Information Processing and Management 000 (2016) 1–17

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Quality versus efficiency in document scoring with learning-to-rank models

Gabriele Capannini[a], Claudio Lucchese[b,*], Franco Maria Nardini[b], Salvatore Orlando[c], Raffaele Perego[b], Nicola Tonellotto[b]

[a] Innovation Design och Teknik (IDT), Mälardalens högskola, Västerås, Sweden
[b] Istituto di Scienza e Tecnologie dell'Informazione (ISTI) of the National Research Council of Italy (CNR), Pisa, Italy and Istella Srl, Cagliari, Italy
[c] University Ca' Foscari of Venice, Italy

ARTICLE INFO

Article history:
Received 30 June 2015
Revised 16 March 2016
Accepted 8 May 2016
Available online xxx

Keywords:
Efficiency
Learning-to-rank
Document scoring

ABSTRACT

Learning-to-Rank (LtR) techniques leverage machine learning algorithms and large amounts of training data to induce high-quality ranking functions. Given a set of documents and a user query, these functions are able to precisely predict a score for each of the documents, in turn exploited to effectively rank them. Although the scoring efficiency of LtR models is critical in several applications – e.g., it directly impacts on response time and throughput of Web query processing – it has received relatively little attention so far.

The goal of this work is to experimentally investigate the scoring efficiency of LtR models along with their ranking quality. Specifically, we show that machine-learned ranking models exhibit a quality versus efficiency trade-off. For example, each family of LtR algorithms has tuning parameters that can influence both effectiveness and efficiency, where higher ranking quality is generally obtained with more complex and expensive models. Moreover, LtR algorithms that learn complex models, such as those based on forests of regression trees, are generally more expensive and more effective than other algorithms that induce simpler models like linear combination of features.

We extensively analyze the quality versus efficiency trade-off of a wide spectrum of state-of-the-art LtR, and we propose a sound methodology to devise the most effective ranker given a time budget. To guarantee reproducibility, we used publicly available datasets and we contribute an open source C++ framework providing optimized, multi-threaded implementations of the most effective tree-based learners: Gradient Boosted Regression Trees (GBRT), Lambda-Mart (λ-MART), and the first public-domain implementation of Oblivious Lambda-Mart (Ω_λ-MART), an algorithm that induces forests of oblivious regression trees.

We investigate how the different training parameters impact on the quality versus efficiency trade-off, and provide a thorough comparison of several algorithms in the quality-cost space. The experiments conducted show that there is not an overall best algorithm, but the optimal choice depends on the time budget.

* Corresponding author.
E-mail addresses: gabriele.capannini@mdh.se (G. Capannini), claudio.lucchese@isti.cnr.it, c.lucchese@isti.cnr.it (C. Lucchese), f.nardini@isti.cnr.it (F.M. Nardini), orlando@unive.it (S. Orlando), r.perego@isti.cnr.it (R. Perego), n.tonellotto@isti.cnr.it (N. Tonellotto).

http://dx.doi.org/10.1016/j.ipm.2016.05.004
0306-4573/

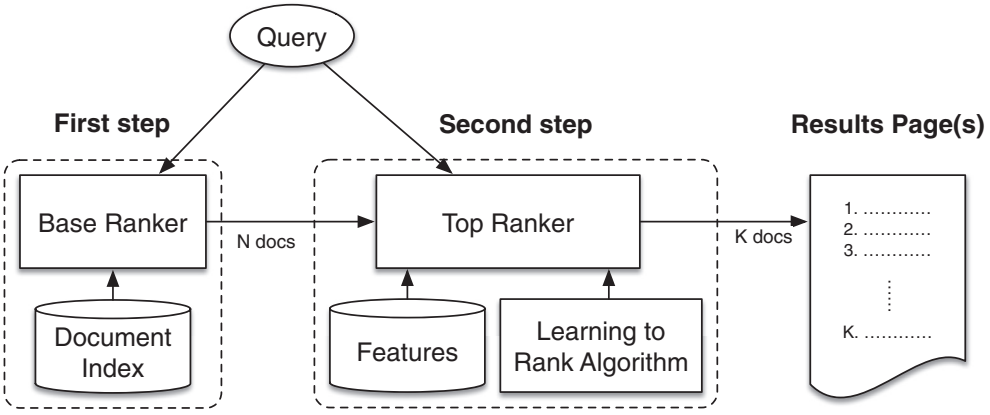

Fig. 1. The architecture of a generic machine-learned ranking pipeline.

1. Introduction

Ranking is a central task of many information retrieval problems, in particular for document retrieval where documents must be ranked according to their relevance to a user query. Indeed, ranking is particularly challenging for large-scale Web Search Engines (WSEs), since it involves effectiveness requirements and efficiency constraints that are not common to other ranking-based applications.

From an effectiveness point of view, a number of machine learning algorithms have been proposed to automatically build high-quality ranking functions able to exploit a multitude of features characterizing the candidate documents and the user query. These algorithms fall under the Learning-to-Rank (LtR) framework (Liu, 2009). The *models*, or *rankers*, generated by such methods are generally quite expensive to use for ranking large sets of documents. For example, methods based on forests of regression trees may generate thousands of trees to be evaluated on hundreds of features modeling a single query-document pair, in order to predict scores used to effectively rank all the candidate documents for a given query (Cambazoglu et al., 2010; Segalovich, 2010). Therefore, even if LtR models are able to provide high quality results, it is not possible to apply such rankers to all the documents matching a user query due to the resulting prohibitive ranking cost.

To overcome this issue, WSEs usually exploit multi-stage ranking architectures (Fig. 1), where top-K retrieval is carried out by a two-step process: (i) candidate retrieval and (ii) candidate re-ranking. The first step retrieves from the inverted index N possibly relevant documents matching the user query, where $N \gg K$. This phase aims at optimizing the recall of the retrieval system, and is usually achieved by a simple and fast *base ranker*, e.g., BM25 combined with some document-level scores (Robertson & Zaragoza, 2009). The assumption is that the base ranker is able to retrieve a large part of the most relevant documents, even it is not able to effectively rank them. In the second step, a complex scoring function is used by the *top ranker* to re-rank the candidate documents coming from the first step. The top ranker is optimized for high precision, i.e., to place the most relevant results in the top positions of the first page of results. LtR models are commonly used in this second step to achieve the desired precision of the top ranker.

Therefore, the top ranker is a crucial component for both effectiveness and efficiency of WSEs. First, the top ranker determines the quality of the results presented to the user. Second, it impacts on the response time of the WSE. We know that both quality and response time largely impact on the click behavior of users (Arapakis, Bai, & Cambazoglu, 2014), and ultimately on the user satisfaction and WSE revenue. Devising a good trade-off between efficiency and effectiveness is thus very important.

This paper investigates such effectiveness vs. efficiency trade-offs. We believe that the problem of devising the right trade-off between the quality and the computational cost of LtR models at query processing time has not yet received enough attention from the Machine Learning (ML) and Information Retrieval (IR) communities. Traditionally, the ML community focused primarily on the accuracy of the learned model, or the scalability of the training phase, while the efficiency of the application of the learned model was considered as unimportant or negligible. On the other hand, strongly motivated by budget considerations that are very important for commercial WSEs, the IR community has started only recently to investigate low-level optimizations to reduce the execution time of some families of LtR rankers. The computational cost of the LtR models must be in fact strictly accounted in the time budget available for processing queries in the incoming stream, as it can impact to a large extent on the throughput of the system. In addition, since each family of algorithms has tuning parameters that can influence both effectiveness and efficiency (e.g., number of trees in tree-based models), even for a given family of algorithms a change in the setting of the parameters can have a deep impact on the performance of the learned model.

Please cite this article as: G. Capannini et al., Quality versus efficiency in document scoring with learning-to-rank models, Information Processing and Management (2016), http://dx.doi.org/10.1016/j.ipm.2016.05.004