Rick Wang
Haoran14

The paper “Recommending What Video to Watch Next: A Multitask Ranking System” Recsys’19, is trying to solve the problem of Modeling for various user feedback signals. In addition to clicks, there are usually many "implicit" feedback signals in a simple recommender system. For example, in the ranking of answers in Quora, in addition to clicking to read, there are signals such as likes, clicks, and user reading time. These signals can indicate how much users like the content. And there is also the problem of selection bias that comes along with ranking; when a user clicks on a certain search result, it may not only be clicked because the quality of the content is good, but the answer may also be ranked first. This will form a cycle, and users will click on the content ranked in the front by the recommendation system; then, because the user click rate is high, the recommendation system will rank this content in the front. Thus, when the developers build a recommendation system, they must distinguish whether a user's click comes from liking this content or just because the system ranked it first. The paper uses YouTube to test the solution to these problems.

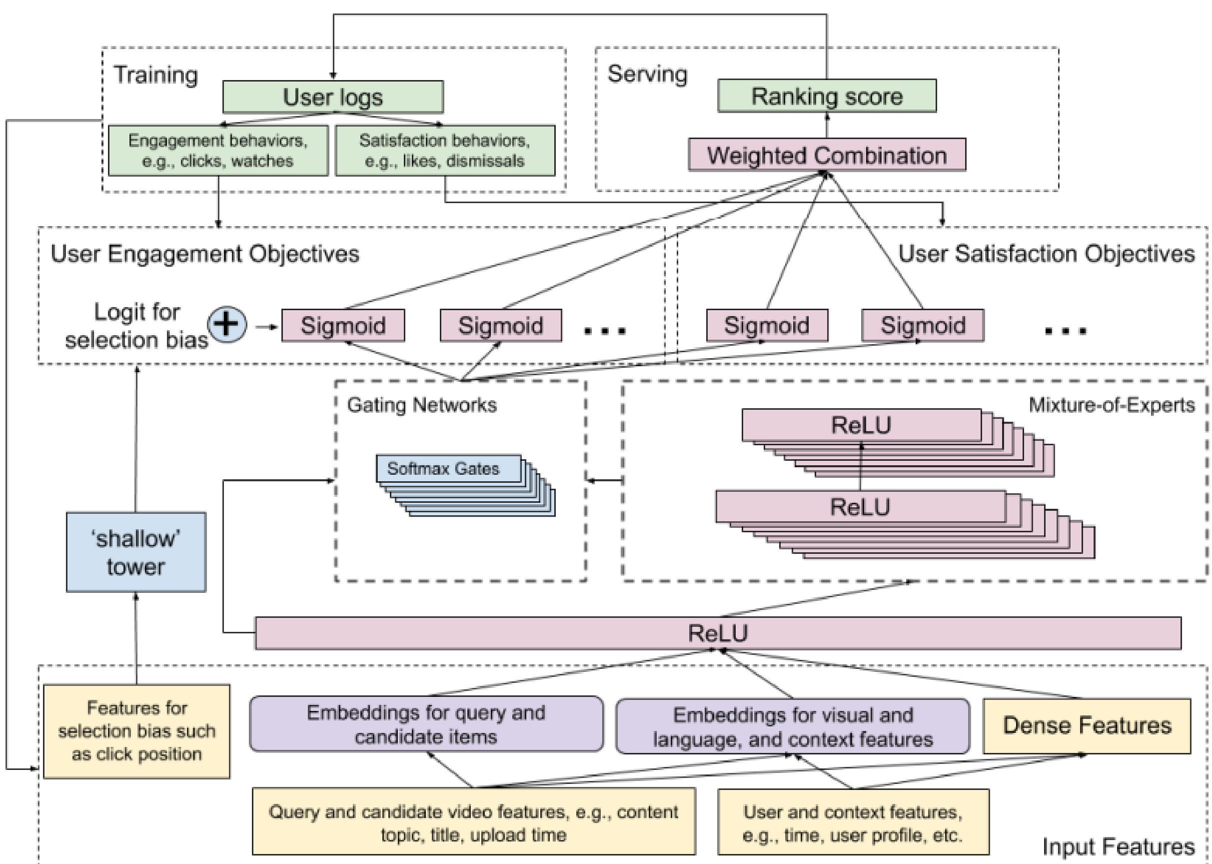

Figure 1(Zhao Z 2019)

This paper uses a NN architecture as a sorting module, as shown in Figure 1—Wide & Deep-based architecture + Multi-gate Mixture-of-Experts (MMOE). MMOE is better than the shared bottom structure for multi-targets that are inconsistent. However, the actual effect is still divided into different application scenarios. As Figure 1 shows, multi-goals are divided into two categories: engagement objectives: engagement goals such as clicking and watching videos, and satisfaction objectives: liking and scoring a video on Youtube. For potentially conflicting goals, the structure of the MMOE is used to solve them, and the gate structure is used to obtain information from the input selectively. To reduce selection bias (such as position bias), the shallow tower on the left of Figure 1 receives selection bias as input, such as sorting position, and output scalar as the final prediction bias term of the main model. This model decomposes the objective into two parts, one is the unbiased user preference, and the other is the propensity score. The model structure can be seen as an extension of Wide&Deep, with shallow towers replacing the Wide part. Because the shallow tower is directly learned, there is no need to obtain the propensity score in the random experimental area.

For the training process, the paper split the interaction target (with the de-bias tower) and the satisfaction target (without the de-bias tower) for training. To avoid overfitting the rank value during the training process, the paper discards 10% of the instance's rank feature, and the rank value is used as a missing value to estimate. The rank feature and device are the input of the shallow tower. The reason is that the position of rank 1 is different for different mobile phone models, so it is necessary to use both device and rank as input. The output of the Gating Network is the weight of different expert networks on different targets. However, due to distributed training, Gating Network degradation occurs in 20% of cases; most of the Gating Network outputs are 0. In this case, the advantages of multi-expert cannot be brought into play. To solve this problem, the paper continues to add dropout: randomly let 10% of the expert output be 0. Dropout is a better way to solve this problem because the essence of Gating Network degradation is that the training is too fit for a certain expert. Hence, the output of a random part of the expert is 0, equivalent to training only part of the network each time and avoiding overfitting to a certain sub-network. It also reduces the variance of the network and optimizes the final Training stability and model performance.

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%

Figure 2 (Zhao Z 2019)

The result of the solution proposed by the paper is shown in figure 2. Based on the result, in order to improve the Satisfaction Metric, the system needs to optimize Engagement Metric first. This is partly because the number of positive samples of the Engagement Metric, like clicks, is usually larger than the number of positive samples of the Satisfaction Metric (such as scoring). Therefore, after joint training & optimization of the model, the gain in the amount of information that the Satisfaction Metric can obtain will obviously be more.

The paper shows that the bias problem is an obvious problem in search results and related recommendations. This means there are also similar problems in a normal recommendation system. For example, if the users are willing to click on the first item in the search results because it ranks first, the other item is left behind. Maybe this happened not because the users don't like it but because the system never recommends it. Thus, from this perspective, does the entire feed recommendation system have this selection bias problem? This could be the topic to be further investigated based on the result of this paper.

Zhao, Z., Hong, L., Wei, L., Chen, J., Nath, A., Andrews, S., Kumthekar, A., Sathiamoorthy, M., Yi, X., & Chi, E. (2019). Recommending what video to watch next. *Proceedings of the 13th ACM Conference on Recommender Systems*. https://doi.org/10.1145/3298689.3346997