Review of Recommending What Video to Watch Next: A Multitask Ranking System

Reviewer: Zhaokuan Chen zc56

Article Reviewed:

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). Association for Computing Machinery, New York, NY, USA, 43–51. https://doi.org/10.1145/3298689.3346997 [1]

Background

The authors mention that there are two challenges in the youtube recommendation system. The first is that using user-log to build training data can lead to the emergence of various biases, such as position-bias, selection-bias, feedback-loop-bias. The second challenge is that there are many variants of feedback-data in the system, such as clicks, watches, likes, etc. It is challenging to combine and use all the metrics efficiently. As shown in Figure 1, to tackle the first challenge, the authors implement a Wide & Deep framework that uses a separate model structure (‘shallow’ tower) to handle bias [1]. To deal with the second challenge, the authors use the Multi-gate Mixture-of-Experts (MMoE) to handle a variety of tasks [1].

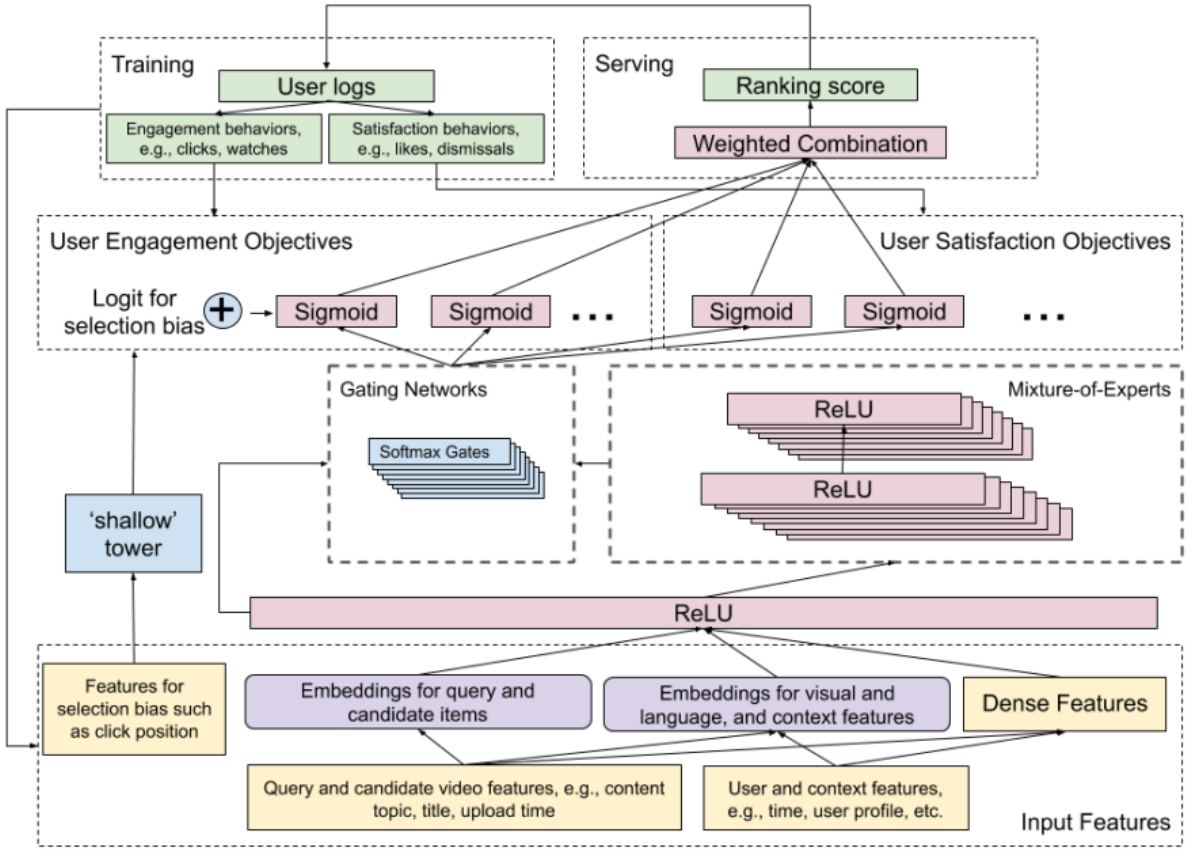

Figure 1

Problem Description

YouTube is mainly aimed at the video recommendation scenario. The problem that needs to be dealt with is to recommend the next video that the user might be interested in. The overall recommendation framework adopts candidate generation and ranking. In the phase of candidate generation, the author will use a variety of methods to obtain a variety of candidate videos, such as video topic-based methods, collaborative filtering (what videos are generally watched by other users who have watched this video), etc [1]. Because the article is mainly aimed at improving the ranking stage, the topic generation phase is not described. In the ranking phase, the system needs to provide a ranked video list.

Model Architecture

This part of the author aims to improve the objective function and model structure for the two challenges mentioned in the background.

1. Ranking Objectives

The feedback data from Youtube includes clicks, likes, ratings, etc. The author first divides these feedback into two types: engagement and satisfaction [1]. Interaction includes clicks and viewing time. Satisfaction includes likes and video ratings. Each feedback information is classified into binary classification and regression according to different types, which are processed with cross entropy loss and squared loss respectively. When online use, a weighted combination is constructed to obtain the final score for each video. The author also mentioned why pair-wise and list-wise sorting models are not used, mainly because point-wised can efficiently handle the case of many candidate sets [1].

2. Model structure

Because there are many targets, and some targets are quite different (such as regression and binary classification), in order to reduce interference, the model structure of share-bottom is changed to MMoE [1]. The author still retains a layer of shared-bottom, because putting MMoE directly on the embedding-layer will have a negative influence on timeliness.

3. Handling Bias

The author believes that the target of the "interaction" class has position-bias, selection-bias, etc., so a shallow-tower is built separately to model the bias, which makes the whole model similar to the structure of Wide & Deep [1].

Experiment

The experiment is mainly carried out on Youtube, and there are two indicators: AUC and complexity (number of multiplications). In the section of multi-task learning evaluation, the authors find out that the AUC of MMoE is better than shared-bottom under the same complexity. In the part of gating stability, the authors find out that training the neural network using multiple machines can lead to frequent divergence, and the gating network polarization. The author adopts the drop-out and re-normalized method for the gated output during the training phase to alleviate this problem. In the evaluation of bias reduction, the authors evaluate the effectiveness of shallow-tower and compare the shallow-tower with other methods. In the first part, the authors show that using shallow-tower can indeed "separate the learning of user utility and position bias" [1]. In the second part, the author compared the method of directly adding position as a feature to the main model, adversarial learning and other methods to deal with bias, and the effect is not as good as using shallow-tower.

Conclusion & Technical Extension

The multitask ranking system described by authors proves its scalability on the biggest media sharing platform Youtube. The direct implementation of this system on Youtube and any other media platform can substantially boost the user experience because the recommended videos can be more accurate than before. In addition to a direct implementation of this system, other variants of implementation can also be useful. For example, users may search for related videos after watching the current video. In addition to recommending related videos, the system can input related keywords in the search bar so that the users do not need to type the keyword themselves. This implementation is especially useful when the user wants to take the initiative of searching but does not know the keyword. A great example would be searching for the

background music that appears in the current video. The user might want to take the initiative of searching but the user may not know the name of the background music. The system can input the name of the music in the search bar so that the user can simply press Enter and view all the results associated with this music.

Reference

[1] 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). Association for Computing Machinery, New York, NY, USA, 43–51. https://doi.org/10.1145/3298689.3346997