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

Akarsh Vankayalapati[1,*]

[1]University of Illinois at Urbana-Champaign, Department of Computer Science, Champaign, IL
[*]e-mail: akarshv2@illinois.edu

Introduction

As users are watching more videos online, platforms begin to develop recommending systems to keep users engaged as long as possible. These systems require multiple factors to recommend the next video and users react to these systems in different ways. Because of this, platform developers need to find a way to build the best recommender system that is tailored to their services.

Zhao et al. aimed to solve this issue for Google as they proposed a large-scale multi-objective ranking system to recommend the next video to watch on YouTube, Google's industrial video sharing platform. This technology review aims at summarizing the system proposed, the results on the group's preliminary experiments, and the industry impact for the system[1].

Proposed System

Related Work

The subject of building a recommendation system can be thought of as returning the highest quality items of query. To build the system, Zhao et al. looked into industrial recommendation systems and multi-objective learning for recommendation systems

Industrial recommendation systems use implicit feedback from large amount of user logs to build their models into two stages: candidate generation and ranking. Implicit feedback assumes user-clicked documents are relevant, and while there is no extra effort form users and is the cost relatively low, there is some level of unreliability between this feedback and true user utility. This paper aims at combating this misalignment by introducing a deep neural network-based ranking model that supports multiple ranking objectives through multitask learning techniques. This multi-objective learning is proposed for the system because it needs into account multiple types of user behaviors to develop a utility score.

Problem Description

The problems associated with developing the proposed system deal are developing in a multi-modal feature space and scalability. The multi-modal feature space deals with learning utility from a combination of different features, like thumbnail, audio, title, etc. Scalability is important because the model needs to scale to billions of items and users and be efficient doing so. There are also problems associated with the stages of the recommendation system: candidate generation and ranking.

Model Architecture

The ranking system takes user feedback in the form of engagement and satisfaction behaviors to help the model of the system learn to predict multiple user behaviors. The ranking problem adopts the learning-to-rank framework to take multiple objectives to predict probabilities.

The system is designed to take multiple ranking objectives to take into account the different behaviors when recommending items. The two objectives stated above predict one type of user behavior and are separated because of it. Engagement objectives deal more with user clicks and watches are split into binary classification tasks, for clicks, and regression tasks, for time spent. Satisfaction objectives deals more with creating binary classification or regression tasks to predict user satisfaction. These tasks and objectives are then used to train a multitask ranking model for prediction.

A Multi-gate Mixture-of-Experts (MMoE) is used to mitigate conflicts of multiple objectives. MMoE models task conflicts and relations through its soft-parameter sharing model structure. This structure takes proponents from the Mixture-of Experts (MoE) structure have experts shared across all tasks combined with a gating network for each task. A MMoE layer is used to take task differences without needing many more model parameters when compared to the baseline method, which uses a shared-bottom model.

Position and selection biases are taken into account for the model because of the effects of using implicit feedback. Position bias occurs when users become more inclined to click on videos closer to the top of the list of recommended videos even when videos with higher utility are displayed lower. To reduce this and selection bias, the model architecture is designed similarly to the Wide Deep model architecture that factorizes the model prediction into two components: a main and a shallow tower. The shallow tower is trained with features contributing to selection bias and added to a logit function to be put into the main tower.

Experimental Results

Setup

The experiment was conducted using YouTube as the platform and a few hundred candidates from multiple candidate generation algorithms. Using TensorFlow, Tensor Processing Units (TPUs) were used to train the model and TFX Servo was used to serve it. the proposed and baseline model were trained sequentially to adapt them to the most recent data.

The experiment was split into two: offline and live. The offline experiments monitored classification tasks' AUC and squared error for regression tasks. The live experiments compared the models with production systems using A/B testing. Both of these experiments used metrics that were used to tune the model's hyper-parameters, like its learning rate. An example engagement metric include time spent on YouTube, and an example satisfaction metric is user survey responses.

Multitask Ranking With MMoE

The proposed model is compared with a baseline model that uses a shared-bottom architecture. When analyzing the results, it was shown that MMoE architectures led to a percent increase in both the engagement and satisfaction metric, both for the 4 and 8 expert models.

The reduction in position bias was also measured by analyzing click through rates (CTR) for different positions. This proved the phenomena that lower position videos received less CTR. When comparing the baseline models, input feature and adversarial loss, and the proposed model, which uses the shallow tower to model and reduce position bias, it was shown that engagement metrics were improved.

Limitations

There are a few limitations highlighted by the paper. These limitations include the uncertainty of scale on the effects of the ranking objectives and the training data being possibly noisy, sparse, and biased.

However, the limitations not highlighted in the paper include the ethical implications by employing such a recommendation system. The first limitation deals with algorithmic awareness and the effect is has on the trust of the user. When analyzing the effect a mismatch transparency has on trust, it can be shown that when expectations are violated by lack of transparency, overall trust is decreased[2]. Another ethical implication not highlighted in the paper was the demographics of the user reports taken. This plays plays a role in automated experiments as setting the gender of a user can impact the results received from a model, especially in a negative manner[3].

Conclusion

Overall, the proposed framework has promise to have an impact on the way ranking systems are used in recommending what video to watch next. From the studies conducted, it was clear that there was an increase in engagement and satisfaction metrics when compared to baseline methods, so it will be interesting to see where these see where Zhao et al. takes the framework. As said in the paper, it would be interesting to if a new architecture can be used to improve stability while not reducing performance and if model compression can be done. Along with that, Zhao et al. should look at taking the framework to other platforms like Facebook and Twitch to see how the metrics differ as well as looking closer into the ethical limitations proposed. As more research is done in the area, the groundwork laid by Zhao et al. will create a better way to have videos recommended to users.

References

1. Zhao, Z. *et al.* Recommending what video to watch next: A multitask ranking system. *RecSys* (2019).

2. Kizilcec, R. How much information? effects of transparency on trust in an algorithmic interface. *CHI* 2390–2395 (2016).

3. Datta, A., Vitagliano, G., Malik, M. & Huber, M. Automated experiments on ad privacy settings. *ArXiv* 72–87 (2015).