2021

Technology Review

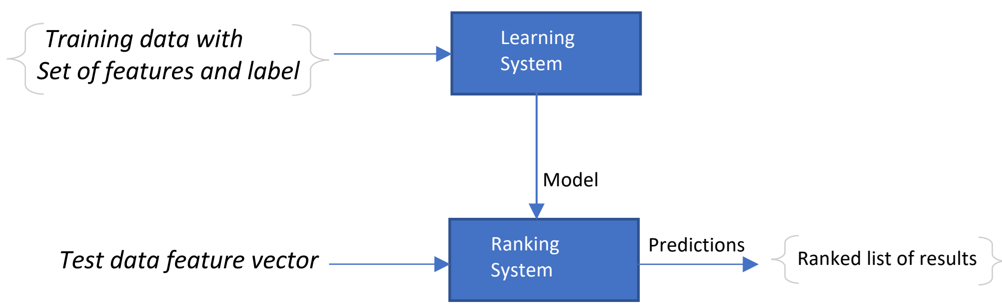
LEARNING TO RANK LISTWISE APPROACH

LEKKALAPUDI, YOGESWARA RAO

Table of Contents

What is Learning to Rank?

Learning-To-Rank is a machine learning technique to solve the ranking problem. Traditional ranking models like BM25 represents the ranking function using probability and statistic techniques based on the words that appears in the query and the document. With the increased availability of data, particularly in the web search, more modern techniques are employed using machine learning techniques to learn the ranking function using training data derived from various signals. Learning to rank has become one of the key technologies for modern web search.



Learning to Rank algorithm can be classified into three major approaches.

1. Pointwise
2. Pairwise
3. Listwise

Pointwise

Predicts the relevance degree of a single document by assigning a score. The relevance degree is a numerical value or an ordinal score. The documents are sorted by the score to produce the final ranked list.

Pros – Standard machine learning algorithms for regressions and classification can be applied.
Cons – The algorithm works on a single document at a time and the order of the documents is lost. This is a major problem for a ranked list.

Pairwise

Predicts the relevance degree of a pair of documents by assigning a score. The relevance degree is a binary value which tells the optimal order for the documents pair which is then used to come up with a final ranked list.

Pros – This can be treated as a classification problem. Standard machine learning techniques can be applied.

Cons – This approach only takes into consideration the relative order of a pair of documents. It is very difficult to derive the final ranked list.

Listwise

Takes into account the entire set of documents associated with an input query. Relevance score of each document in the list is predicted using a scoring function and averaged over all the queries in the training data.

Pros –Takes into consideration the position of the document in the list. Improved performance when compared to pointwise and pairwise approached.
Cons – Optimization problem is difficult because the evaluation methods are not continuous functions with respect to the ranking model's parameters and uses sorting.

Scope of this review

The scope of this review is limited to Listwise Approach for Learning to Rank. Listwise approach for learning to Rank is reviewed with an example based on the published papers.

Listwise Approach for Learning to Rank

Listwise approach uses document lists as instances for both training and prediction. A list of features and list of corresponding scores forms an instance. Ranking function predicts a list of scores for the given list of features. The objective of the learning function is to minimize the total losses with respect to the training data.

Probability models are used to calculate listwise loss function of the ranking task. List of scores is mapped to a probability distribution and the loss between the prediction and the ground truth is calculated using any of the available evaluation metrics. Give two probability distributions, distance between the two distributions can be used as a listwise loss function.

Two probability models are used.

1. Permutation probability
2. Top one probability

Permutation probability

For a give n items in a list, a relevance score is assigned to each item using the ranking function. All possible permutations of the n items are calculated, and the permutation probability is calculated which represents the likelihood of the permutation. Permutation with the highest probability is our desired ranked list.

Probability of a permutation is defined as:

$$P_s(\pi) = \prod_{j=1}^{n} \frac{\phi(s_{\pi(j)})}{\sum_{k=j}^{n} \phi(s_{\pi(k)})}$$

π is a permutation of the n items

ϕ(.) is an increasing and strictly positive function.

$\mathbf{s_{\pi(j)}}$ is the score of the object at position **j** of permutation **π**

Example:

Let's say we have 3 items (n = 3) and s = {s1, s2, s3} representing the score of the items, the permutation probability of a permutation π = {n1, n2, n3} can be defined as

$$P_s(\pi) = \frac{\phi(s_1)}{\phi(s_1) + \phi(s_2) + \phi(s_3)} \cdot \frac{\phi(s_2)}{\phi(s_2) + \phi(s_3)} \cdot \frac{\phi(s_3)}{\phi(s_3)}.$$

For n = 3 all possible permutations are 6 (n!)

$$\overset{6}{\underset{i=1}{\pi}} = \begin{cases} \text{n1 n2 n3} \\ \text{n1 n3 n2} \\ \text{n2 n1 n3} \\ \text{n2 n3 n1} \\ \text{n3 n1 n2} \\ \text{n3 n2 n1} \end{cases}$$

Let S = {0.3, 0.5, 0.2} be the relevance score of n items

Let ϕ(.) be an exponential function

Using the above formula, permutation probability of all the permutations can be computed as follows:

P.No	Permutation	$P_s(\pi)$	$P_s(\pi)$ – computed value
π_1	n1 n2 n3	$\frac{e^1}{e^1+e^2+e^3} * \frac{e^2}{e^2+e^3} * \frac{e^3}{e^3}$	0.18
π_2	n1 n3 n2	$\frac{e^1}{e^1+e^3+e^2} * \frac{e^3}{e^3+e^2} * \frac{e^2}{e^2}$	0.14
π_3	n2 n1 n3	$\frac{e^2}{e^1+e^2+e^3} * \frac{e^1}{e^1+e^3} * \frac{e^3}{e^3}$	0.21
π_4	n2 n3 n1	$\frac{e^2}{e^1+e^2+e^3} * \frac{e^3}{e^3+e^1} * \frac{e^1}{e^1}$	0.19
π_5	n3 n1 n2	$\frac{e^3}{e^1+e^2+e^3} * \frac{e^1}{e^1+e^2} * \frac{e^2}{e^2}$	0.13
π_6	n3 n2 n1	$\frac{e^3}{e^1+e^2+e^3} * \frac{e^2}{e^2+e^1} * \frac{e^1}{e^1}$	0.16

We can sort the above permutations and pick the top one. In the above example, the best ranked list is π_3

Permutation probability can quicky become an expensive task considering the large number of permutations needed for a large n. Total number of permutations is n! For a large n, this is not a viable solution. To address this problem, top one probability is defied.

Top One Probability

Instead of calculating n! permutations, it is enough to calculate just the top one probability of each items in the list. Top one probability represents the probability of an item being ranked on the top, given the scores of all the items.

Top one probability of an item is calculated as shown below

$$P_s(j) = \frac{\phi(s_j)}{\sum_{k=1}^{n} \phi(s_k)},$$

where s_j is the score of object j, $j = 1, 2, ..., n$.

For the above example,

$P_s(n1) = \frac{e^1}{e^1 + e^2 + e^3} = 0.32$

$P_s(n2) = \frac{e^2}{e^1 + e^2 + e^3} = 0.39$

$P_s(n3) = \frac{e^3}{e^1 + e^2 + e^3} = 0.29$

This suggests the ranked list of items as {n2, n1, n3| which is same as π_3 in the Permutation Probability.

References:

http://times.cs.uiuc.edu/course/598f14/l2r.pdf
https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2007-40.pdf