Technical Review for Multitask Ranking System

Recommender systems are widely used in the industry, for example, Google uses recommender systems to recommend advertisements to users, which brings huge amounts of benefits to the company. Besides classical methods taught in class: content-based filtering and collaborative filtering, there are more advanced methods like using deep learning technology. In this review, we will extend the knowledge and explore more in the field of multitask ranking systems based on the learning-to-rank framework.

First, we will introduce the task of recommendations. The goal is to predict users' future behaviors based on historical data. For example, given the information of videos, interactive logs of user's activities and the current video user is watching, we might ask what is the probability of the user's actions such as clicks, likes? Objectives can be grouped into two categories, engagement objectives, and satisfaction objectives, where engagement objectives are those activities that show users' participation and satisfaction objectives are those activities that show users' preference. Each group can further be divided into two types according to whether they are binary classification or regression. For each objective, one type of user behavior related to user utility is predicted [1].

Next, we'll look at some classical learning-to-rank models. Wide&Deep model jointly train wide learn layers and deep neural networks to combine both generalization and memorialization [2]. DeepFM model is a mixed approach between factorization machine and deep neural network, and final prediction is a sigmoid activated sum of FM component and Deep neural component, where FM layer consists of both addition part and inner product part thus can capture both linear and pairwise interactions between sparse features [3]. Covington et al. [4] developed a two-stage deep learning framework for YouTube video recommendation, including a candidate generation stage to generate several hundred candidates and a ranking stage. Zhao et al. [1] further improved the method by addressing two important challenges: the deficiency of the shared-bottom model when the correlation between multiple objectives is low; feedback loop effect due to the implicit bias. In the next two paragraphs, I'd like to introduce how the paper tackles these challenges.

For multi-task learning, one common approach is to have shared hidden layers than have separate higher-level hidden layers for each task. However, when tasks are quite different, shared hidden layers will be quite difficult to train due to the conflict of two tasks. Multi-gate mixture-of-experts (MMoE) was proposed to deal with the above challenge. The idea is to allow partial sharing and let the model itself find out to which extent to share among those different tasks. In more detail, shared bottom layers are split into smaller expert networks, and each task will have a gate consisting of a linear network followed by a softmax. The output of expert layers will be varied by each gate before summing up[5]. In the paper "Recommending what video to watch next: a multitask ranking system", MMoE is adopted and extended to migrate the conflicting objectives, as objectives like predicting if a user will watch a video are quite different from predicting if a user will like a video.

Implicit feedback has been used a lot to train recommended systems, although implicit feedback has some advantages like users don't need to make the extra effort, judgment can be unreliable sometimes. Imagine a user is going to pick a video, it's more likely that the user will select on the top of lists instead of on the bottom, regardless of actual user utility. If not properly addressed, a feedback loop effect will occur and bias would be amplified with the time going on. One way of breaking the feedback loop is to learn inverse propensity score, however, when user behaviors and item popularities change significantly and at a quick speed every day, the method may become unstable. To deal with this problem, Wide&Deep model is adapted to migrate bias information. Except for video information, features that are related to selection bias (such as position feature) are added in a shallow side tower, so the logit for selection bias contributes to the final prediction results[1].

Finally, I'd like to discuss the difference between pointwise, pairwise, and listwise approaches. When using the pointwise or pairwise approach, the ranking problem is transformed into the classification problem or regression problem accordingly. In the pointwise approach, objectives for a single doc/video are predicted for each time. We just need to sort the predicted scores to get the ranking list. Unlike the pointwise approach, the pairwise approach focus on a pair of items each time, the goal of the ranker is to minimize the mistakes in ranking two items. For example, if the specific user likes watching video A more than watching video B, then ranking B before A will be penalized in the loss function. The pairwise approach usually can improve the performance because it better captures the relative order of items. The listwise approach aims to come out with the optimal ordering of the entire list, which is pretty sophisticated, and the training data is hard to get usually[7]. Ma et al. relax restrictions by introducing partitioned preference, where items are split into ordered and disjoint partitions, but the ranking of items within a partition is unknown[8]. Although pairwise and listwise approaches can lead to quite decent results, due to the consideration of efficiency and scalability, pointwise approaches are still quite popular in the industry.

In conclusion, In this technical review, we introduce the topic of multitask ranking system, review some classical models, and talk about how challenges of conflicting objectives and bias selection are addressed. Finally, we discuss loss functions and see the trade-off between performance and efficiency between different methods. Hopefully, the review will let you know the big picture about the recommender system and learn about some state-of-the-art techniques.

[1] Zhao, Zhe, et al. "Recommending what video to watch next: a multitask ranking system." Proceedings of the 13th ACM Conference on Recommender Systems. 2019.
[2] Cheng, Heng-Tze, et al. "Wide & deep learning for recommender systems." Proceedings of the 1st workshop on deep learning for recommender systems. 2016.
[3] Guo, Huifeng, et al. "DeepFM: a factorization-machine based neural network for CTR prediction." arXiv preprint arXiv:1703.04247 (2017).
[4] Covington, Paul, Jay Adams, and Emre Sargin. "Deep neural networks for youtube recommendations." Proceedings of the 10th ACM conference on recommender systems. 2016.
[5] Ma, Jiaqi, et al. "Modeling task relationships in multi-task learning with multi-gate mixture-of-experts." Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018.
[6] Joachims, Thorsten, Adith Swaminathan, and Tobias Schnabel. "Unbiased learning-to-rank with biased feedback." Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. 2017.
[7] https://medium.com/@nikhilbd/pointwise-vs-pairwise-vs-listwise-learning-to-rank-80a8fe8fadfd
[8] Ma, Jiaqi, et al. "Learning-to-Rank with Partitioned Preference: Fast Estimation for the Plackett-Luce Model." International Conference on Artificial Intelligence and Statistics. PMLR, 2021.