Real-time Personalized Ranking and Recommendation System for Linear TV: A Dual Dynamic Queue Approach

Ning Xu, Ph.D.

Adeia Inc., ning.xu@adeia.com

Tao Chen, Ph.D.

Adeia Inc., tao.chen@adeia.com

Written for presentation at the

SMPTE 2022 Media and Technology Summitt

Abstract. *The increasing number of channels on linear TV has amplified the challenge of efficient content discovery for viewers. This paper presents an innovative, real-time personalized ranking and recommendation system designed to address this problem. The objective was to enhance viewer satisfaction and interaction through a personalized, dynamic channel surfing experience. Our approach uses a dual dynamic queue system - the Dynamic History Channel Queue (DHCQ) and the Dynamic Future Channel Queue (DFCQ), each serving a unique purpose in managing the viewer's channel interaction.*

Advanced deep learning models reinforce these queues' dynamism by generating 'global' and 'local' content embeddings and 'user' embeddings. These embeddings are used to provide real-time updates, considering both timestamps and the time-sensitivities of the content and viewer. A 'look-ahead' feature was integrated to account for future content on each channel, adding another layer of personalization.

Preliminary user feedback highlighted a strong interest in this new form of personalized channel surfing, with a majority of respondents indicating a preference for our system over traditional channel navigation methods. The results show that our proposed solution could potentially change the way users interact with the ever-growing number of channels on linear TV.

Keywords. Linear TV, Channel Ranking, Channel Surfing, Real-time Personalization, Dynamic Queues, User Preferences, Personalized Content Delivery.

(The SMPTE disclaimer is on a footer on this page, and will show in Print Preview or Page Layout view.)

Introduction

In an era of digital transformation, traditional linear television is undergoing rapid changes, one of which is the surge in the number of channels available to viewers. According to a report by Nielsen in 2016, the average American adult had access to more than 200 TV channels [1], a significant increase from the past. This explosion of channels often leads viewers to grapple with an overwhelming amount of content, finding it increasingly challenging to navigate through it and discover programming that caters to their preferences. Traditional browsing methods, such as manual channel flipping and program guide scanning, often lead to a suboptimal viewer experience due to their time-consuming and inefficient nature. Moreover, the increased demand for personalized content has rendered these methods insufficient. Thus, there is an urgent need to rethink the way viewers interact with linear TV and enhance their channel-surfing experience.

Over the past decade, the media landscape has witnessed the rise of short-video platforms and streaming services that have successfully adopted personalized ranking and recommendation systems. Utilizing collaborative filtering, content-based filtering, and advanced deep learning techniques, these systems thoroughly analyze user behavior, preferences, consumption patterns, and interactions with diverse content. This analysis not only formulates a dynamic, personalized ranking of recommendations but also ensures users receive real-time adjustments, greatly enhancing user engagement and satisfaction. This personalization has contributed significantly to the success of these platforms. However, the translation of such systems to the real-time, continuously broadcasted world of linear television presents unique challenges and opportunities.

This paper introduces a new, real-time system that personalizes channel recommendations for viewers as they surf through linear television channels. Our system relies on two dynamic queues—the Dynamic History Channel Queue (DHCQ) and the Dynamic Future Channel Queue (DFCQ)—alongside advanced deep learning models. The DHCQ keeps track of the channels a viewer has recently browsed where the content is still playing, while the DFCQ organizes and continually updates the order of all other channels based on real-time changes in their personalized ranking scores.

Furthermore, our system calculates personalized scores using 'global' and 'local' embeddings from the video content, along with 'user' embeddings derived from viewers' past interactions and preferences. A special 'look-ahead' feature is also incorporated, which takes into account not just the current content but also what's coming up next. This way, viewers get a peek into future programs, adding an extra touch of personalization.

The remainder of this paper is organized as follows: Initially, we explain how our proposed channel surfing method interacts with dynamic queues. Following that, we delve into the dynamic mechanisms of the DHCQ and DFCQ, elucidating the core features of our proposed system including the application of deep learning models. Subsequently, we introduce the personalized ranking and recommendation algorithm employed in our system, explaining how personalized ranking scores are computed. Afterward, we discuss the incorporation of 'look-ahead' features in linear TV channel surfing, Finally, we present survey results, discussing the potential impact of our system on viewer engagement and its implications for personalized content delivery strategies across various media platforms, and conclude our findings with a discussion on the implications and potential future developments of our proposed system.

Channel Operations with Dynamic Queues

Our real-time personalized ranking and recommendation system for linear TV channel surfing is built on a novel approach involving two dynamic queues, i.e., the Dynamic History Channel Queue (DHCQ) and the Dynamic Future Channel Queue (DFCQ).

The DHCQ is designed to keep track of the viewer's recently browsed channels, thereby enabling the viewer to navigate channels back and forth with ease. Unlike static video surfing histories found in short-video platforms, the DHCQ refreshes every time a new program starts on a channel in the queue. The DHCQ is therefore not merely a static list but rather a real-time reflection of the viewer's browsing behavior.

The DFCQ, on the other hand, hosts all other channels. Each channel in this queue is assigned a personalized ranking score calculated using the viewer's profile and preferences. This score determines the channel's position in the queue. Similar to the DHCQ, the DFCQ is also dynamic. It continuously reorders channels based on real-time adjustments in their personalized ranking scores. This ensures a more engaging and personalized viewer experience compared to static linear TV line-ups.

We first describe how the channel change operation looks like given a fixed DHCQ and DFCQ. Figure 1 shows when the DHCQ has channels available for channel up or channel down, how the current channel changes in the DHCQ. In this case, the channel change operates similarly to the regular channel history browsing. Figure 2 shows when the current channel is the last channel of DHCQ. In this case, there is no additional channel in DHCQ for channel-up operation, so the first channel in DFCQ will be removed from the DFCQ and be appended to the end of DHCQ. The channel-up operation will result in showing this newly appended channel.

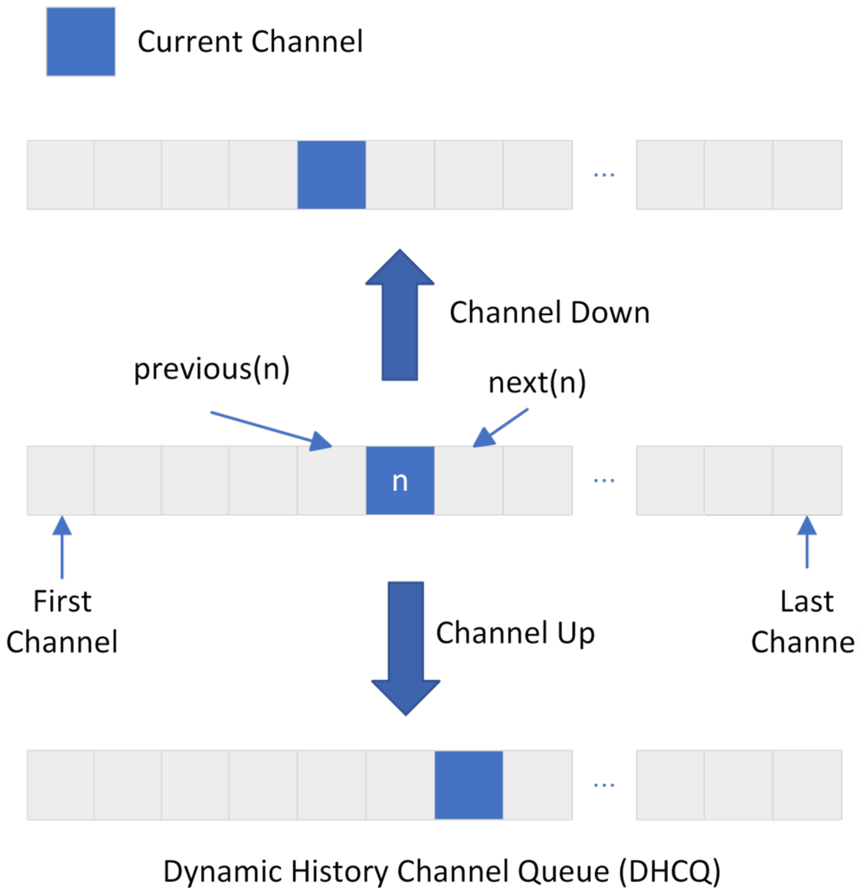

Figure 1. Channel up and channel down with DHCQ when there are available channels.

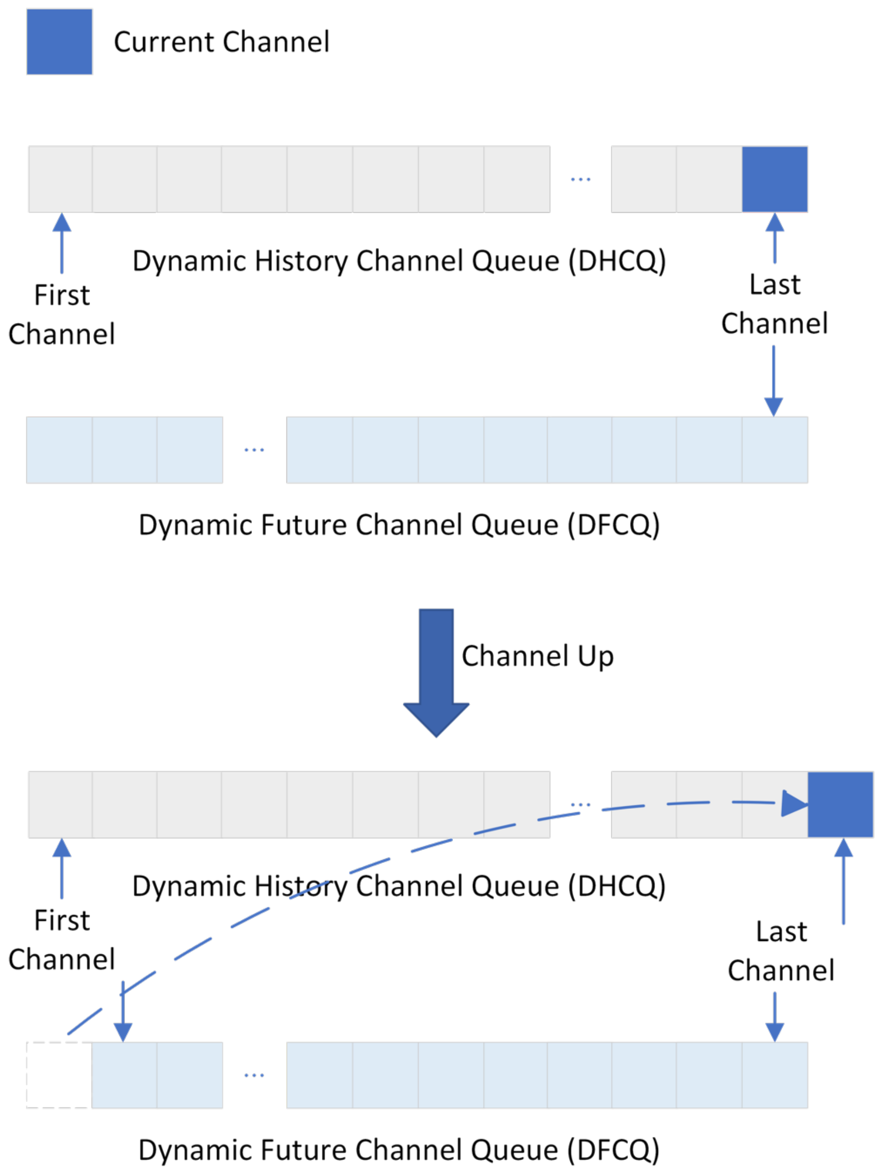

Figure 2. Channel up when the current channel is the last channel in DHCQ. In this case, the first channel of DFCQ will be appended to the DHCQ.

Dynamic mechanism of the DHCQ and DFCQ:

The following elaborates on the initial setup and the subsequent dynamic updating of the Dynamic History Channel Queue (DHCQ) and the Dynamic Future Channel Queue (DFCQ).

Initialization:

To initiate the system, we first set the DHCQ as an empty queue. Concurrently, all channels at time t=0 are ranked using an algorithm, which will be elaborated on later. These channels are then positioned in the DFCQ in descending order based on their ranking scores. The channel with the highest score is placed at the top of the DFCQ. It is then moved from the DFCQ to the DHCQ, becoming the current channel in the DHCQ. Consequently, the viewer's television experience starts with this channel.

Dynamic Updates:

Dynamic updates to the DHCQ and DFCQ are achieved via a set of procedures:

- Periodic Re-ranking: At a pre-set time interval, denoted as dT_1, all channels in the current DFCQ are re-evaluated and re-ranked, followed by sorting them in descending order based on their scores.
- DHCQ Refresh: For each channel in the DHCQ (excluding the current channel being watched), the system records the end time of the current recommended program when they are added to the DHCQ. When each of these programs reaches its end time, the corresponding channel gets removed from the DHCQ. A new ranking score for this channel is then computed, and the channel is placed back in the DFCQ based on this score.
- User Inactivity Management: For every channel in the DHCQ, the last time a viewer visited this channel is recorded. If a channel in the DHCQ is not visited within the time period defined by dT_2, the system removes this channel from the DHCQ. Subsequently, a new ranking score for this channel is calculated, and the channel is placed back into the DFCQ based on its ranking score.
- Channel Up Navigation: When the viewer is watching the last channel in the DHCQ and presses channel-up, the top-ranked channel in the DFCQ is moved to the DHCQ as the new last channel which is also the current channel. If the DFCQ is empty (i.e., all channels are in DHCQ), the system cycles back to the first channel in the DHCQ.
- Channel Down Navigation: Similarly, if the viewer is watching the first channel in the DHCQ and presses channel-down, the top-ranked channel in the DFCQ is inserted as the new first channel in the DHCQ and becomes the current channel. If the DFCQ is empty, the system moves the viewer to the last channel in the DHCQ.

This continuous dynamic management of the two queues ensures a smooth and engaging channel surfing experience for the viewer.

Personalized Ranking and Recommendation Algorithm

In order to implement personalized ranking and recommendations for linear TV channels, various techniques can be incorporated into the system, for example:

- Collaborative Filtering [2] relies on historical user interactions (likes, views, etc.) to recommend videos to similar users.

- Content-based Filtering [3] leverages the attributes of videos (titles, tags, descriptions, etc.) as well as the user's past interactions to offer relevant suggestions.
- Hybrid Recommender Systems [4] combine collaborative and content-based filtering techniques. By considering both video attributes and the user's past interactions, these systems make more accurate recommendations.
- Deep-learning Based Methods [5, 6] employ deep-learning models to detect patterns in user behavior and video data, subsequently making recommendations based on these learned patterns.

In this paper, deep learning models are utilized to compute the embeddings of both the content and the viewers. The proposed system combines the progression of time, the details of the content, and the viewer's preferences to recommend the most suitable content on linear TV at any given moment.

Global and Local Timestamps

Linear TV's dynamic nature means that the content changes over time. Therefore, a successful recommendation system must accurately consider time as a critical parameter. In our model, we've utilized two types of timestamps to address this.

The global timestamp T_g is a universal clock, marking the current time across all channels. For each TV channel, denoted by n, there is specific content being broadcasted, which we denote as $V(T_g, n)$. However, to measure how far along a video content has progressed on a particular channel, we use the local timestamp, $T_l(n)$. Essentially, $T_l(n)$ is calculated by subtracting the content's start time, denoted by global timestamp, from the current global timestamp T_g.

Segmentation of Video Content

For video content pieces longer than a threshold duration T_s , such as a lengthy movie, we split this content into smaller segments, each not exceeding the predefined duration threshold T_s . This ensures that each segment remains under the set duration T_s . A straightforward video segmentation method can be applied by cutting the video into segments where all of them except the last segment have a duration of T_s, while the last segment has a duration shorter than T_s. More sophisticated video segmentation algorithms can also be applied; for example, using scene change detection to cut the video into scenes, and from the beginning of the video, start to aggregate the video scenes until the combined duration approaches T_s. Or, if the initial scene is already longer than T_s, cut the scene into shorter pieces as in the simple method aforementioned.

Feature Embedding via Deep Learning

Deep learning models are crucial for video and user understanding. In our system, two key video embeddings are created for every channel's content: the Global Embedding E_g and the Local Embedding E_l.

The Global Embedding $E_g$is a vector representation of features extracted from the entire video content of $V(T_g, n)$. It considers attributes like titles, tags, descriptions, and other pertinent metadata. On the other hand, the Local Embedding E_l is more detailed, focusing on specific segments of $V(T_g, n)$. Beyond the broader attributes, E_l further delves into nuances like video frames or subtleties derived from audio or captions of that segment. This dual embedding mechanism ensures a balance between general content themes and moment-specific insights.

In parallel, to cater to the user-centric facet of the recommendation system, a User Embedding E_u is generated. This User Embedding captures a spectrum of user behaviors and preferences, consolidating historical interactions and explicitly declared preferences.

Sensitivity Analysis

Beyond content and user preferences, the system recognizes the importance of the sensitivity associated with both the video content and the user's viewing habits. The Video Sensitivity S_v is influenced largely by the genre of the content. A live event, for instance, might evoke a different viewing urgency compared to a recurring documentary.

Complementing this is the User Sensitivity S_u , which is primarily user-specific. Users can adjust this sensitivity based on their viewing preferences. For example, a lower sensitivity might suggest a user's willingness to join a program midway, whereas a high sensitivity would indicate a preference for starting content from the beginning.

Ranking Algorithm

With the above foundations established, the core of the system is the ranking algorithm. For every video content $V(T_g, n)$, the system computes a composite score, $Score_V$ with a machine learning model that considers various parameters: the timestamps T_g and $T_l(n)$, video embeddings E_g and E_l, user embedding E_u, and the sensitivity parameters S_v and S_u. This comprehensive scoring mechanism evaluates the relevance of the content on channel n for a specific user at the timestamp T_g.

Consideration of Look-Ahead Features in Linear TV

The system is not limited to the present. It's forward-looking, taking into account what's coming next. If an upcoming program $V_{next}(T_g, n)$ is slated to begin shortly, specifically within a duration T_{next}, our system factors it into the recommendation pool. The idea here is not just to recommend what's currently airing, but also what will soon be on air, should it align with the viewer's interests.

The look-ahead factor plays a pivotal role in the personalization of linear TV. It's not uncommon for an upcoming movie or program on a channel to yield a significantly higher ranking score than the currently played content on the same channel. For this reason, our system doesn't merely calculate the ranking score of a single program $V(T_g, n)$, but also considers the upcoming video program $V_{next}(T_g, n)$. This is particularly the case when the upcoming program is scheduled to commence within the time period T_{next}. The ranking algorithm will be used to calculate the ranking score for this upcoming video as well, and the higher score of the upcoming video and the current video will be used as the score of this channel.

Should a channel be chosen as the current channel based on the score of the incoming video, the TV will display this channel. Furthermore, an on-screen notification will provide information about the next-up program along with a countdown. This ensures that the user understands why this specific channel is being recommended. If the user is intrigued by the proposed video program, they have the option to either wait on this channel or browse other channels and return later when the recommended program starts.

Survey Results

To understand the acceptance and potential adoption of our proposed channel surfing technique, we surveyed 25 participants from diverse backgrounds in terms of age and gender. This diversity was crucial in order to gauge a comprehensive view of how different users might perceive and value the added functionality.

Preference Over Traditional Channel Surfing

The participants were first introduced to the new channel surfing technique and its intricacies. Upon understanding its nuances, they were asked about their preference between the conventional method and our proposed system. A significant 68% of the respondents indicated their preference for the new system. This inclination towards the proposed approach suggests that a majority of users are open to, or even prefer, our proposed channel surfing methods that cater to their preferences and content consumption habits.

Likelihood of Adoption

To further quantify the acceptance level, participants were asked to rate their likelihood of adopting the proposed method over traditional channel surfing on a scale from 1 (least likely) to 10 (most likely). The average score, calculated from all responses, is 7.48. This score, being significantly above the mid-point, further validates the inclination seen in the preference question and indicates a substantial potential for widespread adoption.

Conclusion

In this paper, we introduced a novel approach to channel surfing in linear television, harnessing the power of advanced deep learning techniques and real-time personalization. The intricate design, which integrates Dynamic History Channel Queue (DHCQ) and Dynamic Future Channel Queue (DFCQ) mechanisms, aims to redefine the way users interact with their television sets. This method brings the precision and adaptability of modern streaming recommendation systems to the age-old platform of linear TV.

A critical cornerstone of our system is the personalization algorithm that makes use of both global and local timestamps. This consideration ensures users are presented with content that's not just aligned with their preferences but is also timely and contextually relevant. The subsequent segmentation of longer programs and the forward-looking features, such as anticipating upcoming high-interest programs, further enhance user engagement and satisfaction.

The survey results suggest a promising acceptance level for the proposed channel surfing technique among a diverse group of participants. While traditional methods have their foundation and familiarity, there seems to be a notable openness towards methods that use intelligent mechanisms, such as our proposal, to enhance the user experience in linear TV consumption. The likelihood score further reinforces this sentiment, presenting an optimistic scenario for the system's real-world adoption and success. The inherent flexibility of our approach, where users can seamlessly toggle between the traditional method and our enhanced technique, further positions it as an attractive and user-friendly proposition.

References

[1] Nanji, Ayaz. How Many TV Channels Do People Actually Watch? https://www.marketingprofs.com/charts/2016/30973/how-many-tv-channels-do-people-actually-watch

[2] Ajitsaria, Abhinav. Build a recommendation-engiine with collaborative filtering. https://realpython.com/build-recommendation-engine-collaborative-filtering/

[3] Lee, Joonseok, Nisarg Kothari, and Paul Natsev. “Content-based related video recommendations.” Advances in Neural Information Processing Systems (NIPS) Demonstration Track (2016): 2.

[4] Verma, Yugesh. A Guide to Building Hybrid Recommendation Systems for Beginners , https://analyticsindiamag.com/a-guide-to-building-hybrid-recommendation-systems-for-beginners/

[5] Covington, Paul, Jay Adams, and Emre Sargin. "Deep neural networks for youtube recommendations." Proceedings of the 10th ACM conference on recommender systems. 2016.

[6] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending what video to watch next: a multitask ranking system. In Proceedings of the 13th ACM Conference on Recommender Systems (RecSys’19)

