Is Learning to Rank Worth it?

A Statistical Analysis of Learning to Rank Methods

Guilherme de Castro Mendes Gomes, Vitor Campos de Oliveira,

Jussara Marques de Almeida, Marcos André Gonçalves

Universidade Federal de Minas Gerais, Brazil
gcm.gomes@dcc.ufmg.br, vitorcoliveira@gmail.com, jussara@dcc.ufmg.br, mgoncalv@dcc.ufmg.br

Abstract. The Learning to Rank (L2R) research field has experienced a fast paced growth over the last few years, with a wide variety of benchmark datasets and baselines available for experimentation. We here investigate the main assumption behind this field, which is that, the use of sophisticated L2R algorithms and models, produce significant gains over more traditional and simple information retrieval approaches. Our experimental results surprisingly indicate that many L2R algorithms, when put up against the best individual features of each dataset, may not produce statistically significant differences, even if the absolute gains may seem large. We also find that most of the reported baselines are statistically tied, with no clear winner.

Categories and Subject Descriptors: H. [Information Storage and Retrieval]

Keywords: Information Retrieval, Learning to Rank, Statistical Analysis

1. INTRODUCTION

Over the last few years, Learning to Rank (L2R) has become a very popular research topic, based on the general and well-accepted assumption that it produces a much better performance than traditional ranking methods, such as BM25 [1] or Language Based Models [2], in information retrieval tasks. Indeed, several new L2R methods [3] and benchmark datasets, including large ones such as the LETOR repository [7], have been developed and made available to the community, in recent years.

However, the development and efficient employment of such methods are not free of costs. Being based on supervised learning, they require labeled datasets in order to properly learn the ranking functions. Moreover, these datasets should be large and heterogeneous enough to be capable of representing the domains upon which they will be applied. Due to such strict requirements, constructing such datasets is not a trivial task. In fact, it is very costly. After building the required data, an usually very computationally demanding learning phase has to be applied to learn the ranking functions, which may also require an expensive parameter tuning for optimal performance. Finally, the use of such functions, in production mode in real search engines for example, is usually a two-stage process, in which traditional methods are first applied and, in a subsequent step, the more expensive learned function is used to re-rank the top results generated by the first step [4]. This implies in an additional overhead to produce query answers.

Given all these issues, as well as the continuous advance and interest in the area, we here take a step back and reevaluate the main assumption upon which Learning to Rank built its foundations, which is that, the use of sophisticated L2R algorithms and models, produce significant gains over more traditional and simple information retrieval approaches. We also investigate, among the many L2R algorithms that have been proposed in the literature, if there is one or more that deliver superior effectiveness in most situations (e.g., different collections, different tasks, etc). In order to do so, we analyze the results for 12 baselines over 6 large datasets of the LETOR 3.0 benchmark [7], as well as 5 baselines over 2 even larger datasets of the LETOR 4.0 benchmark [7], when put up against simple isolated feature rankers, using statistically significant tests. All the datasets and baseline results are available at the benchmark's web page [7]. Our

goal is to verify whether the effectiveness of these methods is better than produced with the best feature of each dataset when used in isolation, as given by some measure of ranking quality (e.g., Mean Average Precision). We also contrast the performance of each method against each other using the same statistical methodology and datasets. To our knowledge no previous work has performed such detailed comparison with a rigorous statistical analysis.

Our experimental results show that: (1) in most datasets, the best single feature, ranked by Mean Average Precision (MAP), produces results that are statistically tied to most of the reported baselines; (2) the absolute differences in effectiveness provided by the L2R algorithms, when compared to single feature rankers, may be large, but, in most cases, not statistically significant; and (3) almost all the baselines have very similar performances, making it unlikely that there is an overall best L2R method. Therefore a clear advantage of L2R solutions may not be confirmed in all situations, mainly considering the costs involved.

The remainder of this paper is organized as follows: Section 2 describes work related to this paper; Section 3 details our experiments and analyses; Section 4 reports our results; Section 5 concludes the paper and describes our future work.

2. RELATED WORK

Despite the great interest in Learning to Rank in recent years, most of the related work focuses on proposing new algorithms for ranking or novel applications of existing ones. After the publication of the LETOR dataset [8], very few studies were made concerning the effects of the public datasets on the task of learning to rank.

In [9], the authors observed that the ways in which documents were selected for each topic of the LETOR benchmark presented on [8] show that the selection has (for each of the three corpora) a particular bias or skewness. This observation has some unexpected effects that may considerably influence any learning-to-rank exercise conducted on these datasets. However, most of these problems were explained and corrected by the benchmark's authors in [10]. Finally, in [11], a comparison of 7 learning to rank algorithms is made on the LETOR 3.0 benchmark. Each algorithm is compared with each other in terms of Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG). In comparison with this previous study, we here compare 12 learning to rank algorithms against not only each other but also against using the best single feature in various datasets of the LETOR 3.0 and 4.0 benchmarks. Moreover we use statistical tests to support our analyses and conclusions.

However, none of these previous efforts effectively evaluated the real gains of learning algorithms over traditional methods like BM25 or Language Models expressed as features of the dataset. Moreover, to our knowledge, our work is the first to do a statistical comparison of learning to rank algorithms and evaluation of their differences against the best single features.

3. EXPERIMENTAL SETUP

For our experimentations, we employed 8 datasets, from the LETOR benchmark [7]. Namely, we used the HP2003, HP2004, NP2003, NP2004, TD2003, and TD2004 datasets, from LETOR 3.0, based on the Gov web page collection. The first four collections are more related to navigational tasks in which a single unique page is the sole best answer for a query while the latter two are related to more traditional informational queries. We also use the larger MQ2007, MQ2008 datasets from LETOR 4.0, based on the Gov2 collection, which are also related to informational tasks. All of the datasets are divided into 5 folds, with the goal of performing 5-fold cross-validation, that is, 3 folds are used for training, one (validation) for parameter tuning, and the remaining one for testing. Below we start by describing in more detail each group of datasets in Section 2.1 and then further describe our experimental methodology in Section 2.2.

3.1 Collections

3.1.1. LETOR 3.0

Each of the many datasets encapsulated by this benchmark is composed of feature vectors for query-document pairs, along with a corresponding relevance judgment indicating whether the document is relevant or not for the query. There are 64 features per pair, which correspond to various pieces of information commonly used by traditional approaches (such as PageRank) or the result of directly applying simpler methods, such as TF*IDF, BM25 and language models, for estimating the document's relevance to the query. Considering all 6 datasets contained within this version of the benchmark, we find 575 queries and over 580.000 labeled documents.

Also available on the benchmark's web page [7], we find 12 different baselines, namely: AdaRank-MAP, AdaRank-NDCG, FRank, ListNet, RankBoost, RankSVM, RankSVM-Struct, RankSVM-Primal, Regression, Regression+L2reg, SmoothRank and SVMMAP. Aside from FRank and RankBoost, all of the algorithms use linear ranking functions.

3.1.2 LETOR 4.0

Similar to the previous benchmark, LETOR 4.0 uses the same structure for its datasets, but with 46 features, instead of 64, per vector. The largest TREC datasets available, from the Million Query Tracks of 2007 and 2008, named as MQ2007 and MQ2008, are based on the Gov2 webpage collection. Both datasets sum over 2.500 queries and roughly 420.000 documents. Unlike in LETOR 3.0, we here find only 5 reported baseline algorithms: AdaRank-MAP, AdaRank-NDCG, ListNet, RankBoost and RankSVM-Struct. Any other information pertaining these baselines and datasets, as well as the datasets themselves, are available at the LETOR website [7].

3.2 Choice of the Best Features

For comparison purposes, we performed the following feature selection procedure. For each fold, the values of each feature of each query-document pairs of the test set are extracted and used as a ranking score for its respective pair, thus obtaining a number of ranked lists equal to the number of features used to describe each document. Afterwards, we use the evaluation tool provided by the benchmark to calculate traditional information retrieval metrics, such as Mean Average Precision (MAP), of the ranked lists previously produced. We then select the feature responsible for the best ranked list, in terms of generated MAP, to use as our isolated ranking feature.

We here compare this MAP result with those obtained by the L2R baselines in the same test sets, using statistical significance tests with a 95% confidence level aiming at quantifying the differences but also verifying whether they are statistically significant. Specifically, to support our analysis and conclusions, we performed a pair-wise comparison of all methods, applying paired difference tests [5] for each pair to verify whether they are statistically different or not.

4. RESULTS

We start by showing the best results obtained with a single feature for each analyzed dataset and comparing these results with the analyzed L2R algorithms (Section 3.1). Next we perform an overall comparison among all methods (section 3.2).

4.1 Feature Ranking Results

As a result of our feature selection procedure, a single unique feature was selected as the best one in each dataset. Some features were also chosen for more than one dataset. In LETOR 3.0, feature 46 (Hyperlink base feature propagation: weighted in-link) was chosen as the best single feature for HP2003, HP2004 and TD2003; for NP2004 and TD2004, feature 42 (Sitemap based score propagation); and for NP2003, feature 9 (IDF of URL). In LETOR 4.0, we found feature 39 (LMIR.DIR of whole document) to be the best ranking feature in both MQ datasets. The MAP scores generated by the evaluation tools for each single feature are displayed in Table 1. In particular, it is interesting to notice the lower MAP values in the TD collections, which are known to be difficult informational datasets.

Table I. Average MAP scores for the best ranked features of the different datasets

Dataset	**Score**	**Best feature**
HP2003	0.7031	Hyperlink base feature propagation: weighted in-link (46)
HP2004	0.6054	Hyperlink base feature propagation: weighted in-link (46)
NP2003	0.5784	IDF of the URL (9)
NP2004	0.5202	Sitemap based score propagation (42)
TD2003	0.1973	Hyperlink base feature propagation: weighted in-link (46)
TD2004	0.1844	Sitemap based score propagation (42)
MQ2007	0.4534	LMIR.DIR of whole document (39)
MQ2008	0.4712	LMIR.DIR of whole document (39)

Table 2 shows the relative MAP difference between the best ranking feature of each dataset and the L2R baselines reported for LETOR 3.0. Next to the dataset's name, in parenthesis, we find the best feature's identifier. When the baseline's performance is statistically better, a (-) sign is present next to the value; when it is statistically equal, (=); and when the method is statistically inferior to the single feature, a (+) is present. In other words, positive values indicate that the isolated feature ranking process has a MAP score higher than the corresponding baseline, negative values indicate a lower score.

By looking at the table, we see that, aside from the NP datasets, less than half of the L2R algorithms are statistically superior to the isolated feature rankings in each dataset and that the absolute differences in performance may not be statistically superior to best feature in isolation. In fact, despite some large (significant) gains, there are a lot of statistical ties and even losses. For instance, in the HP2004 dataset, a difference (on average) of 14% of the RankBoost algorithm over the best single feature is indeed not significant; with 95% confidence both methods are tied. This is very surprising as we expected that **all** or at least **most** of the algorithms would be able to effectively combine the features to deliver a better

performance. However, the variability of the results is so large that relying only on average MAP to determine the best method is not enough.

In contrast, a 9% gain of the Frank method over the best single feature in the same dataset is significant. For the NP2003 and NP2004 datasets, we have that most L2R algorithms (92% and 67%) of the considered methods, respectively, are indeed statistically superior to the best single feature. However, it is interesting to note that, even in these datasets some (apparently) large relative differences (e.g., 16%) are in fact not statistically significant with 95% confidence. It is also worth mentioning the large and significant losses (up to 41.5%) of the regression method over the best feature in the two HP datasets. This may me due to the high correlations among several features in this dataset, which may be detrimental to this particular method, which relies on linear regression [5].

Table II. Relative MAP comparison between feature ranking and L2R algorithms, LETOR 3.0

Algorithm	AdaRank-MAP	AdaRank-nDCG	FRank	List Net	Rank Boost	Rank SVM	RankSVM-Primal	RankSVM-Struct	Regression	Regression+L2reg	Smooth Rank	SVM MAP
HP2003 (46)	8.80% (+)	6.00% (=)	0.89% (=)	8.19% (=)	7.31% (=)	5.08% (+)	8.01% (+)	7.79% (+)	-41.53% **(-)**	6.07% (=)	7.88% (=)	5.25% (=)
HP2004 (46)	14.50% (+)	10.73% (=)	9.46% (+)	10.54% (+)	14.00% (=)	7.53% (=)	8.04% (=)	9.03% (=)	-17.42% **(-)**	2.05% (=)	13.97% (+)	13.99 % (=)
NP2003 (9)	14.73% (+)	13.39% (+)	12.89% (+)	16.11% (+)	17.85% (+)	16.86% (+)	15.97% (+)	14.79% (+)	-2.48% (=)	15.24% (+)	16.87% (+)	15.79 % (+)
NP2004 (42)	16.36% (=)	17.01% (+)	13.41% (=)	22.58% (+)	5.79% (=)	21.03% (+)	22.987 % (+)	23.178 % (+)	-1.16% (=)	24.23% (+)	23.04% (+)	21.41 % (+)
TD2003 (42)	13.57% (=)	16.66% (=)	2.847% (=)	28.32% (+)	13.214 % (=)	24.90% (=)	25.61% (=)	27.26% (+)	18.08% (=)	18.92% (+)	26.80% (+)	19.31 % (+)
TD2004 (46)	15.76% (=)	4.75% (=)	22.81% (+)	17.38% (+)	29.46% (+)	17.59% (=)	10.56% (=)	16.03% (+)	11.28% (=)	7.42% (=)	20.72% (+)	10.02 % (=)

Analogous to Table 2, Table 3 presents results relative to LETOR 4.0. Similarly to the results found in LETOR 3.0, we here see that, not only the relative differences may not be statistically significant (with 95% confidence), such as in MQ2008, but also that, in some cases, the gains may be only marginal (e.g., 2% for the RankSVM-Struct in the MQ2007 dataset. In fact, it is very surprising that in MQ2008, no method is able to surpass the best feature in isolation.

These results lead to interesting conclusions pertaining the effective gains associated with L2R and its aggregated costs. While the performed process of choosing the best features isn't a free process, it is much cheaper than the complex machine learning algorithms. In fact, we may not need to investigate all possible features,. A smaller set of candidates could be used based on results reported in the literature.

Table IIIII. Relative MAP comparison between feature ranking and L2R algorithms, LETOR 4.0

Algorithms	AdaRank-MAP	AdaRank-NDCG	ListNet	RankBoost	RankSVM-Struct
MQ2007 (39)	4.74% (=)	5.58% (+)	6.39% (+)	6.54% (+)	2.39% (+)
MQ2008 (39)	3.20% (=)	3.16% (=)	2.10% (=)	1.85% (=)	0.34% (=)

4.2 Baseline Comparisons

We now turn to our second goal, which is to compare the supervised rankers in the used collections. Tables 4 and 5 show, for each dataset in the LETOR 3.0 and 4.0, respectively, the average MAP results obtained for each baseline, jointly with the corresponding 95% confidence intervals (computed over the results of the 5 folds). Best results for each dataset, along with statistical ties according to paired tests[1] with 95% confidence are shown in bold.

Table IV. Baselines' average MAP and confidence intervals across the different datasets, LETOR 3.0

Algorithm	AdaRank-MAP	AdaRank-nDCG	FRank	List Net	Rank Boost	Rank SVM	RankSVM-Primal	RankSVM-Struct	Regression	Regression+L2reg	Smooth Rank	SVM MAP
HP2003	**0.771 ± 0.063**	**0.748 ± 0.112**	0.710 ± 0.069	**0.766 ± 0.085**	**0.759 ± 0.065**	**0.741 ± 0.061**	**0.764 ± 0.078**	**0.763 ± 0.083**	0.497 ± 0.037	**0.749 ± 0.090**	**0.763 ± 0.086**	0.742 ± 0.098
HP2004	**0.722 ± 0.092**	**0.691 ± 0.048**	**0.682 ± 0.100**	**0.690 ± 0.093**	**0.718 ± 0.020**	**0.668 ± 0.088**	**0.671 ± 0.086**	**0.678 ± 0.076**	0.526 ± 0.067	**0.630 ± 0.076**	**0.717 ± 0.069**	**0.718 ± 0.091**
NP2003	**0.678 ± 0.078**	**0.668 ± 0.092**	**0.664 ± 0.073**	**0.690 ± 0.075**	**0.704 ± 0.017**	**0.696 ± 0.061**	**0.688 ± 0.070**	**0.679 ± 0.065**	0.564 ± 0.086	**0.682 ± 0.061**	**0.696 ± 0.054**	**0.687 ± 0.058**
NP2004	**0.622 ± 0.049**	**0.627 ± 0.041**	**0.601 ± 0.101**	**0.672 ± 0.084**	0.552 ± 0.071	**0.659 ± 0.096**	**0.676 ± 0.110**	**0.677 ± 0.081**	0.514 ± 0.58	**0.687 ± 0.097**	**0.676 ± 0.064**	**0.662 ± 0.087**
TD2003	**0.228 ± 0.095**	**0.237 ± 0.116**	**0.203 ± 0.080**	**0.275 ± 0.090**	**0.227 ± 0.078**	**0.263 ± 0.099**	**0.265 ± 0.098**	**0.271 ± 0.103**	**0.241 ± 0.075**	**0.243 ± 0.089**	**0.270 ± 0.084**	**0.245 ± 0.075**
TD2004	0.219 ± 0.038	0.194 ± 0.031	**0.239 ± 0.038**	0.223 ± 0.007	**0.261 ± 0.031**	0.224 ± 0.031	0.206 ± 0.024	0.220 ± 0..023	0.208 ± 0..031	0.199 ± 0.022	0.233 ± 0.029	0.205 ± 0.022

In the HP2003 dataset, there is a statistical tie for the best method among 8 out of 12 of the baselines, i.e., the differences among them are not statistically significant with 95% confidence. The worst method is Regression, which is inferior to all baselines and even to the best feature in isolation (difference to the best performer, AdaRankMap, of 35%). Notice however that the second worst method (FRank) is only at most 8% worse than the best performer. Thus, in general, except for Regression, the differences among all considered baselines are, if significant, relatively small.

For the HP2004 dataset we have statistically tied results all but 1 baselines. The only baseline that is significantly inferior to the others is (once again) Regression. We here also observe some large differences that are not statistically significant, such as the gap between AdaRank-Map and Regression+L2reg (12.7%). As discussed before, these results clearly reflect the large variability of the methods across the various folds of the datasets.

Like in HP2004, all methods but Regression are statistically tied in the NP2003 dataset, making it once again impossible to single out a best ranking method. In the NP2004, we have a similar situation with 10 out of 12 methods statistically tied. Surprisingly, RankBoost – which had a good performance in the previous datasets and is one of two best rankers in TD2004 (see below) – was tied with Regression as the worst methods.

In the TD2003 dataset, all methods are tied, with no clear winner or loser. An interesting result found here is the very large relative differences (on average) of ListNet over FRank (26.2%), although they are

[1] The tests were performed by computing 95% confidence intervals over the differences between the results obtained for each pair of baselines [5].

still statistically tied with 95% confidence. For the last dataset in LETOR 3.0, TD2004, we have RankBoost and FRank as the best rankers out of all the reported baselines, beating the other 9 baselines. In fact, this dataset is the only one with few (two) methods outperforming most of other algorithms, with some large significant differences in some cases (up to 26%). In contrast, in the other datasets, almost the entirety of the available baselines tied against each other, with 95% confidence.

Table V. Baselines' average MAP and confidence intervals across the different datasets, LETOR 4.0

Algorithms	**AdaRank-MAP**	**AdaRank-NDCG**	**ListNet**	**RankBoost**	**RankSVM-Struct**
MQ2007	0.476	**0.480**	**0.484**	**0.485**	**0.465**
RankBoost	±	±	±	±	±
	0.027	**0.025**	**0.028**	**0.029**	**0.020**
MQ2008	**0.487**	**0.487**	0.481	**0.480**	**0.470**
AdaRank-MAP	±	±	±	±	±
	0.034	**0.035**	0.026	**0.026**	**0.047**

Turning our attention to the baselines and dataset in the LETOR 4.0 benchmark (Table V), we find that there are four statistically tied methods with 95% confidence in the MQ2007 dataset, with AdaRank-MAP being the sole loser. A similar scenario is found in the MQ2008 dataset, but this time ListNet is the worst performer and the only one not statistically tied with the others.

In general, we find that in the vast majority of the analyzed datasets, most of the baselines are statistically tied, with no clear winner, raising a question of whether it is cost-effective to invest on developing new learning-to-rank algorithms, as opposed to combining multiple methods into a single hybrid solution or investing on reducing the costs (particularly in cases where the L2R methods outperform the single best feature).

5. CONCLUSIONS AND FUTURE WORK

After almost a decade of research and development of L2R algorithms, we have here raised two controversial but important questions that should be further discussed by the Information Retrieval community.

First, given all the costs involved in L2R (e.g., labeling, training, tuning) and the overhead introduced by applying such techniques, for instance, for re-ranking search top results at query time, the cost-benefit ratio of applying such techniques should be further investigated.

Secondly, given the similar performance of a dozen different methods across many different datasets, researching and developing new L2R algorithms may not be worth the effort; indeed in few cases there are undisputed best rankers, but it may not be the case in most datasets.

Rather than providing definitive answers, our goal here is to instigate discussion and re-evaluation of many L2R algorithms after having applied solid statistical methods in our own investigation of the subject.

As future work, we intend to expand this study to consider even larger datasets, such as the ones provided by Microsoft Learning to Rank and Yahoo! Labs, as well as new L2R algorithms.

6. ACKNOWLEDGMENTS

This work is supported by INWeb (MCT/CNPq grant 57.3871/2008-6) and by the author grants from CNPq, CAPES and FAPEMIG.

7. REFERENCES

[1] K. S. Jones, S. Walker, and S. E. Robertson, A Probabilistic Model of Information Retrieval: Development and Comparative Experiments. *IP&M*, 36(6): 779-840, 2000.

[2] J. M. Ponte and W. B. Croft. A Language Modeling Approach to Information Retrieval. *In SIGIR*. pp. 275–281, 1998.

[3] Tie-Yan Liu: Learning to Rank for Information Retrieval. Springer 2011: I-XVII, 1-285

[4] B. Cambazoglu, H. Zaragoza, O. Chapelle, J. Chen, C. Liao, Z. Zheng, and J. Degenhardt. Early exit optimizations for additive machine learned ranking systems. In *WSDM*. pp. 411-420, 2010.

[5] R. Jain. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling, Wiley-Interscience, New York, NY, 1991.

[6] C. Zhai and J. Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst*, 22(2): 179-214, 2004.

[7] http://research.microsoft.com/en-us/um/beijing/projects/letor/

[8] Liu, T.Y., Xu, J., Qin, T., Xiong, W.Y., Li, H.: LETOR: benchmark dataset for research on learning to rank for information retrieval. In: SIGIR 2007 Workshop on Learning to Rank for Information Retrieval (LR4IR 2007) (2007)

[9] Minka, T., Robertson, S.:Selection bias in the LETOR datasets.In : SIGIR 2008 Workshop on Learning to Rank for Information Retrieval (LR4IR 2008) (2008)

[10] Qin, T., Liu, T.Y., Xu, J., Li, H.: How to make LETOR more useful and reliable. In: SIGIR 2008 Workshop on Learning to Rank for Information Retrieval (LR4IR 2008) (2008)

[11] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Found. Trends Inf. Retr.* 3, 3 (March 2009), 225-331. DOI=10.1561/1500000016 http://dx.doi.org/10.1561/1500000016