Catch-up TV Recommendations: Show Old Favourites and Find New Ones

Mengxi Xu
University of Sydney and NICTA
Cynthia.Xu@nicta.com.au

Shlomo Berkovsky, Sebastien Ardon
Sipat Triukose, Anirban Mahanti
NICTA
firstname.lastname@nicta.com.au

Irena Koprinska
University of Sydney
irena@it.usyd.edu.au

ABSTRACT

Web-based catch-up TV has revolutionised watching habits as it provides users the opportunity to watch programs at their preferred time and place, using a variety of devices. With the increasing offer of TV content, there is an emergent need for personalised recommendation solutions, which help users to select programs of interest. In this work, we study the watching patterns of users of an Australian nationwide catch-up TV service provider and develop a suite of approaches for a catch-up recommendation scenario. We evaluate these approaches using a new large-scale dataset gathered by the Web-based catch-up portal deployed by the provider. The evaluation allows us to compare the performance of several recommenders that address the discovery of both TV programs already watched by users and new programs that users may find relevant.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*distributed systems, user profile and alert services*; H.5.m [**Information Interfaces and Presentation**]: Miscellaneous

General Terms

Algorithms, Design, Experimentation.

Keywords

Recommender Systems; Catch-up TV; Grouped Video Content Recommendations; Large-Scale Evaluation.

1. INTRODUCTION

Catch-up TV refers to a TV content delivery paradigm where already broadcast programs are uploaded to a dedicated portal, so that users can watch these programs later at their own convenience. Several TV networks, such as the ABC and SBS in Australia, and the BBC in the UK, offer catch-up TV services to their viewers, and this content delivery model is fast evolving as a new approach for engaging users and combating piracy within the TV broadcasting industry. Recent marketing studies found that users tend to watch stored TV programs in addition to live broadcast, and frequently combine the two delivery modes.

With the flexibility of watching any TV program at any time and place, users often face information overload when selecting a program to watch. This challenge becomes particularly acute considering the entertainment context of TV watching, where users prefer to lay-back and relax rather than actively search for a suitable program [6]. Consequently, there is an emergent need for personalised solutions and recommender systems capable of selecting programs on behalf of their users.

The practical challenge of generating accurate recommendations is, however, arduous for several reasons. First, due to privacy restrictions, TV content providers often can only use partial user data they possess. Second, user logs are often noisy and inaccurate, often for technical reasons, i.e. due to lack of a single sign-on. Third, some users may use multiple devices to watch programs and, vice versa, some devices may be used by multiple users. Finally, users' TV content consumption may be strongly affected by external contextual factors, which cannot be captured.

In this work, we partner with a leading Australian national TV network, and study and evaluate a number of recommendation approaches that leverage observed real-life user interactions and viewing logs, to generate personalised TV program recommendations. Specifically, we consider the recommendation for catch-up TV services with *grouped* content. Grouped content refers to multiple programs bound by the same plot or theme (e.g., TV series and football games), or to programs shown at fixed days or certain time slots (e.g., daily news and weekly talk shows). For the sake of simplicity, we denote the groups of content as series and the individual programs belonging to these series as episodes. We disregard in this work standalone individual programs such as movies.

The recommendation engine consists of two components, each responsible for selecting a different type of program. The first component, which we refer to as the *subscribed* series recommender, identifies content regularly watched by a user and recommends unwatched programs that belong to this content. For the sake of simplicity, these recommendations can be considered as recommendations for new episodes of shows the user has been regularly watching. The subscribed recommendation component is implemented as a rule-based recommender aimed to maximise the consump-

RecSys'13, October 12–16, 2013, Hong Kong, China.
Copyright 2013 ACM 978-1-4503-2409-0/13/10 ...$15.00.
http://dx.doi.org/10.1145/2507157.2507204.

tion of the subscribed content. The second component, which we refer to as the *new* series recommender, recommends content that a user is unfamiliar with, i.e., new programs. This component capitalises on the preferences of the entire community of users and builds upon well established collaborative recommendation methods. The goal of this recommender is to expand the users' horizons and expose users to new TV content. Finally, the recommendations generated by the two components are combined using the mixed hybridisation technique [7].

We present an extensive offline evaluation of the aforementioned recommendation components using a dataset gathered by our partner catch-up TV service provider. The dataset contains six months of Australia-wide usage logs that encompass nearly 20 million views by more than 2 million unique users, who collectively watched more than 11,000 unique programs. We evaluate and compare the performance of several recommendation approaches and assess their ability to recommend subscribed content of interest, as well as new unwatched content. Our evaluation shows that we can recommend subscribed content with a high degree of accuracy such that around 75% of the recommended programs are watched by users. The task of recommending new TV content is, however, much more challenging as the ground truth regarding new content that users watch in the catch-up scenario is less reliable. Accordingly, the performance of the new recommender is poorer than of the subscribed recommender. We evaluate several methods for the new recommender, but they reach at best the 12% accuracy level. To summarise, the main contribution of this paper is a thorough evaluation, with a new real-life catch-up TV dataset, of a suite of recommendation approaches for catch-up TV services.

The remainder of this paper is structured as follows. Section 2 presents background on catch-up TV services. Section 3 discusses data collection and presents a high-level characterisation of the dataset used in the evaluation. Section 4 describes the user modelling and recommendation approaches we apply, while Section 5 presents the experimental evaluation and the obtained results. Related work is discussed in Section 6. Finally, Section 7 summarises the work and outlines future research directions.

2. BACKGROUND

Catch-up TV is a video-on-demand service that allows users to watch at their preferred time recently broadcast TV programs that they missed. Catch-up TV services are usually provided via a Web portal and accessible through a variety of devices, such as PCs, tablets, smartphones, and gaming consoles. Catch-up TV differs from traditional video-on-demand services in that the content is typically accessible for a limited time period following the program broadcast. Often, the availability period depends on the licensing terms negotiated between the catch-up service providers and content right holders.

The popularity of catch-up TV has recently sky-rocketed and its consumption figures are approaching those of the traditional TV [9]. Similarly to other video-on-demand services, catch-up TV service providers are looking to maximise user satisfaction, increase user engagement, and increase content consumption. To achieve this, many catch-up TV providers are interested in personalisation and recommendation approaches that allow to have the programs presented to users tailored to their interest and watching preferences.

For this work, we partner with a leading national TV broadcaster in Australia. Each program shown on the TV channels of the broadcaster is made available through its Web-based catch-up portal, typically on the day following the broadcast. Programs remain typically available in the catch-up portal for a period of one, two, or four weeks. However, there are some exceptions; e.g., news and trailers remain available for two days only. The catch-up portal can be accessed through a variety of devices and platforms, e.g., Apple iOS, smart TVs, and Web browsers.

The main user interface of the portal is currently non-personalised. It offers to viewers three lists of programs: two time-based lists ordered by popularity ('just added' and 'about to expire'), and an editorially-curated list ('featured'). It should be highlighted that the selection of programs in these lists is done manually by domain experts in a generic manner, and no personalisation is applied. In addition to the main interface, users can access the programs through several search interfaces. The programs are mapped by domain experts into 12 categories: arts, children (aged 6 to 15), comedy, documentaries, drama, education, lifestyle, news, panel, preschool (children aged under 6), shop, and sport. Users can browse the tree of categories and programs, search for programs through channels on which they were broadcast, or alternatively use a free-text search.

Once a user finds a program of interest and clicks on its thumbnail, a detailed program information is shown. This information includes the program title, category, textual description, broadcast data, publication date (when it was uploaded to the portal), expiry date (when it will be removed from the portal), parental rating, duration, and availability of captions. If the program belongs to the grouped content, other available episodes of the same series are also displayed. The user can then invoke the player and watch the program, share it via online social networks, or email the link to the program. The service allows deep-linking of videos, so that they can be accessed directly from a link on a news article, blog, or social network post, without going through the main or search interface of the portal.

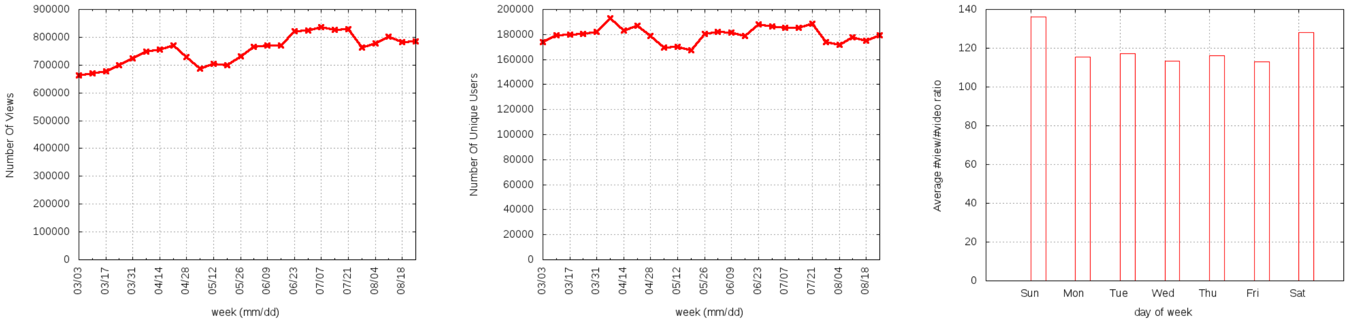
3. OVERVIEW OF DATASET

3.1 Data Collection

As already mentioned, the data used in this work originates from the catch-up TV service of a leading Australian broadcaster that runs a number of national free-to-air channels. It is a popular network, offering both local and international programs. We obtained the complete Australia-wide portal logs for a period of 184 days (26 weeks), from March to August 2012. As shown in Table 1, the dataset includes nearly 20 million views of more than 2 million unique users, who collectively watched 11,699 unique programs.

Time span	184 days
View count	19.8 millions
Unique videos viewed/available	9,114/27,621
Unique cookie	2.02 millions

Table 1: **Dataset high-level description.**



(a) (b) (c)

Figure 1: High-level characterisation: (a) number of videos watched per week; (b) number of unique users per week; (c) average number of views per video as a function of day-of-the-week.

The gathered data includes very little information about the users, as the catch-up service does not include a user sign-on identification. As the service uses HTTP cookies for audience measurement purposes, we use these cookie IDs as a proxy for user IDs. We admit that this mapping of cookie IDs to users is clearly imprecise, as multiple users may use a single device to access the catch-up portal (and have the same cookie ID) and, vice versa, a single user may use multiple devices or browsers (and have multiple cookie IDs). Moreover, users can clear cookies at any time, in which case a new user is created next time they use the service. Alternatively, users can just block cookies, in which case no records of that user are available. It is important to note that this may be one of the key sources of inaccuracy for the recommender. However, these limitations are those typically faced by the industry when deploying catch-up TV services, as requiring users to sign-on before watching TV content may impair user experience. Thus, the gathered logs characterise the real-life usage data accessible by a commercial catch-up TV portal.

For each video that was accessed via the portal, the logs contain a cookie ID, the access date (with no time stamp), and the video ID. No information about the portion of the video that was actually watched is available. In addition, the service provider partially exposed the content meta-data, which was collected on a daily basis. For each video, the meta-data contains a unique video ID, a series ID, the video title, duration, publication date, expiry date, a category tag (chosen from the above 12 categories), and the size of the video file. The logs contain only the observed watching events and do not include any evidence of searches, information accesses, social media posts, and so on.

3.2 High-Level Characterisation

In this section, we present a high-level characterisation of the dataset. Figure 1 (a) shows the number of video accesses per week observed during the data collection period. We observe a slow, but steady, increase in the video access rate over time, with the number of requests increasing from under 700,000 requests/week at the beginning to about 800,000 requests/week at the end. Figure 1 (b) shows the number of unique users, counted using the cookie IDs, seen each week, which steadily hovers around the 180,000 users/week mark. We also study whether there are changes in the the average number of requests per video between weekdays and weekends. Figure 1 (c) shows the average number of video accesses as a function of the day-of-the-week. It can be clearly seen that the average request rate for videos on the weekends is higher than on the weekdays.

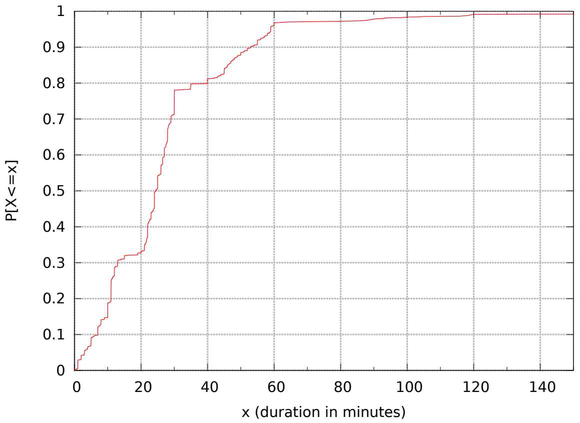


Figure 2: Distribution of video durations (minutes).

Figure 2 shows the empirical cumulative distribution of the duration of the available videos. It can be seen that more than 30% of videos are shorter than 20 minutes, about 45% are between 20 and 30 minutes in length, and only 2% are longer than 60 minutes. The bulk of the content is between 20 and 30 minutes long, corresponding to the typical length of TV programs shown by the broadcaster. The shorter programs in the dataset primarily include news and program trailers.

Next, we measure how the number of views per video changes as a function of their age, where the age is measured as the number of days since the publication date. Earlier, we noted that the portal typically made videos available for a period of one, two, or four weeks. Figure 3 considers these three catch-up availability windows, and shows for these three groups how the average viewing rate varies as a function of the video age. For all three groups considered, the average viewing rate decreases exponentially with the video age, showing that most of the requests are made when the videos are only a few days old. This is likely a function of both the viewing behaviour of the users and the organisation of the portal interface of the catch-up portal interface,

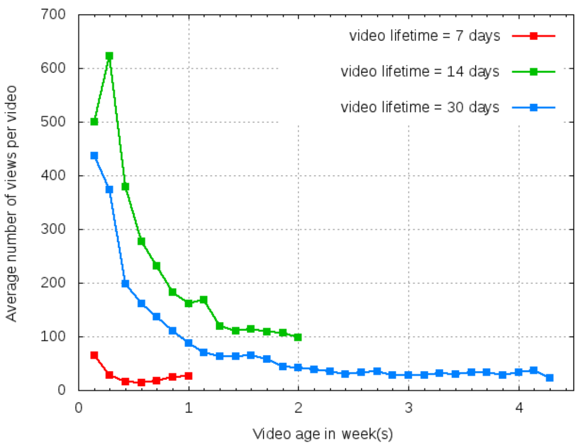


Figure 3: Average viewing rate as a function of video age.

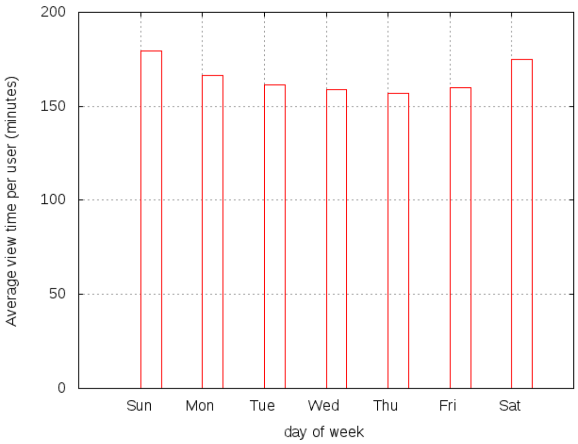


Figure 4: Average viewing time of users as a function of day-of-the-week.

which prominently displays recently uploaded videos. This suggests that most users catch-up on the missed TV content as closely as possible to the content broadcast.

Figure 4 shows, for each day-of-the-week, an upper bound on the average time spent by users watching catch-up TV. Recall that our dataset only includes information on video accesses and does not contain information on the duration of each video playback by the users. This computations assume that each video accessed was watched in its entirety, and, hence, it is an upper bound for the viewing time. Nonetheless, we observe that, on average, users watched content between 150 and 180 minutes per day, with watching time increasing on weekends compared to weekdays. Given that a typical video is 20 to 30 minutes long, this represents an engagement of 5 to 6 videos per day. Note that on any day the catch-up TV portal offers to users between 1000 and 1200 videos. This result highlights the content discovery problem: users are faced with finding 5 to 6 programs from this overwhelming volume of video content.

Related to the above analysis is the typical user stickiness, i.e., the frequency with which users return to the portal. Among the unique users (identified through cookie IDs) seen each week, Figure 5 shows the breakdown of total views into those made by users that requested only one video, between two and nice videos, and ten videos or more. The analysis shows that approximately 40% of users are one-timers, who view only one video and do not return to the portal any more. We also observe that only 10% of users are active enough to view ten videos or more over the course of the logs. It appears that a large fraction of users are accessing the portal as it was conceived, i.e., to catch-up on programs they missed broadcast on TV. A small number of users are frequent visitors, who rely on the catch-up TV service for their viewing needs. In the rest of this paper, we present a recommender focusing on the frequent users.

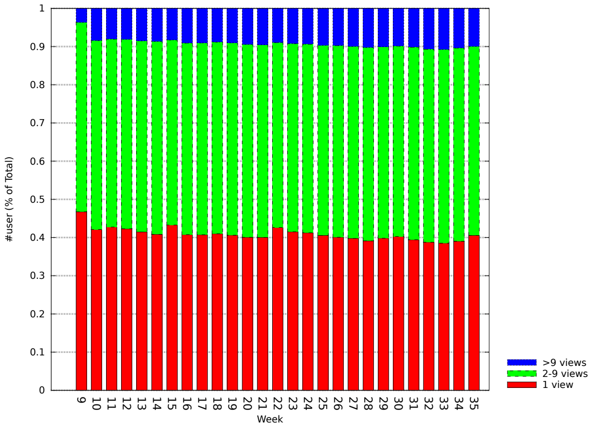


Figure 5: Number of k-time users as a function of time.

4. RECOMMENDATION APPROACHES

We outline a number of recommendation approaches that are evaluated with the gathered dataset. Since the TV content is grouped, the problem is reduced from recommending individual programs to recommending series. The task is further split into two sub-tasks: recommending subscribed series that a user has been regularly watching and recommending new series that a user has not watched yet.

4.1 User Data

We start with defining the data used by the recommenders. Raw logs of from the catch-up portal encapsulate binary viewing ratings $r_{u,i}$:

$$r_{u,i} = \begin{cases} 1 & u \text{ watched } i \\ 0 & \text{otherwise} \end{cases}$$

As discussed in Section 3, the number of available programs is much greater than the average number of programs watched by a user on a given day. Thus, the ratings matrix corresponding to raw viewing logs is sparse. Collapsing episodes into series reduces the dimensionality of the problem and increases the density of the data [5].

Let us denote by S the set of series available at the catch-up portal and by U the set of the portal users. The collapsed rating matrix R is built by computing the implicit $score_{u,s}$ of all users $u \in U$ for all series $s \in S$. This is computed as

$$score_{u,s} = \frac{n_{u,s}}{avail_{u,s}}$$

where $n_{u,s} = \sum_{i \in s} r_{u,i}$ denotes the number of episodes of s that u watched and $avail_{u,s}$ denotes the number of episodes of s that were available since u had joined. The joining date for u is approximated by the the earliest view logged for them. In essence, $score_{u,s}$ reflects the degree of interest of u in s. Then, the user profile P_u represents the ordered set of scores of u for all $s \in S$ series, $P_u = \{score_{u,s}\}$, and the resulting rating matrix $R_{|U| \times |S|}$ represents the collection of all the available user profiles.

We also introduce the binary notion of user subscription to a series, denoted by $sub_{u,s}$, which is computed as

$$sub_{u,s} = \begin{cases} 1 & score_{u,s} \geqslant \alpha \ \wedge \ n_{u,s} \geqslant \beta \\ 0 & \text{otherwise} \end{cases}$$

If the implicit score $score_{u,s}$ of u for s is greater than α and u watched more than β episodes of s, then u is considered *subcribed* to s. The relative threshold α corresponds the implicit score for the series as a proxy for the level of user's interest. The absolute threshold β corresponds to the minimal required number of watched episodes and is useful for long running series that may have a large number of episodes. We define the subscription list SL_u as the set of series to which u is subscribed, $SL_u = \{s \in S \mid sub_{u,s} = 1\}$.

4.2 Subscribed Series Recommender

The goal of the subscribed recommender is to select series that a user regularly watches, but have yet unwatched episodes that the user is likely to watch. In other words, the output of the recommender is restricted to series $s \in SL_u$. Note that this cardinally differs from a typical recommendation scenario, where recommendable items have not been experienced by users, such that their score is unknown and it is predicted by the recommender. On the contrary, in our case the goal is to recommend regularly watched series, the score of which is known at the recommendation time.

Hence, we do not apply any of the state-of-the-art recommendation approaches, but rather two personalised rule-based approaches that use the subscription data to recommend series. The first one, *preference-based subscribed* recommender, selects n series with the highest implicit $score_{u,s}$ from the the subscription list SL_u of the target user u and recommends these. The second, *random subscribed* recommender, selects the n recommended series from SL_u at random rather than according to their scores.

In addition to the two rule-based recommenders, we use two non-personalised recommenders, which consider neither the profile P_u nor the subscription list SL_u when generating recommendations, as baselines for comparison. The first, *most popular*, recommends n series with the highest aggregated score across the entire community. The aggregated score for a series s is computed by averaging $score_{u,s}$ of all the users subcribed to s. The second, *random* recommender, picks at random n series amongst all the series S available in the system and recommends these to u.

4.3 New Series Recommender

Recommending series that a user already watches inherently limits the discovery of new content, as series outside a user's subscription list cannot be recommended. Hence, we use four personalised algorithms (user-to-user collaborative filtering, cluster-based, matrix factorisation, and slope one) to recommend new series. They all predict the score $pred_{u,s}$ for unsubscribed series $s \notin SL_u$ and recommend n series with the highest predicted scores. We briefly overview the four new series recommendation algorithms.

User-to-User Collaborative Filtering. User-to-user CF assumes that users will like items that were liked by similar users [10]. Hence, CF initially computes the degree of similarity between the target user u and all other users. Then, for every series s yet unwatched by u, the recommender selects a set of k most similar users who are already subscribed to s, and computes the predicted score of u for s as a weighted average of the scores of these users. The relative weight of users reflects the degree of their similarity to the target user. Finally, n series with the highest scores are recommended to u.

Cluster-Based Recommender. This recommender is similar to user-to-user CF in the sense that the predicted score $pred_{u,s}$ is also computed as a weighted average of the scores of other users. The users are first partitioned into a discrete set of clusters based on their rating patterns, as described in [17]. The scores of all users belonging to the cluster of the target user u are then taken into account, such that no user-to-user similarity computation is needed. Upon the computation of the $pred_{u,s}$ scores, n top-scoring series are recommended.

Slope-One Recommender. Slope-one recommender is a simplified version of item-to-item CF that does not weigh items according to their similarity [13]. It assumes that a linear relationship between the series scores can be identified. The algorithm computes the average difference between the scores of all pairs of series s_a and s_b using the scores of the entire community of users. It then computes the predicted score $pred_{u,s}$ for user u and series s by adding the computed average difference to all the known $score_{u,s_a}$ of u and averaging the results across all the known scores. Finally, n series with the highest predicted scores are recommended.

Matrix Factorisation. MF recommenders represent a family of model-based recommenders that apply dimensionality reduction techniques to the rating matrix R [12]. The algorithm factorises R into a product of two latent matrices: user matrix p and series matrix q, using the alternative least squares approach (latent matrices p and q are iteratively optimised individually rather than in parallel). Upon the completion of the factorisation process, the latent matrices are multiplied to predict the score $pred_{u,s}$ for yet unwatched series. Again, n series with the highest score are finally recommended to u.

For the new series recommendation task we also evaluate two non-personalised baselines: *most popular* and *random*. These are similar to their non-personalised subscribed recommendation counterparts, but pick the recommended series from the set of series to which u is not subscribed.

4.4 Combining Recommendations

We revisit the different purposes of the subscribed and new series recommenders. The former suggests to users series that they regularly watch and there are new episodes that they are likely to watch. The latter, on the contrary, suggests new series that users are likely to find interesting. Hence, the outputs of both recommenders are of relevance, and should be combined and presented to users. This combination fits the *mixed* hybridisation method [7], where "recommendations from several recommenders are presented at the same time".

5. EXPERIMENTAL EVALUATION

We conducted an offline evaluation of the two recommendation components using the gathered dataset. In this section, we first outline the evaluation setting and then present, analyse, and discuss the obtained results.

5.1 Evaluation Setting and Metrics

We split all the available data into the training set and test set. The recommenders are trained on the training set data and their performance is evaluated on the withheld test set data. In our case, the training set includes the data ranging for the first 136 days of the portal logs and the test set includes three immediately following days.

The training period data consists of 14.9 million views. As discussed earlier, many users had too few views to generate personalised recommendations. Hence, we exclude from the evaluation all the users, who watched less than 30 programs during the training period. The remaining training data contains 11.9 million views of almost $125,000$ unique users and about $9,000$ programs. On average, every user watched 95.1 programs and every program was watched by 14.1 users. During the three day test period, more than $430,000$ views were logged for these users.

Since no explicit ratings for the watched programs are available, we cannot use predictive accuracy and use the classification accuracy metrics of precision and recall [16]. Precision quantifies the ability of the recommender to select the watched and filter out the unwatched programs. Recall quantifies the ability of the recommender to select as many watched programs as possible. Given a user u, who was recommended a set of programs Rec and watched a set of programs $Viewed$ over the course of the test period, precision and recall for u are computed as

$$prec_u = \frac{|Viewed \cap Rec|}{|Rec|} \qquad recall_u = \frac{|Viewed \cap Rec|}{|Viewed|}$$

Finally, we compute the average precision, $\overline{prec}$, and average recall, $\overline{recall}$, across all the users.

We also measure the coverage of the recommendations, which communicates the ability of the system to generate recommendations, regardless of their accuracy. Specifically, we apply user-based coverage, $cover_u = |Rec|/n$ (n is the number of recommendations that the systems needs to generate), and average it across all the users to compute $\overline{cover}$.

5.2 Subscribed Series Recommender

In this section, we present the evaluation of the subscribed series recommender aimed at recommending programs regularly watched by users. Since we conduct an offline evaluation, the important metric for subscribed recommendations is the precision. That is, we assess the portion of the recommended programs that were watched by the users.

We evaluate two personalised (preference-based and random profile) and two non-personalised (most popular and random) subscribed recommenders. We set the series subscription thresholds to $\alpha = 0.3$ and $\beta = 3$. These were determined by cross-validation experiments that are not reported due to space limitations. We included in the test set all the users, who watched ten programs or more during the three day test period. We found 9120 users satisfying this criterion. The number of recommendations n was gradually increased from 1 to 10. For each value of n, we averaged $\overline{prec}$ across the 9120 test users.

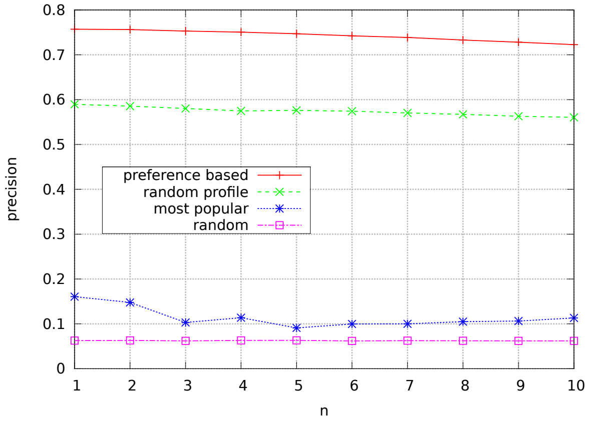


Figure 6: Precision vs. number of recommended series.

Figure 6 shows the average precision for various values of n. As expected, two personalised recommenders clearly outperform the baseline non-personalised ones. We also observe that preference-based recommender is superior to the random profile recommender: the former achieves precision of 0.75, whereas the latter hovers around the $0.55 - 0.6$ mark. Note that the obtained precision scores only slightly decrease with n, e.g., preference-based recommender drops from 0.75 for $n = 1$ to 0.72 for $n = 10$. That is, the two simple rule-based recommenders reliably generate accurate recommendations and obtain a high precision, and their performance is stable despite the growth of the recommendation list.

5.2.1 *User Profile Temporal Span*

One of the key questions in converting the observed viewing logs into user profiles refers to the time span of data considered by the conversion. On the one hand, taking all the available logs into account may lead to reliable profiles and recommendations. On the other hand, outdated logs may lead to imprecise profiles and hamper the accuracy of the recommendations. In this experiment we investigate the temporal aspect of the user profiling.

We evaluate two user profiling methods. The first is denoted as the *complete* profiling and it incorporates all the available user data. The second is the *4-week* profiling and it considers only the four weeks immediately preceding the test period. In both cases, we apply threshold values similar to those used in previous experiments in order determine user subscription to series. Then, we use the preference-based algorithm to generate subscribed series recommendations.

Figure 7 shows the precision and coverage of the recommendations for an increasing from $n = 1$ to $n = 10$ set of recommendations. We observe that precision of recommendations based on the 4-week profiles is superior to the precision of those based on the complete profiles. This is reasonable given that the 4-week profiles are centred on the recent user logs reflecting preferences in a time span close to the test period. However, the improvement in precision comes on the account of coverage. As the 4-week profiles are smaller and contain less subscribed series than the complete profiles, the system struggles to generate as many recommendations as with the complete profiles, and the coverage drops. In both cases, the coverage decreases with n, since

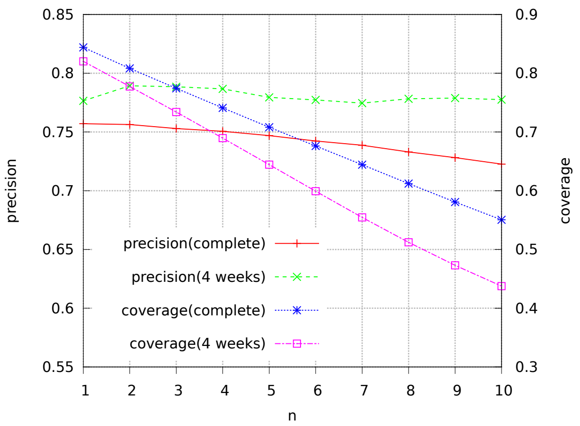


Figure 7: Precision and coverage (complete and 4-week profiles) vs. number of recommended series.

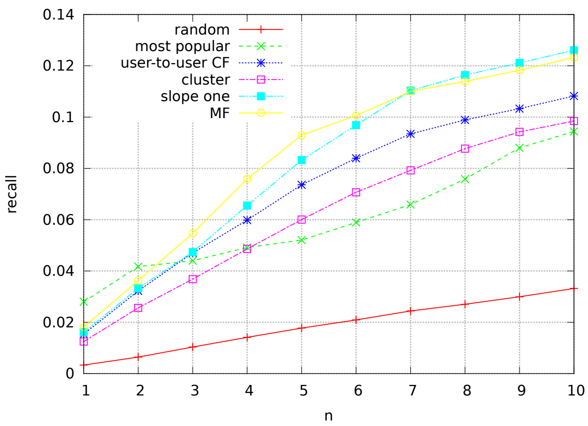


Figure 8: Recall vs. number of recommended series.

the task of generating the required number of recommendations aggravates as the recommendation list grows.

5.3 New Series Recommender

In this section, we present the evaluation of the new series recommender aimed at exposing users to new not yet watched series and virtually subscribing them to new content. Therefore, the evaluation methodology of new series recommender is fairly different. Treating one-off watching events of the recommended series as success indicators may yield a noisy ground truth. Instead, we treat a new subscription to a recommended series as a success. We use the information available beyond the test period to determine "future" subscriptions and measure the ability of the recommender to forecast new subscriptions made in the three day test period. The metric that is measured is recall, i.e., we are interested to assess the portion of new subscriptions that were suggested by the recommender.

We evaluate four personalised (user-to-user CF, cluster-based, slope one, and MF) and two non-personalised (most popular and random) unsubscribed series recommenders. The subscription thresholds are set to $\alpha = 0.3$ and $\beta = 3$, like in previous experiments. The number of similar users is set to $k = 500$ and Pearson's correlation is applied for the user-to-user CF. Clusters used by the clustering recommender are those identified in [17]. Slope one recommender weighs all the items uniformly. Finally, the learning and regularisation parameters of MF were optimised using cross-validation that are not reported due to space limitations.

We included in the test set all the users who subscribed to three new series or more during the three day test period, i.e., users who surpassed the subscription threshold of at least three series during the test period and kept watching these series later on. We found 2803 users satisfying this criterion. The number of recommendations was gradually increased from $n = 1$ to $n = 10$. For each value of n, we averaged $\overline{recall}$ across the 2803 test users.

Figure 8 shows the recall of the six recommenders for various values of n. We observe that from $n = 3$ to $n = 7$, the highest recall is achieved by the MF recommender. This is in line with other works that position MF as the state-of-the-art recommendation technique [12]. Note the high recall achieved by the slope one recommender, which is close to MF and even outperforms it for $n \geq 8$. This supports the findings of [8], which observed that the accuracy of slope one is comparable to that of more complex recommenders, despite its algorithmic simplicity.

The next two algorithms are, respectively, user-to-user CF and cluster-based recommenders. These two are memory-based algorithms and their accuracy is normally lower than of the model-based MF algorithm. Moreover, cluster-based recommender does not use user-to-user similarity scores and this explains its inferiority with respect to user-to-user CF. The lowest recall is achieved by the two non-personalised recommenders. Surprisingly, the top performer for $n \leqslant 2$ is the most popular recommender. This is due to the observed popularity surge of a single series, which attracted during the test period twice the number of views of the second most popular series, and accounted alone for more than 5% of new subscriptions. The most popular recommender identifies the surge and obtains high recall for low values of n.

Overall, the obtained recall scores were low, with the top algorithms achieving $\overline{prec} = 0.12$ for $n = 10$. This is in line with [19], which found that the accuracy of a recommender decreases substantially, if aiming to deliver novel or serendipitous recommendations. Recommendations for not yet subscribed series clearly fall into this category.

5.4 Hybrid Recommender

Finally, we evaluate the overall performance of a hybrid recommender encapsulating the two recommendation components. For each component, we deploy the best performing approach: preference-based for the subscribed and MF for the new recommendations. All the thresholds and parameters are set identically to the standalone experiments. Since we use mixed hybridisation and combine the outputs of the recommenders, each produces five recommendations to generate the final list of $n = 10$ recommendations.

We included in the test set all the users, who satisfy the two criteria of the standalone experiments. That is, the test set contains users, who watched ten programs or more and subscribed to three new series or more during the test period. We found 1907 users satisfying these two criteria. The evaluation metrics we use in this experiment are precision, $\overline{recall}$, and coverage. Overall, we obtained $\overline{prec} = 0.221$, $\overline{recall} = 0.227$, and $\overline{cover} = 0.888$. To analyse the performance of the recommender across various users, we sort the

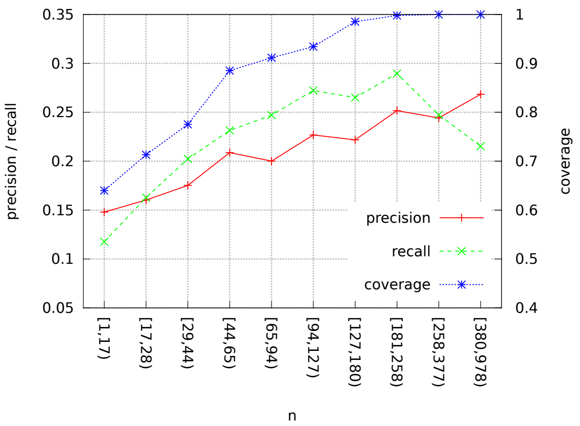


Figure 9: Recall, Precision, and Coverage vs. number of watched programs.

1907 users according to the number of programs watched during the training period (which strongly correlated with the number of programs watched during the test period) and split them into 10 equal-size buckets. We computed the $\overline{prec}$, $\overline{recall}$, and $\overline{cover}$ scores for each bucket.

Figure 9 shows the obtained $\overline{prec}$, $\overline{recall}$, and $\overline{cover}$. We observe that the precision increases with the number of programs watched. This is due to the fact that every watched program improves the accuracy of the user profiles, and, in turn, of the recommendations. An increase in recall is observed for the first eight buckets due to the same reason. However, the recall drops for the last two buckets. We posit that this happens due to the high number of programs watched by users in these buckets. As the number of recommended programs is fixed at 10 and the number of watched programs increases, the recall drops despite the improvement in the accuracy of the user profiles. The coverage of the recommender steadily increases with the number of watched programs, as the list of subscribed series grows and this alleviates the recommendation task.

5.5 Summary

We reported on the evaluation of several approaches for recommending subscribed and new series. Out of the subscribed series recommenders, the one that considers implicit user scores for the subscribed series was found to outperform others. Even a simple rule-based recommender achieved a high precision of 75%. As one may expect, recommending unsubscribed series was harder. We evaluated six recommendation approaches, including the state-of-the-art MF recommender, which achieved the highest recall of 12%. We hybridised the two recommenders and were able to achieve overall average precision and recall scores of around 22%, with a high average user-based coverage of 89%.

It is important to revisit the limitations of the gathered dataset. Note that we have neither reliable indication of the extent of viewing following a program playback nor explicit user feedback for the program. Clearly, obtaining feedback on user experience with the portal is pivotal for the success of catch-up TV recommendations. Finally, it is important to acknowledge the offline nature of our evaluation. In a live system, recommendations can inherently influence user choice, and some programs may receive additional eyeballs only due to the increased exposure through recommendations. Hence, the obtained results underestimate potential uptake of the recommendations, such that results of an online study may surpass those of our offline evaluation.

6. RELATED WORK

We briefly discuss related work pertaining to personalisation and recommendations in the TV domain. These can be grouped into three directions: TV viewer modelling, TV personalisation systems, and group personalisation.

Ardissono et al. [1] investigated the use of hybrid user modelling in the TV domain. Explicit and stereotypical models were combined to capture user preferences for programs. The evaluation showed that enriching the models using community preferences achieved better performance than traditional user modelling. Bellekens et al. [4] introduced the *iFanzy* system with advanced Semantic Web based user modelling capabilities. There, information was extracted from social networks, which resolved the cold start problem and improved the accuracy of the models of new users. Hopfgartner investigated the capture of long-term user interests in the news domain [11]. News items were categorised through their textual content and semantic context, which improved the accuracy of the user models.

O'Sullivan et al. [15] studied the business value of TV personalisation. The *PTVPlus* recommender system they developed applied association rule mining and case-based reasoning methods, and outperformed traditional collaborative filtering recommenders. Zimmerman et al. [20] developed the *Touch and Drag* system, which combined a TV recommender with a usable interface. A user study that was conducted demonstrated the effectiveness of the recommender system. More recent work by Bambini et al. [3] presented a recommender system for *Fastweb*, one of the largest European IPTV providers. The implemented recommender exploited content-based and collaborative filtering techniques and it was observed that up to 30% of recommendations led to the purchasing behaviour.

Masthoff studied how one could cater to the needs of a group of TV viewers [14]. Three different aspects of aggregation of user preferences and the impact of intra-group relationships were investigated. It was shown that aggregated user profiles yielded the highest satisfaction and that users valued the fairness factor. Besides, it was shown that group opinion on a program might fluctuate over time and be affected by other watched programs. Finally, Zhang et al proposed an approach for identifying individual TV viewers behind composite group profiles using subspace clustering [18]. As the number of users behind a composite profile and their preferences were identified, the accuracy of the generated recommendations increased.

Although prior works studied user modelling and personalisation in the TV domain, to the best of our knowledge, none have focussed on the catch-up TV application. Catch-up TV services are still new and a robust catch-up TV recommender requires a dedicated investigation. The fact that in our case the TV programs are grouped, offers an additional domain knowledge that can be incorporated into the recommendation process and differentiates our work from earlier works. To the best of our knowledge, our paper presents the first practical investigation and comparison of recommendation approaches for a catch-up TV.

7. CONCLUSIONS

The convergence of the Social Web and IPTV has exposed users to enormous volumes of content, aggravating the discovery of content of interest. We analysed real-life Australia-wide six month logs of a national catch-up TV provider, which showed the overwhelming volume of the available video content and the choice problem faced by the users. This motivated our work on personalised recommendations that can help the users discover content.

In this work, we evaluated a recommender system for catch-up TV. The recommender, which uses past user interactions to generate TV program recommendations, includes two components. The first recommends subscribed content that the users regularly watch, whereas the second recommends new content that is likely to be of interest for them. We conducted an offline evaluation of several recommendation algorithms and were able to select the best performing algorithms for the two components.

One of the shortcomings of the gathered dataset lies in the unreliable user identification. It is not uncommon for a group of users, e.g., a family, to use the same device to acess the catch-up portal, or, alternatively, for one user to use multiple devices, e.g., computer and tablet. In the future, we aim to develop approaches for identifying composite and duplicate user profiles and evaluate their impact on the accuracy of the generated recommendations.

We will also investigate recommendations for groups of users. TV watching often occurs in groups, e.g., with family or friends, and the recommender should cater for the preferences of the group as a whole and deliver group-based recommendations. Finally, we will investigate various machine learning approaches that can improve the accuracy of the recommendations, and, particularly, of recommendations for new unsubscribed content. We aim to deploy the resulting recommender system into the service offered by our partner broadcaster and their Web-based catch-up portal, and evaluate the performance of the recommender in a longitudinal large scale user study [2].

8. REFERENCES

[1] L. Ardissono, C. Gena, P. Torasso, F. Bellifemine, A. Difino, and B. Negro. User modeling and recommendation techniques for personalized electronic program guides. In *Personalized Digital Television*, pages 3–26. Springer, 2004.

[2] S. Ardon, S. Bensiali, H. Cinis, J. Wang, and S. Berkovsky. Next-generation social TV content discovery. In *International Conference on User Modeling, Adaptation, and Personalization*, 2012.

[3] R. Bambini, P. Cremonesi, and R. Turrin. A recommender system for an IPTV service provider: a real large-scale production environment. In *Recommender Systems Handbook*, pages 299–331. Springer, 2011.

[4] P. Bellekens, G. Houben, L. Aroyo, K. Schaap, and A. Kaptein. User model elicitation and enrichment for context-sensitive personalization in a multiplatform TV environment. In *Proceedings of the European Conference on Interactive TV*, pages 119–128, 2009.

[5] S. Berkovsky. Decentralized mediation of user models for a better personalization. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 404–408. Springer, 2006.

[6] R. Bernhaupt, M. Boutonnet, B. Gatellier, Y. Gimenez, C. Pouchepanadin, and L. Souiba. A set of recommendations for the control of IPTV-systems via smart phones based on the understanding of users practices and needs. In *Proceedings of the European Conference on Interactive TV*, pages 143–152, 2012.

[7] R. Burke. Hybrid recommender systems: survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[8] F. Cacheda, V. Carneiro, D. Fernandez, and V. Formoso. Comparison of collaborative filtering algorithms: limitations of current techniques and proposals for scalable, high- performance recommender systems. *ACM Transcactions on Web*, 5(1):2, 2011.

[9] A. Erlandsson, N. Ronblom, and A. Ericson. TV and video 2011 consumer trends, global version. Technical report, Ericsson ConsumerLab, 2011.

[10] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the SIGIR Conference on Research and Development in Information Retrieval*, pages 230–237, 1999.

[11] F. Hopfgartner. Capturing long-term user interests in online television news programs. *TV Content Analysis: Techniques and Applications*, page 309, 2012.

[12] Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[13] D. Lemire and A. Maclachlan. Slope one predictors for online rating-based collaborative filtering. *Society for Industrial Mathematics*, 5:471–480, 2005.

[14] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction*, 14(1):37–85, Feb. 2004.

[15] D. O'Sullivan, B. Smyth, D. Wilson, K. Mc Donald, and A. Smeaton. Interactive television personalization. In *Personalized Digital Television*, pages 73–91. Springer, 2004.

[16] G. Shani and A. Gunawardana. Evaluating recommendation systems. *Recommender Systems Handbook*, pages 257–297, 2011.

[17] M. Xu, S. Berkovsky, I. Koprinska, S. Ardon, and K. Yacef. Time dependency in TV viewer clustering. In *TVM2P Workshop, International Conference on User Modeling, Adaptation, and Personalization*, 2012.

[18] A. Zhang, N. Fawaz, S. Ioannidis, and A. Montanari. Guess who rated this movie: identifying users through subspace clustering. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, pages 944–953, 2012.

[19] Y. C. Zhang, D. O. Séaghdha, D. Quercia, and T. Jambor. Auralist: introducing serendipity into music recommendation. In *Proceedings of the International Conference on Web Search and Data Mining*, pages 13–22, 2012.

[20] J. Zimmerman, K. Kauapati, A. Buczak, D. Schaffer, S. Gutta, and J. Martino. TV personalization system. In *Personalized Digital Television*, pages 27–51. Springer, 2004.