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

Catch-up TV is revolutionising viewing habits of audiences, as it provides users the opportunity to watch programs at their preferred time and place. With broadcasters offering content through their catch-up TV portals, there is a need for personalised recommendation solutions that help users to select programs of interest. In this work, we study the watching patterns of users of a popular Australian nationwide TV service provider and evaluate a suite of approaches for catch-up TV recommendations. We compare these approaches using a new dataset gathered through the provider’s catch-up TV service. The evaluation allows us to assess the performance of several recommenders that address both the discovery of TV programs already known by users and of new programs they may find relevant.

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 at their own convenience. Several TV networks, such as the ABC and SBS in Australia, and the BBC in the UK, offer catch-up services, and this fast emerging content delivery paradigm is an important means 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 [3].

With the flexibility of watching any program at any time, 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 search for a suitable program [9]. Hence, there is an emergent need for personalised solutions and recommender systems capable of selecting TV 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 one plot or theme, e.g., TV series and football games, or programs shown at regular days or time intervals, e.g., daily news and weekly talk shows. For the sake of simplicity, we refer to the groups of content as series and to the individual programs belonging to the series as episodes. We disregard 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 consumption of the subscribed content. The second component, which we refer to as the new series recommender, recommends content that a user has not been watching, i.e., new programs. This component capitalises on the preferences of the entire community of users and builds upon collaborative recommendation methods. The goal of this recommender is to expand the users horizons and them to new TV content. Finally, the recommendations generated by the two components are combined using the mixed hybridisation [6].

We present an offline evaluation of the above recommendation components using a dataset gathered by our partner TV network. The dataset contains the whole of 6 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 compare the performance of several recommendation approaches and assess their ability to recommend subscribed content of in-
terest, as well as new unwatched content. The results of the evaluation show that we can recommend subscribed content with a high degree of accuracy, such that around 75% of the recommended programs are watched by users. New TV content recommendations are, however, much more challenging and the performance of the new recommender is poorer. We evaluate several recommendation techniques, but they reach the accuracy level of 12% at the best. 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 recently broadcast TV programs at their preferred time. Catch-up TV is usually provided via a Web portal and accessible through various devices: PCs, tablets, smartphones, and gaming consoles. A key difference with respect to traditional video-on-demand services is that catch-up services make content available for a limited period following the program broadcast (depends on the licensing negotiated with the content right holders). The popularity of catch-up TV has recently sky-rocketed and its consumption figures are approaching those of the traditional TV [8]. Similar to other video-on-demand services, catch-up TV providers are looking to increase user satisfaction, user engagement, and content consumption. To achieve this, many catch-up TV providers are considering the use of personalisation and recommendation approaches that facilitate presentation of programs tailored to the users’ interests and preferences.

For this study, we partner with a leading national TV broadcaster in Australia. Each program shown on the TV channels of the broadcaster is made available at the Web-based catch-up portal, typically a day after the broadcast. Programs remain available in the 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 service can be accessed through a variety of devices and platforms, e.g., Apple iOS, smart TVs, and Web browsers.

The main interface of the portal is currently non-personalised. It offers to viewers three lists of programs: two time-based lists (‘just added’ and ‘about to expire’), and one editorially-curated list (‘featured’). Note that the selection of programs for 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, lookup programs through channels on which they were broadcast, or use a free-text search.

Once a user finds a program and clicks its thumbnail, detailed program information is shown. This information includes the program title, category, textual description, broadcast data, publication date (uploaded to the portal), expiry date (removed from the portal), parental rating, and duration. 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 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 social network post, bypassing the main or search interface of the portal.

3. OVERVIEW OF DATASET

3.1 Data Collection

As already mentioned, the data used in this paper originates from the catch-up TV service of a leading Australian broadcaster that runs a number of national free-to-air channels. 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 11,099 unique programs.

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time span</td>
<td>184 days</td>
</tr>
<tr>
<td>View count</td>
<td>19.8 millions</td>
</tr>
<tr>
<td>Unique videos viewed/available</td>
<td>9.114/27.621</td>
</tr>
<tr>
<td>Unique cookie</td>
<td>2.02 millions</td>
</tr>
</tbody>
</table>

Table 1: Dataset high-level description

The gathered data includes very little information about the users, as the catch-up service does not include any sign-on facility. 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, or users can 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.

The logs contain only the observed watching events and do not include any evidence of searches, information accesses, social media posts, and so on. For each video that was viewed, the logs contain the cookie ID, the date, and the video ID only. 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 contains a unique video ID, a series ID, the video title, duration, publication date, expiry date, category (chosen from the above 12 categories), and the size of the video file.

3.2 High-Level Characterisation

In this section, we present a high-level characterisation of the dataset. Figure (a) shows the number of video
accesses per week observed during the data collection period. We observe a slow increase in the video accesses 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, i.e., cookie IDs, seen each week, which steadily hovers around the 180,000 users/week mark. Figure 1 (c) shows the cumulative distribution of the duration of the available videos. More than 30% of videos are shorter than 20 minutes, about 45% are 20 to 30 minutes long, and only 2% are longer than 1 hour. These results resemble the typical length of TV programs shown by the broadcaster.

Figure 2 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 does not contain information on the duration of each video playback. These results assume that each video was watched in its entirety, and, hence, it is an upper bound. Nonetheless, it is observed that, on average, users viewed content between 150 and 180 minutes per day, with viewing time increasing on weekends. Given that a typical video is 20 to 30 minutes long, this represents an engagement of 5 to 6 videos per day. Since on any day, the catch-up TV service catalog contains between 1000 and 1200 videos, these results highlight the content discovery problem: it is a challenge for users to select the right program out of the large number of available programs.

Related to this analysis is the typical user stickiness, i.e., the frequency with which users return to the portal. Among the unique users seen each week, Figure 3 shows the breakdown of total views into those made by users that requested only one video, between two and nine videos, and 10 videos or more. This analysis shows that about 40% of users are one-timers, who view only one video and do not return to the service any more. We also observe that only 10% of users view 10 videos or more over the course of 6 months. It appears that a large fraction of users are accessing the service exactly as the service 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. Clearly, the frequent users account for a large fraction of requests, although the infrequent users are also significant contributors to the total load of the portal. In the remainder of the paper, we present a recommender system focusing on the more frequent users.

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 obtained from the catch-up portal encapsulate binary viewing ratings $r_{u,s}$:

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

The number of available programs is much greater than the average number of programs watched by a user. 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.

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

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

where $n_{u,s} = \sum_{i \subseteq s} r_{u,i}$ denotes the number of episodes of $s$ that $u$ watched and $n_{av}$ denotes the number of episodes of $s$ that were available since $u$ had joined. The joining date of $u$ 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 scores of $u$ for all $s \in S$ series, $P_u = \{score_{u,s}\}$, and the resulting rating matrix $R$ represents the collection of all user profiles.

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

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

If the implicit score $score_{u,s}$ of $u$ for $s$ is greater than $\alpha$ and $u$ watched more than $\beta$ episodes of $s$, then $u$ is considered subscribed to $s$. The relative threshold $\alpha$ corresponds to the series score as a proxy for the level of user’s interest. The absolute threshold $\beta$ corresponds to the minimal 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. That is, the output of the recommender is restricted to $s \in SL_u$. Note that this differs from typical recommendation scenarios, 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, here the goal is to recommend regularly watched series, whose score is known.

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 to $u$. The second, dubbed as the 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 as baselines for comparison. These consider neither the profile $P_u$ nor the subscription list $SL_u$ when generating recommendations. The first, most popular, recommends $n$ series with the highest aggregated score across the entire community of users. The aggregated score for a series $s$ is computed by averaging $score_{u,s}$, of all the users subscribed 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 simplifies the discovery of familiar content, but inherently limits the discovery of new content, since 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 $s \not\in SL_u$ 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 is based on the idea that users will like items that were liked by similar-minded users [9]. Hence, CF initially computes the degree of similarity between the target user $u$ and every other user in the system. Then, for every series $s$ yet unwatched by $u$, the recommender (1) selects a set of $k$ most similar users subscribed to $s$, and (2) computes the predicted score of $u$ for $s$ as a weighted average of the scores of these $k$ users. The relative weight of users reflects the degree of their similarity to the target user. Finally, $n$ series with the highest predicted score 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 [10]. 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 [12]. 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 $s_u$ and $s_v$ 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 $score_{u,s_\ell}$ 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 $R$ [11]. The algorithm factorises $R$ into a product of a user latent matrix $p$ and a series latent matrix $q$ using the alternative least squares approach $(p$ and $q$ are iteratively optimised individually rather than in parallel). Upon the completion of the factorisation, the latent matrices are multiplied to predict
the score $\text{pred}_u,s$ for yet unwatched series. Again, $n$ series with the highest score are finally recommended to $u$.

Also for the new series recommendation task we use two non-personalised baselines: most popular and random. These are analogous to their non-personalised subscribed recommendation counterparts, but rather pick the recommended series from the set of series to which $u$ is not subscribed.

### 4.4 Combining Recommendations

We would like to stress the different purposes of the subscribed and new series recommenders. The former comes to suggest 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 they should be combined and presented to users. According to Burke’s taxonomy of hybrid recommenders [6], this combination fits the mixed hybridisation method, 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 viewed programs are available, we cannot use predictive accuracy and use the classification accuracy metrics of precision and recall [13]. 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 $\text{Rec}$ and watched a set of programs $\text{Viewed}$ over the course of the test period, precision and recall for $u$ are computed as

$$\text{prec}_u = \frac{|\text{Viewed} \cap \text{Rec}|}{|\text{Rec}|} \quad \text{recall}_u = \frac{|\text{Viewed} \cap \text{Rec}|}{|\text{Viewed}|}$$

Finally, we compute the average precision, $\text{prec}$, and average recall, $\text{recall}$, 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, $\text{cover}_u = |\text{Rec}|/n$ ($n$ is the number of recommendations that the systems needs to generate), and average it across all the users to compute $\text{cover}$.

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 $\text{prec}$ across the 9120 test users.

Figure 4 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

![Figure 4: Precision as a function of the number of recommended series.](image-url)
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 5 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 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 [16]. 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 recall across the 2803 test users.

Figure 6 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 [3]. 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 [7], 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 \( n < 2 \) 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 performing MF and slope one algorithms achieving only \( \text{MRE} = 0.12 \) for \( n = 10 \). This is in line with findings of [18], which concluded 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

Figure 5: Precision and coverage (complete and 4-week profiles) as a function of the number of recommended series.

Figure 6: Recall as a functions of the Number of Recommended Series.
Finally, we evaluate the overall performance of a hybrid recommender encapsulating both the subscribed and new series 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 algorithm parameters are set identically to the above standalone experiments. Since we use mixed hybridisation and combine the outputs of the two recommenders, each one produces live recommendations to generate the final recommendation list of size $n = 10$.

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, recall, and coverage. Overall, we obtained $\text{precision} = 0.221$, $\text{recall} = 0.227$, and $\text{coverage} = 0.888$. To analyse the performance variability of the recommender across various users, we sort the 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 $\text{precision}$, $\text{recall}$, and $\text{coverage}$ scores for each bucket.

Figure 7 shows $\text{precision}$, $\text{recall}$, and $\text{coverage}$ obtained for various buckets. We observe that the precision increases with the number of programs watched. This is due to the fact that every watched program adds some information, and 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 during the test period. As the number of recommended programs is fixed at 10 and the number of watched programs increases, the recall drop 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 gathered dataset and refresh its limitations. Note that we have no reliable indication of the extent of viewing following a program playback. Also, note that we do not have any explicit user feedback for the programs. Clearly, obtaining user feedback on their 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 precision and recall underestimate potential uptake of the recommendations, such that results of an online study may surpass those obtained in the 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. [2] 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 [10]. News items were categorised through their textual content and semantic context, which improved the accuracy of the user models.

O’Sullivan et al. [14] 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. [19] 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. [2] 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 for the needs of a group of TV viewers [13]. 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

![Figure 7: Recall, Precision, and Coverage as a function of the Number of Watched Programs in the Training Set](image-url)
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. As the number of users behind a composite profile and their preferences were identified, the accuracy of the generated recommendations increased.

Although prior works have studied user modelling and personalisation in the TV domain, to the best of our knowledge none have focussed on the catch-up TV application as yet. Catch-up TV services are still relatively new and a robust catch-up TV recommender requires a separate investigation and evaluation. 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 scenario.

7. CONCLUSIONS

The convergence of the Web, 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 IPTV provider, which showed the overwhelming volume of the available video content. This motivated our work on personalised recommendations that can help 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 access 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.

8. REFERENCES


