Methodological Approach: Ranking from Information retrieval

Anuj Kumar *Manish Kumar* *Mayank Saxena*

ABSTRACT

Emerging central problem of Information Retrieval is to determine which documents are relevant and which are not to the information need. This problem is handled by a ranking function. In this article, various approaches for learning a ranking function are discussed. Also a comparative analysis is performed which covers pros and cons of these approaches.

Introduction

Information is being created and becoming available in ever growing quantities as the access possibilities to it proliferates. There is currently a great deal of excitement and confusion about the promise of an Electronic Information Superhighway that would enable anybody to access these diverse and large information sources. Many information providers are developing on-line services to provide users with an interface to this emerging rich universe of knowledge stored in the form of multimedia documents, business and financial data, games and entertainment, shopping and consumer information. However, it is not possible to make information available to users almost instantly without better methods to filter, retrieve and manage this potentially unlimited influx of information. Users face an information overload problem and they require tools to explore this vast universe of information in a structured way. So, storage and retrieval of information in a convenient manner is of utmost importance.

2. What is IR?

"Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)".Information Retrieval is different from data retrieval. Data retrieval mainly consists of determining which documents of a collection contain the keywords in the user query which is not enough to satisfy the user information need. In fact, the user of an IR system is concerned more with retrieving information about a subject than with retrieving data which satisfies a given query. For an Information Retrieval system, the retrieved object might be inaccurate and small errors are likely to go unnoticed but for a data retrieval system, however a single erroneous object among a thousand retrieved objects means total failure.

3. Learning to Rank

One central problem of information retrieval (IR) is to determine which documents are relevant and which are not to the information need. This problem is practically handled by a ranking function which defines an ordering among documents according to their degree of relevance to the user query. The process of generating an effective ranking function for IR is referred to as "Learning to Rank for IR" in the field.[2]

The task of "learning to rank" has emerged as an active and growing area of research both in information retrieval and machine learning. The goal is to design and apply methods to automatically learn a function from training data, such that the function can sort objects (e.g., documents) according to their degrees of relevance, preference, or importance as defined in a specific application.

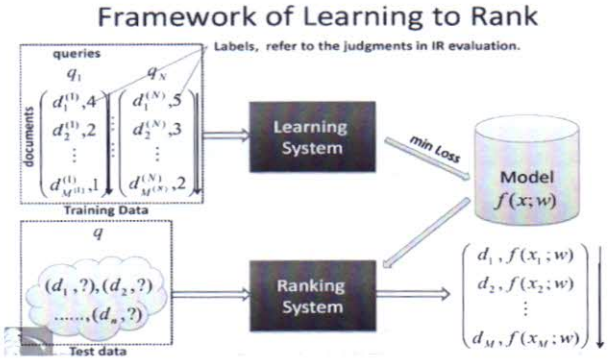

Figure 1: General Framework of Learning- Based methods for IR ranking problem

Figure 1 shows a general framework that most learning-based methods follow to deal with IR ranking problem. The learning process, formalized as follows, consists of two steps: training and test.

Given a query collection Q = { q_1,....., q_N } and a document collection , D = { d_1, ... , d_M } , the training corpus is created as a set of query-document pairs , each (q_i , d_j) ϵ Q × D, upon which a relevance judgment indicating the relationship between q_i and d_j is assigned by a labeler . The relevance judgment given by a label can be:

1) A class label eg. relevant or non-relevant,
2) Rating, eg. definitely relevant, possibly relevant, or non-relevant,
3) An order, eg. k, meaning that d_j is ranked in the kth position of the ordering of all documents when qi is considered,
4) A score, eq. sim(q_i, d_j) specifying the degree of relevance between q_i and d_j.

For each instance (q_i, d_j), a feature extractor produces a vector of features that describe the match between qi and dj. The inputs to the learning algorithm comprise training instances, their feature vectors and the corresponding relevance judgments. The output is a ranking function, f, where f (qi, dj) is supposed to give the "true" relevance judgment for q_i and d_j. During the training process, the learning algorithm attempts to learn a ranking function such that a performance measure (eg. MAP, error rate, NDCG etc.) with respect to the output relevance judgment can be optimized.

In the test phase, the learned ranking function is applied to determine the relevance between each document d_i in D and a new query q. Clearly, factors, such as the form of the training instances, the form of the desired output, and the performance measure, will lead to different design of learning to rank for IR algorithms.

4. Ranking Techniques

In learning to rank for information retrieval, a training set of queries and their associated documents (with relevance judgments) are provided. The ranking model is trained with the data in a supervised fashion, by minimizing certain loss functions. For ranking, the model is applied to new queries and sorts their associated documents. Three major approaches have been proposed, i.e., point-wise, pair-wise and list-wise approaches to learning to rank. Each approach has its pros and cons. [3]

Different approaches for Learning to Rank are:

4.1 Point-wise Approach

The point-wise approach solves the problem of ranking by transforming it to regression, classification or ordinal regression.

E.g. Prank-ing, Ranking with Large Margin Principles etc.

4.2 Pair-wise Approach: The pair-wise approach transforms ranking to classification on document pairs. It takes pairs of documents and their relative preferences as training instances and attempt learning to classify each object pair into correctly ranked or incorrectly ranked.

E.g. Ranking SVM, Rank Boost, RankNet and many more.

4.3 List-wise Approach: The list-wise approach tackles the ranking problem directly, by adopting list-wise loss functions, or optimizing IR evaluation measures. It treats the list of documents associated with the same query as learning instance to obtain rank (position).

Query level information. It takes document collection with respect to query as input space : $\{X_1^{(q)}, \ldots, X_{M(q)}^{(q)}\} \epsilon (R^T)^{M(q)}$ and produces permutation of these documents as output space : $Y \epsilon \Pi_{M(q)}$

E.g. ListNet, RankGP.

Point wise approach is the simplest technique with O(n) complexity which can use existing theories algorithms on regression and classification. But the problem is that it is not obvious to directly compare two documents for the same query. Pair wise technique solves this difficulty of Point wise approach in which an instance is one document pair. The document pair means that for a given query, two returned documents are taken into consideration, so that the two documents can be compared and it is easy to decide the relative position of the documents. This method also has some problems:

It ignores the fact that ranking is a prediction task on list of objects. It formalizes the problem of learning to rank as that of classification. Specifically, in learning it collects document pairs from the ranking lists, and for each document pair it assigns a label representing the relative relevance of the two documents. It then trains a classification model with the labeled data and makes use of the classification model in ranking. Here the objective of learning is formalized as minimizing errors in classification of document pairs, rather than minimizing errors in ranking of documents. Its complexity is O(n2).

	Pointwise Approach	Pairwise Approach	Listwise Approach
No. Of instances	Equal to no. of training elements	Equal to half of the training elements	Take whole list as training instance
Implementation complexity	O(n)	$O(n^2)$	More complex & difficult to implement Time complexity depends on surrogate loss used
Training time	More	Less	Less
Characteristic	More suitable to ordinal regression	More suitable to Learning to rank	More suitable to Learning to rank
Technique	Transform ranking to	Transform ranking to pairwise classificatio	Straightforwardly represent Learning

Finally, Listwise approach came that takes ranked lists of objects as instances & trains a ranking function through the minimization of a Listwise loss function defined on the predicted list and the ground truth list.

Table 1 : Comparative analysis of various approaches

5. Conclusion

In this article, the task of learning to rank for information retrieval is described. Effective ranking function could be generated using various approaches by learning a function from the training data. Comparative analysis of various approaches is performed which shows list wise approach is better