Learning to Rank Models Comparison and Application

Introduction

This paper serves to review the different Learning to Rank (LTR) models and how they compare amongst each other. Learning to rank models use a base machine learning model (e.g. decision tree or neural network) and what differentiates the models is the loss function [1]. A key aspect of this comparison will explore in what situations would one model be preferred over another ultimately deciding what scenario an LTR model will perform the best in.

Pointwise, Pairwise, and Listwise Approaches

The three approaches for handling the loss function are pointwise, pairwise, and listwise. Starting with the simplest approach, the pointwise approach looks at a single document and trains a regressor or classifier on it to predict a value for how relevant it is to the input query [2] . The final ranking is simply ranking the documents based on their score, and the score for a single document is calculated independently from other documents.

The pairwise approach aims to improve upon the pointwise approach by looking at a pair of documents and determining which document is more relevant to the input query [3]. When looking at the result the model checks to see if the order of ranking between the two documents matches the predicted order, ideally keeping the more relevant document at the top of the pair. Ultimately the goal of the model is to minimize the number of wrong orders in the pairing of documents [4].

Finally the listwise approach aims to look at all the documents as a whole resulting in a list of the optimal ordering of relevancy for all the documents. This is done utilizing two primary techniques: direct measure of information retrieval measures such as NDCG

---

[1]Learning to Rank explained,
https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23
[2] Pointwise vs. Pairwise vs. Listwise Learning to Rank,
https://www.linkedin.com/pulse/pointwise-vs-pairwise-listwise-learning-torank-nikhil-dandekar
[3]LTR: A Complete Guide to Ranking using Machine Learning,
https://towardsdatascience.com/learning-to-rank-a-complete-guide-to-ranking-using-machine-learning-4c9688d370d4
[4]Quora Differences in LTR Loss Functions,
https://www.quora.com/What-are-the-differences-between-pointwise-pairwise-and-listwise-approaches-to-Learning-to-Rank

; minimizing the loss function based on understanding the unique properties of the type of ranking the user is trying to achieve [5].

LTR Approach Application

In terms of actual application of the LTR approaches, pointwise appears to be the worse approach mainly based on the principle that the optimization for the model is done based on how close the document is to the label rather than optimizing for ranking of the documents [6]. Because of this, there is a compromise on the ordering of documents. Pointwise approach can be used to solve regression problems or classical clarification implying that the results are easy to evaluate by simply looking the desired statistic measure (e.g, accuracy, recall, etc.) and noticing how many of the results made it into the number of results a user would want to look at [7].

Pairwise approach fixed the biggest problem with pointwise where it will optimize the ranking the of documents based on the relative score differences between a pair of documents rather than looking at a single document and how close it is to the label like in pointwise [8]. Pairwise still had its flaws though as it treated every document equally. This means that it doesn't look at pairs it's trying to fix in relation to where the documents lay in the ranking. Ultimately this means for a corpus size of 50 documents, the ranking is continuously improved all the way to the 50th document, but sacrifices the accuracy of the ranking in the top 10 documents.

Listwise approach improves pairwise even further by adding additional heuristic measures such as NDCG and avoids looking at individual pairs of documents and instead looks at the whole set [9]. An example of an evaluation metric would be a user looking at the DCG or NDCG at N documents. Listwise can be more complex than pairwise, but allows for better customizability. One example of a listwise approach implementation is ListNet which uses probabilistic approximations of ranking. Another popular implementation is LambdaRank which uses heuristics such as DCG. Overall, on most

---

[5] Pointwise vs. Pairwise vs. Listwise Learning to Rank, https://www.linkedin.com/pulse/pointwise-vs-pairwise-listwise-learning-torank-nikhil-dandekar

[6] Pointwise, Pairwise and Listwise Learning to Rank, https://medium.com/@mayurbhangale/pointwise-pairwise-and-listwise-learning-to-rank-baf0ad76203e

[7] LTR Optimizing Relative Ordering, https://datadojo.dev/2020/10/15/pointwise-pairswise-and-listwise-learning-to-rank-models-three-approaches-to-optimize-relative-ordering/

[8] Ibid. Pointwise, Pairwise and Listwise Learning to Rank

[9] Ibid.

ranking problems methods such as LambdaRank and another method called LambdaLoss which provides a generalized framework achieve state-of-the-art ranking evaluation [10].

Conclusion

In terms of deciding what LTR approach would best suit the needs of the user it would mostly depend on the number of documents in the corpus, and understanding the unique properties of the ranking they are trying to achieve. In most cases, a listwise approach will allow for the biggest flexibility and is the most robust of all the approaches. It improves on all the issues that occur in pointwise and pairwise approach, while incorporating solutions to its own problems such as incorporating normalization to an implementation that relies on the DCG heuristic.

[10] LTR: A Complete Guide to Ranking using Machine Learning, https://towardsdatascience.com/learning-to-rank-a-complete-guide-to-ranking-using-machine-learning-4c9688d370d4