

# Distillation based Multi-task Learning: A Candidate Generation Model for Improving Reading Duration

Zhong Zhao, Yanmei Fu, Hanming Liang, Li Ma, Guangyao Zhao, Hongwei Jiang

Tencent Inc.

Shenzhen, China

{zhongzhao,friedafu,meeloliang,listma,issaczha,rockyjiang}@tencent.com

## ABSTRACT

In feeds recommendation, the first step is candidate generation. Most of the candidate generation models are based on CTR estimation, which do not consider user's satisfaction with the clicked item. Items with low quality but attractive title (i.e., click baits) may be recommended to the user, which worsens the user experience. One solution to this problem is to model the click and the reading duration simultaneously under the multi-task learning (MTL) framework. There are two challenges in the modeling. The first one is how to deal with the zero duration of the negative samples, which does not necessarily indicate dislikes. The second one is how to perform multi-task learning in the candidate generation model with double tower structure that can only model one single task. In this paper, we propose an distillation based multi-task learning (DMTL) approach to tackle these two challenges. We model duration by considering its dependency of click in the MTL, and then transfer the knowledge learned from the MTL teacher model to the student candidate generation model by distillation. Experiments conducted on dataset gathered from traffic logs of Tencent Kandian's recommender system show that the proposed approach outperforms the competitors significantly in modeling duration, which demonstrates the effectiveness of the proposed candidate generation model.

## CCS CONCEPTS

• Information systems → Learning to rank.

## KEYWORDS

multi-task learning, knowledge distillation, candidate generation, duration modeling, recommender system

## ACM Reference Format:

Zhong Zhao, Yanmei Fu, Hanming Liang, Li Ma, Guangyao Zhao, Hongwei Jiang. 2021. Distillation based Multi-task Learning: A Candidate Generation Model for Improving Reading Duration. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, July 11–15, 2021, Montreal, Quebec, Canada*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/1122445.1122456>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*SIGIR '21, July 11–15, 2021, Montreal, Quebec, Canada*

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

<https://doi.org/10.1145/1122445.1122456>

## 1 INTRODUCTION

Click through rate (CTR) estimation is a widely adopted method for ranking in many recommender systems. Many models based on deep learning are proposed to estimate CTR in recent years[1, 3, 7, 9]. As for feeds recommendation, the predicted CTR (pCTR) can reflect how likely the user will click the item, but can not reflect how likely the user will like the item after click and read the content. For example, item with low quality and attractive titles (i.e. click baits) usually gets a high pCTR but users never like them. Therefore, only modeling CTR can not ensure the user's satisfaction with the clicked item. To improve user's experience, reading duration should be modeled as well, which is of great importance in industrial applications such as feeds recommendation.

In this paper, we focus on reading duration modeling and its application to large-scaled candidate generation for feeds recommendation. There are two main challenges in our real practice. The first challenge is how to deal with the zero duration of the negative samples. The negative samples get zero duration just because they are un-clicked, which doesn't necessarily indicate that the user dislikes the item. It is quite different from the positive samples with short duration which indicates dislike. Directly use the zero duration as target for modeling may lead to the inaccurate estimation. The second challenge is caused by the first challenge. In order to solve the problem of the first challenge, multi-task learning is employed. However, it is difficult to perform multi-task learning in the candidate generation model. As we know, most deep learning based candidate generation models adopt the double tower structure[2, 4]. They have a user tower and a item tower for computing the user vector and item vector respectively, and use user-item inner product as similarity metric for ANN search to generate candidate items in a very efficient way. Since the inner product can only model one single task, it makes multi-task learning difficult to be applied directly to the candidate generation model. To our knowledge, few papers discuss duration modeling. In the real practice, the commonly used method is to model duration by regression in one single task, in which the duration of all negative samples are set to zero and square loss is used. As mentioned before, fitting the duration of negative sample to zero may treat the dislikes (short duration) and the un-clicks (zero duration, but not necessarily dislike) similarly, which may mislead the model training.

To tackle the challenges stated above, we propose a distillation based multi-task learning approach, which we refer to as DMTL, to model reading duration for candidate generation. We overcome the problems of the existing duration models by considering duration's dependency to click within the multi-task learning framework which simultaneously models CTR and CTCVR for the click task and the duration task respectively. Then, we use the distillation

technique to transfer knowledge learned by the multi-task model to the double tower candidate generation model, which makes the candidate generation model obtain the ability of modeling reading duration while keeping its high efficiency in candidate generation.

To evaluate the performance of the proposed approach, we conducted experiments on the dataset gathered from the traffic logs of Tencent Kandian's recommender system. The results of the offline and online experiments show that the proposed approach outperforms the competing duration models significantly, which demonstrates the effectiveness of the proposed approach in modeling reading duration for candidate generation.

## 2 THE PROPOSED APPROACH

### 2.1 Multi-task Learning for Duration Modeling

The purpose of candidate generation is to select hundreds or thousands of items that are relevant to user's interests from the whole item corpus which may have millions or even billions of items. In this paper, the proposed DMTL improves the quality of candidate generation by modeling click and reading duration simultaneously, rather than modeling click only. For the click task, positive samples are clicked impressions, and negative samples are randomly selected from all items according to their frequency of being clicked. This is quite different from ranking model which uses clicked impressions as positive sample and un-clicked impressions as negative samples. For the duration task, positive samples are clicked impressions with duration more than 50 seconds (i.e., the median of all durations), and the rest are negative samples.

Let  $u_i$  and  $v_i$  be the user features and item features respectively, both of which are usually concatenation of embeddings of multiple fields, and  $x_i$  be the concatenation of  $u_i$ ,  $v_i$  and other dense features. Let  $y_i$  be the label of click task with  $y_i = 1$  representing the item is clicked and  $y_i = 0$  representing the item is randomly selected. Let  $z_i$  be the label of duration task with  $z_i = 1$  representing the item has been read for more than 50s, and  $z_i = 0$  otherwise. The duration modeling problem can be formulated as estimating the probability of  $z_i = 1$  given  $x_i$ , i.e.,  $p(z_i = 1|x_i)$ . As mentioned before,  $z_i$  is dependent of  $y_i$  since  $y_i = 0$  will cause  $z_i = 0$ . To better model this probability, we make good use of the dependency of click and reading. Specifically,  $p(z_i = 1|x_i)$  can be rewritten as

$$p(z_i = 1|x_i) = p(y_i = 1|x_i)p(z_i = 1|y_i = 1, x_i) \quad (1)$$

where  $p(y_i = 1|x_i)$  is the predicted click-through rate(pCTR),  $p(z_i = 1|y_i = 1, x_i)$  is the predicted conversion rate(pCVR) and  $p(z_i = 1|x_i)$  is the predicted click-through and conversion rate (pCTCVR). To reduce the influence of selection bias and data sparsity when modeling duration, we adopt the approach proposed in ESMM[6] which fits CTR and CTCVR simultaneously under the multi-task learning framework. In our model, click task and duration task fit the CTR and the CTCVR respectively by minimizing the binary cross entropy.

We employ the multi-task learning framework MMoE[5, 8] to model CTR and CVR. Let  $f_k$  be the  $k$ -th expert network which is usually a DNN and  $f_k(x_i)$  be the output vector of the  $k$ -th expert. For the modeling of CTR, the gate is computed as  $g_c(x) = [g_{c1}(x_i), \dots, g_{cK}(x_i)]$ , where  $g_c(\cdot)$  is the gate function defined as  $g_c(x) = \text{softmax}(W_c x_i)$  with  $W_c$  being a trainable matrix,  $K$  is the

number of experts, and  $g_{ck}(x_i)$  is the  $k$ -th element of  $g_c(x_i)$ . The output of the experts for modeling CTR is computed as

$$e_c(x_i) = \sum_{k=1}^K g_{ck}(x_i) f_k(x_i) \quad (2)$$

For the modeling of CVR, the gate function  $g_d(\cdot)$  with trainable parameter matrix  $W_d$  can be defined similarly, and the output of the experts for modeling CVR is computed as

$$e_d(x_i) = \sum_{k=1}^K g_{dk}(x_i) f_k(x_i) \quad (3)$$

where  $g_{dk}$  is the  $k$ -th element of  $g_d(x_i)$ . The pCTR and pCVR for sample  $x_i$  are modeled as

$$p_{ctr}(x_i, \theta_t) = \text{sigmoid}(h_c(e_c(x_i))) \quad (4)$$

$$p_{cvr}(x_i, \theta_t) = \text{sigmoid}(h_d(e_d(x_i))) \quad (5)$$

where  $h_c(\cdot)$  and  $h_d(\cdot)$  are DNNs that map  $e_c(x_i)$  and  $e_d(x_i)$  to the logit of pCTR and pCVR respectively, and  $\theta_t$  is all trainable parameters in the above formulations. According to 1, 4 and 5, the pCTCVR can be written as

$$p_{ctcvr}(x_i, \theta_t) = p_{ctr}(x_i, \theta_t)p_{cvr}(x_i, \theta_t) \quad (6)$$

As modeling reading duration is to fit the CTCVR, the loss of the duration task is the following binary cross entropy

$$L_d(\theta_t) = - \sum_{i=1}^N z_i \log p_{ctcvr}(x_i, \theta_t) + (1 - z_i) \log(1 - p_{ctcvr}(x_i, \theta_t)) \quad (7)$$

Equation (6) and (7) have modeled the dependency of click and reading by introducing  $p_{ctr}(x_i, \theta_t)$  to compute  $p_{ctcvr}(x_i, \theta_t)$ . However, only fitting  $p_{ctcvr}(x_i, \theta_t)$  to CTCVR can not ensure  $p_{ctr}(x_i, \theta_t)$  fits to CTR. Therefore, we need a auxiliary task to make sure that  $p_{ctr}(x_i, \theta_t)$  fits to the CTR. The loss function of this auxiliary click task is formulated as

$$L_c(\theta_t) = - \sum_{i=1}^N z_i \log p_{ctr}(x_i, \theta_t) + (1 - z_i) \log(1 - p_{ctr}(x_i, \theta_t)) \quad (8)$$

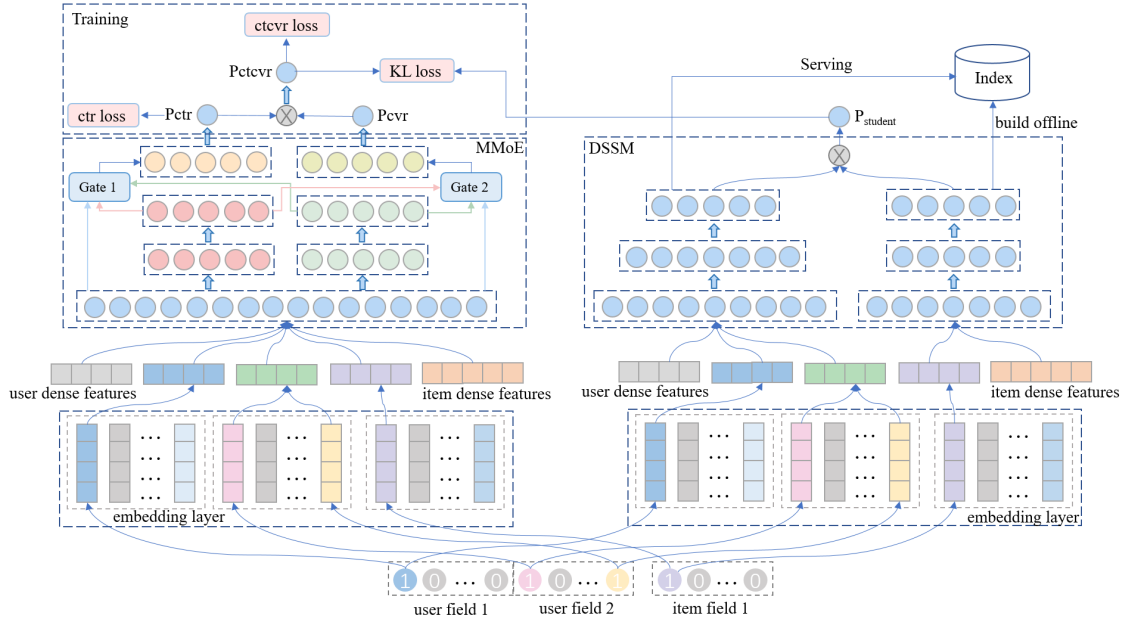
By summing (7) and (8), we get the multi-task learning loss function for the duration model as

$$L_{teacher}(\theta_t) = w_1 L_d(\theta_t) + w_2 L_c(\theta_t) \quad (9)$$

where  $w_1$  and  $w_2$  are the weights for each loss.

### 2.2 Distillation for Candidate Generation

In most deep learning based candidate generation models, double tower structure is employed to compute user vectors and item vectors, where the item vectors are used to build the item index. For a given user vector, the user-item inner product is used as similarity for ANN search in the item index, and the top-k items are returned as candidate items. However, the candidate generation models are unable to model duration by multi-task learning since the inner product can only model one task. To make the candidate generation model obtain the extra ability of modeling reading duration within its high efficient double tower structure framework, we use distillation technique to transfer knowledge learned by the MTL model in section 2.1 to the candidate generation model.



**Figure 1: Network structure of the proposed distillation based multi-task learning model. The teacher model (left) is a multi-task learning model which models reading duration. It considers the dependency of click and reading by minimizing the CTR loss and CTCVR loss simultaneously. The student model (right) is a candidate generation model with double tower structure. Knowledge of the teacher model is transferred to the student model by distillation so that the student model can obtain the ability to model reading duration while keeping its high efficiency for candidate generation.**

The proposed candidate generation model uses the double tower structure, and computes the user vector and item vector by DNNs. Let  $R(u_i)$  and  $S(v_i)$  be the user vector and the item vector respectively, where  $R(\cdot)$  and  $S(\cdot)$  are the DNNs that map input embedding to output vector. Given  $R(u_i)$  and  $S(v_i)$ , the CTCVR predicted by the candidate generation model can be formulated as

$$p(z_i = 1 | R(u_i), S(v_i), \theta_s) = \text{sigmoid}(R(u_i)^T S(v_i)) \quad (10)$$

where  $R(u_i)^T S(v_i)$  is the inner product of  $R(u_i)$  and  $S(v_i)$ , and  $\theta_s$  is the trainable parameters in  $R(u_i)$  and  $S(v_i)$ . We expect that  $p(z_i = 1 | R(u_i), S(v_i), \theta_s)$  can be similar to  $p_{ctcvr}(x_i, \theta_t)$  as much as possible, so that we can use (10) to estimate the CTCVR of duration accurately while keeping the high efficiency of the candidate generation model. To this end, we treat the multi-task learning model (9) as the teacher model and the double tower candidate generation model (10) as the student model, and use distillation to transfer knowledge from the teacher model to the student model. The loss of the distillation can be formulated as the following KL-divergence

$$L_{student}(\theta_s) = p_{ctcvr}(x_i, \theta_t) \frac{p_{ctcvr}(x_i, \theta_t)}{p(z_i = 1 | R(u_i), S(v_i), \theta_s)} + (1 - p_{ctcvr}(x_i, \theta_t)) \frac{1 - p_{ctcvr}(x_i, \theta_t)}{1 - p(z_i = 1 | R(u_i), S(v_i), \theta_s)} \quad (11)$$

By summing the loss of the teacher model and the student model, we obtain the loss of the distillation-based multi-task learning model as follow

$$L(\theta_t, \theta_s) = L_{teacher}(\theta_t) + L_{student}(\theta_s) \quad (12)$$

To prevent the teacher model from being influenced by the student model in the training stage, the parameters in the student model are separated from those in the teacher model, and the pCTCVR of teacher model is frozen when computing  $L_{student}(\theta_s)$ . Therefore, minimizing loss (12) is equivalent to minimizing the teacher loss and the student loss alternatively. In the inferring stage, we only use the student model to compute user vectors and item vectors, where item vectors are used for building the index, user vector is used as the query, and ANN search is performed to fetch top-k candidate items from the index for the user. The network structure and training/serving framework of the proposed model is illustrated in figure 1.

### 3 EXPERIMENT

#### 3.1 Offline experiment

**3.1.1 Dataset.** Experiments are conducted on dataset gathered from traffic logs of Tencent Kodian's feeds recommender system. The dataset has billions of training samples and millions of test samples. For each user, the positive samples are the clicked impressions, and the negative samples are randomly selected from all items according to their frequency of being clicked. Each clicked impression has a positive reading duration, and each randomly selected item has a zero duration.

**3.1.2 Competitors.** We compare the proposed DMTL to the existing candidate generation models. The competitors are listed as follow

**DSSM-Regression:** user vectors and item vectors are computed by DNNs, and their inner product regresses the reading duration by square loss. The duration of the negative samples is treated as zero for training.

**DSSM-Classification:** user vectors and item vectors are computed by DNNs, and their inner product is used to compute binary cross entropy for training a classification model, where positive samples are clicked-impressions with reading duration more than 50s and the rest are negative samples.

**DSSM-Click:** user vectors and item vectors are computed by DNNs, and their inner product is used to compute binary cross entropy for training a classification model, where positive samples are clicked-impressions and the rest are the negative samples.

**3.1.3 Metric.** We compare all approaches by evaluating their performance in a binary classification task, in which positive samples are true clicked-impressions with duration larger than 50s and the rest are negative samples. Area under ROC curve (AUC) is used as metric for performance evaluation. Higher AUC represents stronger ability of modeling reading duration.

**3.1.4 Parameter settings.** For the teacher model, the experts are DNNs with hidden layer size  $1024 \times 512 \times 256$  and the number of experts is 2. The DNN for each task is  $256 \times 256$ . For the student model, the hidden layer size for each tower is  $512 \times 256 \times 128$ . For both models, the embedding size for the categorical variables is 30.

**3.1.5 Results and Analysis.** Table 1 shows the performance of each candidate generation model. The regression model performs the worst among all models, which may be due to its direct fitting to large number of zero durations. The classification model is a little better than the regression model but still performs much poor than the proposed DMTL. This is due to the lack of modeling dependency between click and reading, which leads to the confusion between un-clicks and short duration. Compared to the duration DSSM, the click DSSM performs much better, which indicates the importance of click to duration. As the occurrence of duration depends on click, modeling duration without considering the dependency of click may miss a lot of important information for training. Among all models, the proposed DMTL achieves the highest AUC and significantly outperforms the competing methods. The improvement is attributed to the knowledge distilled from the teacher model which models duration in a more reasonable way that considers dependency of the click.

**Table 1: Offline performance of different duration models.**

models	AUC
DSSM-Regression	0.7374
DSSM-Classification	0.7562
DSSM-Click	0.9359
<b>DMTL</b>	<b>0.9544</b>

## 3.2 Online experiment

Online A/B tests are also conducted to compare the proposed DMTL to the competing candidate generation models. For the online experiment, we only change the candidate generation step by using different candidate generation models and keep all other steps unchanged. The online evaluation metric is the average reading duration defined as  $T/M$ , where  $T$  is the total sum of all reading durations (seconds) and  $M$  is total number of impressions. Table 2 shows the online performance of different models. DSSM-Regression and DSSM-Classification perform much worse than the DSSM-click and DMTL, which is consistent with the result of the offline experiment. Directly modeling duration without modeling its preceding step (click) may lead to the inaccurate estimation which causes the returned candidate items to be less relevant to user’s interests. The proposed DMTL overcomes this problem, and thus achieves the best performance compared to competing methods.

**Table 2: Online performance of different duration models.**

models	average reading duration (s)
DSSM-Regression	2.73
DSSM-Classification	1.74
DSSM-Click	3.93
<b>DMTL</b>	<b>4.27</b>

## 4 CONCLUSION AND FUTURE WORK

In this paper, we proposed a distillation based multi-task learning approach for modeling reading duration in candidate generation stage. The teacher model is a multi-task learning model which models reading duration by using ESMM to consider the dependency of click and reading. The student model is a DSSM candidate generation model with double tower structure. Knowledge Distillation technique is employed to make DSSM obtain the ability of modeling reading duration while keeping its high efficiency in candidate generation. Offline/online experiments conducted on real world dataset demonstrated the effectiveness of the proposed approach in modeling reading duration for candidate generation. The proposed approach can be easily generalized to other scenario in which there are multiple tasks with dependency. In the future, we will study how to develop multi-task learning model in the case when only part of the tasks are related, and how to fuse the output score for distillation.

## REFERENCES

- [1] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishu Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.
- [2] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.
- [3] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).
- [4] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 2333–2338.

- [5] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1930–1939.
- [6] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1137–1140.
- [7] Ruoxi Wang, Rakesh Shivanna, Derek Z Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed H Chi. 2020. DCN-M: Improved Deep & Cross Network for Feature Cross Learning in Web-scale Learning to Rank Systems. *arXiv preprint arXiv:2008.13535* (2020).
- [8] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending what video to watch next: a multitask ranking system. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 43–51.
- [9] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1059–1068.