

Article

A Television Recommender System Learning a User's Time-Aware Watching Patterns Using Quadratic Programming

Noo-ri Kim ¹, Sungtak Oh ² and Jee-Hyong Lee ^{1,*}¹ Department of Electrical and Computer Engineering, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon-si, Gyeonggi-do 16419, Korea; pd99j@skku.edu² Samsung Electronics, 1-1 Samsungjeonja-ro, Hwaseong-si, Gyeonggi-do 18448, Korea; sungtak.oh@samsung.com

* Correspondence: john@skku.edu; Tel.: +82-31-290-7154

Received: 10 July 2018; Accepted: 5 August 2018; Published: 8 August 2018



Abstract: In this paper, a novel television (TV) program recommendation method is proposed by merging multiple preferences. We use channels and genres of programs, which is available information in standalone TVs, as features for the recommendation. The proposed method performs multi-time contextual profiling and constructs multiple-time contextual preference matrices of channels and genres. Since multiple preference models are constructed with different time contexts, there can be conflicts among them. In order to effectively merge the preferences with the minimum number of conflicts, we develop a quadratic programming model. The optimization problem is formulated with a minimum number of constraints so that the optimization process is scalable and fast even in a standalone TV with low computational power. Experiments with a real-world dataset prove that the proposed method is more efficient and accurate than other TV recommendation methods. Our method improves recommendation performance by 5–50% compared to the baselines.

Keywords: context awareness; recommender systems; time-aware recommendation; time context; TV program recommendation; quadratic programming

1. Introduction

The recommender system has been recognized as an important issue in the data mining field and actively researched over decades [1–29]. It is a kind of information filtering system that predicts users' ratings or preferences for items and provides users with items that they may be interested in. It is applied in a variety of domains such as movies, music, books, news, and any other contents in general [1,4,18,20,22,25–29]. As a result, many recommendation algorithms have been developed for these domains in recent years. Since each domain has different characteristics, a clear grasp of the properties of a target domain is essential to develop a good recommendation algorithm.

Television (TV) is the most popular viewing device for multimedia content distribution, so it is one of the most important domains for recommender systems. The number of channels is increasing, such as drama, entertainment, sports, and news, since the broadcast has changed from analog to digital. Usually, more than several tens of channels are provided by broadcast service providers. Some broadcast service providers provide more than one hundred channels. Users have a lot of choices, but they may spend a lot of time and feel confused when they try to search their favorite TV programs.

Because of this issue, some broadcast service providers provide TV program information using an electronic program guide (EPG) [2,3]. An EPG provides viewers with continuously updating menus displaying schedule information for current and upcoming programs. The schedule information

includes the details of TV programs such as title, genre, synopsis, and actors. An EPG provides useful information to help channel selection, but it does not provide personalized recommendations.

A lot of personalized TV content recommendation approaches are based on collaborative filtering (CF) or content-based filtering (CBF), which is also popular regarding movie recommendations [3–12]. However, recommending programs in live TVs is more difficult and challenging compared to movies because of the following issues.

First, the movie domain is unconstrained with time context, but the TV domain is time-constrained. Users can access any movies whenever they want to, but programs on live TVs are broadcast on a scheduled day and time. Since users' favorite TV programs are broadcast periodically, users tend to watch TV programs in periodic patterns. So, TV recommender systems need to consider time context to analyze users' watching patterns [14]. Moreover, a user's watching pattern can have multiple periodicities because TV programs have different broadcasting schedules. For example, if a user tends to watch a news program at 9 p.m. every day, it is regarded as a periodic pattern in hour-context. If a user is likely to watch sports programs every Saturday, the user's watching pattern may have a periodicity in day-context. Various periodicities are mixed up together in the watching patterns of users. Therefore, TV recommender systems need to model multiple time-contextual preferences of each user.

Second, external user watch logs and TV program metadata are hard to be collected in standalone set-top box environments, in contrast to movie domains. Most TVs are not connected to the internet or cloud servers that store various external information. In general, the available information of TV programs in standalone environments are watching time, watching channels, and brief information on the programs, such as title and genre, from the EPG. Existing recommendation techniques such as CF and CBF, which require external users' feedback or metadata on programs, are not suitable to the TV domain. Thus, further studies for TV recommender system specialized in standalone environments are needed.

In this paper, the issues in the TV domain mentioned above are considered. The proposed method recommends TV programs by merging different time contexts without additional external information with minimum computation costs, which is a great advantage for standalone TVs compared to the conventional TV recommender systems.

We choose channels and genres of programs as features for the recommendation. In general, channels and genres are available information in standalone TVs. The channels chosen by users are known in TVs and the genres of chosen programs are usually available in the EPG. Also, they have strong correlations with user preferences on programs. For example, a user who frequently watches news tends to watch news channels or programs in the news genre. So, user preferences on TV programs can be effectively modeled based on channels and genres of watched programs.

In order to effectively model complicated periodic patterns, the proposed method performs multi-time contextual profiling and constructs several time contextual preference matrices of channels and genres. The proposed method combines those time contextual preference matrices to infer user preferences on channels and genres. Since multiple preference models are constructed with different time contexts, there can be conflicts among them. To merge the preferences with the minimum number of conflicts, we develop a quadratic programming (QP) model. The optimization problem is formulated with a minimum number of constraints so that the optimization process is scalable and fast even in a standalone TV with low computing power. The channel preferences and the genre preferences are merged to calculate recommendation scores of live programs. The proposed method is validated with a real-world dataset.

The remainder of this paper is organized as follows. Section 2 surveys the related works. Section 3 presents the proposed TV recommender system and explains each part of the system. Section 4 reports the experimental results and offers some discussions. The conclusions are presented in Section 5.

2. Related Work

Many researchers have studied TV contents recommendations. The first type of TV recommender system enhances the performance of recommendations by adapting the traditional recommendation methods such as CBF and CF [3–12]. Most of these systems disregarded users' time context and time-dependent watching patterns. As users' time context is an important factor for TV watching, other TV recommender systems use time context as input, which is a time-aware system [13–21]. In this section, these two types of TV recommender systems are surveyed.

2.1. Context-Free TV Recommender Systems

A number of works tried to enhance the performance by adapting CBF approaches [3–7]. Ehrmantraut et al. proposed a personalized EPG based on the individual view preferences for attributes of TV programs such as category, title, participants, and producers [3]. Engelbert et al. proposed a personal video recorder-based TV recommender system calculating users' preference with a Bayesian classifier based on both explicit feedback and users' recording history [4]. Quan and Cho proposed a group recommender system by using Bayesian networks and an analytic hierarchy process [5]. Chaudhry et al. proposed a heterogeneous information network-based TV program recommendation method that utilized the relationship information between the target user and the TV program based on meta-paths of its genre, channel, and timestamp [6]. Hsu et al. considered viewers' properties, such as activities, interests, moods, and these data were fed into a neural network model to predict viewers' program preferences [7]. These methods need a lot of detailed descriptions of programs such as titles, genres, actors, synopsis, and so on. Such information is difficult and costs a lot to be gathered in standalone TVs.

Also, some researchers adapted CF approaches for TV recommender systems [8–12]. Smyth and Cotter proposed an Internet-based personalized television listing system that provided a daily TV guide by adapting CF approaches [8]. Antonelli et al. and Martinez et al. proposed TV recommender systems based on matrix factorization, which is a model-based CF method [9,10]. Hölbling et al. proposed a tag-based collaborative TV recommender system based on user-added tags for TV programs and collaborative tag generation [11]. Kim et al. proposed an automatic TV program recommendation scheme based on CF [12]. They first clustered users based on their genre preferences and performed CF for the group that the active user belongs to. However, the CF approach has a problem that it requires other users' information, which is difficult to be gathered in standalone TV environments.

Fewer works integrated social information into TV recommender systems for better performance [22–24]. Chang et al. and Wang et al. assumed that a user's preference was learnable from viewing history and information in social networks [23,24]. However, this approach also has a problem that personal social data are hard to be gathered in off-line TV environments.

The problem of context-free methods mentioned above is that they calculate only a user's overall preference without consideration of the time context. However, time contextual preferences are different from overall preference, so that these methods can be biased to each user's average favorites.

2.2. Time-Aware TV Recommender Systems

In recent years, TV recommender systems considering time-aware consumption patterns have been actively researched [13–21]. For example, Oh et al. proposed a time-dependent user profiling technique that splits a user's watching logs into certain time slots and re-merges consecutive time slots by using a hierarchical clustering method [13]. Turrin et al. proposed a time-aware TV recommendation method that smoothed a user's time-aware preference based on a distance between time slots [14]. Yuan et al. proposed a recommendation algorithm with a context-aware probabilistic graphical model [15]. They extended the original latent Dirichlet allocation topic model reflecting contextual variations such as time of day context. Oh et al. addressed a new problem for TV recommender systems that determined the timing to recommend [16]. They proposed profit and cost models considering the

remaining-time of TV programs. Oh and Lee proposed a personalized TV channel recommendation method that learnt each user's time-dependent propensity by optimizing with constraints for each viewing instance in logs [17]. Park et al. proposed a real-time recommender system for online broadcasting. They developed user preferences considering broadcasting times and staying times. Then, a 4-D factor factorization is applied [21].

However, most of them did not consider various periodicities in multiple time contexts that are usually observed in users' watching patterns as mentioned in Section 1. They have used one type of time context, so there is a limitation about considering complicated periodic patterns. Some of them were not scalable or required a lot of computing costs, which are not adequate for standalone TVs.

3. Proposed Method

In this section, the proposed live TV recommender system is presented and then each part of the system is explained. The proposed system includes three parts: multi-time contextual profiling, optimizing weights of time contextual factors for each user, and recommending programs by combining time contextual scores.

3.1. Multi-Time Contextual Profiling

The proposed system recommends relevant live TV programs to users based on their preference for the features of broadcasted programs. In general, channels and genres are the available features of programs in standalone TVs as explained in Section 1. So, we recommend TV programs based on the channel and the genre preferences of users. Users' channel and genre preferences are inferred from the channels and genres of TV programs the user watched.

We assume that users' watching logs include channels, time stamps, and genres of programs. Since there is no direct users' preference on channels or genres, we estimate the implicit preference on those using the time stamps. The channel preference is calculated based on the total watching time for the channel. A user u 's preference for channel c in time slot s , notated as $p_{u,c,s}$, is calculated as follows:

$$p_{u,c,s} = \frac{\sum_{l_{c,s}} (end_{l_{c,s}} - begin_{l_{c,s}})}{\sum_{n \in \text{Channels}} \sum_{l_{n,s}} (end_{l_{n,s}} - begin_{l_{n,s}})}, \quad (1)$$

where $l_{c,s}$ is a user u 's log watching channel c in time slot s . $begin_{l_{c,s}}$ and $end_{l_{c,s}}$ stand for the viewing start time and the viewing end time, respectively. Time slot s can be any period such as Monday, 1–2 p.m., etc. The preference is a normalized value, so its range is from 0 to 1. The value of channel preferences can be high if a user frequently watches a specific channel. For example, if a user watches channel 1 for total of 20 h and channel 2 for 30 h on all Mondays, the preference for channel 1 is 0.4 and channel 2 is 0.6 on Monday. In the same way, a user u 's preference for genre g in time slot s , notated as $p_{u,g,s}$, is calculated as follows:

$$p_{u,g,s} = \frac{\sum_{l_{g,s}} (end_{l_{g,s}} - begin_{l_{g,s}})}{\sum_{n \in \text{Genres}} \sum_{l_{n,s}} (end_{l_{n,s}} - begin_{l_{n,s}})}, \quad (2)$$

where $l_{g,s}$ is a user u 's log watching a program of genre g in time slot s . $begin_{l_{g,s}}$ and $end_{l_{g,s}}$ stand for the viewing start and end time, respectively.

Most TV programs are broadcasted periodically. For example, some programs are broadcast at 9 p.m. every day, and others are broadcast every Sunday. The main temporal factors of periodicity are day and hour, so a user's preference is highly associated with them. We model user preferences in four-time context: usual context, day context, hour context, and day-hour context. Usual context is a contextual factor covering all duration of a user's watching history. So, the preference of usual context refers to a user's ordinary taste. In day context, there are seven-time slots from Monday to Sunday to model user preference on each day. In hour context, there are 24-time slots from 0 to

23, which represent time slots of each hour. In day-hour context, there are 168 (7×24) time slots, which represent combinations of days and hours to model user preferences at a specific time on a specific day. For example, $p_{u,c,\text{Mon}}$ is user u 's preference on channel c on Monday time slot.

Thus, the proposed system constructs four time-aware preference matrices for channels and genres as shown in Figure 1. It shows a user's four matrices of time contextual channel preferences. The rows and columns of matrices in Figure 1 denote time slots and TV channels, respectively. User u 's channel preference matrices in usual, day, hour, and day-hour contexts will be denoted as U^u , D^u , H^u , and T^u . For example, D_{Mon,c_1}^u is a user u 's preference for channel 1 on Monday in day matrix, and its value is 0.1 in Figure 1b. $H_{2-3\text{h},c_2}^u$ is user u 's preference for channel 2 at 2–3 a.m. in hour matrix and its value is 0.5 in Figure 1c.

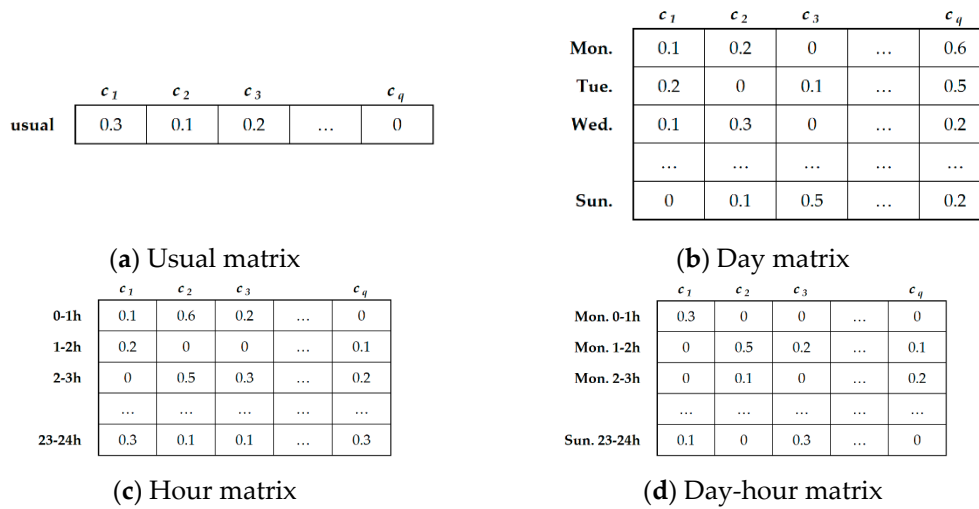


Figure 1. An example of a user's time-aware channel preferences: (a) usual matrix composed of only channels, (b) day matrix composed of channels and days, (c) hour matrix composed of channels and hours, and (d) day-hour matrix composed of channels and days-hours.

Matrices of time contextual preferences on the genre, U^u , D^u , H^u , and T^u , are constructed in the same way. So, the proposed system constructs eight-time contextual matrices with four channels and four genres.

3.2. Optimizing the Weights of Time Contextual Factors

It is assumed that a user's channel preference can be described by combining the defined time contextual matrices, U^u , D^u , H^u , and T^u . That is, the preference of user u for channel c at time t , such that $m_{u,c,t}$ can be defined as follows:

$$m_{u,c,t} = w_U^u \cdot U_c^u + w_D^u \cdot D_{S_D,c}^u + w_H^u \cdot H_{S_H,c}^u + w_T^u \cdot T_{S_T,c}^u, \quad (3)$$

where S_D , S_H , and S_T are the time slot of day, hour, and day-hour matrix that include time t . w_U^u , w_D^u , w_H^u , and w_T^u are the weights of each matrix. For example, the preference of user u for channel 1 at 1:30 a.m. on Monday can be defined as the linear combination of $U_{c_1}^u$, D_{Mon,c_1}^u , $H_{1-2\text{h},c_1}^u$, and $T_{\text{Mon:1-2h},c_1}^u$. If the weights of matrices w_U^u , w_D^u , w_H^u , and w_T^u are 0.1, 0.2, 0.3, and 0.4, respectively, $m_{u,c_1,(1:30 \text{ A.M. on Monday})}$ can be calculated as follows:

$$\begin{aligned} m_{u,c_1,(1:30 \text{ A.M. on Monday})} &= 0.1 \times 0.3 + 0.2 \times 0.1 + 0.3 \times 0.2 + 0.4 \times 0.0 \\ &= 0.11 \end{aligned} \quad (4)$$

Since the weight of the day-hour context is higher than any others, we can say that the user tends to choose TV channels under the day-hour context than other time contextual factors.

Similarly, the preference on genres is defined using time contextual matrices, U^u , D^u , H^u , T^u , and their weight parameters, v_U^u , v_D^u , v_H^u , and v_T^u as follows:

$$m_{u,g,t} = v_U^u \cdot U_g^u + v_D^u \cdot D_{sD,g}^u + v_H^u \cdot H_{sH,g}^u + v_T^u \cdot T_{sT,g}^u \quad (5)$$

In order to find each user's optimal weight parameters w^u 's and v^u 's, a QP technique is used, which is a kind of constrained optimization method. The parameters for channels and genres, w^u 's and v^u 's, are optimized separately with similar QP formulas. If user u watched channel c at time t , it can be considered that user u most preferred channel c at time t . This means that the preference score of channel c at time t must be higher than those of the other channels. This consideration can be formulated as simple inequality constraints. The best weights can be defined as ones that minimize the violation of those constraints for all the viewing records in user u 's history. The optimal values of weight parameters are found by solving the following constrained optimization problem. The following is for weights of channel preferences, w^u 's.

$$\begin{aligned} & \underset{w_U^u, w_D^u, w_H^u, w_T^u}{\operatorname{argmin}} \sum_{i,c} \varepsilon_{u,i,c}^2 \\ & \text{subject to} \\ & w_U^u + w_D^u + w_H^u + w_T^u = 1 \\ & w_U^u \geq 0, w_D^u \geq 0, w_H^u \geq 0, w_T^u \geq 0 \\ & \varepsilon_{u,i,c} \geq 0 \\ & m_{u,c_i,t_i} \geq m_{u,c,t_i} - \varepsilon_{u,i,c} \\ & \text{for } i = 1, \dots, l, c = 1, \dots, q \text{ and } c \neq c_i \end{aligned} \quad (6)$$

ε 's are slack variables to be minimized. The total number of channels is q and the length of user u 's watching history is l . t_i and c_i are the viewing time and channel of the i th record in user u 's watching history. Especially, this optimization process does not require any external information because it only needs a target user's viewing logs.

However, if there are many records in a user history, the optimization process can be very slow because the constraints in the formula will increase proportionally to the number of records and the time complexity of QP is $O(n^3)$. To cope with this problem, we model user preferences based on day-hour time slots instead of records in watching history. As the maximum number of day-hour time slots is 168, it leads to the great decrease of the constraints and needs a constant time for optimization, which is very important for a standalone TV with low computation power.

Let us assume that the present time is 1:30 a.m. on Monday and we recommend a channel for the user in Figure 1. Which channel will the user prefer? From the viewpoint of usual context, the user may prefer channel 1 because channel 1 has the highest score in U^u . From the viewpoint of day context, the user will choose channel q which has the highest score in D_{Mon}^u . From the viewpoint of hour context, it is channel 1, which is the most preferred channel in H_{1-2h}^u , and from the viewpoint of day-hour context, channel 2, the most preferred channel is $T_{\text{Mon:1-2h}}^u$. We may conclude that the user will choose one of channels 1, 2, and q . So, we try to maximize the preference scores of channels 1, 2, and q . That is, the preference scores of channels 1, 2, and q should be higher than those of the other

channels in Mon: 1–2 h time slot. This rule is applied to all day-hour time slots. This alternative idea is formulated as the following constrained optimization problem:

$$\begin{aligned}
 & \underset{w_U^u, w_D^u, w_H^u, w_T^u}{\operatorname{argmin}} \sum_{i,c} \varepsilon_{u,s,c_s,c}^2 \\
 & \text{subject to} \\
 & w_U^u + w_D^u + w_H^u + w_T^u = 1 \\
 & w_U^u \geq 0, w_D^u \geq 0, w_H^u \geq 0, w_T^u \geq 0 \\
 & \varepsilon_{u,s,c_s,c} \geq 0 \\
 & m_{u,c_s,s} \geq m_{u,c,s} - \varepsilon_{u,s,c_s,c} \\
 & c \in \text{Channels and } c \notin C_s, \\
 & c_s \in C_s \\
 & C_s = \bigcup_{P=U,D,H,T} \underset{c \in \text{Channels}}{\operatorname{argmax}} P_{s_p,c}^u \\
 & \text{for } s = \text{Mon : 0–1h}, \dots, \text{Sun : 23–24h}
 \end{aligned} \tag{7}$$

s_p is the time slot including s in P , and C_s is the most preferred channels of U^u, D^u, H^u, T^u in time slot s . For example, if s is Mon: 1–2 h, then s_U, s_D, s_H , and s_T are All Times, Mon., 1–2 h and Mon: 1–2 h, respectively.

In the formula, the number of constraints is bounded by a constant because the number of day-hour time slots is 168 and the number of inequality constraints for a time slot is bounded by four times the number of channels. Since it is fixed to a constant, the running time of the optimization process is also fixed to a constant time, which means that the proposed method is highly scalable regardless of the size of watching history.

In the same way, v^u 's are also optimized. With this optimization step, each user's weight parameters for channel preferences and genre preferences are computed and their watching patterns associated with time contextual factors can be inferred.

3.3. Calculating Recommendation Scores

A user's channel preferences and genre preferences are aggregated to calculate recommendation scores of candidate programs. They are weighted summed to calculate $S_{u,p,t}$, which is a user u 's recommendation score for program p at time t as follows:

$$S_{u,p,t} = \alpha \cdot m_{u,c_p,t} + (1 - \alpha) \cdot m_{u,g_p,t} \tag{8}$$

where α is the weight parameter for the combination. c_p and g_p are the channel and the genre of p , respectively. A user can adapt the value of α . If α becomes larger, the system recommends TV programs depending more on the channel preference than the genre preference. Finally, top k live TV programs are recommended for user u at time t using the recommendation scores.

4. Evaluation

4.1. Experimental Setting

For the evaluation of TV recommender systems, Korean users' watching history dataset and Korean TV programs dataset were used. They were collected from TNmS, which is a Korean audience measurement corporation. The watching history dataset included a user's ID, channel number, and time stamps of each watching record from January to June in 2013. The TV program dataset also included the detailed information, such as titles and genres, of TV programs. The number of available channels was 24, including Korean public and cable channels. The number of the genre was

43, covering drama, entertainment, news, sports, cartoon, and so on. The watching history dataset contained 3318 users and 4,587,478 records. The average number of records per user was about 1383.

The watching records of which length was less than 10 min are removed and users who have less than 200 viewing records were also removed for experiments.

Seven baseline methods were compared with the proposed one: *kNN*, *SVD*, *Pop_usual*, *Pop_day*, *Pop_hour*, *Pop_dayhour*, and *QP_log*. *kNN* and *SVD* are based on CF approaches. *kNN* is a user-based CF. *SVD* is a singular vector decomposition-based CF. *kNN* and *SVD* use the user-program matrix of all users' viewing records. *Pop_usual*, *Pop_day*, *Pop_hour*, and *Pop_dayhour* are popularity-based methods on each user. *Pop_usual* recommends based on matrix U^u , which does not consider time context and others recommend based on a matrix D^u , H^u , and T^u , respectively. *QP_log* is the approach which optimized weights of time-contextual factors of each viewing instance in logs with quadratic programming [17]. We note that it only uses the channel information and does not use the genre as a feature. *Pop_usual*, *kNN*, and *SVD* are time context-free and the others are time context dependent.

The number of neighbors in *kNN* was set to 200; the rank of *SVD* was set to 100; and the parameter α of the proposed method was set to 0.8. We choose the values that showed the best performances. We tried six different numbers of neighbors—50, 100, 150, 200, 250, and 300—for *kNN*. For *SVD*, three different ranks—50, 100, and 150—were searched, and for our method, we tested six values of α : 0.0, 0.2, 0.4, 0.6, 0.8, and 1.0. *kNN* and *SVD* showed very similar performances regardless of parameter changes. We used each user's first 80% of watching logs as the training data and the rest was used as the test data. The recommender system recommended from one to five TV programs. As the evaluation metrics, the hit rate and the average rank are used. The hit rate (HR) is the average value of hits. If the watched program is included in the recommendation list, it is a hit recommendation. The average rank (AR) is the average of the ranks of the watched program in each user's recommendation list. A lower AR means that the watched program is highly recommended by the system.

4.2. Performance Comparison

Tables 1 and 2 show the performance of each TV recommendation method in terms of HR and AR as the size of recommendation list, K , changes from 1 to 5. The methods are divided into three groups, which are CF methods, popular item-based methods, and QP-based methods.

Table 1. Hit rate of TV recommender systems (%).

Method	$K = 1$	$K = 2$	$K = 3$	$K = 4$	$K = 5$
kNN	24.31	41.59	53.16	60.15	65.73
SVD	25.93	39.31	48.20	54.69	59.49
Pop_usual	29.51	51.08	67.09	75.89	80.79
Pop_day	30.30	51.15	65.84	74.14	79.15
Pop_hour	33.20	53.93	67.78	75.68	80.19
Pop_dayhour	34.02	52.61	64.83	73.57	78.38
QP_log	35.49	55.41	69.04	76.87	81.41
Proposed	37.31	56.91	69.90	76.56	80.35

Table 2. Average rank of TV recommender systems.

Method	$K = 1$	$K = 2$	$K = 3$	$K = 4$	$K = 5$
kNN	1.00	1.30	1.62	1.87	2.09
SVD	1.00	1.24	1.53	1.79	2.02
Pop_usual	1.00	1.43	1.81	2.07	2.26
Pop_day	1.00	1.41	1.77	2.03	2.24
Pop_hour	1.00	1.39	1.73	1.97	2.15
Pop_dayhour	1.00	1.36	1.68	1.96	2.16
QP_log	1.00	1.37	1.69	1.94	2.12
Proposed	1.00	1.35	1.66	1.88	2.03

In CF methods, *kNN* was generally better than *SVD* in terms of HR. In popular based methods, *Pop_hour* and *Pop_dayhour*, which considered more detailed time context, were better than others when *K* was small. Otherwise, when *K* was large, *Pop_dayhour* was worse than others. All QP-based methods outperformed other methods. This implied that multi-time contextual profiling was more effective to increase the quality of the system than using only a detailed time context such as *Pop_dayhour*.

When comparing with *QP_log*, the overall performance of the proposed method was prominent, though it had fewer constraints. In terms of AR, the proposed method outperformed *QP_log* for any value of *K*. The improvement of AR was increasing as *K* became large. When *K* was 1–3, the proposed method improved AR by 1–2%, but, when *K* was 4–5, it improved by 3–4%. In terms of HR, the proposed method was better when *K* was 1–3 and was slightly lower when *K* became large. When *K* was 1–3, the improvement was around 1–5%, but around –1% when *K* was 4–5. Even though HR slightly decreased when *K* was large, the improvement of AR, which was about 4%, could compensate for it. Thus, we could conclude that the proposed method more accurately recommends TV programs than the other approaches. Interestingly, the proposed methods showed the maximum improvement of HR when *K* = 1. It improved by 5–50% compared to the baselines. This means that the proposed method very effectively pinpoints TV programs that the target user will watch.

Also, the proposed method was more practical because it required very little computing cost. Figure 2 shows the optimization time of *QP_log* and the proposed method as the size of watching logs increases. Since *QP_log* is based on records in watching history, the optimization time very rapidly increased as the number of records increases. However, the optimization time of the proposed method was bounded by a constant as mentioned in Section 3. If there are some day-hour time slots in which the target user did not watch TV, we could remove constraints for the time slots. So, the actual optimization time was lower than an upper bound as shown in Figure 2. It proves that the proposed method is scalable.

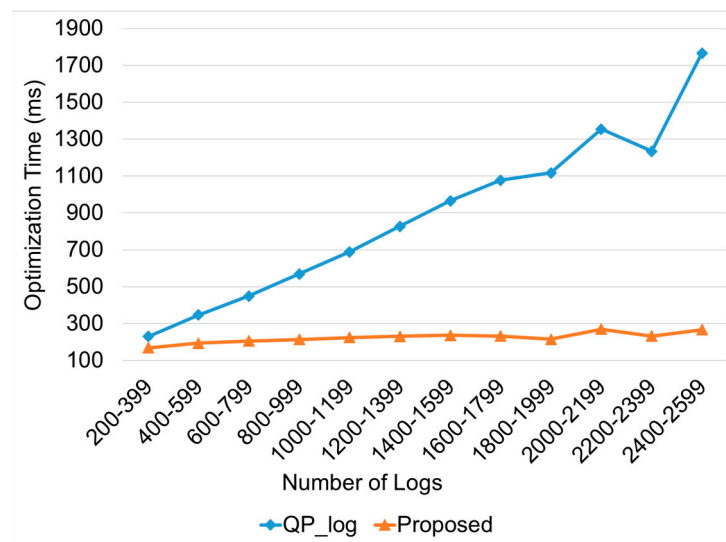


Figure 2. The optimization time as the size of logs increases.

4.3. Parameter Analysis

In this section, we analyze the impact of parameters. Our method was based on one pre-fixed parameter, α , and eight learned parameters, four weight values for channel preferences, w_U^u , w_D^u , w_H^u , and w_T^u , and four weight values for genre preferences, v_U^u , v_D^u , v_H^u , and v_T^u .

Figures 3 and 4 show the performance with different values of the weight parameter α in Equation (8) from 0 to 1. *K* is the number of recommended TV programs. In all cases of *K*,

the recommendation performance was the best when both channels and genres were considered together. In terms of HR, the performance was high when α was 0.6–1.0 and tended to improve as α increased for any value of K . This meant that both the channels and the genre preference were necessary, but the channel preference was more effective for better HR. In terms of AR, the values of α for the best performance were 0.1–0.6 and tended to improve as α decreased for any value of K . This implied that the genre preference was effective to improve AR.

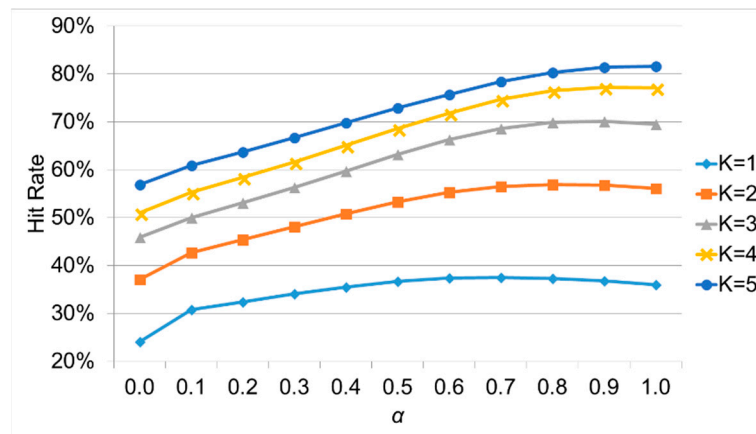


Figure 3. Hit rate as the weight parameter changed.

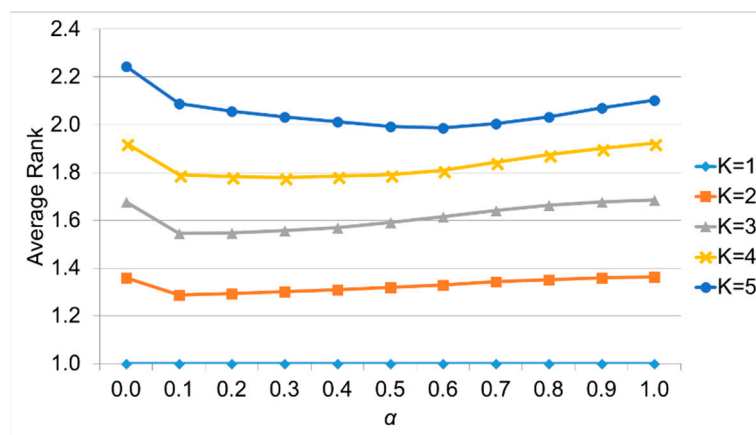


Figure 4. Average rank as the weight parameter changed.

When $K = 5$, HR was the best under $\alpha = 1.0$. This may be related to users' watching patterns. In Table 1, the top five channels that a user frequently watched cover more than 80% of TV watching. So, there was little improvement using genre features.

Figure 5 shows the average values of weight parameters of channel preferences and genre preferences. The blue bars were weight values for channel preferences, w_U^u , w_D^u , w_H^u , and w_T^u in Equation (3), and the red ones for genre preferences, v_U^u , v_D^u , v_H^u , and v_T^u in Equation (5). For both channel preferences and genre preferences, the weight value of usual context was larger than that of other time contextual factors. So, it could be inferred that most of users were likely to be influenced by the usual context when they selected a program. In other words, users tended to select TV programs that they usually watch regardless of time. Hour context was the second influential time-contextual factor, followed by day context and day-hour context. Overall, it was inferred that users selected a program mainly based on their usual preference and other time contextual preferences that function as a variation on users' usual preference.

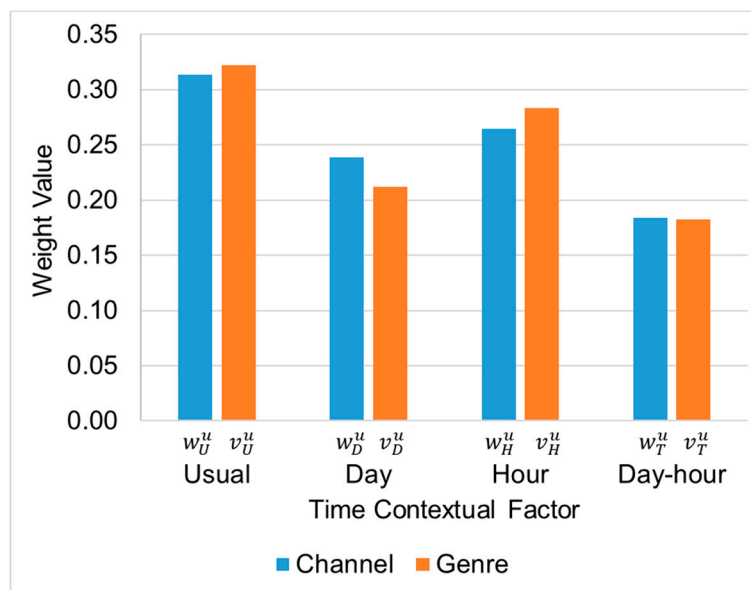


Figure 5. The average value of weight parameters, w^u 's and v^u 's.

4.4. Effects of External User Information

CF methods, which utilize external user information, usually show good performance in recommendations. Since our methods depend on only the target user's watching history, we analyzed the additional gain of our methods by incorporating external user information.

Tables 3 and 4 show the performance of the proposed method extended with external user information. QP_global was the same as the proposed method but it used only global time contextual matrices that are constructed from all users' watching history. It recommended the same TV programs for any users if time context was the same. QP_fusion assembles the proposed method and QP_global . Thus, it recommended TV programs considering not only the target user information but also the external users' information. The recommendation score, $S_{u,p,t}$, was calculated as follows:

$$S_{u,p,t} = \beta \cdot (\alpha \cdot m_{u,c_p,t} + (1 - \alpha) \cdot m_{u,g_p,t}) + (1 - \beta) \cdot (\alpha \cdot \mu_{c_p,t} + (1 - \alpha) \cdot \mu_{g_p,t}) \quad (9)$$

where $m_{u,c_p,t}$ and $m_{u,g_p,t}$ are the channel and genre scores by the proposed method for user u ; $\mu_{c_p,t}$ and $\mu_{g_p,t}$ are the channel and genre scores by QP_global ; and α and β are the weight parameters for the combination, which are set to 0.8 and 0.9, respectively.

Table 3. Hit rate of TV recommender systems with external user information (%).

Method	K = 1	K = 2	K = 3	K = 4	K = 5
QP_global	25.42	47.52	62.78	70.38	73.22
QP_fusion	37.32	57.05	70.20	76.85	80.63
Proposed	37.31	56.91	69.90	76.56	80.35

Table 4. Average rank of TV recommender systems with external user information.

Method	K = 1	K = 2	K = 3	K = 4	K = 5
QP_global	1.00	1.47	1.85	2.10	2.23
QP_fusion	1.00	1.35	1.67	1.88	2.03
Proposed	1.00	1.35	1.66	1.88	2.03

Compared to the proposed method, both HR and AR of *QP_global* were much worse. This meant that the personal information was essential to the personalized TV recommender system. The performance of *QP_fusion* was almost the same as that of the proposed method. Although the personal and global information were fused, there was little improvement compared to the proposed method, which means that there was little room for global information to improve. It can be noted that our proposed method very effectively recommends TV programs with a minimum amount of information.

5. Conclusions

We presented a novel time-aware TV program recommendation method specialized for standalone TVs. The proposed method profiled a user's history by constructing multiple time contextual preference matrices of channels and genres to consider various periodicities. We also developed a quadratic programming model to effectively combine time-contextual preference matrices.

The proposed system has some merits. One is that the system used the available information in standalone TVs, which are channels and genres of programs. The other is that the maximum number of constraints of QP is a constant regardless of the size of watching history so that the system is highly scalable.

Evaluation results with real-world dataset proved that the proposed method was more efficient and accurate than other TV recommendation methods. Our method improved recommendation performance by 5–50% compared to the baselines. The learning process was fast and scalable so that it can operate within a standalone TV with low computation power.

Author Contributions: J.-H.L. provided guidance and supervision. N.-r.K. and S.O. conceived the idea of this paper, performed the experiment, and wrote the paper. All authors contributed equally to the analysis, experiment, and result discussions.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2016R1A2B4015820). Also, this research was supported by the Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2016 (R2016030046).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Portugal, I.; Alencar, P.; Cowan, D. The use of machine learning algorithms in recommender systems: A systematic review. *Expert Syst. Appl.* **2018**, *97*, 205–227. [[CrossRef](#)]
2. Harrison, C.; Amento, B.; Stead, L. iEPG: An ego-centric electronic program guide and recommendation interface. *ACM Int. Conf. Proc. Ser.* **2008**, *291*, 23–26. [[CrossRef](#)]
3. Ehrmantraut, M.; Härder, T.; Wittig, H.; Steinmetz, R. The personal electronic program guide—Towards the pre-selection of individual TV programs. In Proceedings of the Fifth International Conference on Information and Knowledge Management (CIKM '96), Rockville, MD, USA, 12–16 November 1996; pp. 243–250.
4. Engelbert, B.; Blanken, M.B.; Kruthoff-Bruwer, R.; Morisse, K. A user supporting personal video recorder by implementing a generic Bayesian classifier based recommendation system. In Proceedings of the 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), Seattle, WA, USA, 21–25 March 2011; pp. 567–571.
5. Quan, J.-C.; Cho, S.-B. A Hybrid Recommender System Based on AHP That Awares Contexts with Bayesian Networks for Smart TV. In *International Conference on Hybrid Artificial Intelligence Systems (HAIS 2014)*; Springer: Cham, Switzerland, 2014; pp. 527–536.
6. Chaudhry, M.U.; Oh, S.; Kim, N.; Lee, J.H. Heterogeneous Information Network based TV Program Recommendation. In Proceedings of the 16th International Symposium Advanced Intelligent Systems, Mokpo, Korea, 4–7 November 2015; pp. 808–815.
7. Hsu, S.H.; Wen, M.-H.; Lin, H.-C.; Lee, C.-C.; Lee, C.-H. AIMED—A Personalized TV Recommendation System. In *Interactive TV: A Shared Experience*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 166–174.

8. Smyth, B.; Cotter, P. A personalised TV listings service for the digital TV age. *Knowl. Based Syst.* **2000**, *13*, 53–59. [[CrossRef](#)]
9. Antonelli, F.; Francini, G.; Geymonat, M.; Lepsoy, S. DynamicTV: A Culture-aware Recommender. In Proceedings of the Third ACM Conference on Recommender Systems, New York, NY, USA, 23–25 October 2009; pp. 257–260.
10. Martínez, A.B.B.; Arias, J.J.P.; Vilas, A.F.; Duque, J.G.; Nores, M.L. What's on TV Tonight? An Efficient and Effective Personalized Recommender System of TV Programs. *IEEE Trans. Consum. Electron.* **2009**, *55*, 286–294. [[CrossRef](#)]
11. Hölbling, G.; Thalhammer, A.; Kosch, H. Content-based tag generation to enable a tag-based collaborative tv-recommendation system. In Proceedings of the 8th International Conference on Interactive TV and Video (EuroITV '10), Tampere, Finland, 9–11 June 2010; p. 273. [[CrossRef](#)]
12. Kim, E.; Pyo, S.; Park, E.; Kim, M. An automatic recommendation scheme of TV program contents for (IP)TV personalization. *IEEE Trans. Broadcast.* **2011**, *57*, 674–684. [[CrossRef](#)]
13. Oh, J.; Sung, Y.; Kim, J.; Humayoun, M.; Park, Y.H.; Yu, H. Time-dependent user profiling for TV recommendation. In Proceedings of the 2nd International Conference on Cloud and Green Computing and 2nd International Conference on Social Computing and Its Applications, CGC/SCA 2012, Xiangtan, China, 1–3 November 2012; pp. 783–787.
14. Turrin, R.; Condorelli, A.; Cremonesi, P.; Pagano, R. Time-based TV programs prediction. In Proceedings of the 1st Workshop on Recommender Systems for Television and Online Video (RecSysTV) 2014, Foster City, CA, USA, 6–10 October 2014.
15. Yuan, J.; Sivrikaya, F.; Hopfgartner, F.; Lommatzsch, A.; Mu, M. Context-aware LDA: Balancing Relevance and Diversity in TV Content Recommenders. In Proceedings of the 2nd Workshop on Recommender Systems for Television and Online Video (RecSysTV) 2015, Vienna, Austria, 16–20 September 2015.
16. Oh, J.; Kim, S.; Kim, J.; Yu, H. When to recommend: A new issue on TV show recommendation. *Inf. Sci.* **2014**, *280*, 261–274. [[CrossRef](#)]
17. Oh, S.; Lee, J.H. Personalized TV Channel Recommendation Considering Viewer's Time Dependent Propensity Using Constrained Optimization Technique. In Proceedings of the 16th International Symposium Advanced Intelligent Systems, Mokpo, Korea, 4–7 November 2015; pp. 817–824.
18. Cremonesi, P.; Turrin, R. Time-evolution of IPTV recommender systems. In Proceedings of the 8th International Conference on Interactive TV and Video (EuroITV '10), Tampere, Finland, 9–11 June 2010; pp. 105–114. [[CrossRef](#)]
19. Oh, S.; Kim, N.R.; Lee, J.; Lee, J.H. Comparison of techniques for time aware TV channel recommendation. In Proceedings of the 2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS 2014) and 15th International Symposium on Advanced Intelligent Systems (ISIS 2014), Kitakyushu, Japan, 3–6 December 2014; pp. 989–992.
20. da Silva, F.; Alves, L. PersonalTVware: An Infrastructure to Support the Context-Aware Recommendation for Personalized Digital TV. *Int. J. Comput. Theory Eng.* **2012**, *4*, 131–136. [[CrossRef](#)]
21. Park, Y.; Oh, J.; Yu, H. RecTime: Real-Time recommender system for online broadcasting. *Inf. Sci.* **2017**, *409–410*, 1–16. [[CrossRef](#)]
22. Barraza-Urbina, A.; Hromic, H.; Heitmann, B.; Hulpus, I.; Hayes, C. Using social media data for online television recommendation services at RTÉ Ireland. In Proceedings of the 2nd Workshop on Recommender Systems for Television and Online Video (RecSysTV) 2015, Vienna, Austria, 16–20 September 2015.
23. Chang, N.; Irvan, M.; Terano, T. A TV program recommender framework. *Procedia Comput. Sci.* **2013**, *22*, 561–570. [[CrossRef](#)]
24. Wang, X.; Sun, L.; Wang, Z.; Meng, D. Group recommendation using external followee for social TV. In Proceedings of the 2012 IEEE International Conference on Multimedia and Expo, Melbourne, Australia, 9–13 July 2012; pp. 37–42.
25. Rosaci, D.; Sarné, G.M.L.; Garruzzo, S. MUADDIB: A distributed recommender system supporting device adaptivity. *ACM Trans. Inf. Syst.* **2009**, *27*, 1–41. [[CrossRef](#)]
26. Rosaci, D.; Sarné, G.M.L. Cloning mechanisms to improve agent performances. *J. Netw. Comput. Appl.* **2013**, *36*, 402–408. [[CrossRef](#)]
27. Chung, Y.; Kim, N.; Park, C.; Lee, J. Improved Neighborhood Search for Collaborative Filtering. *Int. J. FUZZY Log. Intell. Syst.* **2018**, *18*, 29–40. [[CrossRef](#)]

28. Wei, J.; He, J.; Chen, K.; Zhou, Y.; Tang, Z. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Syst. Appl.* **2017**, *69*, 1339–1351. [[CrossRef](#)]
29. Rosaci, D. Finding semantic associations in hierarchically structured groups of Web data. *Form. Asp. Comput.* **2015**, *27*, 867–884. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).