Ranking with Deep Multi-Objective Learning

Xuezhi Cao, Sheng Zhu, Biao Tang, Rui Xie, Fuzheng Zhang, Zhongyuan Wang
Meituan-Dianping Group
caoxuezhi, zhusheng, rui.xie, zhangfuzheng, wangzhongyuan02@meituan.com, biao.tang@dianping.com

ABSTRACT

Ranking is the central part of plenty information retrieval problems. Existing learning to rank methods mostly employ one of the following learning methodologies: pointwise, pairwise and listwise learning. In this paper, we conduct analysis to demonstrate that these learning methodologies perform well in different scenarios respectively, according to corresponding evaluation metrics. Theoretically, all these learning methods aim at capturing relation between query context and candidate documents, the knowledge they extracted should be able to benefit each other and further improve the performance. Follow this idea, we propose multi-objective learning to combine their strengths. We extend existing deep ranking models into multi-prediction networks, and conduct training using both pointwise and listwise objectives simultaneously. Experiments using real search logs indicate that we can further improve the performance of existing models according to both global-level and query-level evaluations using our learning methodology.

KEYWORDS

Learning to Rank, Deep Learning, Multi-Task Learning

ACM Reference Format:
Xuezhi Cao, Sheng Zhu, Biao Tang, Rui Xie, Fuzheng Zhang, Zhongyuan Wang. 2020. Ranking with Deep Multi-Objective Learning. In *Proceedings of DLP-KDD 2020*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Ranking is one of the fundamental problems in information retrieval. The goal is to rank the candidate documents/items according to their importance/relevance/preference towards the query context, thus provide the useful information for the users [7]. Learning to rank has been successfully applied in various scenarios including search engines, recommender systems, expert finding, etc.

Existing learning to rank methods can be categorized into the following three types: pointwise, pairwise and listwise learning. Pointwise learning methods tackle the ranking problem using regression models, which estimates the likelihood of each data instance being a positive one, e.g. estimating the user's click through rate. Researchers in this direction focus on document/item/user understanding and model structures. As deep learning thrives, researchers propose deep ranking models such as DeepFM [4], PNN [8], DSSM [6] etc. Despite their success, pointwise learning only considers candidate documents independently and does not explicitly capture the relevant ranking information among them.

Pairwise learning is then proposed to capture the comparison between documents. The model is trained to estimate whether we should rank document i higher than document j under given context c. The representative works include Ranking SVM [3], RankNet [1] etc. The computational cost of pairwise training is rather large as it requires document pairs for training. Another drawback is that such models consider all mis-ranked pairs equally, while top documents in final ranking obviously draw more attention.

To tackle the aforementioned limitations, researchers further propose listwise learning. LambdaMart [2] is one of the representative works in this direction, which speeds up the training by aggregating pairwise losses within each query and also achieves direct optimization for listwise metrics such as NDCG (normalized discounted cumulative gain). As listwise methods can be considered as an upgrade for pairwise methods with larger model capability, we focus on pointwise and listwise learning in this paper.

Different learning methodologies have distinctive advantages and limitations. Pointwise learning considers each training instance (context-document pair, a.k.a. impression) independently and equally, hence reaches for global optimization. Listwise learning on the other hand, is great at optimizing top-k ranking but does not provide point estimation (e.g. ctr estimation, which could be extremely useful for scenarios such as computational ads). Also, they may neglect long tail part of ranking when focusing on top-k due to the nature of listwise metrics.

We conduct detailed experimental analysis to gain further insight of different learning methods. Experiments indicate that pointwise learning performs better according to global-level metric AUC, while listwise methods perform better according to query-level metric NDCG. Online experiments also conclude that listwise method achieves higher click through rate for top slots while pointwise method performs better for the rest. Another interesting finding is that models trained with different methods can convert to each other with only few training iterations, indicating that both learning methods extract similar lower-level general knowledge while having distinctive upper-level prediction-related knowledge.

Inspired by these findings, we attempt to combine the strength of both pointwise and listwise learning by proposing a novel deep multi-objective learning methodology. We extend existing deep ranking models by introducing two parallel prediction networks after hidden networks to serve pointwise and listwise learning respectively. Therefore, both objectives can share lower-level general knowledge by simultaneous updating the shared embeddings and hidden networks, while maintaining distinctive prediction properties in parallel prediction networks. Experiments using large-scale search log indicate that multi-objective learning can further improve the performance of existing deep ranking models according to both global-level and query-level evaluations.

DLP-KDD 2020, August 24, 2020, San Diego, California, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 POINTWISE VS LISTWISE

In general, the goal of learning to rank is to train a predictive model that automatically ranks the candidate documents effectively according to the given request context, e.g. in document search we rank the candidates according to their relevance towards the query keywords. Specifically, we train the model to score each document according to the given request context to capture the relevance/importance/preferences, given features regarding candidate document i and request context c as input. Formally, we have $\hat{y}_{ic} = \mathcal{M}(\mathcal{F}_{ic})$, where $\mathcal{F}$ indicates the feature representation and $\mathcal{M}$ represents the model. Then, we rank the candidates according to the predicted scores $\hat{y}_{ic}$.

The learning is carried out by fitting the model $\mathcal{M}$ with observed data $\mathcal{D} = \{y_{ic}, \mathcal{F}_{ic}\}$ by minimizing a pre-defined loss function $\mathcal{L}(\cdot)$, where y_{ic} is the ground truth indicator (e.g. user click).

$$\mathcal{M}(\mathcal{D}) = \arg\min_{\mathcal{M}} \mathcal{L}\left(\{y_{ic}, \hat{y}_{ic} = \mathcal{M}(\mathcal{F}_{ic})\}\right) \tag{1}$$

The loss function $\mathcal{L}(\cdot)$ is defined in general form, where it takes the prediction and the ground truth for all instances as input. In practice, we always break it down to pointwise or listwise loss functions depending on use scenarios, leading to pointwise and listwise learning respectively.

Pointwise Learning aims at estimating the absolute correlation of each document-context pair, e.g. ctr estimation. Hence, in these works we consider each training instance individually and equally, and break the general loss function down to pointwise loss:

$$\mathcal{L}\left(\{y_{ic}, \hat{y}_{ic}, \mathcal{F}_{ic}\}\right) = \sum_{ic} \ell_{point}(y_{ic}, \hat{y}_{ic}) \tag{2}$$

in which log loss and square error are the two widely used functions for pointwise loss function ℓ_{point}.

Listwise Learning focus on optimizing the ranking directly and breaks the general loss function down to listwise loss function:

$$\mathcal{L}\left(\{y_{ic}, \hat{y}_{ic}, \mathcal{F}_{ic}\}\right) = \sum_{c} \ell_{list}\left(\{y_{ic}, \hat{y}_{jc}\}\right) \tag{3}$$

A typical choice for listwise loss function ℓ_{list} is NDCG, which leads to LambdaMART [2] and its variations.

The fundamental difference between pointwise learning and listwise learning is that the former considers each training instance equally, while the latter emphasis on list, especially top ranked documents within each query. Therefore, they may out-perform each other when applied to different scenarios. For scenarios where global optimization is needed (e.g. display ads), we prefer pointwise learning. In these cases, AUC and RMSE are normally used for global-level evaluation. On the other hand, when focusing on top-k optimization we may prefer listwise learning and employ NDCG or MAP for evaluation.

To analyse the influence of learning methodologies, we train same model structure using both pointwise and listwise learning. Unless indicated otherwise, we use DeepFM [4] as the representative model structure for experiments in this paper. We use restaurant search logs from Meituan-Dianping, one of the most popular platforms for restaurant review and food delivery in China. Results showed in Fig. 1 indicate that pointwise learning achieves better performance according to global-level metrics AUC and RMSE while listwise learning gives better results according to query-level metric NDCG and MAP.

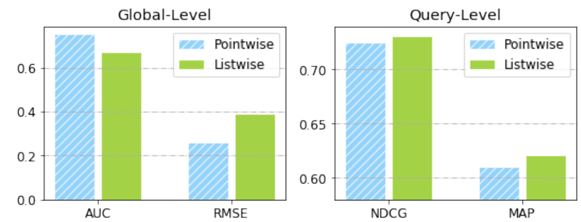

Figure 1: Pointwise Learning vs Listwise Learning

Table 1: CTR at Different Positions, Pointwise vs Listwise

Position	1-3	4-6	7-9	10-12	13-15	16-18	19-21
Pointwise	18.49%	9.20%	7.49%	6.89%	**5.71%**	**5.26%**	**5.46%**
Listwise	**18.66%**	**9.25%**	**7.51%**	**6.90%**	5.70%	5.20%	5.31%
Relative	+0.90%	+0.55%	+0.32%	+0.19%	-0.11%	-1.21%	-2.86%

We also conduct online A/B experiment to show how user behaviour differs when facing models trained with different learning methods. We show the results in Table. 1, where we average the click through rate by every 3 display slots to eliminate the influence of display advertisements. Results indicated that listwise learning achieves better performance on top slots (roughly top 10 positions), while pointwise learning performs better on the rest.

3 DEEP MULTI-OBJECTIVE LEARNING

Previous analysis indicates that pointwise and listwise learning have distinctive advantages for global-level and query-level optimization respectively, indicating that they can extract different but useful knowledge from training instances. This inspired us to combine their strengths by using the integrated knowledge.

Specifically, we aim at training the target model using both pointwise and listwise learning, leading to a multi-objective learning methodology. The rationale behind is that both learning methods aim at capturing the relation between query context and candidate documents, i.e. their objectives align with each other. Theoretically speaking, a perfect model should be optimal according to both global-level and query-level evaluations. Hence, optimizing on both dimensions together may close the gap towards such perfect model.

There exist a few research works that follow similar idea and intuition. In early years, Sculley first stated that ranking and regression learning has distinctive advantage and limitations respectively, and combine them can give 'best of both' performance [9]. Recently Hu et al. also point out that using only point-wise, row-wise or column-wise training alone can be problematic, and extend pairwise learning by combining different types of training samples [5]. Nevertheless, these works achieve multi-objective learning follow simple idea that randomly choose the objective in each training step, which has no theoretical support hence may be further improved.

3.1 Iterative Multi-Objective Learning

A straight-forward methods to conduct multi-objective learning is to apply pointwise and listwise learning iteratively, as demonstrated in Fig. 2 (C). We depict the training curves of pointwise, listwise

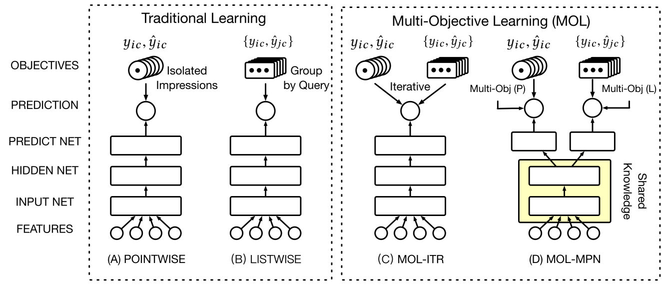

Figure 2: Learning Methodologies: (A) Pointwise Learning, (B) Listwise Learning, (C) Iterative Multi-Objective Learning, (D) Multi-Objective Learning with Multi-Predict Networks

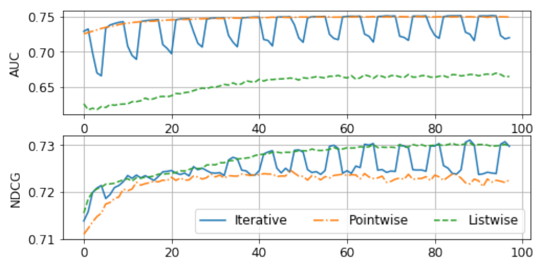

Figure 3: Training Curve of Iterative Multi-Obj Learning

and iterative multi-objective learning in Fig. 3, in which we plot according to both global-level and query-level evaluations.

Although iterative multi-objective learning does not provide significant performance gain, we do get some interesting findings from this experiment. From the training curve we may notice that when we iteratively train the model, its behaviour oscillates between the two original learning methods. Each time when a new iteration begins, the model's training curve quickly converges to the corresponding original training curve. This phenomenon indicates that lower-level general knowledge extracted using both learning methods are mostly consistent, while the upper-level knowledge, which can be quickly learnt, differs with each other. The reason behind could be that both learning methods share similar document/context understanding and feature representations in embeddings and lower-level layers, and differs in upper-level layer as the two objectives may lay distinctive emphasis on the predictions.

3.2 Multi-Prediction for Multi-Objective

Guided by previous findings, we prefer if both pointwise and listwise learning can contribute to the training of lower-level general knowledge while maintaining different upper-level knowledge to preserve distinctive prediction properties. Hence, we extend existing deep ranking models into multi-prediction networks.

Existing ranking models in deep learning family (e.g. DeepFM, PNN, DSSM, etc.) can be partitioned into input network (embedding representations), hidden network and prediction network. As we do not focus on model structures, we skip the details here.

We extend existing models by duplicating their prediction network and then train each prediction node using one of pointwise learning and listwise objectives respectively. We refer these two prediction node by Multi-Obj (P) and Multi-Obj (L) for pointwise and listwise respectively. We demonstrate the model structure as well as the training methodology in Fig. 2 (D). Within each iteration, we batch the training instances by two strategies, one organized by query to serve listwise learning and the other by random shuffle for pointwise learning. Then we put these training batches together to form the training set. Note that each training instance will appears twice. For each training step, we choose one training-batch from the candidates and train through corresponding network.

Following this design, the lower-level networks (input and hidden networks) can absorb training information from both optimizers, capturing the general shareable knowledge such as document or context understanding, feature representations etc. In the meanwhile, upper-level parallel prediction networks may focus on modelling the scoring function according to different objectives, keeping individual prediction properties. Ideally, the shared knowledge in lower-level networks can contribute to the improvement of both prediction networks hence perform better according to both global-level and query-level evaluations.

4 EXPERIMENTS

Dataset. We conduct experiments using restaurant search logs from Meituan-Dianping, one of the most popular platforms for restaurant review and food delivery in China. We use search log of 10 consecutive days in one city for training, and the following day for evaluation. In total we have 23,427,546 instances (impressions) from 1,890,762 queries for training and 1,924,915 instances from 163,657 queries for evaluation. Each instance is represented by 50-dimension feature vector, regarding user, shop, query, statistics (e.g. ctr, review count) etc.

Experimental Settings. As we do not have additional requirement on the ranking models besides it belongs to deep family, with out loss of generality we conduct experiments using DeepFM [4] and PNN [8] as representative models. For all evaluated models, we use embeddings with 12 dimensions, hidden network with 2 layers of 1,024 neurons, and prediction network with 1 layer of 1,024 neurons. Logarithmic loss is used for pointwise learning and NDCG is used for listwise learning.

Evaluating Multi-Objective Learning. We first use different learning methodologies to train DeepFM and PNN, and present the results in Table. 2. Results indicate that pointwise learning achieves better performance according to global-level metric AUC while listwise learning performs better according to query-level metric NDCG. For multi-objective learning, we evaluate according to both parallel predictions, where Multi-Obj (P) and Multi-Obj (L) indicates the one serving pointwise and listwise objective respectively.

We first discuss according to experiment using DeepFM. The best performance according to global-level metric AUC is provided by Multi-Obj (P), with 0.18% relative improvement compared to traditional pointwise learning. The improvement is considered significant in ranking scenario, especially when considering we only make alternation in learning methodology without adding additional model structures nor feature representations. And for query-level metric NDCG, Multi-Obj (P) achieves 0.16% relative improvement comparing to traditional listwise learning. This indicates our multi-objective learning with multi-predict network can successfully share knowledge between pointwise and listwise objectives

Table 2: Experimental Results for Multi-Objective Learning

	DeepFM		PNN	
	AUC	NDCG	AUC	NDCG
Pointwise	0.750006	0.724719	0.750656	0.724082
Listwise	0.669403	0.730435	0.670266	0.730732
Multi-Obj (P)	**0.751366**	0.724973	**0.751167**	0.724537
Multi-Obj (L)	0.725542	**0.731575**	0.725917	**0.731308**

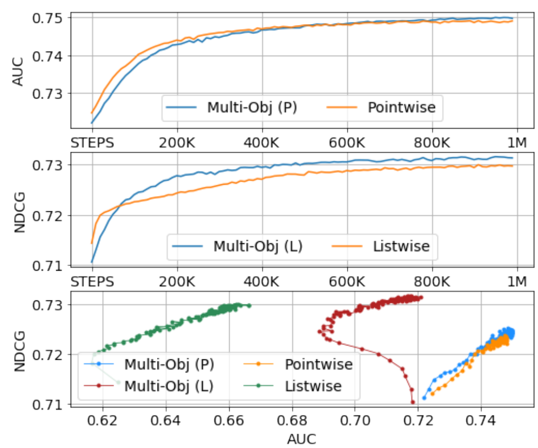

Figure 4: Training Curve of Different Training Methods

thus improving model performance according to both evaluation metrics. Experiments using PNN also present similar conclusion, indicating that the improvement can benefit ranking models from deep learning family in general.

By comparing Multi-Obj (L) with listwise learning, we may notice that our model achieves better performance according to both evaluation metrics. The relative improvement is rather significant for global-level evaluation AUC. The reason is that traditional listwise learning do not leverage global-level information while Multi-Obj (L) can acquire such knowledge via knowledge sharing in hidden and input networks. Hence, we alleviate the disadvantage of traditional listwise learning.

Training Curve. We depict training curve of multi-objective learning in Fig. 4. The first two plots correspond to AUC and NDCG versus training iterations respectively, and in third plot we plot 2-D training curve with x-axis representing AUC and y-axis representing NDCG. The curves indicate that using multi-objective learning achieves similar convergence rate with traditional methods hence does not lead to computational cost concerns.

Varying Model Capacities. We vary the embedding dimensions and present the results in Table 3. Result indicates that performance gain enlarges when dimension is over 12, indicating that multi-objective learning requires slightly larger model capacity as the amount of knowledge we extracted increases.

Balancing Parallel Prediction Networks. Previous experiments are conducted by feeding same amount of training data to both prediction networks, hence pointwise and listwise learning have equal influence. Now we break the balance by feeding more training samples to one of the optimizers (e.g. have each instance go through pointwise optimizer twice and listwise optimizer once). The results showed in Table 4 indicate that multi-objective learning

Table 3: Varying Embedding Dimension (AUC)

Dimension	6	12	24
Pointwise	0.749123	0.750006	0.750068
Multi-Obj (P)	0.749997	0.751366	0.751413
Relative	+0.11%	+0.18%	+0.18%

Table 4: Varying Training Ratio Between Optimizers

Point vs List	2 : 1	1 : 1	1 : 2
AUC	0.751439	0.751366	0.751131
NDCG	0.731364	0.731575	0.731635

is rather robust to such adjustment. Increase the ratio towards pointwise learning leads to slight improvement over global-level metric (AUC) and decline over query-level metric (NDCG), and vice versa. This also supports the finding that lower-level knowledge extracted using both optimizers are consistent, hence reducing training ratio for one optimizer does not lead to severe model vibration.

5 CONCLUSION & FUTURE WORKS

In this paper, we first conduct experimental analysis on pointwise and listwise learning to examine their individual strength and properties. Results indicate that they have distinctive advantage according to global-level and query-level evaluations respectively. We also notice that knowledge they extract only differ on upper-level prediction-related knowledge, while consistent in lower-level general knowledges. Based on these findings, we propose multi-objective learning methodology for deep ranking models to combine the power of both pointwise and listwise learnings. We extend existing deep models to multi-prediction networks to better serve the multi-objective learning. Experiments indicate that our learning methodology can successfully improve the performance of existing models according to both global-level and query-level evaluations, which close the gap towards general ranking optimization. For future works, we may consider integrating the parallel prediction networks to achieve united prediction for general scenario.

REFERENCES

[1] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*. ACM, 89–96.
[2] Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11, 23-581 (2010), 81.
[3] Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon. 2006. Adapting ranking SVM to document retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 186–193.
[4] 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).
[5] Jun Hu and Ping Li. 2018. Collaborative Multi-objective Ranking. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1363–1372.
[6] 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 Conference on information & knowledge management*. ACM, 2333–2338.
[7] Tie-Yan Liu et al. 2009. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval* 3, 3 (2009), 225–331.
[8] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 1149–1154.
[9] David Sculley. 2010. Combined regression and ranking. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 979–988.