CS410 Technology Review

For this assignment, I reviewed this paper “Recommending What Video to Watch Next: A Multitask Ranking System” (Zhao et al., 2019) which can be found at this link https://daiwk.github.io/assets/youtube-multitask.pdf, because I’m interested to learn about how ranking algorithms work for video recommendations. The context is given the video that the user is currently watching, how should the candidate videos be ranked to maximize the user’s utility. Here the utility is defined by two objectives: 1) the user engagement objectives; 2) the user satisfaction objectives. The user engagement objectives are defined by the user interactions with the videos such as clicks, watches and comments to indicate the degree of engagement with the recommended videos. The user satisfaction objectives are defined as users’ ratings on the recommended videos (likes or dislikes) and whether the user would recommend the videos.

The model architecture discussed in this paper uses multiple task layers to predict the user behaviors for user engagement and user satisfaction (Zhao et al., 2019). The model consumes user logs as training data and extracts features to train and optimize for the different tasks. To optimize for the user engagement objective, binary classification task is used for the click prediction and regression task is used to predict the degree of engagement such as time spent on the video. To optimize for the user satisfaction objective, binary classification task is used to predict how likely the user will click like on the video and regression tasks is used to predict the user’s rating on the video. Cross entropy loss is computed for binary classification tasks and squared loss is computed for regression tasks. Then a multi-task ranking model is trained to output a weighted score and the weights are tuned to balance between the user engagement and user satisfaction objectives.

One of the problems for video recommendation ranking is that there is implicit feedback in the training data given the position of the recommended videos. Due to the fact that the implicit feedback is collected based on the currently ranked videos in the recommended list, users will only engage with videos in the recommended list and will only provide implicit feedback for videos in the recommended list. So the implicit feedback is biased towards the position of videos upon which the implicit feedback is provided for both user engagement and user satisfaction.

To solve this problem, the authors of this paper proposed a “wide and deep” model architecture with a “shallow tower” extension to correct ranking selection bias by taking features such as click position and device information as model inputs to produce a scalar term to correct this

selection bias in the final ranking score (Zhao et al., 2019). Based on the Figure 1 included in the paper, the output from the “shallow tower” is fed into the weighted combination of ranking scores at serving time. From what I gathered from the paper, the “shallow tower” is learned alongside with the main model, therefore I’m not sure why the “shallow tower” is not reflected in the training process in Figure 1.

Another problem discussed in the paper is the hard-parameter sharing techniques for the shared bottom model architecture. The hard parameter is not very effective for learning when there is low correlation between the tasks, therefore the paper proposed the “Multi-gate Mixture-of-Experts” approach (Zhao et al., 2019) to use soft parameter sharing model architecture to resolve the conflicts of different training objectives. The expert tasks are introduced to replace the shared bottom model layer to allow for more flexibility on the parameters so the task differences can be captured without more parameters. The gating layer is introduced to learn and select the expert tasks used based on the specific inputs.

Overall, the paper demonstrated the benefits of leveraging the “shallow tower” and “Multi-gate Mixture-of-Experts” model architecture by providing detailed analysis on the experiment results for online and offline experiments. In addition, the paper also discussed a few insights from the experiments. One of them being using neural network model architecture for video recommendation and ranking. There are a few challenges for using this model architecture specifically for video recommendation and ranking, namely the multimodal features (features vary from texts to images), multiple conflicting ranking objectives, noisy training data and distributed training challenges (Zhao et al., 2019). For video recommendation and ranking, the training data is the user logs that capture the user engagement and satisfaction on the videos which could be highly specific to the context and leads to high variance in the user feedback data. Whether a user clicks or likes the video may depend on the query context and therefore it’s difficult to generalize to include less common query contexts.

In summary, the paper explains in detail the multi-task learning model for video recommendation and learning and showcases the experiment results using the videos on the YouTube platform. The complexity of the model architecture allows me to understand better the challenges of information retrieval and ranking.

Reference:
Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, Ed Chi. 2019. Recommending What Video to Watch Next: A Multitask Ranking System. In Thirteenth ACM Conference on Recommender Systems

(RecSys ’19), September 16–20, 2019, Copenhagen, Denmark. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3298689.3346997