# Optimizing Video Recommender System with Bandit-Based Ensemble from Online User Action

**Minseop Lee, Sanghyeon Lee, Yeonghwan Jeon, Hyuncheol Jo, Byoung-Ki Jeon**
LG Uplus Corp.
oper0815@lguplus.co.kr, shlee145@lguplus.co.kr, yeonghwan@lguplus.co.kr, hcjo@lguplus.co.kr, bkjeon@lguplus.co.kr

**Abstract**

In the modern era of video content platforms like YouTube and Netflix, personalized recommendations are essential. This study introduces an ensemble recommendation model that considers unique platform characteristics. We segment recommendation scenarios and fine-tune models based on real user actions, resulting in personalized recommendation lists. Our contributions include developing recommendation models for each characteristic, the use of the Multi-Armed Bandit (MAB) algorithm for ensemble each model with user feedback, and improved video recommendations with enhanced user satisfaction in real-world services.

## Introduction

The modern era offers a plethora of services capable of delivering a wide array of video content, as exemplified by platforms such as YouTube (Davidson et al. 2010) and Netflix (Steck 2018). Given the extensive diversity of videos available, it becomes crucial to tailor recommendations to individual users' tastes. Each service employs its unique methodology for video recommendation, which may include suggestions based on similar viewing patterns, trending content, or sequential consumption. The efficacy of these recommender systems hinges on the insightful analysis of user viewing history, interactive behaviors, and video metadata, which collectively aid in understanding and predicting user preferences. Such systems are instrumental in presenting users with video selections that are contextually relevant and personally appealing.

In this research, we propose an ensemble recommendation model that not only accommodates but also capitalizes on the distinct characteristics of each service. By segmenting the context of video recommendations, we have devised models that resonate with each specific aspect of user preference. These models are fine-tuned based on real user actions, allowing us to formulate a recommendation list that is optimally weighted according to the user's unique disposition.

In summary, our contributions are as follows:

- The development of various characteristic recommendation models, combined in an ensemble tailored to the user's profile.
- Implementation of the MAB algorithm in our ensemble method, facilitating immediate integration of user feedback.
- Optimization of a personalized recommendation list for each user, utilizing the MAB-based ensemble approach.
- Application of our methodology to actual services, demonstrating improvements in both offline and online performance metrics in the realm of video recommendations. This includes a notable enhancement in user satisfaction indicators.

## Related Work

This study examines the application of ensemble MAB techniques in video recommendation systems, focusing on their adaptive content diversity to potentially enhance customer satisfaction. Considering the variable nature of user preferences, this research area is notably significant. Our review includes key themes such as bias mitigation and diverse methodologies in video recommender systems (Bottou et al. 2023, Christakopoulou et al. 2013, Gruson et al. 2019, Joachims et al. 2017, Schnabel et al. 2016), the impact of content diversity on platform models (Anderson et al. 2020, Abdollahpouri, Burke, and Mobasher 2019, Kaminskas and Bridge 2016, Liu et al. 2021, and Rahman and Oh 2018), the integration of ensemble algorithms (Burke 2002, Ma and Wang 2017, Kanakia et al. 2019, Brodén et al. 2019), and the influence of online user feedback on system performance (Li et al. 2010 and Sohail, Siddiqui, and Ali 2014). In-depth discussions on these topics are available in the appendix.

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

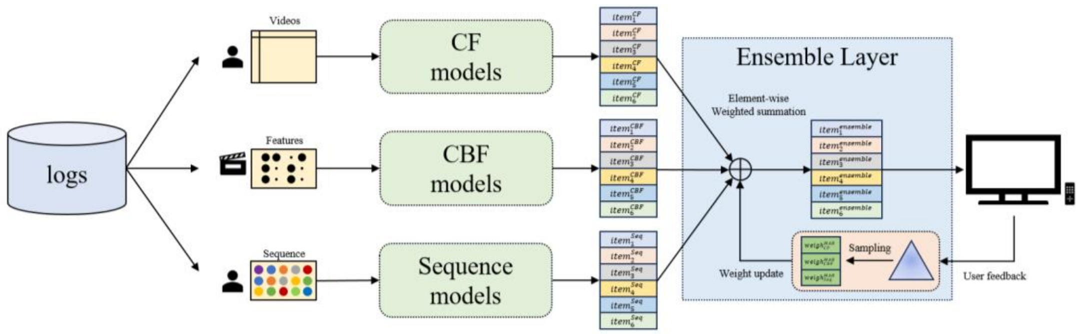


Figure 1: Illustration of Bandit-based ensemble framework

## Datasets

There are usually two types of datasets used in video recommendations: explicit feedback data with clear ratings and implicit feedback data, which is difficult to know whether user prefers items or not. It is important to understand the data in recommendations because the pre-processing method and model used are different depending on what characteristics and distribution the data has. In this paper, among the video data, the video dataset of IPTV is used which is serviced by LG Uplus in Korea.

For this research, we have utilized a dataset from IPTV services offered by LG Uplus in South Korea. As a result, it contains about 100 million interactions with 29,177 videos and 2,252,946 users. Detailed information about the dataset, including its structure and specific interaction data, is provided in the appendix of this paper.

## Methods

Our methodology, illustrated in Figure 1, capitalizes on the diversity of models in recommendation systems, reflecting the multifaceted nature of user preferences and behaviors. We integrate various models into an ensemble architecture, aiming to capture a broader spectrum of user interactions. This ensemble is dynamically refined through a bandit-based approach, which continuously updates model weights based on real-time user data.

### Bandit Ensemble

In this chapter, a detailed ensemble recommendation method is provided. Since the recommendation is not a classification task, we present an ensemble method from the perspective of the recommendation service and describes how to update it based on MAB to optimize the ensemble.

### Recommend Rank based Ensemble

In a recommendation system based on batch inference, a method of Voting-based ensemble of predicted recommendation lists in each model is often used instead of model ensemble to achieve ensemble modeling with minimal computational cost. The first important consideration in voting-based ensembles is that recommendation lists based on model-specific characteristics are exposed. When utilizing a Strong Voting-based ensemble approach, there is an advantage in excluding outlier recommendation lists, as items predicted multiple times by various models are more likely to be recommended. However, in the case of strong voting ensembles, there is a disadvantage that a list recommended based on the unique characteristics of the model can be excluded because items that come out a lot from model predicting usually come out. To address this issue, weighted voting ensemble is utilized, a form of soft voting, which mitigates the drawbacks of strong voting by assigning weights to models.

Another important aspect of ensemble methods is the handling of prediction scores. In recommendation tasks, which involve ranking rather than straightforward classification, disparities in the scale of prediction scores between models can diminish the ensemble's effectiveness. Therefore, understanding and adjusting these score scales is crucial. Our proposed ensemble method, a rank-based weighted ensemble, addresses this issue. It assigns new scores based on the reciprocal of the ranking in each model's recommendation list, elevating the importance of top-ranked items in the final ensemble recommendations while diminishing the impact of lower-ranked lists. Among the various ways to generate ensemble scores, including sequential scores, the performance of the proposed method is the best.

This methodology is particularly advantageous when dealing with mixed user histories, a common scenario in our IPTV data, which aggregates histories based on customer IDs, often representing multiple household members. Our ensemble approach, which combines various models like CF-based, sequence, and content-based models, is designed to accommodate the varying interaction patterns of different users. This diversity in the ensemble allows for a broader coverage of user preferences, especially in situations where the identity of the active user is uncertain. By utilizing a diverse set of top-ranking lists from various models, the ensemble is better equipped to cater to a mixed user base.

Finally, the effectiveness of the ensemble is maximized by adjusting the weights of each model based on their newly generated rank scores. To facilitate understanding, we provide pseudocode through Algorithm 1 in Appendix, outlining the entire process of our ensemble method.

### Bandit based Ensemble Approach

In this section, we approach the determination of optimal ensemble weights from the perspective of "What ensemble weight would be the most effective?" As mentioned earlier, the ensemble requires weight values, and the simplest approach is to define them based on offline metrics. However, offline tests do not always align with online tests, and the optimal weight values can change over time. Therefore, we propose a method of updating weights online through MAB based on user feedback.

The MAB method we propose involves utilizing N weights and updating them using Thompson sampling based on user Click Through Rate (CTR) which refers to the ratio of entry into the content details for an impression. While traditional Thompson sampling is known for updating bandits using a beta distribution, determining the arm's ratio is necessary for ensembles. This requirement leads to the need for a multivariate distribution, where the sum of all weights is 1, allowing for exploration and exploitation. To address this, we employ the Dirichlet distribution. The Dirichlet distribution is a generalization of the beta distribution, and when k equals 2 in the Dirichlet distribution, it is equivalent to the beta distribution.

After several tests, we recognize the importance of the initial values of the Dirichlet distribution. Initially, it is crucial to minimize opportunity costs and providing appropriate initial values to the distribution allows for a rapid convergence of weight values. In order to set the initial distribution, N weights corresponding to N models (arm) are set to be the largest, one for each N trial, and the alpha value of the largest arm based on the user feedback is set to be slightly larger than that of other arms. Through this approach, we achieve a reduction in opportunity costs and a faster convergence speed compared to uniformly setting alpha values.

Another crucial point to consider is the alpha parameter update in the reward update part of Dirichlet distribution. We introduce three variables for updating alpha values. The first variable is the ctr weight, which is applied from the previous batch time. If the feedback (ctr) is lower than baseline, we use the value of the opposite index of weight. The second variable is the index score, which is generated based on rank values of ensemble weight. If the feedback is lower than baseline, we use the inverse rank value. The weight is more meaningful when the ctr difference is more than ±4% of the baseline, and we assign a parameter based on this. Lastly, a scaling parameter is needed to reflect the service characteristics. The alpha of the corresponding batch time is obtained by adding the product of the above three parameters and the previous alpha value. Finally, weights are obtained by sampling from Dirichlet distribution with the updated alpha.

The Algorithm 2 in Appendix D is a detailed description of how to adjust the ensemble weight based on the mab, and the case where there are three arms is described.

## Experiments

We evaluate our proposed method on LG Uplus IPTV dataset by analyzing multiple metrics. First, we would like to mention the directionality of the evaluation. Afterward, we conduct an evaluation to determine if it aligns with the mentioned directionality by answering four research questions. We demonstrate the utility of ensembles through a comparison between ensembles and individual models in offline experiments (RQ1). Next, we verify whether the system effectively induces user interest in online scenarios from an accuracy perspective (RQ 2). Following that, we explain from a diversity perspective whether the recommendation system adequately reflects various preferences (RQ3). Finally, we evaluate overall user satisfaction (RQ4).

### Evaluation Direction of Recommendation System

Recommendation system evaluation is split into offline and online assessments. While offline and online metrics may sometimes be correlated, this relationship can vary depending on the characteristics of the data and the service. Therefore, it is essential to consider both metrics comprehensively. Additionally, since both evaluation methods involve multiple metrics, a comprehensive analysis is necessary.

Our first evaluative direction is aimed at examining various metrics, with focusing on accuracy and diversity among the multiple metrics. The reason is that good recommendations are not solely determined by accuracy, and recommending a diverse set of items does not guarantee excellence either. Ideally, what we consider as an ideal recommendation is a scenario where a wide variety of items is selected by many users. The second evaluative direction is focused on online testing. Online evaluation is inherently crucial in recommendations, and since MAB allows for online evaluation, we validate our model through online testing.

### RQ 1. Does Ensemble Model Improve Recommendation Performance?

To demonstrate whether the ensemble has improved recommendation performance, we compare its performance with that of a single model. Furthermore, since we are proposing an ensemble method, we establish its effectiveness through a comparison with strong voting ensemble. Table 1 presents the results of the offline tests encompassing these aspects. Bert4rec is included as a single model with the highest overall metrics among those used in the ensemble. AWS Personalize obtained through the AWS console which is based on Hierarchical RNN (HRNN) model. Ensemble(vote) denotes metrics extracted from an ensemble based on strong voting, while Ensemble(rank) refers to the proposed method that generates predict scores based on ranking and then ensembles them. In the experiments, all ensemble-based models adjusted the ensemble weights for each model to the same ratio.

We observe that some metrics show a decrease after the voting ensemble compared to the single model. However, there is an increase in coverage, indicating the ensemble effect. This outcome can be attributed to the fact that the strong voting ensemble is not based on a separate score criterion. In situations where the voting scores are identical, it appears that an arbitrary lower-ranked list is recommended after the ensemble, resulting in slightly lower metrics. The results of proposed ensemble method based on rank score, on the other hand, demonstrate higher metric values compared to the single model and the strong voting ensemble. In particular, the notable increase in coverage without a decline in accuracy metrics shows that ensemble method is efficient.

| | P@25 | Recall@25 | MRR@25 | Cov@25 |
|---|---|---|---|---|
| Ensemble(rank) | 0.0298 | 0.7445 | 0.5165 | 0.4551 |
| Ensemble(voting) | 0.0238 | 0.5952 | 0.2499 | 0.3118 |
| Bert4rec | 0.0286 | 0.714 | 0.5133 | 0.2418 |
| Personalize | 0.0284 | - | 0.2561 | 0.1862 |

Table 1: Comparison of results from ensemble and single model with the Precision (P@25), Recall, MRR and Coverage (Cov@25)

## RQ 2. Can Ensemble MAB Engage User Interest?

In this section, we evaluate whether updating the ensemble with MAB induces more user interest from an accuracy perspective. While there are various accuracy-related metrics, we use CTR, commonly used in recommendations. CTR indicates how many clicks occurred per impression and is less influenced by the recommendation position, making it suitable for accuracy measurement. In the experiment, a click refers to a user pressing the video in the carousel, signifying entry into the detailed page. Experimental results show that after applying MAB to the ensemble, CTR increases by approximately 4%. The comparison here refers to the case where the ensemble is rank-based and weights are assigned in equal proportions. Consequently, updating the ensemble with MAB leads to an increase in recommendation accuracy and, consequently, a greater engagement of user interest.

## RQ 3. Can Ensemble MAB Reflect Diverse User Preferences?

Providing recommendations that reflect users' diverse preferences and lead to a greater variety of consumed videos is ideal, rather than recommending predominantly popular videos and consuming them. In this section, to evaluate this aspect, we address the coverage of videos that users have watched. Video coverage refers to the unique videos for which watching has occurred. Experimental results demonstrate a 6% increase in coverage after applying MAB to the ensemble. This confirms that users have consumed a diverse range of content, contributing to long-term user retention by encouraging users to consume a variety of content.

## RQ 4. Does Ensemble MAB Increase User Satisfaction?

Satisfaction with recommendations should be evaluated through various metrics, we will evaluate them through online indicators other than CTR and Coverage previously evaluated. For instance, it is challenging to answer questions about whether it is satisfactory from a user satisfaction perspective if the transition to watching does not occur after entering the detailed page of a video. The result of increasing user satisfaction is not only increasing CTR but also increasing conversion rate (CVR). CVR refers to the ratio from viewing content details to watching content. In this experiment, to demonstrate this, we address various metrics such as CVR, watching page views (PV), and other indicators. The experimental results indicate an increase in all online metrics, including CVR, contributing to an overall improvement in user satisfaction. Finally, we present a comparison of the overall online metrics before and after applying MAB in Table 2. As evidenced by the overall increase in metrics, it can be inferred that our proposed model optimizes the ensemble weights based on MAB. The update history of the Dirichlet parameters of alpha and ensemble weights used in the experiment is provided in the appendix.

| Method | CTR | CVR | PV | UV | Coverage |
|---|---|---|---|---|---|
| Ensemble (same weight) | X1 | X1 | X1 | X1 | X1 |
| Ensemble (applying mab) | X1.04 | X1.05 | X1.21 | X1.21 | X1.06 |
| Ratio | ▲4% | ▲5% | ▲21% | ▲21% | ▲6% |

Table 2: Comparison of results from online metric after applying mab with ensemble

## Future Works

In this study, we have introduced a method for optimizing the ensemble of several recommendation models using MAB techniques within the context of Video Recommender Services (IPTV). In our future work, we aim to enhance our recommendation models by incorporating real-time user feedback and real-time contextual information through contextual bandit-based methods. Furthermore, we intend to advance our approach by leveraging causal information to make recommendations that consider user interests and popularity trends.

## References


Abdollahpouri, H.; Burke, R.; and Mobasher, B. 2019. Managing popularity bias in recommender systems with personalized reranking. *arXiv preprint. arXiv:1901.07555*.

Anderson, A.; Maystre, L.; Anderson, I.; Mehrotra, R.; and Lalmas, M. 2020. Algorithmic effects on the diversity of consumption on spotify. In *Proceedings of the web conference.* 2155-2165.

Bottou, L.; Peters, J.; Quiñonero-Candela, J.; Charles, D.; Chickering, D.; Portugaly, E.; Ray, D.; Simard, P.; and Snelson, E. 2013. Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising. *The Journal of Machine Learning Research.* 14(11).

Brodén, B.; Hammar, M.; Nilsson, B. J.; and Paraschakis, D. 2019. A bandit-based ensemble framework for exploration/exploitation of diverse recommendation components: an experimental study within e-commerce. *ACM Transactions on Interactive Intelligent Systems (TiiS).* 10(1). 1-32.

Burke, R. 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction.* 12. 331-370.

Christakopoulou, K.; Lalama, A.; Adams, C.; Qu, I.; Amir, Y.; Chucri, S.; Vollucci, P.; Soldo, F.; Bseiso, D.; Scodel, S.; Dixon, L.; Chi, E.; and Chen, M. 2023. Large Language Models for User Interest Journeys. *arXiv preprint. arXiv:2305.15498*.

Davidson, J.; Liebald, B.; Liu, J.; Nandy, P.; Van Vleet, T.; Gargi, U.; Gupta, S.; He, Y.; Lambert, M.; Livingston, B; and Sampath, D. 2010. The YouTube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems.* 293-296.

Gruson, A.; Chandar, P.; Charbuillet, C.; McInerney, J.; Hansen, S.; Tardieu, D.; and Carterette, B. 2019. Offline evaluation to make decisions about playlist recommendation algorithms. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining.* 420-428.

Joachims, T.; Swaminathan, A.; and Schnabel, T. (2017, February). Unbiased learning-to-rank with biased feedback. In *Proceedings of the tenth ACM international conference on web search and data mining.* 781-789.

Kaminskas, M.; and Bridge, D. 2016. Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems (TiiS),* 7(1), 1-42.

Kanakia, A.; Shen, Z.; Eide, D.; and Wang, K. 2019. A scalable hybrid research paper recommender system for microsoft academic. In *The world wide web conference.* 2893-2899.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web.* 661-670.

Liu, Y.; Shen, Z.; Zhang, Y.; and Cui, L. 2021. Diversity-promoting deep reinforcement learning for interactive recommendation. In *5th International Conference on Crowd Science and Engineering.* 132-139.

Rahman, M.; and Oh, J. C. 2018. Graph bandit for diverse user coverage in online recommendation. *Applied Intelligence.* 48. 1979-1995.

Ma, L.; and Wang, X. 2017. An ensemble algorithm used in video recommendation system. In *Proceedings of the 1st International Conference on Big Data Research.* 67-71.

Schnabel, T.; Swaminathan, A.; Singh, A.; Chandak, N.; and Joachims, T. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *international conference on machine learning.* 48: 1670-1679.

Sohail, S. S.; Siddiqui, J.; and Ali, R. 2014. User feedback scoring and evaluation of a product recommendation system. In *2014 seventh international conference on contemporary computing (ic3).* IEEE. 525-530.

Steck, H. 2018. Calibrated recommendations. In *Proceedings of the 12th ACM conference on recommender systems.* 154-162.

# Appendix

## Appendix A: Literature Review

The YouTube designed the recommender system (Davidson et al. 2010) using various viewing histories and demographic data of YouTube users. The Netflix designed calibrated recommendations (Steck 2018) provided not only accuracy, but also recommendations considering diversity. Google uses Large Language Models (Christakopoulou et al. 2023) to express customer exploration intentions in a natural language to increase the reliability of recommendations that improves the video recommender system from a variety of perspectives.

This study examines the application of ensemble Multi-Armed Bandit (MAB) techniques in video recommendation systems. The focus is on assessing how these techniques adaptively provide a diverse range of content, with potential implications for enhancing customer satisfaction. Given the variable nature of user preferences in video content consumption, this area warrants significant research attention.

### Video Recommender Systems: Approaches to Bias Mitigation and Model Perspectives

In the realm of video content services, user engagement is predominantly for entertainment (Christakopoulou et al. 2023). The decision-making process in content selection by users is influenced by an array of factors ranging from personal interests to societal trends. This often leads to a concentration on popular content, causing a 'long-tail' effect where a few items gain substantial popularity, overshadowing a majority. The studies explore the use of Inverse Propensity Scoring (IPS) as a strategy to mitigate this, aiming to balance the prominence given to popular items and address data scarcity resulting from popularity (Bottou et al. 2013, Gruson et al. 2019, Joachims et al. 2017, Schnabel et al. 2016). However, this method has limitations in recognizing users' explicit preferences, potentially underestimating those who enjoy a mix of popular and less popular content.

### Exploring Content Diversity in Recommendation Systems and Its Impact on Platform Business Models

The pursuit of diversity in content recommendation is a vibrant field of study across various service domains (Anderson et al. 2020, Abdollahpouri et al. 2019, Kaminskas et al. 2016, Liu et al. 2021). For instance, research conducted by Spotify underscores the importance of diversifying content recommendations. It reveals that a higher diversity score in recommendations can improve long-term user engagement across both niche audiences and the general user base (Abdollahpouri et al. 2019). Another approach focuses on post-processing techniques, incorporating popularity de-biasing in the re-ranking process to enhance diversity(Kaminskas et al. 2016). However, this method often faces challenges in striking a balance between popularity and diversity. Advanced techniques, such as reinforcement learning policies that account for both diversity and item relevance, are emerging as promising solutions to these challenges(Liu et al. 2021). On the other hand, offering a wide range of products with extensive coverage is crucial for both users and customers in platform businesses (Rahman et al. 2018). Sustaining a diverse content exposure not only stimulates user engagement but also enhances the visibility of various contents, which in turn encourages more active participation from content providers. For example, there are various services such as YouTube and Netflix. The videos offered in these services are very diverse, so it is very important to recommend videos that suit your taste. In fact, each service recommends videos in various ways. The video recommender system allows users to find videos that suit their personal preferences. In recommender system, the user's viewing history, user actions, and video metadata are very important for understanding user preferences. With this system, users can be provided suited videos in various contexts.

### The Emergence of Ensemble Algorithms in Recom-mender Systems

In the field of recommender systems, hybrid models have emerged as a solution to the limitations inherent in singular recommendation methods. Collaborative filtering (CF), with techniques like Neural Collaborative Filtering (NCF) and ALS (Alternating Least Squares), excels in analyzing user interactions, yet it struggles with the 'cold start' problem for new users or items. In contrast, content-based filtering (CBF), using tools such as TF-IDF (Term Frequency-Inverse Document Frequency and NLP techniques, offers personalized recommendations based on content analysis but often lacks diversity in its suggestions. To address these challenges, hybrid systems blend the strengths of CF and CBF (Burke 2002). For example, weighted ensembles can integrate the robust user-item interaction analysis of NCF with the detailed content understanding from decision trees. This synergistic approach aims to enhance recommendation accuracy while ensuring a broader variety of content. Current research is focused on fine-tuning these hybrid systems for greater adaptability and precision. The evolution of these sophisticated hybrid methodologies is a key trend in the advancement of recommender systems (Ma and Wang 2017, Kanakia et al. 2019, Brodén et al. 2019).

### Enhancing Recommender Systems through Online User Feedback

Recent advancements in recommender systems have seen a growing focus on leveraging real-time online user feedback for model optimization. Studies demonstrate the effective use of user interactions, such as content clicks and engagement patterns, as contextual information. This data is then employed to enhance the accuracy and relevance of recommendations, thereby maximizing user engagement metrics like click-through rates (Li et al. 2010). Another innovative approach involves the construction of feedback vectors from user service statistics, which are then utilized to re-rank recommendation outputs, aligning them more closely with user preferences (Sohail et al. 2014).

## Appendix B: Data Source

The historical data used in this paper is log data that occurs in the Set-Top Box of IPTV. About 350 million logs are generated per day, including video browsing history, video watching history, dislike, search, real-time channel, remote control action log, and more. Uplus IPTV data is basically a household ID unit, which is a set of family members, and is called a Customer ID. Users can create profiles if they want, which can be used to manage their personal history. When creating a profile, the user makes a preference choice, so that the cold start can be resolved. Finally, the tower box log data has a coexistence of household unit history and profile history. Video metadata consists of general video information (genre, director, keyword, actor, etc.) and detailed metadata. Video detailed metadata is produced directly by Uplus as unique information of the content.

To create the entire IPTV log into train data, watch history and initial preference evaluation data among user logs are used. The watching history includes all the user's history of watching after searching, the history of watching as a recommendation, and the history of watching after purchasing. In addition, video impression data which refers to exposure to a user is also processed for MAB feedback. The history data including payment amount and watching time consists of six months of video history to provide recommendations to as many people as possible. As a result, it contains about 100 million interactions with 226,127 videos and 2,252,946 users.

## Appendix C: Base Models

In the realm of recommendation systems, the heterogeneity of models mirrors the complex tapestry of user preferences and viewing behaviors. Each model, with its unique architectural design, interprets user interaction patterns differently, thereby contributing distinctively to the recommendation output.

**Collaborative Filtering Model**. It utilizes dimensionality reduction techniques to learn latent variables from extensive user-item interactions. This approach is adept at inferring user preferences, even in scenarios lacking direct user-item interactions, by generating interaction scores that predict user likings. Its strength lies in offering relevant suggestions across diverse user interests, particularly effective in environments dominated by popular content.

**Content-based Model**. This model, through a detailed analysis of item attributes, effectively addresses the 'cold start' problem for new users. Capable of integrating multimodal metadata, it allows for detailed item profiling. It aligns item features such as genre, director, and cast with users' established preferences and behavioral patterns, thereby ensuring effective recommendations for users with limited interaction data.

**Sequence-based Model**. It focuses on learning the sequential nature of item consumption. It is tailored for users inclined towards series or trending content, considering viewing order and patterns to enhance the accuracy of its recommendations. By analyzing a user's viewing history in a chronological order, this model is proficient in predicting future viewing choices, adapting to changes in user preferences over time.

Each of these models is strategically designed to target specific user behaviors and preferences. The ensemble of these diverse models paves the way for a more nuanced and satisfying recommendation system. This system not only refines the granularity of recommendations but also significantly elevates customer satisfaction by offering a varied range of content that aligns closely with personalized user experiences.

## Appendix D: Algorithm Pseudo Codes

This appendix provides the details of the key algorithms developed in this research in the form of pseudocode. Algorithm 1 corresponds to the rank-based ensemble method introduced in the Methods section of the main text. Algorithm 2 delineates the bandit-based ensemble update technique.

Algorithm 1: Rank based Ensemble

**Input**: Recommendation lists for $U$ users ($M$ models)
**Parameter**: Ensemble weights $w \in W$
**Output**: Ensemble recommendation lists
1: **for** each user $u \in U$ **do**
2: **for** each recommendation list of $M$ models **do**
3: Calculate the rank of item $i$ in the recommendation list based on prediction score
4: Calculate the rank_score as the inverse value of rank:
$rank\_score_i = 1/rank_i$
5: **end for**
6: **for** each item $i$ in recommendation list **do**
7: Calculate the ensemble score:
$ensemble\_score = \sum_{m=1}^{M} w_m * rank\_score_i$
8: **end for**
9: Generate the final recommendation list based on the $ensemble\_score$ for each item
9: **end for**
10: **return** Final recommendation list

Algorithm 2: Bandit based updating ensemble weights

**Input**: CTR, Baseline CTR
**Output**: alpha $[\alpha_1, \alpha_2, \alpha_3] \in \alpha_t$, (Dirichlet distribution), weights $[w_1, w_2, w_3] \in W_t$
1: **Step 0: INIT**
2: Initialize parameters: $W, \alpha$
3: Set initial $W$ with the biggest ensemble weight for each model(bandit) during initialization batches (3 days), e.g., $W_{init} = ([0.5, 0.25, 0.25], [0.25, 0.5, 0.25], [0.25, 0.25, 0.5])$
4: Evaluate CTR for 3 days and set the initial alpha with bigger n-th value (largest ensemble weight's order when it is the largest CTR), e.g., $a_0 = [7, 5, 5]$ *or* $[5, 7, 5]$ *or* $[5, 5, 7]$
5: **for** every batch $t = 1, 2, \ldots, T$ **do**
6: **Step 1**: Evaluate $CTR_t$
7: **Step 2:** $ctr\_weight$ **based on** $(CTR_t - baseCTR)$
8: **if** $(CTR_t - base\ CTR) \geq 0$ **then**
9: $ctr\_weight = W_{t-1}$
10: **else**
11: $ctr\_weight = reverse\_index(W_{t-1})$, e.g., $W = [0.7, 0.1, 0.2]$ then $reverse\_index(W) = [0.1, 0.7, 0.2]$
12: **end if**
13: **Step 3:** $index\_score$ **based on** $|CTR_t - baseCTR|$
14: **if** $|CTR_t - baseCTR| > 0.04 * baseCTR$ **then**
15: $index\_score = rank\_based\_score(W_{t-1})$
e.g., $W = [0.5, 0.2, 0.3]$ then $rank\_based\_score(W) = [3, 1, 2]$
16: **else**
17: $index\_score = [1, 1, 1]$
18: **end if**
19: **Step 4: Update** $alpha$
20: $\alpha_t = \alpha_{t-1} + ctr\_weight * index\_score * scale$
($scale$ is a parameter for adjusting the update rate value, default $scale = 2$)
21: **Step 5: Update** $weights$ of ensemble
22: $sampling\ weights$ from Dirichlet distribution based on updated alpha:
$[w_1, w_2, w_3] \in W_t \sim Dirichlet(\alpha_1, \alpha_2, \alpha_3) \in \alpha_t$
23: **end for**

## Appendix E: Parameters

Figure 2 represents the updated Dirichlet parameters of alpha from the experiment, while Figure 3 depicts the weight of ensemble which is extracted from the alpha. In all figures, the x-axis represents the batch cycle (days). Through this, it is evident that the ensemble weights converge to their optimal values as the alpha values are updated. Experimentally, the weights tend to show a tendency after 20 days, and they show convergence after 60 days.

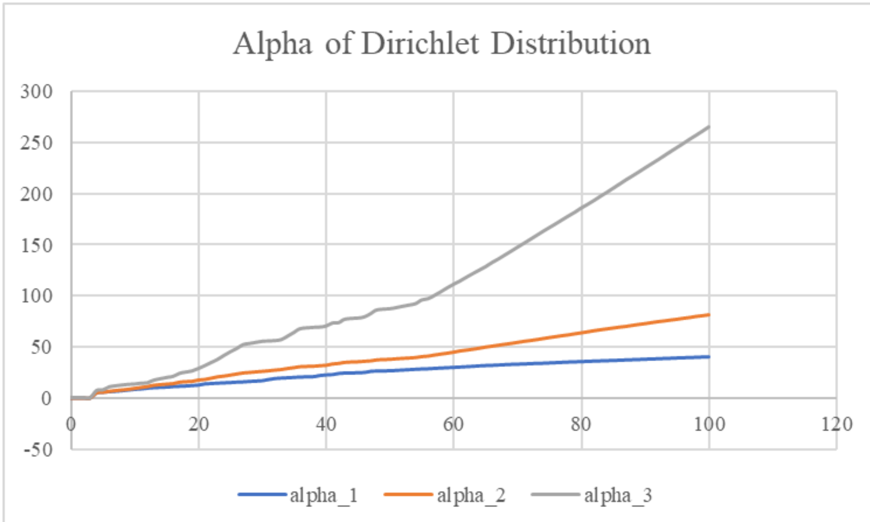


Figure 2: Dynamic parameters updated over time based on user feedback.

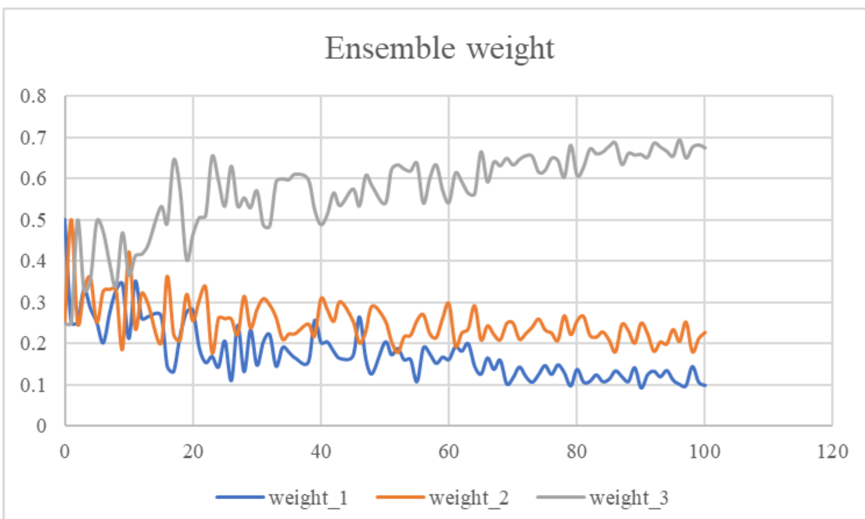


Figure 3: Optimization of Ensemble weights for IPTV service from Dirichlet distribution