows are usually implicit (viewed/not viewed). Therefore it becomes difficult to implement pure CF algorithms to generate recommendations for linear TV since we do not get explicit ratings of the TV shows from the users. Fourthly, it is not possible for a user to watch multiple shows at the same time. Thus any recommendation algorithm must consider the programs which are scheduled simultaneously so that the most interesting shows can be recommended to the users. In this paper, we address the TV shows recommendation problem by analyzing the implicit feedback of the users in order to find the most important features and then provide appropriate weightage to those features. In other words, we try to convert the implicit feedback of a user-program interaction pair into an explicit rating by assigning proper weightage to the different features of the said user-program interaction pair.

3.1 Proposed Approach

Note that the two most important features of any user-show interaction are (a) frequency- the number times a user U has watched a show P over a given period of time, and (b) duration- it is the amount of time a user U spends watching a unique instance of a show P. First, we calculate the frequency of each unique user-show interaction pair. The average duration of view for an unique user-show interaction pair is calculated by summing the durations of each view of the pair and then dividing the sum by the frequency of that pair. Further, let us state two important points regarding the behavior of TV users.
(1) Most users tend to skip advertisements during a break which reduces the actual running time of the show. For example, a show that has been scheduled to run for 30 minutes actually has only 22 minutes of content. The rest 8 minutes might be spent on commercials and other promotional activities during which the users tend to switch channels.
(2) A lot of shows are broadcasted multiple times a day and most of the users have a tendency to watch it only once. Hence, the total number of unique instances of a particular show is crucial.

In order to get better and accurate results the above two points need to be considered before proceeding with any further computations. Since, the information related to the above two points cannot be inferred directly from the data set we are using, we need to make the following assumptions:

- Since the actual running time of a show excluding the commercials is not available to us, we consider the maximum average duration of that show from all the users, and use it as the actual running time (ART) of that show.
- Since an instance (episode) of a show may be broadcasted multiple times, we need to find the actual frequency of that show. For this, we count the number of unique instances of that show and use it as a measure for the total frequency (TF) of that show. An instance of the above process is shown in Table 1 and Table 2.

In Table 1, we can observe that user $U1$ watched the show $P5$ four times having unique event ids $E1, E2, E3$, and $E4$. Accordingly its average duration is 21.5 and frequency is 4 (see Table 2). Similarly we calculate the average duration and frequency for all other users who watched the show $P5$. In Table 2, we compute ART (22) and TF (4) for the show $P5$ by taking the maximum of average duration and frequency respectively.

Table 1: Sample User-Show Interaction

User Id	Program Id	Event Id	Duration of view
U1	P5	E1	22
U2	P5	E1	15
U3	P5	E1	23
U1	P5	E2	23
U2	P5	E2	5
U3	P5	E2	21
U1	P5	E3	22
U2	P5	E4	20
U1	P5	E4	19
U4	P5	E4	5

Table 2: Calculation of ART and TF

User Id	Program Id	Average Duration	Frequency
U1	P5	21.5	**4**
U2	P5	13.33	3
U3	P5	**22**	2
U4	P5	5	1

3.2 Mapping Implicit Feedback into Explicit Rating

To convert the available implicit feedback into explicit ratings we need to scale the feedback obtained in terms of duration of view and frequency of view. In this work, we define two ratios namely, Duration Ratio (DR) and Frequency Ratio (FR), which will help us to compare the individual implicit feedback with the overall available feedback.

Duration Ratio (DR): It is the ratio of the average duration of view of each unique user-show interaction to the ART of that show. The values will range from 0 to 1.

Frequency Ratio (FR): It is the ratio of the frequency of each unique user-show interaction to the TF of that show. The values will range from 0 to 1.

We calculate the frequency ratio and duration ratio corresponding to each unique user-show interaction. The above ratios will help us to estimate the explicit ratings from the available implicit feedback as follows. The first step is to 'bin' the range of values obtained as DR and FR above. The bins are usually specified as consecutive, non-overlapping intervals of a variable. The bins (intervals) must be adjacent but need not be of equal width. In our case, the number of bins will depend on the rating scale (1 - 5). We divide the entire range of values (DR and FR) into a series of intervals and then assign a bin to each of the intervals. We consider the bins for the FR to be of equal width while the bins for the DR to be of unequal width. The width of the bins are decided based on the findings derived from the available dataset. In this work, we limit the number of bins to 5 since we aim to derive ratings on a scale of 1 to 5 from the available implicit feedback. An example of this process is shown in Table 3 and Table 4.

Table 3: Duration Rating

Bin no.	Duration Ratio (DR)	Rating
1	> 0.0 & ≤ 0.05	1
2	> 0.05 & ≤ 0.25	2
3	> 0.25 & ≤ 0.50	3
4	> 0.5 & ≤ 0.75	4
5	> 0.75 & ≤ 1.0	5

Table 4: Frequency Rating

Bin no.	Frequency Ratio (FR)	Rating
1	> 0.0 & ≤ 0.2	1
2	> 0.2 & ≤ 0.4	2
3	> 0.4 & ≤ 0.6	3
4	> 0.6 & ≤ 0.8	4
5	> 0.8 & ≤ 1.0	5

In order to assign ratings to each of the unique user-show interaction, we take help of the binning process discussed above. The corresponding bin number will tell us the derived rating. As for example, in Table 3, if the DR of a user-show interaction falls in the range of $0.01 - 0.05$, then its bin no. is 1 and accordingly assigned a rating of 1. Similarly, a DR in the range of $0.76 - 1.0$ corresponds

Table 5: Final Rating Calculation

User Id	Program Id	DR	FR	$R_{Duration}$	$R_{Frequency}$	R_{Final}
U1	P5	0.97	1	5	5	5
U2	P5	0.6	0.75	4	4	4
U3	P5	1.0	0.5	5	3	3.87
U4	P5	0.22	0.25	2	2	2

to bin no. 5 and as a result a rating of 5. We derive similar ratings using FR as shown in Table 4. Thus, for each unique user-show pair we actually get two kinds of ratings, one rating corresponding to the frequency of the view (obtained from the bins for FR) and another one corresponding to the duration of the view (obtained from the bins for DR). Henceforth, we will refer to these ratings as frequency rating ($R_{frequency}$) and duration rating ($R_{duration}$) respectively. Although we have derived two ratings $R_{frequency}$ and $R_{duration}$ for each user-show pair, our aim is to find a single rating combining the two ratings, since none of the state-of-the-art recommendation algorithms allow multiple ratings for a unique user-item pair. We term this rating as R_{final} and is calculated using the following equation.

$$R_{final} = (R_{frequency})^n \times (R_{duration})^{(1-n)}, \ where \ 0 \leq n \leq 1 \tag{1}$$

Here n is the weightage of $R_{frequency}$ and $(1-n)$ is the weightage of $R_{duration}$. We determine the value of n experimentally to maximize the accuracy of the final ratings. We will discuss more about it later on in the Experimental section. An example of this rating calculation using $n = 0.5$ is shown in Table 5. Once we obtain the ratings for the different user-show interactions we can use any standard CF based algorithm to produce recommendations. In this work, we have tested our scheme using User-based CF, Item-based CF, SVD, NMF, and PMF methods of recommendation. We have depicted our framework pictorially in Figure 1.

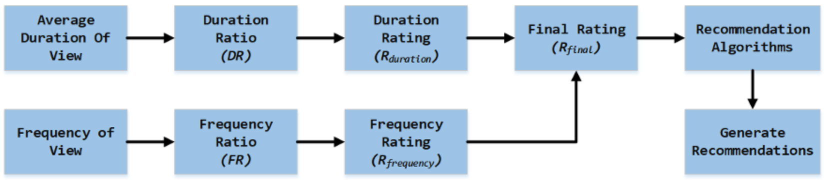


Fig. 1: Flowchart of our Framework

4 Experimental Settings

4.1 Data Description

We use a dataset containing viewing history of around 13,000 users over 217 channels[3]. The data has been recorded over a period of 12 weeks. There are

[3] http://recsys.deib.polimi.it/?page_id=76

14,000 programs/shows available to the users. The data set consists of information such as user id, program id, channel id, slot and duration of each view. We have considered only those user-program interactions, where the duration of view was greater than one minute. There are about 12.3 million such interactions in our dataset. We have divided the dataset into five disjoint sets. Each set has been used separately for testing and then the rest four for training, so that there were five different training/testing sets. We have repeated our experiment with each set and then considered the average of the results.

4.2 Evaluation Metric Discussion

The prediction accuracy of our algorithm is measured in terms of Root Mean Square Error (RMSE) [13]. The objective of any recommendation algorithm is to minimize the RMSE value. However, only RMSE cannot correctly evaluate a Top-k recommendation list. Therefore in this work, we use Precision, Recall and F1 measure metric [9] to evaluate the quality of the recommended list.

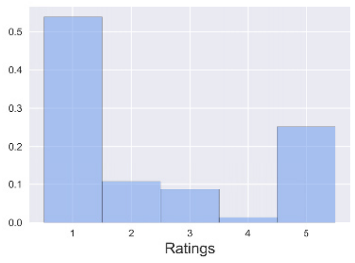
$$Precision = \frac{t_p}{t_p + f_p} \qquad Recall = \frac{t_p}{t_p + f_n} \qquad F1 = \frac{2 * Precision \ * Recall}{Precision + Recall}$$

True Positive (t_p) or a hit means a relevant product is recommended to a customer by the recommender system. On the contrary, False Positive (f_p) denotes the case when an irrelevant item is recommended, and when an item of customer's liking has not been recommended then we term the case as False Negative (f_n). F1 measure combines Precision and Recall with equal weightage making the comparison of algorithms across datasets easy.

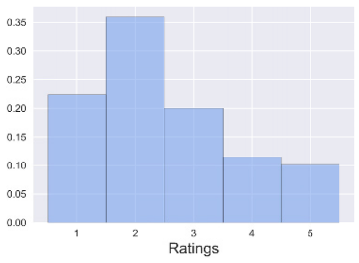
5 Results and Discussion

We report the frequency rating and duration rating distributions in Figure 2(a) and 2(b) respectively. From the distribution reported in Figure 2(a), we can infer that about 54% of the time the frequency rating $R_{frequency}$ is 1. This is due to the fact that a lot of users tend to surf through different channels looking to explore new content but they watch only a handful of the TV shows on a regular basis. On the other hand only around 25% of the time users watched the show on a regular basis, which is indicated by a $R_{frequency}$ value of 5. Similarly from Figure 2(b), we can observe that about 22.5% of the time the users watched the show for less than 5% of the actual running time of the show as indicated by a duration rating $R_{duration}$ of 1. This happens mostly when a user is surfing through the channels looking for new shows. We can further notice that around 10% of the time users watched the show for more than 75% of the actual running time of the show indicated by an $R_{duration}$ value of 5.

From the above observations, we can conclude that the situation for linear TV is quite different from the online streaming services and VOD services like Netflix, Amazon Prime, etc. Users do not have the freedom to watch any show at any given time. Therefore, a lot of users are not able to watch TV shows that frequently (they might not always be available when a particular show gets broadcasted). Moreover, users dont have the freedom to rewind or pause a show



(a) Frequency Rating



(b) Duration Rating

Fig. 2: Normalized Distribution of Ratings

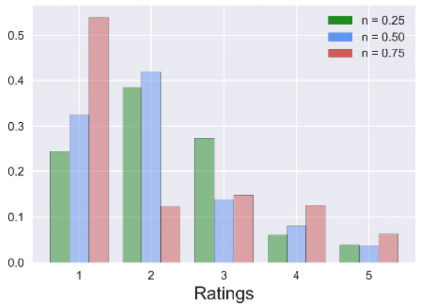


Fig. 3: Normalized Distribution of Final Ratings

whenever they want to. Thus, users may not be able to completely watch a show.

5.1 Finding Right Weightage

A common feature of linear TV viewership is that a lot of users do not watch TV frequently and also tend to watch it for shorter duration. Therefore, depending solely on frequency or duration to make recommendations will be unwise. A balanced approach needs to be taken where the right weightage is assigned to both frequency and duration to make the suitable recommendations. Therefore, in the calculation of R_{final}, we give weightage to both frequency and duration (see equation 1). We vary the value of n from 0 to 1 in order to find the optimum weightage that will maximize the accuracy of the final rating. An example of derived final rating using $n = 0.25, 0.5$, and 0.75 is shown in Figure 3.

5.2 Comparisons

In this work, we used two popular memory based CF algorithms - User-based and Item-based [13], and three matrix factorization techniques namely Singular Value Decomposition (SVD) [11], Non-negative Matrix Factorization (NMF) [4] and Probabilistic Matrix Factorization (PMF) [12]. These recommendation methods are combined with our framework to verify whether their performance has improved or not. We compute user-user and item-item similarities using cosine-based similarity measure. In SVD, the user-item matrix is decomposed into three matrices with n features: $R = U_n S_n V_n^T$. The prediction score for the i-th customer on the j-th product is given by $P_{i,j} = \bar{r_i} + U_n \sqrt{S_n}^T(i).\sqrt{S_n} V_n^T(j)$, where

Table 6: Recommendation Performance Comparisons in terms of RMSE, Precision, Recall and F1. The bold numbers indicate best results

Recommendation Method	n	$(1-n)$	$RMSE$	$Precision$@10	$Recall$@10	$F1$@10
User-Based	0	1	1.093	0.703	0.313	0.433
	0.25	0.75	0.836	0.603	0.324	0.421
	0.5	0.5	0.664	0.768	0.626	0.689
	0.75	**0.25**	**0.51**	**0.985**	**0.692**	**0.812**
	1	0	0.544	0.978	0.645	0.777
Item-Based	0	1	1.065	0.886	0.191	0.314
	0.25	0.75	0.866	0.952	0.087	0.16
	0.5	0.5	0.7	0.879	0.368	0.518
	0.75	**0.25**	**0.568**	**0.992**	**0.587**	**0.737**
	1	0	0.57	0.987	0.569	0.721
SVD	0	1	1.051	0.73	0.322	0.462
	0.25	0.75	0.794	0.65	0.327	0.449
	0.5	0.5	0.613	0.79	0.569	0.68
	0.75	**0.25**	**0.434**	**0.986**	**0.661**	**0.815**
	1	0	0.474	0.979	0.588	0.734
PMF	0	1	1.051	0.739	0.316	0.458
	0.25	0.75	0.797	0.66	0.316	0.442
	0.5	0.5	0.617	0.797	0.554	0.673
	0.75	**0.25**	**0.438**	**0.987**	**0.658**	**0.813**
	1	0	0.478	0.98	0.585	0.733
NMF	0	1	1.071	0.744	0.272	0.413
	0.25	0.75	0.825	0.656	0.258	0.384
	0.5	0.5	0.651	0.773	0.558	0.666
	0.75	**0.25**	**0.485**	**0.981**	**0.662**	**0.814**
	1	0	0.52	0.973	0.593	0.736

$\bar{r_i}$ is the i-th row average. For both SVD and PMF methods, we consider 40 features while NMF is implemented using 15 features.

We report and compare the recommendation performance using different recommendation methods in Table 6. Note that, we present Precision (P@10), Recall (R@10) and F1 ($F1$@10) score on position 10. The bold numbers indicate the best results for that particular recommendation method. In Table 6, n is the weightage of $R_{frequency}$ and $(1-n)$ is the weightage of $R_{duration}$. A study of Table 6 clearly reveals that when $n = 0.75$, we get the best results for all the recommendation methods irrespective of the different evaluation metrics. This indicates that frequency of view is a more significant factor than duration of view in order to generate effective recommendations. We can further notice that when only frequency is considered ($n = 1$) or when only duration is considered ($n = 0$), then the recommendation results are worse than the case when $R_{frequency}$ is given the weightage (n) of 0.75 and $R_{duration}$ is given the weightage $(1-n)$ of 0.25. This result is consistent across all the recommendation methods. Thus we can conclude that assigning weightage to both frequency and duration helped us in achieving more accurate recommendations.

6 Conclusion and Future Work

In this paper, we have presented an approach to tackle the problem of recommending shows on linear TV by converting the implicit feedback from the users (collected in terms of frequency and duration of user-show interactions) to explicit ratings. For each unique user-show interaction we derive two ratings, frequency rating and duration rating and then the two ratings are combined together to obtain a final rating. Experimentally we have verified that recommendations generated are more accurate when both frequency and duration ratings are given some weightage to compute final rating than when only one of them are considered separately. The focus of our future work is to make the final rating calculation more accurate by assigning optimum weightage to frequency and duration ratings using some machine learning technique.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Ardissono, L., Gena, C., Torasso, P., Bellifemine, F., Difino, A., Negro, B.: User modeling and recommendation techniques for personalized electronic program guides. In: Personalized Digital Television. Human-Computer Interaction Series, vol. 6, pp. 3–26 (2004)
3. Breese, J., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the fourteenth Conference on Uncertainty in Artificial Intelligence (UAI'98). pp. 43–52 (1998)
4. Cai, D., He, X., Han, J., Huang, T.S.: Graph regularized non-negative matrix factorization for data representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(8), 1548–1560 (2011)
5. Chang, N., Irvan, M., Terano, T.: A tv program recommender framework. Procedia Computer Science 22, 561–570 (2013)
6. Cotter, P., Smyth, B.: Ptv: Intelligent personalised tv guides. In: Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence. pp. 957–964 (2000)
7. Cremonesi, P., Modica, P., Pagano, R., Rabosio, E., Tanca, L.: Personalized and context-aware tv program recommendations based on implicit feedback. In: Stuckenschmidt H., Jannach D. (eds) E-Commerce and Web Technologies. LNBIP. vol. 239 (2015)
8. Das, D., Horst, H.: Recommender systems for tv. In: Workshop on Recommender Systems, Proceedings of 15th AAAI Conference, pp. 35–36 (1998)
9. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems 22(1), 5–53 (2004)
10. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM '08). pp. 263–271 (2008)
11. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Computer Society 42(8), 30–37 (2009)
12. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. Advances in Neural Information Processing Systems 20, 1257–1264 (2008)
13. Su, X., Khoshgoftaar, T.: A survey of collaborative filtering techniques. Advances in Artificial Intelligence 2009 (2009)
14. Turrin, R., Condorelli, A., Cremonesi, P., Pagano, R.: Time-based tv programs prediction. In: RecSysTV Workshop at ACM RecSys 2014, pp. 957–964 (2014)