Learning to Rank

PRI 23/24 · Information Processing and Retrieval
M.EIC · Master in Informatics Engineering and Computation

Sérgio Nunes
Dept. Informatics Engineering
FEUP · U.Porto

Based on Learning to Rank for Information Retrieval, Tie-Yan Liu (2009)

Learning to Rank

➔ *Learning to Rank (LtoR) (Liu, 2009)*

- ➔ *is a task to automatically construct a ranking model*
- ➔ *using training data, such that*
- ➔ *the model can sort new objects according*
- ➔ *to their degrees of relevance, preference, or importance.*

➔ Motivated by the large volume of information available

- ➔ Challenge in ranking millions of documents
- ➔ Opportunity to use existing data for ranking

Ranking Problems

- Many information retrieval problems are by nature ranking problems, such as:
  - document retrieval,
  - collaborative filtering,
  - key term extraction,
  - definition finding.
- Many different ranking scenarios:
  - Rank document purely according to their relevance with regards to the query.
  - Considering the relationships of similarity and diversity between documents (e.g. relational ranking).
  - Aggregate several candidate ranked lists (e.g. meta search, unified ranking).
  - Find to what degree a document property influences the ranking result.

Ranking in Information Retrieval

- Conventional document ranking models:
  - Query-dependent models:
    - Boolean model, set based model (no degree of relevance)
    - Vector space model, using term weighting such as TF-IDF.
    - Probabilistic models, such as language models and BM25.
  - Query-independent models:
    - Link-based models for the web, such as PageRank.

Learning to Rank

➔ Many of the existing ranking models contain parameters that need to be tuned.

➔ Different ranking models can be combined to create a new ranking.

➔ In addition, models perfectly tuned using the validation set do not necessary perform well on unseen queries.

➔ How to define and combine parameters and models to produce an effective ranking?

➔ Machine learning methods have demonstrated their effectiveness in automatically tuning parameters combining multiple evidences.

➔ Learning to rank is the application of machine learning methods to information retrieval problems.

Typical Learning to Rank Flow

- The training data consists of:
  - n queries (q1, q2, …),
  - their associated documents represented as features vectors (x1, x2, …),
  - and the corresponding relevance judgements.
- A learning algorithm is employed to learn the ranking model h.
- The ranking model h is then evaluated with standard IR measures by producing ranked lists for the training data.

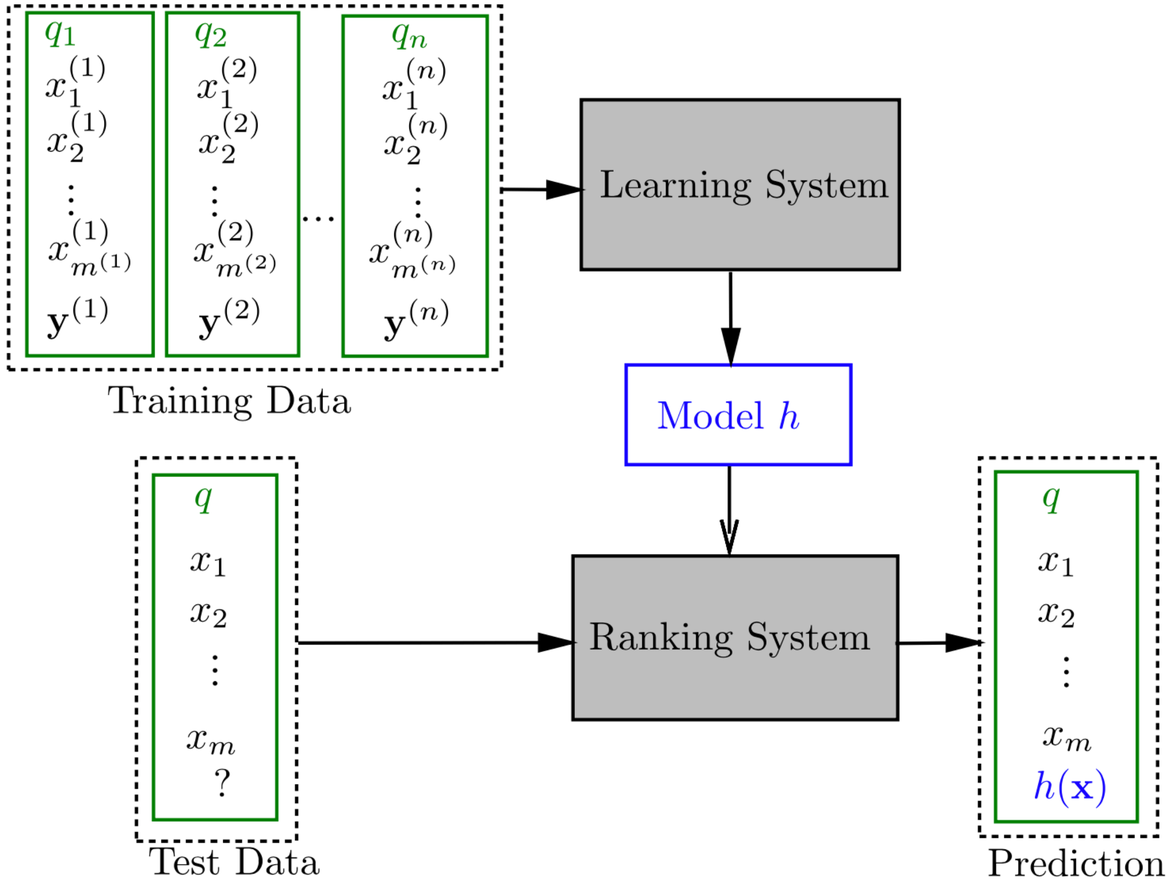

Fig. 1.1 Learning-to-rank framework.

Image from Learning to Rank for Information Retrieval, Tie-Yan Liu (2009)

Popular Features

Table 6.2 Learning features of TREC.

ID	Feature description
1	Term frequency (TF) of body
2	TF of anchor
3	TF of title
4	TF of URL
5	TF of whole document
6	Inverse document frequency (IDF) of body
7	IDF of anchor
8	IDF of title
9	IDF of URL
10	IDF of whole document
11	TF*IDF of body
12	TF*IDF of anchor
13	TF*IDF of title
14	TF*IDF of URL
15	TF*IDF of whole document
16	Document length (DL) of body
17	DL of anchor
18	DL of title
19	DL of URL
20	DL of whole document
21	BM25 of body
22	BM25 of anchor
23	BM25 of title

Table 6.2 (*Continued*)

ID	Feature description
40	LMIR.JM of whole document
41	Sitemap based term propagation
42	Sitemap based score propagation
43	Hyperlink base score propagation: weighted in-link
44	Hyperlink base score propagation: weighted out-link
45	Hyperlink base score propagation: uniform out-link
46	Hyperlink base feature propagation: weighted in-link
47	Hyperlink base feature propagation: weighted out-link
48	Hyperlink base feature propagation: uniform out-link
49	HITS authority
50	HITS hub
51	PageRank
52	HostRank
53	Topical PageRank
54	Topical HITS authority
55	Topical HITS hub
56	Inlink number
57	Outlink number
58	Number of slash in URL
59	Length of URL
60	Number of child page
61	BM25 of extracted title
62	LMIR.ABS of extracted title
63	LMIR.DIR of extracted title
64	LMIR.JM of extracted title

Table from Learning to Rank for Information Retrieval, Tie-Yan Liu (2009)

Learning to Rank Approaches

- ➔ The learning to rank process can be modeled in different ways.
  - ➔ Pointwise approach
  - ➔ Pairwise approach
  - ➔ Listwise approach

- ➔ Different learning techniques can be employed, including SVM, Boosting, Neural Nets, etc.

- ➔ Effectiveness from literature: Listwise > Pairwise > Pointwise
- ➔ Different methods can be combined with successful results.

Pointwise Approach

➔ Input: feature vectors for single documents, e.g. scores for query-document pairs.

➔ Output: relevance degree of each single document (e.g., relevant / non-relevant).

➔ Methods: OC SVM, McRank, Prank.

➔ Treat the problem as a regression problem, i.e. estimate a continuous variable (i.e. each document's score).

➔ Straightforward approach.

➔ Ambitious goal — produce document scores, but only rankings are needed.

Pairwise Approach

- ➔ Pairwise classification problem, i.e. decide pairwise preferences for documents.
- ➔ Input: pairs of documents, both represented as feature vectors.
  - ➔ Manually annotated data, e.g. (q,d,d') — d is more relevant to q than d' (harder to get, high quality).
  - ➔ Log data, e.g. if, for query q, document d and d' are listed, and user clicked d and not d', then (q,d,d') can be inferred (easy, lower quality).
- ➔ Output: pairwise preferences (ranging from 1 to -1) between document pairs.

- ➔ The problem is treated as a classification problem, i.e. given a query and two documents, determine which is better.
- ➔ Methods: Ranking SVM, LambdaMART, LambdaRank

Listwise Approach

- ➔ Input: entire group of documents associated with a query, i.e. ranking lists.
  - ➔ Offline: obtain relevance judgements (q,d,s), where s is a score indicating the relevance of document d to query q.
  - ➔ Online: present different rankings to users (or interleave lists) and observe (from logs) which users prefer.
- ➔ Output: full document predicted ranking for query.
- ➔ Methods: PermuRank, AdaRank, SoftRank.

- ➔ Problem is modeled in a more natural “IR” way but not directly adaptable to conventional machine learning techniques.
- ➔ Best performing approaches.

Reference Datasets for Learning to Rank

- ➔ LETOR: Learning to Rank for Information Retrieval
  - ➔ www.microsoft.com/research/project/letor-learning-rank-information-retrieval
- ➔ Microsoft Learning to Rank Datasets
  - ➔ www.microsoft.com/research/project/mslr

Support in Solr

➔ Solr includes support for Learning to Rank.

➔ https://solr.apache.org/guide/learning-to-rank.html

  ➔ Includes a step-by-step practical example.

References

➔ Liu, Tie-Yan. Learning to rank for information retrieval. Foundations and Trends in Information Retrieval 3.3 (2009): 225-331.