Learning to Rank Pointwise approach - technology review

Introduction

In the emergence of waves of new information each day, Information retrieval is becoming more and more crucial in helping people get the information they want accurately and efficiently, from piles of unorganized documents. In the heart of every information retrieval system, a ranker is used to intake a user query, match it against its collection of documents and output relevance judgments. Many conventional ranking algorithms are based on heuristics [1] such as term frequency and inverse document frequency. As an increasing amount of user behavior data are becoming available, machine learning technologies are being adopted to generate the state of art ranking models.

Learning-to-rank refers to the methods where machine learning is applied to learn the best combinations of predefined features for ranking[1]. For example, machine learning could be applied to learn the optimal parameters in BM25 for k1, b and k3[1]. Different ranking algorithms could also be combined to make a decision, the weights for the combination could also be learned using machine learning.

Pointwise learning to rank approaches

The three common learning to rank approaches are pointwise, pairwise and listwise. Pointwise approaches take a query and a document as input, and measure the relevance of this document and the query. To perform ranking, one computes the relevance of the query and each document in the collection, and sorts the documents by their relevance in the descending order. A pairwise approach takes a pair of documents, and learns the preference of one over the other. Listwise approaches take a list of documents and optimize for ranking metrics that usually requires a list of documents to be computed. It is worth noting that many popular ranking metrics are non-smooth and non-differentiable so listwise approaches often use some relaxation or heuristics of these metrics to facilitate learning. The rest of this article will mostly focus on pointwise learning to rank methods, and review different approaches and their performance. Pointwise approaches gain their popularity from the relatively low requirement on computing resources, compared with the other two. They are therefore easier to scale [4]. Pointwise approaches are also more robust to labeling noise in training data[6] because it is less impacted by the deteriorated quality of document pairs. This is important to consider because implicit feedback may bring some labeling noises when a user mistakenly clicks on an item.[7]

Random forest-based pointwise learning to rank methods

One approach to achieve better ranking is to combine a list of traditional rankers' scores to get a more accurate ranking. Random forest methods provide a list of advantages including computational simplicity, the ability to be parallelized, robust to outliers and missing values, etc.[2] Random forest methods could capture complex nonlinear interactions among features.[2]

On large datasets, a random forest with a regression setting is significantly more effective than a classification setting[8] [2]. On the contrary, on small datasets, a classification setting always performs better than a regression setting. Mapping of relevance labels could be used to stretch out the difference between highly relevant documents, but research[2] showed that this may or may not help with ranking performance, depending on the datasets size and evaluation metrics (ERR, NDCG@10). In random forest methods, the best number of candidate features at each node is also a hyperparameter that we could adjust. This study showed that on smaller datasets, the optimal setting is to use the default number, whereas for large datasets, increasing this number could lead to better results, since the training data could contain many weak features. There has also been a comparison of different ways to ensemble the trees in the forest. It turns out that assigning weights to individual trees might not have better results than using the standard non-weighted average of individual trees.

McRank pointwise learning to rank methods

The McRank[3] pointwise learning algorithm was developed in the context of web search. It treats the ranking problem as a multiple classification problem ("Mc"). To convert the classification results to ranking results, "expected relevance" is used as scores for each query-document pair based on learned class probabilities. The scores are then sorted to provide a ranked list. Results showed that when using NDCG for evaluation, McRank outperforms both regression-based and pair-based approaches.[3]

Deep learning in pointwise learning to rank methods

Researchers also explored using deep learning algorithms in ranking, which allows extraction of high level features. One study [4] compared three deep learning models when treating ranking as a classification problem (probability distribution over five classes). They are (1) DeepNet, which is composed of only fully connected layers, (2) CovNet, which replaced the first few fully connected layers of DeepNet with convolutional layers, and (3) FeatNet, where intermediate features are learned through convolutions. Results show that DeepNet and ConvNet achieved good performance on all experimented datasets [4]. DeepNet outperformed the other two on larger datasets because of its possibility to learn a more complex model. Noticeably, ConvNet was able to achieve better results than RankNet [5] and ListNet

Adaptive pointwise-pairwise learning to rank methods

Adaptive pointwise-pairwise methods incorporate the performance advantage of both pointwise and pairwise approaches. In One study [7], a model is trained to make a decision on which approach to use (thus “adaptive”). It built a continuum between the two approaches and learned a coefficient that indicates to what extent each approach is used. Results showed that it significantly outperforms solely pointwise, pairwise and listwise approaches. It also achieved better results than the combined pointwise-pairwise baselines.

Conclusion

Pointwise learning to rank method incorporates a wide variety of machine learning algorithms, all developed to give the most accurate relevance rating for a single query-document pair. In this article, we presented some recent and interesting research articles, including random forest, McRank and Deep Neural Net. We also look into adaptive pointwise-pairwise which shows the strong potential for future studies where different learning to rank approaches could be integrated. Results from these researches show a promising future of using machine learning algorithms in information retrieval.

Reference

[1] Liu, Tie-Yan. Learning to Rank for Information Retrieval. Berlin ;: Springer, 2011. Print.
[2] Cinar, Y. G., and J. -. Renders. Adaptive Pointwise-Pairwise Learning-to-Rank for Content-Based Personalized Recommendation, 2020. SCOPUS, www.scopus.com, doi:10.1145/3383313.3412229.
[3] Li, P. (. 1.)., et al. “McRank: Learning to Rank Using Multiple Classification and Gradient Boosting.” Advances in Neural Information Processing Systems 20 - Proceedings of the 2007 Conference, p. 8p. EBSCOhost,search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-84858773949&site=eds-live&scope=site. Accessed 16 Nov. 2020.
[4]Prakash, C., and A. Sarkar. “Ranking with Deep Neural Networks.” Proceedings of 5th International Conference on Emerging Applications of Information Technology, EAIT 2018. EBSCOhost, doi:10.1109/EAIT.2018.8470423. Accessed 16 Nov. 2020.
[5]Burges, C. (. 1.)., et al. “Learning to Rank Using Gradient Descent.” ICML 2005 - Proceedings of the 22nd International Conference on Machine Learning, pp. 89–96. EBSCOhost, doi:10.1145/1102351.1102363. Accessed 16 Nov. 2020.
[6]Niu, S., et al. “Which Noise Affects Algorithm Robustness for Learning to Rank.” Information Retrieval, vol. 18, no. 3, pp. 215–245. EBSCOhost, doi:10.1007/s10791-015-9253-3. Accessed 16 Nov. 2020.
[7] Cinar, Y. G., and J. -. Renders. Adaptive Pointwise-Pairwise Learning-to-Rank for Content-Based Personalized Recommendation, 2020. SCOPUS, www.scopus.com, doi:10.1145/3383313.3412229.
[8] Geurts P, Louppe G (2011) Learning to rank with extremely ran- domized trees. In: JMLR: workshop and conference proceedings, vol 14