Technology Review

Recommending What Video to Watch Next: A Multitask Ranking System

Introduction：

The paper[1] focuses on designing a recommendation system on large-scale video-streaming platform, which recommend the videos that audience might watch and enjoy.Typically a recommendation system can be divided into two part, generating candidates and ranking system.Ranking system is the key focus of this paper, which rank the candidate videos according to multiple factors and ideally give high rank to videos that users like most.The main contribution of the paper[1] is that it tackled two complex problems in ranking systems, namely, multitask learning and selection bias.

Challenges:

There are two key challenges existing in the recommendation design existing in a real-world video streaming platform.

The first difficulty is that there might be multiple high priority target that have conflict interest.More specifically, instead of recommending the videos that audience might get interested in, we want to promote the videos that rated highly, so that ensures the videos watched by the users are high quality.For this example, the target of recommending videos that interested the users might conflict with the target of recommending the high rated videos.

The second challenge is that there are implicit bias existing in the recommendation system. For example, user click the video might simply because the video is ranked higher rather than matching their interest.If system regard this click as a positive feedback and recommend more videos related to this topics, the whole recommendation system might not be useful and can fail to recommend the videos that interested the audiences.

Solutions:

In order to solve those problems, the team proposed new techniques MMOE(Multi-gate Mixture-of-Experts) and shallow tower to solve multitask and selection bias.

The previous technique used in ranking system use a shared-bottom model architecture. The main drawback of this architecture is that it is inefficient to learn multiple objectives when correction between tasks are low.In order to mitigate and improve the performance, MMOE design is introduced.MMOE is a type of a soft-parameter sharing model structure specifically designed to eliminate multitask issues.More specifically, the architecture substitute the shared layer in shared-bottom model to a MOE layer, with adding individual gate for each tasks.The main purpose of gating is to control information retrieval, and to eliminate the useless

information, which significantly encourage sharing of experts and improve the efficiency for training.

To solve the issue of selection bias, the team propose an architecture that is similar to Wide & Deep model architecture.The first step is to factorized the model into two parts. The first part is the main tower and the other is the shallow tower. More specifically, the main tower used to handle user preferences and for the shallow tower, it used to eliminate selection bias, which includes position bias and location preferences.

Experiments:

In order to test the performance of the ranking system, the team conducted several experiments to check the correctness.

In the initial experimental setup stage, the team uses multiple candidate generation algorithms to generate several candidates. Then apply the technology Tensor Processing Units to train the models and create both proposed model and baseline models, which enables the models to adopt to the most current information.

To judge the performance of MMOE architecture, the team conduct the experiments on live streaming platform YouTube, and compare the result with the baseline.For baseline method, the team simply use the shared bottom model architecture and compute the complexity based on main computing cost operation: multiplication.For live experiment, the team simply calculate the data of engagement metric and satisfaction metrics [engagement matrix indicates the amount of time spent on watching videos, and satisfaction matrix captures user's responses with a rating score] and compare the result for both MMOE and baseline models.

To evaluate the effectiveness of shallow tower in reducing position bias. The team divide the experiments into several stages.At the initial stage, the team will analysis users' implicit feedback, which can verify whether position bias existed.Then the team can started to do baseline test. Baseline experiments includes either use position as an input feature or imply adversarial learning.In the next stage,the team can conduct live experiment test just like the test in MMOE architecture, to compare the value of engagement metrics by adjusting the position bias. The last stage is the learned Position Biases. This stage can estimate the propensity value by using the biased implicit feedback, and the team can reuse the value to train the database.

Conclusion:

In conclusion, the paper[1] discuss the challenges existed in real-world ranking system.To solve those challenges, the author propose the techniques shallow tower architecture and MMOE architectures.In order to check the effectiveness of those

two architecture, the author conduct live experience in Youtube and showed a great improvement for both engagement and satisfaction of the users.

Reference

[1] Zhao, Zhe, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. "Recommending what video to watch next: a multitask ranking system." In Proceedings of the 13th ACM Conference on Recommender Systems, pp. 43–51. 2019.