Mingqing Teng

Technical Review: (Google's) multitask ranking system: Video recommendation on a large scale

1 Introduction

YouTube is the world's largest platform for creating, sharing, and discovering video content. It impacts most of the people's daily life and we live closely with YouTube. One of the most important features is the YouTube recommendation system functionality: give a video which a user is currently watching, recommend the next video that a user may expect and like to watch. This powerful functionality of YouTube is based on YouTube large-scale ranking system for video recommendation.

YouTube's video recommendation evolves from simple models to complicated models: Adsorption [1], ItemCF based method [2], combination of techniques based on collaborating filtering and content analysis [3]. From year 2016, the recommendation system was based on the deep learning solutions. The typical recommendation system follows the two-stage design with a candidate generation stage and ranking stage. Covington et al. [4] pioneered the work and laid down the skeleton of the current YouTube recommendation system and Zhao et al. 2019 [5] further developed and extended the YouTube model in the ranking stage. In this technical review, YouTube video recommendation system was reviewed.

2 Challenges faced by the large-scale multiple task industrial recommendation system

Scalability

Many existing recommendation models proven to work well for small problems but will fail for the large-scale video platform like YouTube which has more than 2.0 billion monthly active users. A successful model should be effective at training and efficient at serving. The current recommendation uses the point-wise approach which means it makes the predictions for each candidate based on only itself. In contrast, pair-wise or list-wise approaches learn to make predictions on ordering of two or multiple candidates. Pair-wise or list-wise approaches can be used to potentially improve the diversity of the recommendations. However, the current model chooses to use point-wise approach in order to increase the efficiency when it severs the large scale system. It is also easy to scale to a large number of candidates.

Freshness

YouTube has a very dynamic corpus and the topics of videos changes significantly even in seconds. The recommendation system should be able to model the new uploaded contents as well as the latest action taken by the users.

Noise

The sparsity and a variety of unobservable external factors make it very difficult to predict the users' historical behaviors. Due to the property of the metadata associated with the content, i.e., poorly structured, the recommendation system algorithm should be very robust to handle this.

Multiple objectives

The recommendation system is facing multiple objectives to optimize for and these objectives are different and even conflicts sometimes.

Position bias

It is possible that the users clicked the and watched a video simply because it was ranked high and not because the video is the one which the users like the most. This is due to the implicit bias in the system.

Multimodal feature space

The candidate videos with feature generated from multiple modalities, e.g. video content, thumbnail, audio, title description and demographics. These new modalities make the recommendation system more learning more challenging.

The current YouTube recommendation system is trying to overcome all these challenges mentioned above.

3 Model Architecture

Like other recommendation ranking system, the current YouTube recommendation system has two states, i.e., candidate generation stage and ranking stage. The candidate generation stage will retrieve hundreds of candidates from the huge corpus and ranking stage will give a score for each candidate and a final list will be provided.

3.1 Candidate Generation

In this stage, multiple algorithms are used and each of them is used to capture one aspect of the similarity between the query video and the candidate videos. One sequence model similar to [6] is used for generating personalized candidate given user history. The techniques mentioned in [7] to generate context-aware high recall relevant candidates. Finally, all the candidates are put in a pool for ranking stage.

3.2 Ranking

The model categorizes the user behaviors into two types: engagement behavior like clicks and watches; Satisfaction behaviors like clicking like and ratings. Correspondingly, the model formulates the multiple objectives into two categories as engagement objectives and satisfaction objectives.

Multiple objectives

The current ranking model extended the Wide and Deep Model by using Multi-gate Mixture of Experts (MMOE) as shown in the figure below:

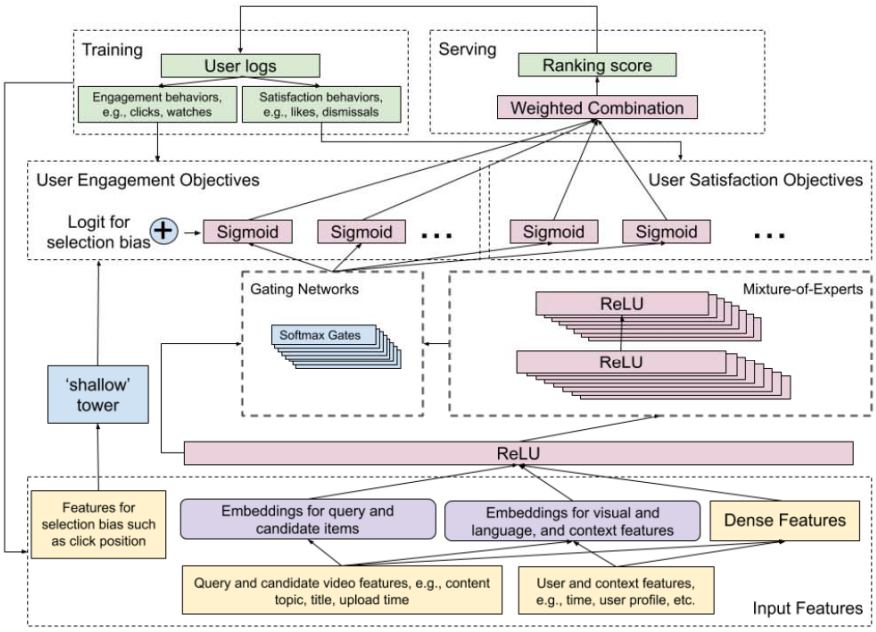

Figure 1: Model architecture of our proposed ranking system. It consumes user logs as training data, builds Multi-gate Mixture-of-Experts layers to predict two categories of user behaviors, i.e., engagement and satisfaction. It corrects ranking selection bias with a side-tower. On top, multiple predictions are combined into a final ranking score.

One MMoE layer replaces a shared layer, and this allows to create multiple experts where each of them learns a specific feature from the input. A separate gating network was added to each task. The gating network is trained and will choose particular experts for each task. MMOE is soft-parameter sharing model structure to model task conflicts and relationship. In contrast, the old shared-bottom model is hard parameter model which harms the learning of multiple objectives when the correlation between tasks is low. Thorofare, the MMoE is very appropriate in this regard.

Position bias

The recommendation system commonly relies on implicit feedback such as clicks and engagement with the recommended items. As pointed out by some researchers [8], the interactions between users and the current system create selection bias (position bias) in the feedback. It is possible that the users clicked the and watched a video simply because it was ranked high and not because the video is the one which the users like the most. To avoid this, the current YouTube reformation system add one shallow tower in addition to the main model. In this way, the shallow tower plus the main model can factorize the prediction into two components: a user-utility component from the main tower, and a bias component from the shallow tower. Then the output of this shallow tower is combined with the output from the main tower and it is used to rank the video list.

4 Experiments

YouTube recommendation system team designed and conducted series of experiments to test the performance of the recommendation system. These experiments are both offline and live ones.

Model Architecture	Number of Multiplications	Engagement Metric	Satisfaction Metric
Shared-Bottom	3.7M	/	/
Shared-Bottom	6.1M	+0.1%	+ 1.89%
MMoE (4 experts)	3.7M	+0.20%	+ 1.22%
MMoE (8 Experts)	6.1M	+0.45%	+ 3.07%

Table 1: YouTube live experiment results for MMoE.

The above table shows the MMoE performance improved a lot compared to the baseline models on both engagement metric and satisfaction metric.

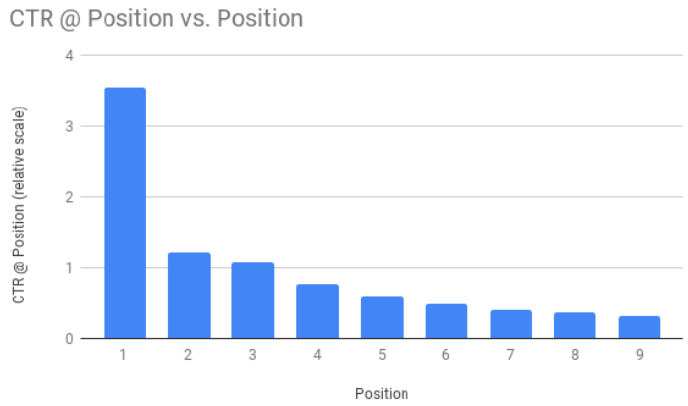

Figure 6: CTR for position 1 to 9.

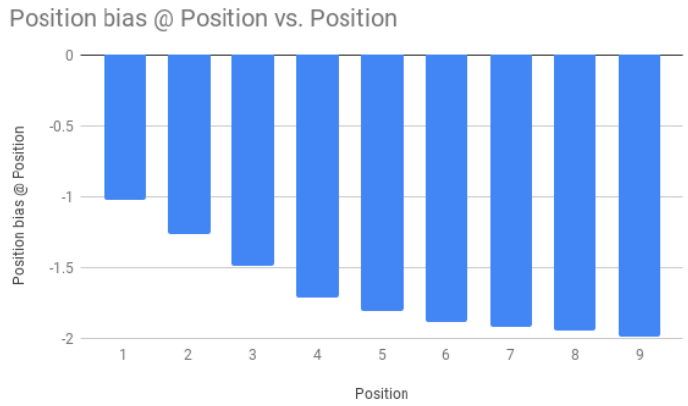

Figure 7: Learned position bias per position.

The first figure above shows the MMoE can accurately capture the bias of each positions and the second figure show MMoE can effectively and correctively learn the bias in each position.

Method	Engagement Metric
Input Feature	-0.07%
Adversarial Loss	+0.01%
Shallow Tower	+0.24%

Table 2: YouTube live experiment results for modeling position bias.

The table above shows MMoE can significantly improves engagement metrics by modelling and reducing the position bias compared to other methods.

5 Conclusions

The current YouTube recommendation system is based on the extension of Multi-gate Mixture-of-Experts (MMoE) model architecture to utilize the soft-parameter sharing. It has a light-weight and effective method to model and reduce the selection biases, especially position bias. The offline and live experiments show the substantial improvements on both engagement and satisfaction metrics.

Reference

[1]. [Baluja et al., 2008] Shumeet Baluja, Rohan Seth, D. Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, Mohamed Aly. Video suggestion and discovery for YouTube: Taking random walks through the view graph. WWW: 895-904, 2008.

[2]. [Davidson et al., 2010] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, Dasarathi Sampath: The YouTube video recommendation system. RecSys, pages 293-296, 2010.

[3]. [Bendersky et al., 2014] Michael Bendersky, Lluis Garcia Pueyo, Jeremiah J. Harmsen, Vanja Josifovski, Dima Lepikhin. Up next: retrieval methods for large scale related video suggestion. KDD: 1769-1778, 2014.

[4]. [Covington et al., 2016] Paul Covington, Jay Adams, Emre Sargin. Deep Neural Networks for YouTube Recommendations. RecSys: 191-198, 2016.

[5]. [Zhao et al., 2019] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, Ed H. Chi. Recommending what video to watch next: A multitask ranking system. RecSys: 43-51, 2019.

[6]. Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for YouTube Recommendations. In Proceedings of the 10th ACM conference on recommender systems. ACM, 191–198.

[7]. Walid Krichene, Nicolas Mayoraz, Stefen Rendle, Li Zhang, Xinyang Yi, Lichan Hong, Ed Chi, and John Anderson. 2018. Efcient training on very large corpora via gramian estimation. arXiv preprint arXiv:1807.07187 (2018).

[8]. Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. ACM Transactions on Information Systems (TOIS) 25, 2 (2007), 7.