Personalized recommendation of linear content on interactive TV platforms: beating the cold start and noisy implicit user feedback

Dávid Zibriczky, Balázs Hidasi, Zoltán Petres, and Domonkos Tikk

Gravity R&D Ltd.
Expo tér 5–7, Budapest, Hungary
{david.zibriczky,balazs.hidasi,zoltan.petres,domonkos.tikk}@gravityrd.com
http://www.gravityrd.com

Abstract. Recommender systems in TV applications mostly focus on the recommendation of video-on-demand (VOD) content, although the major part of users' content consumption is realized on linear channel programs (live or recorded), termed EPG programs. The accurate collaborative filtering algorithms suitable for VOD recommendation cannot be directly carried over for EPG program recommendation. First, EPG program recommendation features the cold start problem; a significant part of EPG programs are new in the system. Second, and more importantly, without explicit user feedbacks (ratings) the algorithms have to model user preference based on the noisy and less directly interpretable implicit user feedbacks. In this paper, we present several approaches that overcome these difficulties, by applying pre-filtering on noisy low-level data and taking into account channel preferences of users and program metadata if available to cope with the cold start. Using time-dependent tensor factorization approaches, the temporal preferences of users are also reflected in recommendation, that also hints on the person watching the TV. Experiments were performed on a dataset of SaskTel, a Canadian IPTV service provider using Microsoft Mediaroom middleware.

Keywords: implicit feedback collaborative filtering, matrix factorization, tensor factorization, alternating least squares, time-dependent modeling, cold start problem

1 Introduction

Content consumption trends changed significantly in the last 10 years with the rise of digital evolution and Internet. Netflix and YouTube were the two main innovators in the last decade. Netflix streaming service allows reaching thousands of blockbuster and premium Hollywood contents, a selection that never before was accessible from a digital device. YouTube also became the de facto main content hub where all kinds of content can be found. These and many new related Internet services have a very strong effect on the very big screen in our living room—people still spend much more time at the TV set than on streaming video internet sites—and the services it offers.

The race to control the very big screen forced cable TV (television) and telecommunication companies to expand their media offerings. Therefore, they added more live channels, typically thematic programs that offer programs for a particular interest group, enabled on-demand content catalogues, and added premium functions like PVR (personal video recordings) to allow time-shift content watching. The current advanced cable TV and IPTV (Internet Protocol television) services offer high quality content distribution of over 500 live channels and radio programs, and provides video libraries of blockbuster contents, catch-up clips, and more with ten thousands titles.

However, as more and more channels and on-demand titles are added, the navigation and search become more difficult and challenging, just like for its Internet counterparts Netflix and YouTube. In order to manage this problem, EPG (electronic program guide) was first introduced. It provided a quick overview of the programs the live channels aired. Typically, it covered 8–10 channels for 3–4 hours timespan. This solution was comfortable up to 30–40 channels as scrolling became uncomfortable. Then, thematic filtering and coloring (assign different colors to different genre or topic of content) is applied that allowed to narrow down the interested channel from the available 100+ to the range of 20–30, when again the scrolling was option for navigation and selection. Though, when there are over 500 available live channels a completely different approach is needed, likely a recommendation based solution using advanced data mining and user profiling techniques.

The importance of this is vital for telecommunication and cable companies. If the subscribers do not see the value of the service offering, they just leave and use other services. The generated income from subscription fees is still much larger than from the on-demand services, but adding more channels is not an attractive solution any more either, because of the troublesome navigation issues.

A Canadian IPTV operator was looking for a solution that addressed the following problems:

1. user should find the interested on-going live program with minimal navigation
2. user should be able to use the thematic filters as well
3. the solution should allow the personal promotion of those channels that the user is not subscribed to
4. the solution should drive on-demand sales
5. the solution should work well for time-shift users

Gravity offered a unified recommendation application that can be easily deployed on the operator's interactive Microsoft Mediaroom IPTV platform. The application merges and combines all content sources (live TV and on-demand) and presents the user in a unified way. The unification enables to handle all live TV programs similarly to on-demand contents and to follow the same recommendation pattern that the user is familiar with in the Internet. Based on the user's personal taste and preferences, the application recommends relevant live TV programs. There is no need to browse and channel zap through the hundreds of channels and stick to something okay, but the user can easily select from the relevant and interesting live TV recommendations. The unification allows to up-sell non-subscribed channels by presented relevant teaser programs and to promote content from on-demand catalogues.

This concept fulfills the operator's requests from the user experience side, but opens many challenging questions on the recommendation algorithm side. Live TV and on-demand content and the way users consume it is very different, especially in the shared, multi-user, and very laid-back environment like the very big screen living room TV. This paper is focusing on and provides solution to the following recommendation problems:

- **Noisy input:** as TV watching is a very, very laid-back environment, users do not provide explicit feedback. The taste is expressed implicitly by channel zapping, recording, recording playback events. One key challenge is how to pre-process and identify the relevant user actions that can be efficiently is used for user profiling.
- **Live TV programs are always new:** by nature, live TV programs are always new. Recommender algorithms that require user feedback for preference modeling cannot be used, since such data is not available before the program start, and when it becomes available, it is too late, the program is over. The metadata associated with live programs is also much less detailed and relevant to on-demand programs. About 60% of live TV programs belong to news, sports, reality, talk shows, and many of them are aired in live. The program descriptions are either the same or very similar that makes the recommendation problem very challenging.
- **Time-based recommendation:** consumption patterns changes on daily and weekly bases. Users are not interesting in the same type of content in the morning and in the evenings, and it is also true for weekends.
- **Shared, multi-user device:** TV is a shared, multi-user device. The algorithms should be aware to this problem and if there is no explicit indication, who is in front of the device, it should also predict.

In this paper we focus on the solution above listed tasks. In particular, we will show how to extract information useful for preference modeling from the vast IPTV user data logs, propose several algorithms to handle the cold start, and present algorithms that account for time-dependent consumption patterns, which can indirectly use as a hint to predict the person watching the screen.

2 Methods

Recommender algorithms are classified into content based filtering (CBF) and the collaborative filtering (CF). CBF algorithms use user metadata (e.g. demographic data) and item metadata (e.g. author, genre, etc.) and predicts items with similar metadata the user liked in the past. In contrast, CF methods do not use metadata, but only data of user–item interactions, thus they are domain independent, and usually offer better quality than CBF methods. Relying only on historical interaction data, they are particularly affected by the cold-start problem.

The extent of the cold-start issue is domain specific. For domains with dynamically changing item catalog, such as online news and TV programs, the percentage of items already interacted with can be as low as 20% [3]. In such cases, naive implementations of CF algorithms fail to produce accurate recommendation, and therefore some either CBF or hybrid (combined CF and CBF) approaches are favored.

Here, we first analyze the EPG program data that largely differs in nature from VOD consumption data, since (i) the number of user feedbacks is by several order of magnitude larger than for VOD data (per user few thousand vs a couple [4]), (ii) but much noisier, (iii) the number of recommendable items is smaller at a given moment (only now or soon aired programs) (iv) and the lifetime of individual items is shorter. Therefore, we need to preprocess user log data to extract information useful for user preference modeling. Then we briefly describe the CF and CBF approaches we used in our evaluation, and we present a latent factor based hybrid approach that integrates item metadata into the model to cope with the cold start problem.

Next, we propose alternative algorithms to overcome the cold-start problem. Modeling channel preferences of user is a straightforward way to approach this problem. We present two tensor factorization based channel recommenders that take into account the time of watching in the recommendation and compare them to the baseline favorite channel method.

2.1 Preprocessing

The IPTV application offers several control options for the customers such as changing channels, setting the box for recording or playbacking TV programs, rewinding, fast-forwarding, pausing, etc. All of these options can be interpreted as feedbacks from the user about the current or next TV program (e.g. shutting down the applications during a TV program can be considered as negative feedback). The amount of information in each feedback is different. While playbacking a previously recorded TV program is a strong positive feedback, pausing an aired program tells much less about the preference of user. One can assign a confidence score to each feedback based on their interpreted relevance in describing user preference. While feedback types with high confidence score are valuable input for the recommender system, the ones with low score may reduce its efficiency due to the relatively high noise. Therefore, one may opt for filtering out feedbacks with low confidence (like pausing, changing channels rapidly, watching programs for short time) in order to reduce the noise of input. Such a pre-filtering yields smaller training data set that enables to use more complex algorithms if available the training time and computational resources are limited in an application; being typical in an IPTV setting.

The usefulness of feedback types are deemed based on their strength in contributing to the modeling of the user–item relationship. Therefore, event types with presumably small relevance in characterizing the user–item relationship can be removed. Events of the same types may have different different descriptive power as well. For instance in the case of most frequent EPG watch event type, very short events should not be interpreted as positive user feedback. Also, when the user gets inactive—e.g. falls asleep or does not simply switch off the TV set when leaving—and lets a channel on for a longer period is not informative either about the user tastes, and hence needs to be further filtered out.

2.2 Basic algorithms

As for baseline, we apply standard CBF and CF algorithms for the EPG program recommendation problem. The CBF method represents items by the vector of item attributes. We considered all item attributes and selected those ones for item representation that yielded the best performance: title, genre, channel, category, parental rating. In contrast with [4], the factorization of the item matrix (applying LSA) did not improve the model either, so the raw representation was kept. Some attributes, for instance program description, were sparsely available, therefore could not contribute to build better model. User feature vectors are calculated as the average of items in the user history, and item prediction are computed as the cosine similarity of user and item vectors (CosineSim).

We used IALS1, a fast variant [10] of the alternating least squares for implicit feedback (IALS, [6]) as the baseline latent factor CF method. The prediction formula of IALS1 for user u and item i is $\hat{r}_{ui} = p_u^T q_i$. The parameters of the model $p_1, \ldots, p_U \in \mathbb{R}^F$ and $q_1, \ldots, q_I \in \mathbb{R}^F$ are called user and item feature vectors; F denotes the number of features, U and I are the number of users and items, respectively. Further, P and Q denotes the matrices that contain user and item feature vectors as rows. IALS1 calculates the user and item features by iteratively minimizing the P with a fixed Q (Q with a fixed P) the sum of weighted squared error between the predicted ($\hat{r}_{ui}$) and true ratings ($\hat{r}_{ui}$), that its goal is to approximate the ratings to keep the approximation error low.

2.3 Hybrid filtering

Let we consider the original (implicit) ratings matrix $R \in \mathbb{R}^{U \times I}$ and a matrix $S \in \mathbb{R}^{M \times I}$ that contains the item metadata, that is, $s_{m,i} = 1$ if metadata m characterizes item i, 0 otherwise. Let us create a new matrix $R^\star \in \mathbb{R}^{(U+M) \times I}$ by stacking matrices R and S along the dimension I. Now an element of an $R^\star$ row is not zero, if a user has an event on i or metadata m is assigned to i. We can consider metadata as pseudo-users and the metadata related events as pseudo-events, and apply IALS1 to factorize matrix $R^\star$. In practice, there is significant difference between the number of events and the number of metadata values for an item. For keeping the balance in factorization we can assign different weight to user and metadata based events (real and pseudo-events) to reflect the different importance of events. This hybrid method guarantees that even when only pseudo-user events are available meaningful recommendations can be generated, and as more real events are present the user-event based part becomes dominant in the model, providing a smooth and principled transformation from a CBF to CF method.

2.4 Channel recommendation

One way to overcome the aforementioned item cold start problem is the aggregation of items. When items are merged along a fixed range or categorical attribute (or attribute set), new items can be immediately characterized by the training event of the relevant item group. Straightforward candidate attributes for item merging are item category and/or genre. However, using these attributes may have the adverse effect, since having only typically a few tens of categories, the resulting user preferences model will be too coarse or differences between user models may be blurred.

An alternative solution for aggregation is using the channel info. Hence, we basically predict the users' preferences over the TV channels. The preferences of a user over channels are not constant in time therefore the above blurring effect may re-occur here as well. However, users tend to watch the same channels at similar time periods: e.g. one watches her favorite series on the afternoon on channel A then switches to channel B to watch news, etc. This phenomenon is termed the *seasonality* of the watching behavior. Note that genre based aggregation would benefit less from seasonality because genres are less specific than channels.

Channel based aggregation has two steps. First, instead of programs, channels are considered as items. Then, at merging events, two events of a user is merged if one event immediately precedes the other and the corresponding channel is the same.

Seasonality consists of a season that defines its periodicity, and time bands within the season [5]. The season is a time interval (e.g. one day, one week) wherein periodicity in user behavior can be observed. We assume that users behave similarly at the same time offset in different seasons. Within a season, time bands specify non-overlapping time offset intervals, which can be of equal (e.g. hours of the day) or non-equal length (e.g. morning, afternoon, night). The time band can be assigned to each event based on the time stamp.

Our channel recommendation uses three seasonality-aware algorithms. The baseline algorithm (FavoriteChannel) simply ranks the channels within each time band using the total count of events therein. Depending on the available information it can be personalized (ranked based on events of an individual user) or non-personalized (ranked based on all user events). This method serves as a baseline for channel recommendation. Next we present two algorithms to improve the baseline method.

Time-based recommendation with seasonality-aware iTALS The user preferences can be organized into a three dimensional tensor using users, items, and time (seasonality) as dimensions. iTALS [5] is a powerful context-aware algorithm for the implicit feedback based recommendation task. It performs tensor factorization on user–item–context tensor; here context is the time. The tensor is decomposed into three (feature) matrices (one for each dimension). The prediction for preference of the u^{th} user on the i^{th} item in the t^{th} time band is the sum of elements in the Hadamard product of the u^{th}, i^{th} and t^{th} columns of the feature matrices:

$$\hat{r}_{u,i,t} = 1^T \left(P_{\bullet,u} \circ Q_{\bullet,i} \circ T_{\bullet,t}\right) \tag{1}$$

Compared to the common matrix factorization model, one can view this variant as the time-based reweighting of the user–item relations in the feature space.

Considering the duration of the watching events Unlike the point-like purchase events in the simpler VOD recommendation scenario, EPG watch events have duration. We already use the duration to filter insignificant events (see Section 2.1). So far the time band corresponding to an event was identified by its starting time. However, the tensor can be modified so that watching event counts as positive feedback in every time band that duration of the event overlaps with. Furthermore, one can assign weights to instead simple counts based on the overlap between the time band and duration of the event. This approach results a denser preference matrix and therefore the learning becomes slower, but we expect that the user preference model is thus improved.

3 Experiments and results

The raw dataset from SaskTel consists of $\sim$ 305M user events[1] for a 5 week period with the following event types: PVR start, playback, delete, schedule, cancel, abort recording; stop, rewind, fast forward; channel tune, watch, browse. We deemed EPG watch, DVR start and playback as the most relevant event types.[2] After discarding all but EPG watch, PVR playback and PVR start recording events, the number of events is reduced to 82M. For further denoising, we removed events of that account for zapping and leave-on. For removing zapping, we assume that user preference are best described by long enough events that immediately start after a channel tune (channel change) event, so we kept events longer than 5 minutes. Furthermore, we found that keeping only the first three events after a channel tune can efficiently remove events generated by leave-on. After applying these filters, the remaining dataset had $\sim$ 23M events, which also allows to potentially use more complex algorithms (e.g. matrix and tensor factorization with more factors). After preprocessing, the dataset was split into a train and test sets; the latter contained 686k events of the last day.[3]

[1] The data used in this experience were anonymous.

[2] DVR events is used by only a smaller group of users, however, these are highly valuable user feedback. In particular, DVR playback confirms DVR start and has the strongest relevance in describing user preferences.

[3] For VOD recommendation, the testing period is usually longer (1 week, [12, 6]). Note that for EPG program recommendation, it is better to test on a shorter period, since there are several orders of magnitude more events, and the set recommendable items change faster.

To characterize the cold-start phenomenon in the dataset, we measured new item and new event ratio in the test set. We used various heuristics to group items together along various item attributes (see Figure 1). Genre based merging obviously generates training data for all items. Interestingly, about 71% of items (62% of events) in the test set have training data, which is explained by the large amount of re-airing (e.g. same episodes of series), and that ProgramId maps together regular programs (news/weather). Grouping the episodes of series, already almost 90% of the TV programs are not new, and performing both groupings, only about 5.5% of the items are new. This means that an unexpected repetitiveness can be found in the aired programs; nevertheless finding the right users for the new items is still an important recommendation task, since these are typically feature movies or new series. For CF and hybrid approaches we use ProgramId and SeriesId based grouping of events in the experiments.

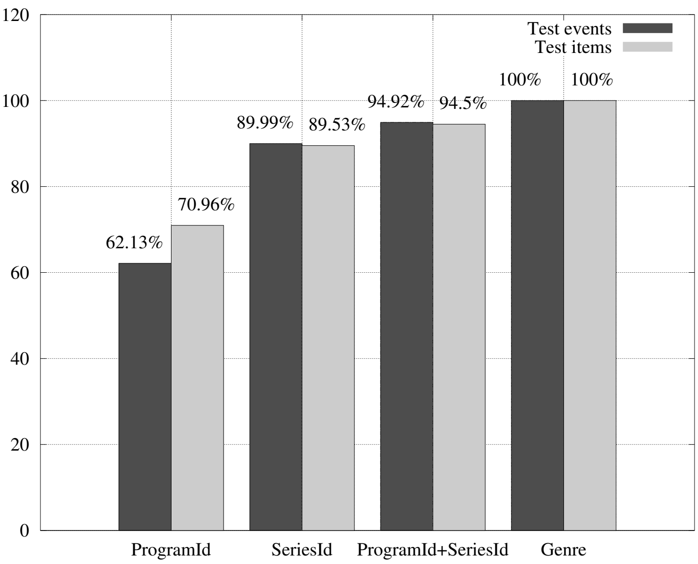

Fig. 1. The ratio of supported events/items after grouping items by metadata keys

We used two evaluation metrics recall@N and coverage@N. Predictions of algorithms are ranked and the top-N items are considered. If a test item appears within the top-N, we have a hit. Recall@N measures ratio of hits on the test set. Coverage@N calculates the ratio of the unique elements in top-N lists and the total number of items. High coverage infers novelty and serendipity effect that contributes to the users' perceived quality of recommendations [2]. Since the GUI of the IPTV application we tested displays 9 programs at once (recommended program are assigned numbers 1–9 on the remote), we set $N = 9$ in our experiments.

We also experimented with two baseline algorithms. TopPopSeries counts events for each series in the training set and recommends the most popular ones aired at the given moment. The personalized AlreadySeen algorithm recommends only programs or episodes of series the user already watched at least once. Blend is a linear combination of CosineSim, AlreadySeen and Hybrid IALS1. In Table 1 summarizes the results for different algorithms. We ran channel recommenders with and without duration based weighting, but show only the better result.

Algorithm	Type	Recall@9	Coverage@9
CH global (w/o duration)	Pop	0.1151	0.1916
TopPopSeries	Pop	0.1722	0.0497
CH user (w duration)	Pop	0.2773	0.9996
CosineSim	CBF	0.4285	0.9596
CH ITALS (w duration)	CF	0.3140	0.9146
CH ITALS (w/o duration)	CF	0.3360	0.8170
IALS1	CF	0.3612	0.7867
AlreadySeen	Pop	0.3911	0.8403
Hybrid IALS1	HF	0.4054	0.7911
Blend (linear combination)		0.4534	0.8941

Table 1. Evaluation of various algorithms, type indicates the methodology used (Pop – popularity based, HF – hybrid filtering)

We also investigated how algorithms perform on certain subsets of items. For that we partitioned the item set as (i) old vs. new items (ii) popular (20%) vs. long-tail (80%) and (iii) series vs. non-series. Table 2 includes the result for the all the three partitions.

4 Discussion

Considering only the global accuracy of the algorithms, AlreadySeen performs surprisingly well, but this can be attributed to the bias towards re-aired programs in the dataset. On new, long tail and non-series items its performance is particularly poor, with a novelty/serendipity effect of practically zero. On the other hand, content metadata based and channel recommenders perform equally well on new, long tail and non-series items, among them the most complex ITALS outperforms the popularity based variants significantly. The duration weighting improves only the performance of the user variant of the channel-based recommender. In accordance with other studies in the literature, the simple CosineSim produced the best recall on new items. On the other end, Hybrid IALS1 optimized for the global recall was not able to recommend new items, suggesting that for each item subset different algorithm parametrization may be needed. The overall best recall was achieved by the linear combination of the best approaches.

Algorithm	old item	new item	popular	long-tail	series	non-series
Number of events	651300	34887	579047	107140	636409	49778
Number of items	2187	216	481	1922	2103	300
CH global (w/o duration)	0.1172	0.0737	0.1325	0.0206	0.1227	0.0161
CH global (w duration)	0.0861	0.0608	0.0926	0.0430	0.0877	0.0472
TopPopSeries	0.1815	0.0000	0.2037	0.0022	0.1857	0.0000
CH user (w/o duration)	0.2674	0.1524	0.2767	0.1802	0.2758	0.0800
CH user (w duration)	0.2837	0.1574	0.2916	0.1997	0.2921	0.0867
CosineSim	0.4369	0.2670	0.4419	0.3560	0.4448	0.2220
CH ITALS (w/o duration)	0.3441	0.1829	0.3620	0.1949	0.3555	0.0857
CH ITALS (w duration)	0.3215	0.1749	0.3367	0.1914	0.3326	0.0770
IALS1	0.3806	0.0000	0.4411	0.1254	0.3886	0.0117
AlreadySeen	0.3970	0.0000	0.4239	0.1221	0.4054	0.0104
Hybrid IALS1	0.4266	0.0317	0.4495	0.1643	0.4358	0.0105
Blend	0.4656	0.2154	0.4725	0.3485	0.4742	0.1871

Table 2. Recall of the algorithms on various item subsets of the test set

The coverage of recommendations is high or very high (except CH global and TopPopSeries), which can be attributed to the relatively low number of recommendable programs at a time (few hundreds). In this regards, CH user has the best performance recommending basically all available programs. The second best coverage of CosineSim can be also attributed to the use of ChannelId among the metadata. In general, algorithms considering user based popularity provide the most diverse recommendation which suggests that user preferences largely differ, justifying the application of personalized recommenders to overall improve user experience and satisfaction.

Using channel recommendation combined with seasonality can efficiently solve the item cold start problem while maintaining the accuracy of the recommendation, if the periodicity assumption on aired programs holds. This is usually the case of morning and afternoon programs, series, and programs of thematic channels with recurring program blocks, but may fall short for recommending evening feature movies since they are of different type and channel independent.

5 Related work

In the field IPTV related recommendation, studies are mainly focused on VOD program recommendation. Cremonesi et al evaluated CBF, CF (both neighbor and latent factor based approaches) and various hybrid methods [4, 3]. Regarding the cold-start problem, they pointed out that CBF approaches outperformed CF in recall only when the overwhelming part (80–90%) of the item set is new [4]. Furthermore, they observed the time-evolution of recommender system with the increase of user feedbacks and found that in early phase, neighbor-based methods are more accurate than latent factor models, but with sufficiently many factor and enough historical data, latent factor models may outperform neighbor models if the dataset does not exhibit long-tail behavior. In a field study on recommending VOD titles that involved 210 subjects [2], they also showed that users' perceived quality of recommendation does not correlate well with a single accuracy metric, but rather a compound of accuracy and novelty/serendipity, furthermore the popularity based non-personalized recommendation performs surprisingly well for the first few recommendations.

Park et al [8] compared CF and CBF approaches on an IPTV (EPG program) dataset where regular (news, weather, traffic) and low support programs were removed. They measured no significant difference between the two approaches, but in a interleaved combination, a slight improvement (4.3%) in recall was observed.

For taming the beast of the cold start problem while still providing good accuracy for not new items, several authors proposed different hybrid CF and CBF methods; see survey [1]. On the IPTV domain, [4] tested several simple hybrid approaches which did not have latent factor component. They reported that low recall for interleaved and SimComb [7] (item-to-item similarity values are computed as the linear combination of content-based and collaborative similarities), while their Filtered Featured Augmentation (FFA), and Similarity Injected knn are on par with CF method when the ratio of new items are small. When there are many new items (80–90%) the interleaved and knn based ones are on par with the CBF approach, while SimComb and FFA fall shorter.

The most similar to our hybrid model are combination of latent factor models and content based feature proposed for explicit feedback data. In this regard, Zhang et al [13] filled movie feature matrix with metadata, user features were estimated by a least square solver, and alternating iteration is performed. They reported that even on explicit data the method is slow. Singh et al. [11] proposed Collective Matrix Factorization (CMF) that factorize the matrices describing user-movie relation and the movie-metadata simultaneously, where the movie feature vectors are shared. The model is able to find efficiently similarity in movie metadata. A similar model was proposed in [9] where a linear transformation was used to transform item descriptions to be as close as possible to the feature vectors generated by a matrix factorization.

6 Conclusion

EPG program recommendation is important for service providers since most of the consumption is realized on such content. Accurate and trustworthy EPG program recommendations provide a straightforward up-selling opportunity, since it can motivate users to upgrade their monthly plan. Despite its practical importance, this topic was almost neglected in scientific and field studies. Recommending EPG content compared to VOD differs largely, due to the difference in the characteristics of EPG and VOD usage. In this paper, we present a case study performed on data of a Canadian IPTV service provider. We demonstrate that a massive denoising is required first to extract useful information for user preference modeling from user log data. For the cold-start problem, we propose various approaches using only content metadata, time-dependent channel watching behavior data and a hybrid (CBF and CF) approach, and showed that the former two can be efficient applied to recommend new, less popular or non-regular items. Best recall and coverage values were attained by a simple content metadata based model. Results suggest that item partition based combination of algorithms can be used to further boost accuracy, but its practical applicability may be limited by the required response time of the recommendations.

References

1. Burke, R.: Hybrid web recommender systems. In: The adaptive web. pp. 377–408. Springer (2007)
2. Cremonesi, P., Garzotto, F., Negro, S., Papadopoulos, A., Turrin, R.: Looking for "good" recommendations: A comparative evaluation of recommender systems. In: Human-Computer Interaction INTERACT 2011, LNCS, vol. 6948, pp. 152–168. Springer, Berlin/Heidelberg (2011)
3. Cremonesi, P., Turrin, R.: Time-evolution of IPTV recommender systems. In: EuroITV '10: Proc. of the 8th Int. Interactive Conf. on Interactive TV&Video. pp. 105–114 (2010)
4. Cremonesi, P., Turrin, R., Airoldi, F.: Hybrid algorithms for recommending new items. In: Proc. of the 2nd Int. Workshop on Information Heterogeneity and Fusion in Recommender Systems. pp. 33–40. HetRec '11 (2011)
5. Hidasi, B., Tikk, D.: Fast ALS-based tensor factorization for context-aware recommendation from implicit feedback. arXiv preprints arXiv:1204.1259, 1–12 (2012)
6. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: ICDM-08: IEEE Int. Conf. on Data Mining. pp. 263–272. Pisa, Italy (2008)
7. Mobasher, B., Jin, X., Zhou, Y.: Semantically enhanced collaborative filtering on the web. In: EWMF'03: the 1st European Web Mining Forum. pp. 57–762 (2003)
8. Park, K., Choi, J., Lee, D.: A single-scaled hybrid filtering method for IPTV program recommendation. Int. J. of Circuits, Systems and Signal Processing 4(4), 161–168 (2010)
9. Pilászy, I., Tikk, D.: Recommending new movies: Even a few ratings are more valuable than metadata. In: RecSys'09: Proc. of the 2009 ACM Conf. on Recommender Systems. pp. 93–100 (October 23–25, 2009)
10. Pilászy, I., Zibriczky, D., Tikk, D.: Fast ALS-based matrix factorization for explicit and implicit feedback datasets. In: RecSys'10: Proc. of the 4th ACM Conf. on Recommender Systems. pp. 71–78. Barcelona, Spain (Sept 26–30, 2010)
11. Singh, A.P., Gordon, G.J.: Relational learning via collective matrix factorization. In: KDD-08: Proc. of the 14^{th} ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (2008)
12. Xin, Y., Steck, H.: Multi-value probabilistic matrix factorization for IP-TV recommendations. In: RecSys '11: Proce. of the 5th ACM Conf. on Recommender Systems. pp. 221–228 (2011)
13. Zhang, Y., Koren, J.: Efficient Bayesian hierarchical user modeling for recommendation system. In: SIGIR-07: Proc. of the 30^{th} Annual Int. ACM SIGIR Conference on R&D in Information Retrieval. pp. 47–54 (2007)