Modeling Label Ambiguity for Neural List-Wise Learning to Rank

Rolf Jagerman
University of Amsterdam
Amsterdam, The Netherlands
rolf.jagerman@uva.nl

Julia Kiseleva
UserSat.com & University of Amsterdam
Amsterdam, The Netherlands
j.kiseleva@uva.nl

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
derijke@uva.nl

1 INTRODUCTION

One of the most important components to any search engine is the Learning to Rank (LTR) model. It considers many relevance signals and determines in what order to show the documents to the user based on these signals. Three main directions have emerged in the field of LTR: point-wise, pair-wise and list-wise methods. In our work, we focus on list-wise LTR, since these methods are the current state-of-the-art. In particular, we focus on ListNet [1] and ListMLE [3]. One of the major difficulties with these list-wise methods is that there is no consideration for the ambiguity that exists in LTR data that uses relevance scores. We solve this problem by introducing a new list-wise loss function called *ListPL* [2].

2 LABEL AMBIGUITY

The problem of *label ambiguity*, refers to the phenomenon that multiple documents may be assigned the same relevance label for a given query, so that no preference order should be learned for those documents. Learning a preference where none exists may lead to overfitting or limitations in the learner's ability to generalize.

3 LIST-WISE LOSS FUNCTIONS

Our main contribution, the ListPL loss function, is based on insights from both ListNet [1] and ListMLE [3]. We will briefly introduce those two functions before arriving at ListPL. We refer the reader to [2] for more information about our notation and the Plackett-Luce (PL) probability distribution.

3.1 ListNet

ListNet [1] attempts to optimize the following loss function, which is derived from the Kullback-Leibler divergence of a PL probability distribution of the relevance labels and the prediction function scores:

$$\mathcal{L}\left(f(\mathbb{D}), \mathbb{Y}\right) = -\sum_{\pi \in \Omega} PL\left(\pi \mid \mathbb{D}; \psi_{\mathbb{Y}}\right) \log\left(PL\left(\pi \mid \mathbb{D}; f\right)\right), \tag{1}$$

This cross-entropy accurately models label ambiguity, but is too expensive to compute in practice because $|\Omega| = O(n!)$. Instead the authors resort to a top-k approximation where k is usually 1, which mitigates much of the attractive properties of the list-wise loss function.

3.2 ListMLE

ListMLE [3] makes a simplifying assumption and uses a single permutation $\pi \in \{\pi \mid y_{\pi_i} \geq y_{\pi_j}; i < j\}$, which is then assumed to be *the* ground truth labeling. The loss function then becomes:

$$\mathcal{L}\left(f(\mathbb{D}), \mathbb{Y}\right) = -\log PL\left(\pi \mid \mathbb{D}; f\right). \tag{2}$$

A drawback is that this learns overly specific relations, which is harmful to the generalization power of the learning algorithm. If the chosen permutation happens to contain $\boldsymbol{d}_1 \succ \boldsymbol{d}_2$, even if both $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ have the same relevance scores, the algorithm will attempt to learn this relation, when in fact there is none.

3.3 ListPL

Instead of naively choosing a single permutation π of the documents and considering that permutation to be the ground truth, we propose a more sophisticated sampling method. The main idea is to directly sample a ranking from the PL distribution of the relevance labels during every stochastic update. We arrive at the ListPL loss function, which is our main contribution:

$$\begin{aligned} \mathcal{L}\left(f(\mathbb{D}), \mathbb{Y}\right) &= -\log\left(PL\left(\pi \mid \mathbb{D}; f\right)\right) \\ \pi &\sim PL\left(\pi \mid \mathbb{D}; \psi_{\mathbb{Y}}\right) \end{aligned} \tag{3}$$

This loss function approximates the true ListNet loss in expectation. Furthermore, it is easy to compute because sampling π directly from the PL distribution can be done efficiently since it is equivalent to sampling without replacement from a set of item-specific scores.

4 EXPERIMENTS

We compute a basic neural network ranking function using the three loss functions: ListNet (top-1 approximation), ListMLE and ListPL. We use the MSLR-WEB10k data set. ListPL performs similar to ListNet during training, but significantly outperforms both ListNet and ListMLE during validation and testing. We observe performance degradations on the validation set and test set for ListNet and ListMLE after 100 epochs, which indicates that these methods are effectively learning noise coming from label ambiguity. ListPL does not suffer from this problem. These results are in line with our expectations because ListPL properly deals with the ambiguity in the relevance scores and thus generalizes better. For more details about our experimental setup, we refer the reader to our paper [2].

5 CONCLUSION

We introduced a new list-wise loss function, *ListPL*, which handles the problem of label ambiguity better. Our experiments using a neural ranking function, show that the loss function generalizes better because it avoids learning noise coming from label ambiguity.

REFERENCES

[1] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *ICML*. ACM, 129–136.
[2] Rolf Jagerman, Julia Kiseleva, and Maarten de Rijke. 2017. Modeling Label Ambiguity for Neural List-Wise Learning to Rank. *arXiv preprint arXiv:1707.07493* (2017).
[3] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *ICML*. ACM, 1192–1199.