ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10445973

ENHANCING MULTI-TASK MODELS FOR RECOMMENDATION WITH TENSOR TRACE NORM

Boqi Dai[⋆], *Kai Ouyang*[⋆], *Jun Yuan*[†], *Miaoxin Chen*[⋆], *Xingyu Lu*[⋆] , *Weiwen Liu*[†], *Rui Zhang*[*], *Hai-Tao Zheng*[⋆‡]

[⋆] Tsinghua Shenzhen International Graduate School
[†] Huawei Noah's Ark Lab
[*] ruizhang.info
[‡] Pengcheng Laboratory, Shenzhen, China

ABSTRACT

Noise is a pervasive issue in recommendation systems, which can stem from user behaviors that do not align with their intentions. As a result, noise reduction has become a prominent area of research in the field of recommendation systems. However, existing noise reduction techniques in recommendation tend to compromise the performance of certain task objectives. Moreover, they require modifying the structure of the model, which introduces inference latency and additional space cost. In this paper, we propose a straightforward yet powerful approach, **M**ulti-layer **T**ensor trace **N**orm (MTN), to address noise-related challenges. Our method achieves this by promoting information sharing across different tasks using tensor trace norms. By leveraging norms, MTN effectively reduces noise without modifying the model's structure or incurring substantial time and space complexities. Extensive experiments on public datasets and generated noisy datasets demonstrate the effectiveness of MTN on several of the most popular multi-task models.

***Index Terms*—** Multi-task learning, recommendation, noise reduction, low-rank

1. INTRODUCTION

Recommendation systems are widely used for personalized information filtering in online services such as E-commerce [1], social media [2], and instant video. Furthermore, the concept of multi-task learning is extensively applied to various tasks to address the issue of data sparsity [3, 4, 1].

However, recommendation models face significant challenges in capturing true user intent due to the pervasive presence of noisy interactions [6]. For instance, many user clicks are driven by curiosity rather than indicating their positive views towards the products. Additionally, there are instances where an item is exposed to users, but they do not pay attention to it, resulting in user not interacting with the potentially user-interested item.

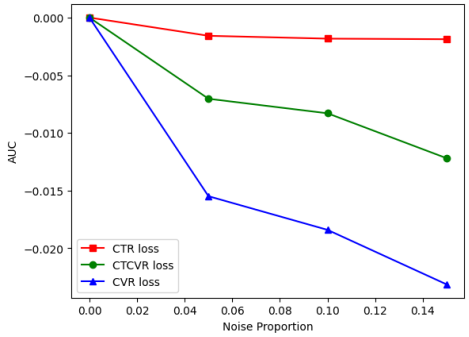

Fig. 1. CTR, CTCVR and CVR task (see details in Section 2.1) performance degradation of the MMoE [5] model under different noise proportions.

Multi-task learning can effectively leverage shared knowledge among tasks, thereby mitigating the impact of noise on each individual task. However, existing multi-task learning models are susceptible to noise interference. Figure 1 demonstrates that as the proportion of noisy samples increases, the performance of CTR, CVR, and CTCVR prediction tasks is significantly affected.

Unfortunately, there are currently limited researches on addressing the noise issue in multi-task recommendation models, and the few existing methods tend to compromise the performance of certain objectives within the multi-task framework [1]. For example, CSTWA [1] pays more attention to the latent structure information and incorporates a sample weight assignment algorithm biased towards CVR modeling to alleviate the interference caused by introducing a large number of CTR samples. However, although CSTWA partially alleviates the noise issue in multi-task recommendation models, it only focuses on a specific objective within the multi-task framework and fails to achieve consistent improvement across multiple objectives.

To address this issue, we propose to introduce low-rank regularization methods into the multi-task recommendation framework. Low-rank regularization methods are commonly

used in image denoising tasks [7, 8, 9], for they can identify the structures more robust to noise. Tensor Trace Norm (TTN) is a low-rank regularization commonly used in multi-task learning of computer vision field [10, 11, 12]. We believe TTN can reduce the noise impact of multi-task recommendation models.

In summary, we propose the **M**ulti-layer **T**ensor trace **N**orm (MTN) method to integrate TTN into multi-task models for recommendation. MTN improves the performance of all tasks and the robustness against noise by enhancing the sharing of common knowledge among task-specific parameters. Our method introduces only a few additional parameters and does not add any extra inference latency. Moreover, it can be applied to many multi-task recommendation models.

To evaluate the effectiveness of our method, we applied MTN to MMoE [5], ESMM [3] and Shared-bottom [4] models. Experimental results on public datasets demonstrate the enhancement of the aforementioned models by our method. We conducted extensive experiments to further highlight the advantages of our method.

The contributions of our work are as follows:

- To the best of our knowledge, our work is the first to apply tensor trace norm to multi-task recommendation systems in order to improve model performance and reduce the impact of noisy samples.
- The experimental results demonstrate that our MTN method can generally enhance the performance of several most popular multi-task models and mitigate the impact of noise on all tasks.
- Through extensive experiments, we demonstrated that applying tensor trace norm to task-specific parameters, as well as multilayer structures , leads to a greater improvement in model performance.

2. METHODOLOGY

2.1. CTR, CVR and CTCVR prediction tasks

Our definitions of CTR, CVR and CTCVR follow the work by Ma *et al* [3]. In a real-world purchasing session, user actions follow a pattern of $impression \longrightarrow click \longrightarrow conversion$. The click-through rate (CTR) task aims to predict the probability from impression to clicking. The click-through & conversion rate (CTCVR) task aims to predict the probability of from impression to clicking and subsequently conversion. The conversion rate (CVR) task aims to predict the probability from impression to conversion. According to dependency among tasks, CVR can be calculated from the following formula:

$$P_{CVR} = P_{CTCVR}/P_{CTR}. \tag{1}$$

2.2. Tensor Trace Norms

TTN is a generalization of matrix trace norm to higher dimensional spaces. To calculate tensor trace norm, a tensor should be unfolded to a matrix. Inspired by the work of Zhang *et al* [10], we defined TTN of parameter tensor $\mathcal{W}$, $|||\mathcal{W}|||_*$, as:

$$|||\mathcal{W}|||_* = \sum_{\mathbf{s}\subset[p],\mathbf{s}\neq\emptyset} \alpha_{\mathbf{s}} \left\|\mathcal{W}_{\{\mathbf{s}\}}\right\|_*, \tag{2}$$

$$\mathcal{W}_{\{\mathbf{s}\}} = \text{reshape}\left(\text{permute}(\mathcal{W}, [\mathbf{s}, \neg\mathbf{s}]), \left[\prod_{i\in\mathbf{s}} d_i, \prod_{j\in\neg\mathbf{s}} d_j\right]\right), \tag{3}$$

where $[p]$ denotes all possible combinations of dimension indexes, $\mathbf{s}$ represents a nonempty subset of $[p]$, $\neg\mathbf{s}$ represents the complement of $\mathbf{s}$ with respect to $[p]$. Through reshaping the tensor and permuting the dimensions (d in Formula 3), Formula (3) can flatten the high-dimensional tensor into a two-dimensional matrix. , Formula (2) then uses the sum of singular values of the matrices as matrix trace norm, weighted by α_s, to obtain the trace norm of the tensor. In our experiments, $\alpha_{\mathbf{s}}$ is a non-negative learnable parameter, and satisfies the condition $\sum_{\mathbf{s}\subset[p],\mathbf{s}\neq\emptyset} \alpha_{\mathbf{s}} = 1$.

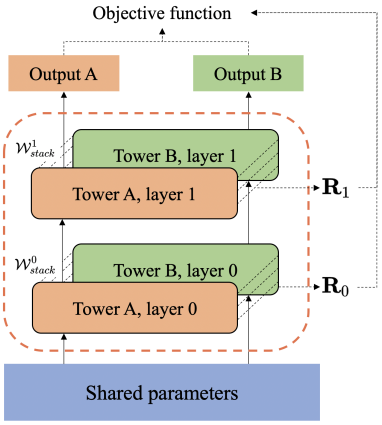

Fig. 2. Architecture of MTN method

2.3. Multi-layer Tensor Trace Norms

As shown in Figure 2, MTN is employed on all the layers of task-specific structures, such as task towers. The steps to use MTN method are as follows:

First, calculate the regular norm of layer i, $\mathbf{R}_i$. We stack all the layers of depth i from different task towers to compose a high-dimensional tensor $\mathcal{W}^i_{stack}$. After forward propagation, each stacked parameter tensor calculates the regular norm $\mathbf{R}_i$ through Formula (4), where the $|||\mathcal{W}^i_{stack}|||_*$ can be calculated by Formula (2). $|||\mathcal{W}^i_{stack}|||_F$ represents the Frobenius norm, which can prevent the extremely unbalanced learning of $\alpha_{\mathbf{s}}$ in Formula (2).

$$\mathbf{R}_i = |||\mathcal{W}^i_{stack}|||_* + |||\mathcal{W}^i_{stack}|||_F, \tag{4}$$

$$\min_{\mathbf{\Theta}} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} l\left(f_i\left(\mathbf{x}^i_j; \mathbf{\Theta}\right), y^i_j\right) + \lambda \frac{1}{n_l} \sum_{i}^{n_l} \mathbf{R}_i. \tag{5}$$

Table 1. Comparison of baseline models combined with MTN method. Each experiment was repeated ten times, the average was taken as metric. "*" indicates that the improvement of baseline after applying MTN method is statistically significant at p-value<0.05 over paired t-test, "**" indicates the p-value<0.01

	AliExpress_NL			AliExpress_ES			AliExpress_US			Ali-CCP		
	CTR	CVR	CTCVR	CTR	CVR	CTCVR	CTR	CVR	CTCVR	CTR	CVR	CTCVR
MMoE	0.7258	0.7934	0.8616	0.7281	0.8313	0.8901	0.7061	0.8310	0.8751	0.6187	0.6739	0.6521
+MTN	**0.7267**	**0.7951***	**0.8630***	**0.7303***	0.8304	**0.8912***	0.7058	**0.8326***	**0.8763***	**0.6198***	**0.6768****	**0.6548***
ESMM	0.7259	0.7946	0.8620	0.7288	0.8331	0.8916	0.7065	0.8357	0.8765	0.6157	0.6656	0.6471
+MTN	0.7252	**0.7966***	**0.8634***	**0.7306***	0.8323	**0.8926***	**0.7082***	0.8324	**0.8770**	**0.6170***	**0.6748****	**0.6559****
Shared-bottom	0.7245	0.7939	0.8605	0.7291	0.8231	0.8903	0.7053	0.8312	0.8749	0.6144	0.6416	0.6268
+MTN	**0.7259***	**0.7950***	**0.8620***	**0.7311***	**0.8316****	**0.8924***	**0.7076****	**0.8326***	**0.8773****	**0.6159***	**0.6520****	**0.6289****

After computing the norm $\mathbf{R}_i$ of all layers, the objective function is obtained by combining $\mathbf{R}_i$ with the cross-entropy loss of labels, as shown in Formula (5). In Formula (5), m represents the number of tasks, and n_i represents the number of samples, l represents cross entropy loss, f_i represents the network, n_l represents the number of the layers with TTN, λ is a hyperparameter controlling regularization degree.

In Formula (5), we utilize the average of $\mathbf{R}_i$ as the optimization objective. We also explored using learnable weights or gating networks to control the weights of $\mathbf{R}_i$. However, both approaches suffered from training instability issues. Instead, using the average of $\mathbf{R}_i$ or applying exponential decay weights to $\mathbf{R}_i$ resulted in better model performance. In our experiments, we used the average of $\mathbf{R}_i$ as optimization objective.

3. EXPERIMENTAL SETUP

3.1. Dataset and Metrics

We conducted our experiments on two public datasets, AliExpress dataset [13] and Ali-CCP dataset [3]. AliExpress is a dataset gathered from real-world traffic logs of the search system in a global e-commerce platform. Based on the different countries where the traffic data is collected, the dataset is divided into multiple sub-datasets, categorized as AliExpress_NL, AliExpress_ES and AliExpress_US. Ali-CCP is a dataset gathered from real-world traffic logs of the recommender system in Taobao, which has over 42 million records. For each dataset, we randomly selected 20% of the training data as the validation set.

We utilized Area Under the Curve (AUC) as the evaluation metric for CTR, CVR and CTCVR prediction tasks, each experiment was repeated 10 times, and the average of these results was taken as the final metric. We employed early-stop training, the training process stopped if AUC of CVR decreased for 3 consecutive epochs.

We conducted experiments on both original datasets and generated noisy Ali-CCP dataset. To experiment with the denoising effect of the MTN method under defferent noise proportions, we sequentially selected 5%, 10% and 15% of the samples to add noise. For each noisy sample, there was a 20% probability for each feature to be replaced by another valid value of that feature.

3.2. Baselines and Methods

To test the generalization ability of our method, we conducted experiments on the following popular multi-task models:

MMoE [5]: MMoE contains multiple sharing bottom, named experts. Every expert is expected to output from distinctive perspective. For each task, MMoE assigns a specific gating network to generate the corresponding weights for the expert networks. The weighted representations are then inputted into the task-specific towers.

ESMM [3]: This model includes shared embedding layer, and two separate task-specific towers for CTR task and CVR task. It leverages the sequential dependency between tasks to model the entire sample space.

Shared-bottom [4]: The Shared-bottom model consists of a shared bottom network and task-specific towers.

3.3. Experiments Implement

Our experiments were implemented based on the open-source recommendation library MTReclib [1] [14, 15]. The hyperparameters of the model architectures followed the same as those in the MTReclib. We did not employ batch normalization layer and dropout layer, and used TTN as the only regular norm. After tuning on validation dataset, the hyperparameter λ mentioned in section 3 was set to 1e-5 .

4. RESULTS AND ANALYSIS

Analysis 1: The generalization ability of MTN over models. We found MTN can enhance the performance of all baselines over all datasets mentioned in Section 3.1 after applying MTN method on the task towers of baseline models. In Table 1, all bold data indicates that models combined with MTN method outperform the baselines. For the AliExpress

[1]https://github.com/easezyc/Multitask-Recommendation-Library

Table 2. The performance of baseline models after applying MTN method under different noise proportions. "*" indicates that the improvement is statistically significant at p-value <0.05 over paired t-test, "**" indicates the p-value <0.01

Noise Proportion	0.05			0.10			0.15		
	CTR	CVR	CTCVR	CTR	CVR	CTCVR	CTR	CVR	CTCVR
MMoE	0.6171	0.6584	0.6451	0.6169	0.6555	0.6438	0.6168	0.6508	0.6399
+MTN	**0.6183***	**0.6669****	**0.6520****	**0.6182***	**0.6591***	**0.6454***	**0.6177**	**0.6582****	**0.6444****
ESMM	0.6166	0.6665	0.6459	0.6162	0.6547	0.6421	0.6164	0.6534	0.6315
+MTN	**0.6204****	**0.6732****	**0.6537****	**0.6180***	**0.6626****	**0.6468****	**0.6172**	**0.6601****	**0.6456****
Shared-bottom	0.6167	0.6582	0.6355	0.6155	0.6539	0.6314	0.6159	0.6502	0.6306
+MTN	**0.6188***	**0.6640****	**0.6406****	**0.6180****	**0.6585****	**0.6371****	**0.6171***	**0.6586****	**0.6365****

dataset, most of the baselines show improvements in various metric, with the highest improvement of 2.4 per mille points observed in AUC. There are only a few metrics that show a decrease, with not significant performance losses. For the Ali-CCP dataset, all metrics show significant improvements.

Specifically, for the Shared-bottom model, which is the backbone of many popular multi-task models, all tasks showed improved AUC across all datasets, indicating that our method has strong generalization capabilities.

Analysis 2: The performance of models combined with MTN method on noisy dataset. We applied the method mentioned in Section 3.1 to introduce noise into the Ali-CCP dataset, and the baseline performance after applying MTN under different noise proportions is shown in Table 2.

According to Table 2, the MTN method demonstrates better robustness against noise across all tasks when combined with all baselines.

After combined with MTN method, all baseline models show significant improvement on CTCVR and CVR prediction tasks, the maximum improvement can reach up to 1.4 percent of AUC.

CTR prediction tasks of all baselines shows smaller improvement under the current noise proportion. As the noise proportion further increases, the MTN method shows a greater improvement.

Analysis 3: Adopting MTN on shared parameters and single layer. We observed that applying the MTN method to shared parameters or a single layer does not yield the same level of effectiveness as when applied to task-specific parameters and multiple layers.

We compared the performance of applying MTN to the shared experts of the MMoE model and the task-specific towers. The results show that when MTN is applied to task-specific parameters, all metrics show improvement compared to the baseline. However, when applied to the shared experts, some metrics decrease.

We hypothesize that this phenomenon may result from MTN making the output logits of the individual experts more similar, thereby reducing their specificity. To validate this hypothesis, we visualized the outputs of the mentioned experts. Figure 3 shows the Euclidean distance between the outputs of the 8 experts. It can be observed that when MTN is applied to the experts, the outputs tend to converge towards each other, while when applied to the towers, the experts are able to learn more distinct information.

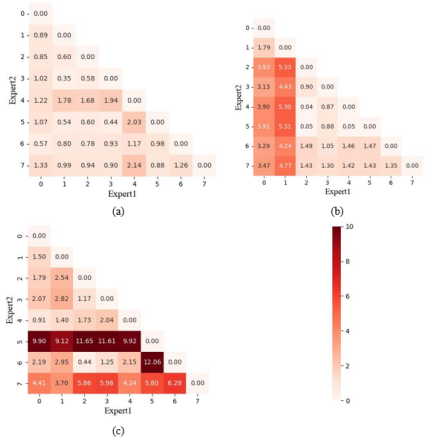

Fig. 3. Heatmap of difference between experts output after adopting MTN on MMoE towers and experts, (a): $\mathbf{MTN_{expert}}$, (b): baseline, (c): $\mathbf{MTN_{tower}}$

To test the effect of applying TTN on a single layer, we conducted experiments on an MMoE model with two-layers tower. When TTN is applied on single layer, the metrics of CTR and CTCVR tasks decrease significantly. We speculate that applying TTN to a single layer may interfere with the outputs of other layers.

5. CONCLUSION

In this paper, we propose a simple but effective method, MTN, to enhance information sharing among multiple tasks thereby improve model performance and noise resilience. Through empirical studies, we have found that MTN is generally effective for many popular multi-task models, and improves the performance of all tasks. For better performance, it is recommended to apply tensor trace norm to multiple layers of task-specific parameters.

6. REFERENCES

[1] Kai Ouyang, Wenhao Zheng, Chen Tang, Xuanji Xiao, and Hai-Tao Zheng, "Click-aware structure transfer with sample weight assignment for post-click conversion rate estimation," *arXiv preprint arXiv:2304.01169*, 2023.

[2] Kai Ouyang, Xianghong Xu, Chen Tang, Wang Chen, and Haitao Zheng, "Social-aware sparse attention network for session-based social recommendation," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022, pp. 2173–2183.

[3] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai, "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*, 2018, pp. 1137–1140.

[4] Rich Caruana, *Multitask learning*, Springer, 1998.

[5] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi, "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*, 2018, pp. 1930–1939.

[6] Yunjun Gao, Yuntao Du, Yujia Hu, Lu Chen, Xinjun Zhu, Ziquan Fang, and Baihua Zheng, "Self-guided learning to denoise for robust recommendation," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 1412–1422.

[7] Noam Yair and Tomer Michaeli, "Multi-scale weighted nuclear norm image restoration," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3165–3174.

[8] Jun Xu, Lei Zhang, David Zhang, and Xiangchu Feng, "Multi-channel weighted nuclear norm minimization for real color image denoising," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1096–1104.

[9] Tao Huang, Weisheng Dong, Xuemei Xie, Guangming Shi, and Xiang Bai, "Mixed noise removal via laplacian scale mixture modeling and nonlocal low-rank approximation," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3171–3186, 2017.

[10] Yi Zhang, Yu Zhang, and Wei Wang, "Multi-task learning via generalized tensor trace norm," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2254–2262.

[11] Yongxin Yang and Timothy M Hospedales, "Trace norm regularised deep multi-task learning," *arXiv preprint arXiv:1606.04038*, 2016.

[12] Lei Han and Yu Zhang, "Multi-stage multi-task learning with reduced rank," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016, vol. 30.

[13] pengcheng Li, Runze Li, Qing Da, An-Xiang Zeng, and Lijun Zhang, "Improving multi-scenario learning to rank in e-commerce by exploiting task relationships in the label space," in *proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2020, Virtual Event, Ireland, October 19- 23,2019*, New York,NY,USA, 2020, ACM.

[14] Yongchun Zhu, Yudan Liu, Ruobing Xie, Fuzhen Zhuang, Xiaobo Hao, Kaikai Ge, Xu Zhang, Leyu Lin, and Juan Cao, "Learning to expand audience via meta hybrid experts and critics for recommendation and advertising," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 4005–4013.

[15] Dongbo Xi, Zhen Chen, Peng Yan, Yinger Zhang, Yongchun Zhu, Fuzhen Zhuang, and Yu Chen, "Modeling the sequential dependence among audience multi-step conversions with multi-task learning in targeted display advertising," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 3745–3755.