Modern Information Retrieval

Learning to rank

Hamid Beigy

Sharif university of technology

December 19, 2022

Table of contents

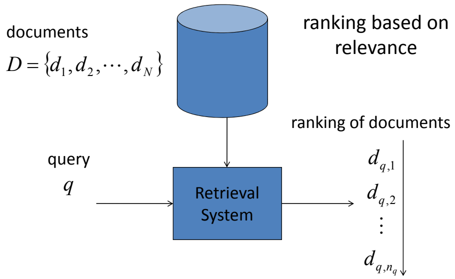
Introduction

Introduction

1. There are many tasks in information retrieval (IR) and natural language processing (NLP), for which the central problem is ranking.
2. These include
   - document retrieval,
   - entity search,
   - question answering,
   - meta-search,
   - personalized search,
   - online advertisement,
   - collaborative filtering,
   - document summarization, and
   - machine translation.

Document retrieval

1. Document retrieval includes web search, enterprise search, desktop search.
2. Document retrieval can be described as the following task in which ranking plays a key role.
3. The retrieval system maintains a collection of documents.
4. Given a query from the user, the system retrieves documents containing the query words from the collection, ranks the documents, and presents the top ranked list of documents to the user.
5. Ranking is performed mainly based on the relevance of the documents with respect to the query.



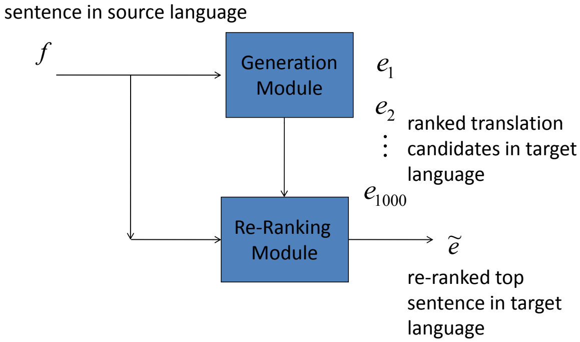
Collaborative filtering

1. Collaborative filtering is the most fundamental model for computer systems to make recommendations to users in electronic commerce, online advertisement.
2. For example, if the users' preferences on some of the movies in a database are known, then we can employ collaborative filtering to recommend to the users movies which they might have not seen and might be interested in.

	Item1	**Item2**	**Item3**	**...**	**ItemN**
User1	5	4			
User2	1		2		2
...		?	?	?	
UserM	4	3			

3. The question is how to determine the unknown elements of the matrix.
4. One assumption is: similar users may have similar ratings on similar items.
5. When a user is specified, the system suggests a ranking list of items with the high grade items on the top.

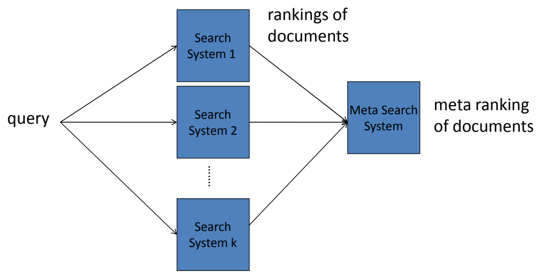
1. Machine translation can help people to access information cross languages and thus is very important.
2. Given a sentence in the source language, usually, there are a large number of possible translations (sentences) in the target language.
3. The quality of translations can vary, however. How to select the most plausible translation(s) is the key question.
4. A popular approach to machine translation consists of two phases: candidate generation and re-ranking.



1. A popular approach to machine translation consists of two phases: candidate generation and re-ranking.
2. Given a sentence in the source language,
   - the system first generates and ranks all possible candidate translations in the target language using a generative model.
   - then it conducts re-ranking on the top candidate translation using a discriminative model, and,
   - finally, it chooses the top ranked candidate as output.
3. The re-ranking process is performed based on the likelihood of candidates' being good translations, and it is critical to the performance of machine translation.

Meta-search system

1. A meta-search system is a system that sends the user's request to several search systems and aggregates the results from those search systems.
2. Meta-search is becoming more and more important when web continues to evolve, and more and more search systems become available.
3. In meta-search, the query is submitted to several search systems and ranking lists of documents are returned from the systems.
4. Meta-search system then combines all the ranked lists and generates a new ranked list, which is better than all the individual ranking lists.
5. The sets of documents returned from different systems can be different. One can take the union of the sets of documents as the final set of documents.



Learning to rank

1. Recently, a new area called learning to rank has emerged in intersection of
   - machine learning,
   - information retrieval,
   - natural language processing
2. Learning to rank is about performing ranking using machine learning techniques.
3. It is based on previous work on ranking in machine learning and statistics, and it also has its own characteristics.

Rank creation

1. Let $Q = \{q_1, q_2, \ldots, q_M\}$ be the set of requests/queries.
2. Let $O = \{o_1, o_2, \ldots, o_N\}$ be the set of offerings/documents.

Application	The set of requests	The set of offerings
Document retrieval	a set of queries	a set of documents
Collaborative filtering	a set of users	a set of items
Machine translation	a set of source sentences	a set of target sentences

3. Note that Q and O can be infinite sets.
4. Given a $q \in Q$ and a $o \in O$, rank elements in Os using q and o.

Definition (Ranking using scoring function)

Ranking using ranking/scoring function is $f(q, o) : Q \times O \mapsto \mathbb{R}$ with

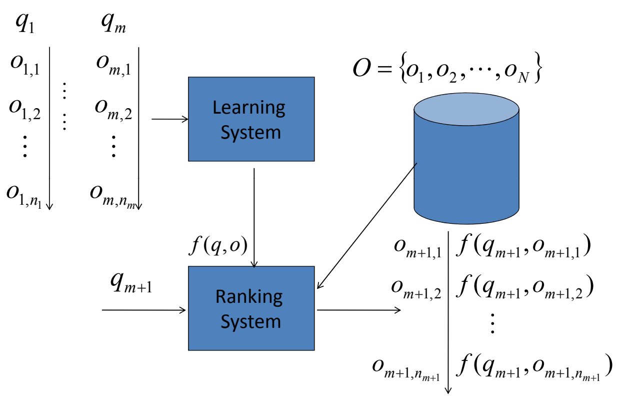
$$s_o = f(q, o)$$
$$\pi = Sort_{s_o, o \in O}(O)$$

where s_o is score of o, and π is a ranking list.

Learning for ranking creation

1. When learning to rank is mentioned, it usually means ranking creation using supervised learning.
2. There are two systems: a learning system and a ranking system.
3. The learning system takes training data as input.
4. The training data consists of requests and their associated ranking lists of offerings.
   - For each request $q_i \in \{q_1, \ldots, q_m\}$, there is
   - an associated set of offerings $O_i \in \{O_1, \ldots, O_m\}$, where $O_i = \{O_{i,1}, \ldots, O_{i,n_i}\}$, and
   - true ranking list on the offerings $\pi_i \in \{\pi_1, \ldots, \pi_m\}$.
5. The learning system constructs a ranking model, $f(q, o)$ on the basis of the training data.

1. The ranking system then makes use of the learned ranking model for ranking prediction.
2. Given a new request q_{m+1},
   - the ranking system receives a subset of offerings O_{m+1},
   - assigns scores to the offerings using the ranking model,
   - and sorts the offerings in descending order of the scores, obtaining a ranking list π_{m+1}.



Learning task

1. Learning for ranking creation is comprised of training and testing, as a supervised learning task.
2. The training data contains queries and documents.
   - Each query is associated with a number of documents.
   - The relevance of the documents with respect to the query is also given.
   - Here, we assume that the relevance of a document with respect to a query is represented by a label.
   - The labels are at several grades (levels). The higher grade a document has, the more relevant the document is.

1. Suppose that Q is the query set and D is the document set.
2. Suppose $Y = \{1, 2, \ldots, l\}$ is label set, where labels represent grades.
3. A total order exists between grades $l \succ l-1 \succ \ldots \succ 1$, where $\succ$ is order relation.
4. Let $\{q_1, q_2, \ldots, q_m\}$ be set of queries for training and q_i is i-th query.
5. Let $D_i = \{d_{i,1}, d_{i,2}, \ldots, d_{i,n_i}\}$ be set of documents associated to query q_i and $y_i = \{y_{i,1}, y_{i,2}, \ldots, y_{i,n_i}\}$ is set of labels associated to query q_i, where
   - n_i is sizes ofD_i and y_i;
   - $d_{i,j}$ is j-th document in D_i; and
   - $y_{i,j} \in Y$ is j-th grade label in y_i, showing relevance degree of $d_{i,j}$ with respect to q_i.
6. The original training set is denoted as $S = \{(q_i, D_i), y_i\}_{i=1}^{m}$.

Learning task

1. The original training set is denoted as $S = \{(q_i, D_i), y_i\}_{i=1}^{m}$.
2. A feature vector $\mathbf{x}_{ij} = \phi(q_i, d_{i,j})$ is created for query-document pair $(q_i, d_{i,j})$.
3. Now, the training set becomes $S' = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$.
4. We aim to train a ranking model $f(q, d) = f(\mathbf{x})$ that can assign a score to a given query document pair q and d, or equivalently to given feature vector $\mathbf{x}$.
5. The learning to rank methods can be categorized as the point-wise approach, pair-wise approach, and list-wise approach.

Point-wise approach

1. In the point-wise approach, the ranking problem is transformed to classification, or regression and existing methods for classification, or regression are applied.
2. Suppose that the learned model $f(\mathbf{x})$ outputs real numbers.
3. Then, given a query, we can use the model to rank documents (sort documents according to the scores given by the model).
4. The loss function in learning is point-wise in the sense that it is defined on a single object (feature vector).
5. Example algorithms: Prank, OC SVM, McRank, and Subset Ranking.

Pair-wise approach

1. In the pair-wise approach, ranking is transformed into pairwise classification or pairwise regression.
2. From the labeled data of query q_i, $(\mathbf{x}_{i,1}, y_{i,1}), \ldots, (\mathbf{x}_{i,n_i}, y_{i,n_i})$, it creates preference pairs of feature vectors (documents).
3. For example, if $\mathbf{x}_{i,j}$ has a higher grade than $\mathbf{x}_{i,k}$ $(y_{i,j} > y_{i,k})$, then $\mathbf{x}_{i,j} \succ \mathbf{x}_{i,k}$ becomes a preference pair.
4. We can create the training set as
   - $(\mathbf{x}_i, \mathbf{x}_j, +1)$, i.e. $\mathbf{x}_i \succ \mathbf{x}_j$.
   - $(\mathbf{x}_i, \mathbf{x}_j, -1)$, i.e. $\mathbf{x}_i \prec \mathbf{x}_j$.
   - $(\mathbf{x}_i, \mathbf{x}_j, 0)$, i.e. no preference.
5. From the above training set, scoring function $f(\mathbf{x})$ is learned.
6. Example algorithms: Ranking SVM, RankBoost, RankNet, IR SVM, GBRank, Frank, LambdaRank, and LambdaMART.

List-wise approach

1. The list-wise approach addresses the ranking problem in a more natural way.
2. Specifically, it takes ranking lists as instances in both learning and prediction.
3. The group structure of ranking is maintained and ranking evaluation measures can be more directly incorporated into the loss functions.
4. It views the labeled data $(x_{i,1}, y_{i,1}), \ldots, (x_{i,n_i}, y_{i,n_i})$ associated with query q_i as one instance.
5. The approach learns a ranking model $f(x)$ from the training data that can assign scores to feature vectors and rank the feature vectors using the scores, such that feature vectors with higher grades are ranked higher.
6. This is a new problem for machine learning and conventional techniques in machine learning cannot be directly applied.
7. Example algorithms: ListNet, ListMLE, AdaRank, SVM MAP, Soft Rank, and AppRank.

Learning to rank evaluation

Table 2.10: NDCG on TD2003 Dataset

Method	NDCG@1	NDCG@3	NDCG@5	NDCG@10
Regression	0.32	0.31	0.30	0.33
Ranking SVM	0.32	**0.34**	**0.36**	**0.35**
RankBoost	0.28	0.32	0.31	0.31
FRank	0.30	0.27	0.25	0.27
ListNet	0.40	0.34	0.34	**0.35**
AdaRank-MAP	0.26	0.31	0.30	0.31
AdaRank-NDCG	**0.36**	0.29	0.29	0.30
SVMMAP	0.32	0.32	0.33	0.33

Learning for ranking aggregation

Learning for ranking aggregation

1. Ranking aggregation is aimed at combining multiple rankings into a single ranking, which is better than any of the original rankings in terms of an evaluation measure.
2. Learning for ranking aggregation is about building a ranking model for ranking aggregation using machine learning techniques.
3. In meta-search, the query from the user is sent to multiple search systems, and the ranking lists from the search systems are then combined and presented to the user in a single ranking list.
4. Since the ranking lists from individual search systems may not be accurate enough, meta-search actually takes a majority voting over search ranking lists.
5. The question is then how to effectively perform the majority voting.
6. Here we call the rankings from individual search systems basic rankings, and the ranking in meta search final ranking.

Learning for ranking aggregation

1. Learning for ranking aggregation can be performed either as unsupervised learning or supervised learning.
2. In traditional IR, ranking aggregation is usually based on unsupervised learning.
3. Recently, supervised methods for ranking aggregation have also been proposed.
4. In supervised learning for ranking aggregation, the training data contains
   - queries,
   - their associated documents,
   - basic rankings on the documents, and
   - the corresponding final rankings.
5. The testing data includes query, associated documents, and basic rankings on the documents.

Learning for ranking aggregation

1. Suppose Q is the query set, and D is the document set.
2. Suppose $\{q_1, q_2, \ldots, q_m\}$is the set of queries in training data.
3. Suppose $D_i = \{d_{i,1}, d_{i,2}, \ldots, d_{i,n_i}\}$ is the set of documents associated with query q_i.
4. Suppose $\Sigma_i = \{\sigma_{i,1}, \sigma_{i,2}, \ldots, \sigma_{i,k}\}$ is the set of basic rankings on the documents in D_i with respect to query q_i.
5. Suppose π_i is the final ranking on the documents in D_i with respect to query q_i.
   - $d_{i,j}$ denotes the j-th document in D_i,
   - $\sigma_{i,j}$ denotes the j-th basic ranking in Σ_i,and
   - k denotes the number of basic rankings.
6. The training set is represented as $S = \{(q_i, \sigma_i), \pi_i\}_{i=1}^{m}$.

Definition (Learning model for ranking aggregation)

In learning, a model for ranking aggregation is constructed, which takes the form of $f(q, \Sigma) : Q \times \Pi^k \mapsto \mathbb{R}^n$, where

- q is a query,
- D is a set of associated documents,
- Σ is a set of basic rankings on the documents in D with respect to q,
- n denotes the number of documents, and
- k denotes the number of basic rankings.

$f(q, \Sigma)$ assign scores to the documents in D, sort the documents according to the scores, and generate a final ranking.

$$
\begin{aligned}
s_D &= f(q, \Sigma) \\
\pi &= sort_{s_D}(D)
\end{aligned}
$$

Learning for ranking aggregation

Existing methods for ranking aggregation includes

unsupervised learning methods Borda Count and Markov Chain

supervised learning methods Cranking

References

Reading

1. Chapters 1 through 3 of Learning to Rank for Information Retrieval and Natural Language Processing[1]

[1] Hang Li (2011). *Learning to Rank for Information Retrieval and Natural Language Processing*. Morgan & Claypool Publishers.

References

Li, Hang (2011). *Learning to Rank for Information Retrieval and Natural Language Processing*. Morgan & Claypool Publishers.

Questions?