A Cross-Benchmark Comparison of 87 Learning to Rank Methods

Niek Tax[a,c,1,*], Sander Bockting[a], Djoerd Hiemstra[b]

[a] *Avanade Netherlands B.V., Versterkerstraat 6, 1322AP Almere, the Netherlands*
[b] *University of Twente, P.O. Box 217, 7500AE Enschede, the Netherlands*
[c] *Eindhoven University of Technology, P.O. Box 513, 5600MB Eindhoven, the Netherlands*

Abstract

Learning to rank is an increasingly important scientific field that comprises the use of machine learning for the ranking task. New learning to rank methods are generally evaluated on benchmark test collections. However, comparison of learning to rank methods based on evaluation results is hindered by non-existence of a standard set of evaluation benchmark collections. In this paper we propose a way to compare learning to rank methods based on a sparse set of evaluation results on a set of benchmark datasets. Our comparison methodology consists of two components: 1) Normalized Winning Number, which gives insight in the ranking accuracy of the learning to rank method, and 2) Ideal Winning Number, which gives insight in the degree of certainty concerning its ranking accuracy. Evaluation results of 87 learning to rank methods on 20 well-known benchmark datasets are collected through a structured literature search. ListNet, SmoothRank, FenchelRank, FSMRank, LRUF and LARF are Pareto optimal learning to rank methods in the Normalized Winning Number and Ideal Winning Number dimensions, listed in increasing order of Normalized Winning Number and decreasing order of Ideal Winning Number.

Keywords: Learning to rank, Information retrieval, Evaluation metric

1. Introduction

Ranking is a core problem in the field of information retrieval. The ranking task in information retrieval entails the ranking of candidate documents according to their relevance to a given query. Ranking has become a vital part of

*Corresponding Author: Niek Tax, Eindhoven University of Technology, Department of Mathematics and Computer Science, P.O. Box 513, 5600MB Eindhoven, the Netherlands; Email, n.tax@tue.nl; Phone, +31634085760

Email addresses: n.tax@tue.nl (Niek Tax), sander.bockting@avanade.com (Sander Bockting), d.hiemstra@utwente.nl (Djoerd Hiemstra)

[1] Author is affiliated with Eindhoven University of Technology, but this paper was written during his stay at Avanade Netherlands B.V.

web search, where commercial search engines help users find their need in the extremely large collection of the World Wide Web. Among useful applications outside web search are automatic text summarisation, machine translation, drug discovery and determining the ideal order of maintenance operations (Rudin, 2009). In addition, McNee et al. (2006) found the ranking task to be a better fit for recommender systems than the regression task (continuous scale predictions), which is currently still frequently used within such systems.

Research in the field of ranking models has long been based on manually designed ranking functions, such as the well-known BM25 model (Robertson and Walker, 1994). Increased amounts of potential training data have recently made it possible to leverage machine learning methods to obtain more effective ranking models. Learning to rank is the relatively new research area that covers the use of machine learning models for the ranking task.

In recent years, several learning to rank benchmark datasets have been proposed with the aim of enabling comparison of learning to rank methods in terms of ranking accuracy. Well-known benchmark datasets in the learning to rank field include the *Yahoo! Learning to Rank Challenge* datasets (Chapelle and Chang, 2011), the *Yandex Internet Mathematics 2009* contest[2], the LETOR datasets (Qin et al., 2010b), and the MSLR (Microsoft Learning to Rank) datasets[3]. There exists no agreement among authors in the learning to rank field on the benchmark collection(s) to use to evaluate a new model. Comparing ranking accuracy of learning to rank methods is largely hindered by this lack of a standard way of benchmarking.

Gomes et al. (2013) analyzed the ranking accuracy of a set of models on both LETOR 3.0 and 4.0. Busa-Fekete et al. (2013) compared the accuracy of a small set of models over the LETOR 4.0 datasets, both MSLR datasets, both the Yahoo! Learning to Rank Challenge datasets and one of the datasets from LETOR 3.0. Both studies did not aim to be complete in benchmark datasets and learning to rank methods included in their comparisons. To our knowledge, no structured meta-analysis on ranking accuracy has been conducted where evaluation results on several benchmark collections are taken into account. In this paper we will perform a meta-analysis with the aim of comparing the ranking accuracy of learning to rank methods. The paper will describe two stages in the meta-analysis process: 1) collection of evaluation results, and 2) comparison of learning to rank methods.

[2]http://imat2009.yandex.ru/en
[3]http://research.microsoft.com/en-us/projects/mslr/

2. Collecting Evaluation Results

We collect evaluation results on the datasets of benchmark collections through a structured literature search. Table 1 presents an overview of the benchmark collections included in the meta-analysis. Note that all these datasets offer feature set representations of the to-be-ranked documents instead of the documents themselves. Therefore, any difference in ranking performance is due to the ranking algorithm and not the features used.

Benchmark collection	# of datasets
AOL	1
LETOR 2.0	3
LETOR 3.0	7
LETOR 4.0	2
MSLR	2
WCL2R	2
Yahoo! Learning to Rank Challenge	2
Yandex Internet Mathematics 2009 contest	1
Total	**20**

Table 1: Included learning to rank evaluation benchmark collections

For the LETOR collections, the evaluation results of the baseline models will be used from LETOR 2.0[4], 3.0[5] and 4.0[6] as listed on the LETOR website.

LETOR 1.0 and 3.0, Yahoo! Learning to Rank Challenge, WCL2R and AOL have accompanying papers that were released with the collection. Authors publishing evaluation results on these benchmark collections are requested to cite these papers. We collect evaluation measurements of learning to rank methods on these benchmark collections through forward literature search. Table 2 presents an overview of the results of this forward literature search performed using Google Scholar.

The LETOR 4.0, MSLR-web10/30k and Yandex Internet Mathematics Competition 2009 benchmark collections are not accompanied by a paper. To collect evaluation results for learning to rank methods on these benchmarks, a Google Scholar search is performed on the name of the benchmark. Table 3 shows the results of this literature search.

2.1. Literature Selection

Table A.5 in the appendix gives an overview of the learning to rank methods for which evaluation results were found through the described procedure.

[4] http://research.microsoft.com/en-us/um/beijing/projects/letor/letor2.0/baseline.aspx
[5] http://research.microsoft.com/en-us/um/beijing/projects/letor/letor3baseline.aspx
[6] http://research.microsoft.com/en-us/um/beijing/projects/letor/letor4baseline.aspx

Benchmark	Paper	# of forward references
LETOR 1.0 & 2.0	Liu et al. (2007)	307
LETOR 3.0	Qin et al. (2010b)	105
Yahoo! Learning to Rank Challenge	Chapelle and Chang (2011)	102
AOL dataset	Pass et al. (2006)	339
WCL2R	Alcântara et al. (2010)	2

Table 2: Forward references of learning to rank benchmark papers

Search Query	Google scholar search results
"LETOR 4.0"	75
"MSLR-web10k"	16
"MSLR-web30k"	15
"Yandex Internet Mathematics"	3

Table 3: Google scholar search results for learning to rank benchmarks

Occurrences of L2, L3 and L4 in Table A.5 imply that these algorithms are evaluated as official LETOR 2.0, 3.0 and 4.0 baselines respectively.

Some studies with evaluation results found through the literature search procedure were not usable for the meta-analysis. The following enumeration enumerates those properties that made one or more studies unusable for the meta-analysis. The references between brackets are the studies to which these properties apply.

1. A different evaluation methodology was used in the study compared to what was used in other studies using the same benchmark (Geng et al., 2011; Lin et al., 2012)
2. The study focuses on a different learning to rank task (e.g. rank aggregation or transfer ranking) (De and Diaz, 2011; De et al., 2010; Derhami et al., 2013; De et al., 2012; Chen et al., 2010; Ah-Pine, 2008; Wang et al., 2009a; De, 2013; Miao and Tang, 2013; Hoi and Jin, 2008; De and Diaz, 2012; Duh and Kirchhoff, 2011; Argentini, 2012; Qin et al., 2010a; Volkovs and Zemel, 2013; Desarkar et al., 2011; Pan et al., 2013; Lin et al., 2011a; Volkovs and Zemel, 2012; Dammak et al., 2011)
3. The study used an altered version of a benchmark that contained additional features (Bidoki and Thom, 2009; Ding et al., 2010)
4. The study provides no exact data of the evaluation results (e.g. results are only in graphical form) (Wang et al., 2008; Wang and Xu, 2010; Xu et al., 2010; Kuo et al., 2009; Li et al., 2008; Xia et al., 2008; Zhou et al., 2011; Wu

et al., 2011; Zhu et al., 2009; Karimzadehgan et al., 2011; Swersky et al., 2012; Pan et al., 2011; Ni et al., 2008; Ciaramita et al., 2008; Stewart and Diaz, 2012; Petterson et al., 2009; Agarwal and Collins, 2010; Chang and Zheng, 2009; Qin et al., 2008b; Adams and Zemel, 2011; Sculley, 2009; Huang and Frey, 2008; Alejo et al., 2010; Sun et al., 2011; He et al., 2010a; Benbouzid et al., 2012; Geng et al., 2012; Chen et al., 2012; Xu et al., 2012; Shivaswamy and Joachims, 2011)

5. The study reported evaluation results in a different metric than the metrics chosen for this meta-analysis (Yu and Joachims, 2009; Thuy et al., 2009; Pahikkala et al., 2009; Kersting and Xu, 2009; Mohan et al., 2011)
6. The study reported a higher performance on baseline methods than official benchmark runs (Dubey et al., 2009; Banerjee et al., 2009; Peng et al., 2010a; Song et al., 2014; Bian et al., 2010; Bian, 2010; Carvalho et al., 2008; Acharyya et al., 2012; Peng et al., 2010a; Tran and Pham, 2012; Asadi, 2013)
7. The study did not report any baseline performance that allowed us to check validity of the results (Chakrabarti et al., 2008; Wang et al., 2012b; Buffoni et al., 2011).

3. A Methodology for Comparing Learning to Rank Methods Cross-Benchmark

Qin et al. (2010b) state that it may differ between datasets what the most accurate ranking methods are. They propose a measure they call *Winning Number* to evaluate the overall performance of learning to rank methods over the datasets included in the LETOR 3.0 collection. Winning Number is defined as the number of other algorithms that an algorithm can beat over the set of datasets, or more formally

$$\mathrm{WN}_i(M) = \sum_{j=1}^{n} \sum_{k=1}^{m} I_{\{M_i(j) > M_k(j)\}}$$

where j is the index of a dataset, n the number of datasets in the comparison, i and k are indices of an algorithm, $M_i(j)$ is the performance of the i-th algorithm on the j-th dataset, M is a ranking measure (such as NDCG or MAP), and $I_{\{M_i(j) > M_k(j)\}}$ is an indicator function such that

$$I_{\{M_i(j) > M_k(j)\}} = \begin{cases} 1 & \text{if } M_i(j) > M_k(j), \\ 0 & \text{otherwise} \end{cases}$$

The LETOR 3.0 was a comparison on a *dense* set of evaluation results, in the sense that there were evaluation results available for all learning to rank algorithms on all datasets included in their comparison. The Winning Number evaluation metric relies on the denseness of the evaluation results set. In contrast to the LETOR 3.0 comparison, our evaluation results will be a *sparse* set. We propose a normalized version of the Winning Number metric to enable comparison of a sparse set of evaluation results. This Normalized Winning Number

takes only those datasets into account that an algorithm is evaluated on and divides this by the theoretically highest Winning Number that an algorithm would have had in case it would have been the most accurate algorithm on all datasets on which it has been evaluated. We will redefine the indicator function I in order to only take into account those datasets that an algorithm is evaluated on, as

$$I'_{M_i(j)>M_k(j)} = \begin{cases} 1 & \text{if } M_i(j) \text{ and } M_k(j) \text{ are both defined and } M_i(j) > M_k(j), \\ 0 & \text{otherwise} \end{cases}$$

From now on this adjusted version of Winning Number will be references to as *Normalized Winning Number (NWN)*. The formal definition of Normalized Winning Number is

$$\text{NWN}_i(M) = \frac{\text{WN}_i(M)}{\text{IWN}_i(M)}$$

where IWN is the Ideal Winning Number, defined as

$$\text{IWN}_i(M) = \sum_{j=1}^{n} \sum_{k=1}^{m} D_{\{M_i(j),M_k(j)\}}$$

where j is the index of a dataset, n the number of datasets in the comparison, i and k are indices of an algorithm, $M_i(j)$ is the performance of the i-th algorithm on the j-th dataset, M is a ranking measure (such as NDCG or MAP), and $D_{\{M_i(j),M_k(j)\}}$ is an evaluation definition function such that

$$D_{\{M_i(j),M_k(j)\}} = \begin{cases} 1 & \text{if } M_i(j) \text{ and } M_k(j) \text{ are both defined,} \\ 0 & \text{otherwise} \end{cases}$$

NDCG@$\{3, 5, 10\}$ and MAP are the most frequently used evaluation metrics in the used benchmark collections combined, therefore we will limit our meta-analysis to evaluation results reported in one of these four metrics.

4. Results of Learning to Rank Comparison

The following subsections provide the performance of learning to rank methods in terms of NWN for NDCG@$\{3, 5, 10\}$ and MAP. Performance of the learning to rank methods is plotted with NWN on the vertical axis and the number of datasets on which the method has been evaluated on the horizontal axis. Moving to the right, certainty on the performance of the method increases. The Pareto optimal learning to rank methods, that is, the learning to rank methods for which it holds that there is no other method that has 1) a higher NWN and 2) a higher number datasets evaluated, are identified as the best performing methods and are labeled. Table B.6 in the appendix provides raw NWN data for the learning to rank methods at NDCG@$\{3, 5, 10\}$ and MAP and their cross-metric weighted average.

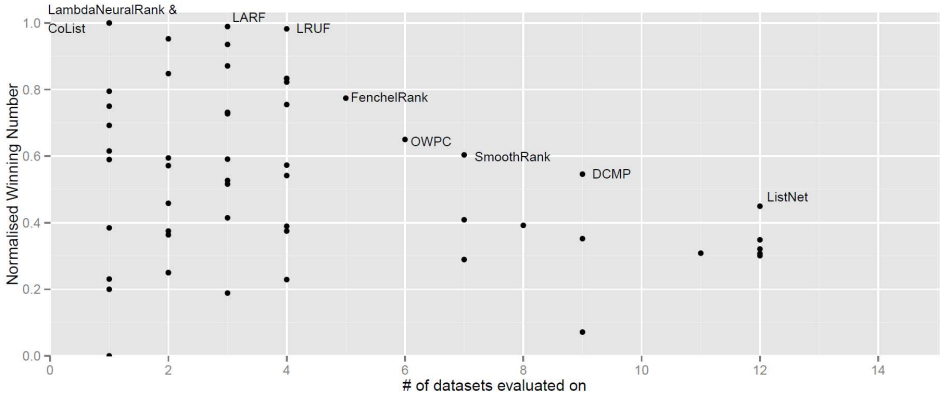

Figure 1: NDCG@3 comparison of 87 learning to rank methods

4.1. NDCG@3

Figure 1 shows the NWN of learning to rank methods based on NDCG@3 results. LambdaNeuralRank and CoList both acquired a NWN score of 1.0 by beating all other algorithms on one dataset, with LambdaNeuralRank winning on the AOL dataset and CoList winning on Yahoo Set 2. LARF and LRUF scored very high scores of near 1.0 on three of the LETOR 3.0 datasets, which results in more certainty on these methods' performance because they are validated on three datasets that additionally are more relevant than AOL and Yahoo Set 2 (number of evaluation results for LETOR 3.0 are higher than those for AOL and Yahoo set 2). FenchelRank, OWPC, SmoothRank, DCMP and ListNet are ordered decreasingly by NWN and at the same time increasingly in number of datasets that they are evaluated on, resulting in a higher degree of certainty on the accuracy of the algorithms.

LambdaNeuralRank, CoList, LARF, LRUF, OWPC and DCMP evaluation results are all based on one study, therefore are subjected to the risk of one overly optimistic study producing those results. FenchelRank evaluation result are the combined result from two studies, although those studies have overlap in authors. SmoothRank and ListNet have the most reliable evaluation result source, as they were official LETOR baseline runs.

4.2. NDCG@5

Figure 2 shows the NWN of learning to rank methods based on NDCG@5 results. LambdaNeuralRank again beat all other methods solely with results on the AOL dataset scoring a NWN of 1.0. LARF, LRUF, FenchelRank, SmoothRank, DCMP and ListNet are from left to right evaluated on an increasing number of datasets, but score decreasingly well in terms of NWN. These results are highly in agreement with the NDCG@3 comparison. The only modification compared to the NDCG@3 comparison being that OWPC did show to be a method for which there were no methods performing better on both axes in

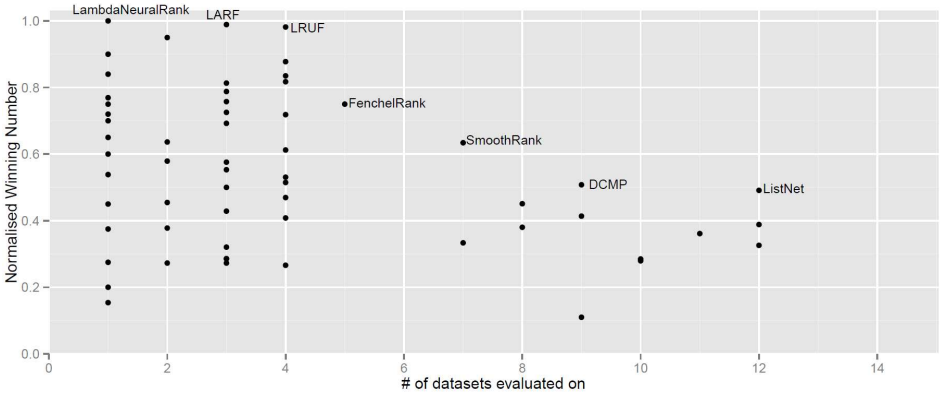

Figure 2: NDCG@5 comparison of 87 learning to rank methods

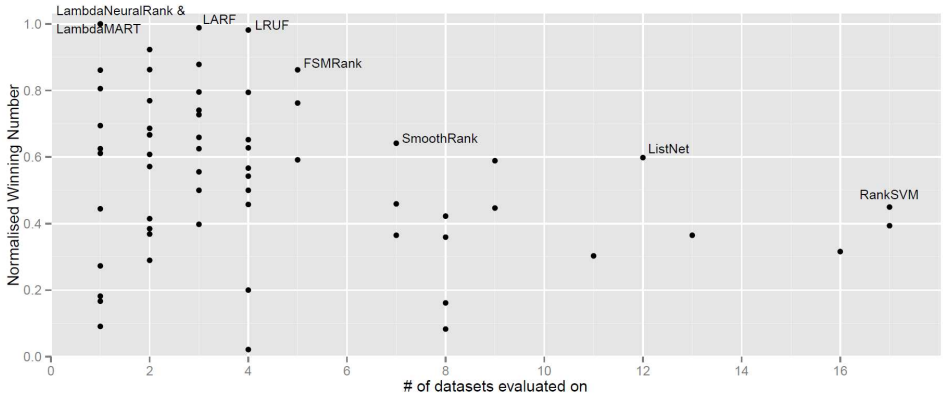

Figure 3: NDCG@10 comparison of 87 learning to rank methods

the NDCG@5 comparison, but not in the @3 comparison. Like in the NDCG@3 comparison, SmoothRank and ListNet can be regarded as most reliable results because the evaluation measurements for these methods are based on LETOR official baselines.

4.3. NDCG@10

Figure 3 shows the NWN of learning to rank methods based on NDCG@10 results. LambdaMART and LambdaNeuralRank score a NWN of 1.0 on the NDCG@10 comparison. For LambdaNeuralRank these results are again based on AOL dataset measurements. LambdaMART showed the highest NDCG@10 performance for the MSLR-WEB10k dataset. The set of Pareto optimal learning to rank algorithms is partly in agreement with the set of Pareto optimal methods for the NDCG@3 and @5 comparisons, both include LARF, LRUF, FSMRank, SmoothRank, ListNet, RankSVM. In contrast to the NDCG@3 and @5 com-

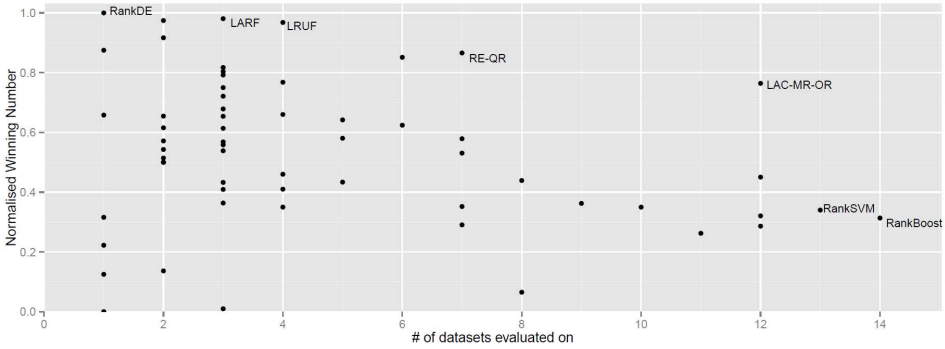

Figure 4: MAP comparison of 87 learning to rank methods

parisons, DCMP is not a Pareto optimal ranking method in the NDCG@10 comparison.

4.4. MAP

Figure 4 shows the NWN of learning to rank methods based on MAP results. Comparisons on the NDCG metrics where highly in agreement on the Pareto optimal algorithms, MAP-based NWN results show different results. RankDE scores a NWN of 1.0 on one dataset, which is achieved by obtaining highest MAP-score on the LETOR 2.0 TD2003 which has many evaluation results are evaluated.

LARF and LRUF score very high NWN scores, but based on only few datasets, just as in the NDCG-based comparisons. Notable is the low performance of SmoothRank and ListNet, given that those methods were top performing methods in the NDCG-based comparisons. Table B.6 in the appendix shows that LAC-MR-OR is evaluated on more datasets on MAP than on NDCG, thereby LAC-MR-OR obtained equal certainty to ListNet with a higher NWN. SmoothRank performed a NWN of around 0.53 on 7 datasets, which is good in both certainty and accuracy, but not a Pareto optimum. RE-QR is one of the best performers in the MAP comparison with a reasonable amount of benchmark evaluations. No reported NDCG performance was found in the literature search for RE-QR. There is a lot of certainty on the accuracy of RankBoost and RankSVM as both models are evaluated on the majority of datasets included in the comparison for the MAP metric, but given their NWN it can said that both methods are not within the top performing learning to rank methods.

4.5. Cross-Metric

Figure 5 shows NWN as function of IWN for the methods listed in Table A.5. The cross-metric comparison is based on the NDCG@$\{3, 5, 10\}$ and MAP comparisons combined. Figure 5 labels the Pareto optimal algorithms, but also the Rank-2 Pareto optima, which are the labels the algorithms with exactly one

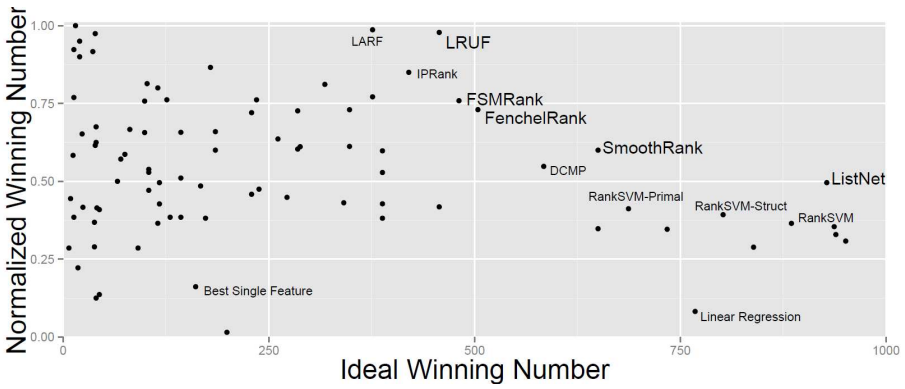

Figure 5: Cross-benchmark comparison of 87 learning to rank methods

algorithm having a higher value on both axes. Pareto optimal are labeled in large font while Rank-2 Pareto optima are labeled using a smaller font size. In addition, Linear Regression and the ranking method of simply sorting on the best single feature are labeled as baselines.

LRUF, FSMRank, FenchelRank, SmoothRank and ListNet showed to be the methods that have no other method superior to them in both IWN and NWN. LRUF is the only method that achieved Pareto optimality in all NDCG comparisons, the MAP comparison as well as the cross-metric comparison. With FenchelRank, FSMRank, SmoothRank and ListNet being Pareto optimal in all NDCG comparisons as well as in the cross-metric comparison, it can be concluded that the cross-metric results are highly defined by the NDCG performance as opposed to the MAP performance. This was to be expected, because the cross-metric comparison input data of three NDCG entries (@3, @5, and @10) enables it to have up to three times as many as many weight as the MAP comparison.

LARF, IPRank and DCMP and several variants of RankSVM are the Rank-2 Pareto optima of the cross-metric comparison. LARF was also a Pareto optima on the NDCG and MAP comparisons and DCMP was a Pareto optimal ranker in a few of the NDCG comparisons. C-CRF, DirectRank, FP-Rank, RankCSA, LambdaNeuralRank and VFLR all have a near-perfect NWN value, but have a low IWN value. Further evaluation runs of these methods on benchmark datasets that they are not yet evaluated on are desirable. The DirectRank paper (Tan et al. (2013)) shows that the method is evaluated on more datasets than the number of datasets that we included evaluation results for in this meta-analysis. Some of the DirectRank measurements could not be used because measurements on some datasets were only available in graphical form and not in raw data.

LAC-MR-OR and RE-QR showed very good ranking accuracy in the MAP comparison on multiple datasets. Because LAC-MR-OR is only evaluated on two datasets for NDCG@10 and RE-QR is not evaluated for NDCG at all, LAC-MR-OR and RE-QR are not within the Pareto front of rankers in the cross-metric comparison.

5. Sensitivity Analysis

In this section we evaluate the stability of the obtained results when one of the evaluation measures (5.1) or one of the datasets (5.2) are left out of the comparison. We scope this sensitivity analysis to those ranking methods that showed to be Pareto optimal in the trade-off between IWN and NWN: ListNet, SmoothRank, FenchelRank, FSMRank and LRUF.

5.1. Sensitivity in the evaluation measure dimension

To analyze the sensitivity of the comparison method in the evaluation measure dimension we repeated the NWN and IWN calculation while leaving one evaluation measure. Table 5.1 shows the NWN and IWN results when all evaluation measures are included in the computation and when MAP, NDCG@3, NDCG@5 or NDCG@10 are left out respectively. From this table we can infer that FSMRank is not a Pareto optimal ranking method when MAP is left out of the comparison (LRUF scores higher on both NWN and IWN) and FenchelRank is not a Pareto optimal ranking method when either NDCG@3 or NDCG@5 are left out (FSMRank scores higher on both NWN and IWN). All other orderings of ranking methods on NWN and IWN stay intact when one of the evaluation measures is left out of the comparison.

Notable is that all Pareto optimal ranking methods have the largest increase in IWN as well as the largest decrease in NWN when the MAP measure is left out of the comparison. The NWN score of FSMRank increased almost 0.1 when the MAP evaluation measure was left out, which is the highest deviation in NWN score seen in this sensitivity analysis. Note that MAP uses a binary notion of relevance, where NDCG uses graded relevance. The fact that all Pareto optimal rankers obtain an even higher NWN score when the MAP measure is left out shows that apparently the Pareto optimal rankers perform even better on ranking on graded relevance, compared to non-Pareto-optimal rankers.

5.2. Sensitivity in the dataset dimension

In Table 1 we showed to include 20 datasets in our comparison, originating from eight data collections. We analyzed the variance in NWN and IWN scores of the Pareto optimal rankers for the situations where one of the 20 datasets is not included in the NWN and IWN computation. The results are visualized in Figure 6 in a series of bagplots, which is a bivariate generalization of the boxplot proposed by Rousseeuw et al. (1999). Bagplot extends the univariate concept of rank as used in a boxplot to a halfspace location depth. The *depth median*,

	All		MAP		NDCG@3		NDCG@5		NDCG@10	
	NWN	IWN	NWN	IWN	NWN	IWN	NWN	IWN	NWN	IWN
ListNet	0.4952	931	0.5127	669	0.5099	710	0.4965	707	0.4625	707
SmoothRank	0.6003	653	0.6266	474	0.5988	491	0.5900	500	0.5870	494
FenchelRank	0.7307	505	0.7628	371	0.7158	380	0.7244	381	0.7206	383
FSMRank	0.7593	482	0.8585	311	0.7403	385	0.7292	384	0.7268	366
LRUF	0.9783	460	0.9821	335	0.9767	344	0.9772	351	0.9771	350
LARF	0.9868	379	0.9891	275	0.9859	283	0.9861	288	0.9863	291

Table 4: NWN and IWN scores of the Pareto optimal rankers on all evaluation metrics, and with MAP, NDCG@3, NDCG@5 or NDCG@10 left out of the comparison respectively

shown in orange, is the deepest location. Surrounding it is a *bag*, the dark blue area in Figure 6, containing $\frac{n}{2}$ observations with the largest depth. The light blue area represents the *fence*, which magnifies the bag by a factor 3.

Note that the number of unique observations on which the bagplots are created is equal to the number of dataset on which a ranking method is evaluated (in any of the evaluation measures), as removing a dataset on which a ranking algorithm is not evaluated does not have any effect on the NWN and IWN scores. The difference between the leftmost and the rightmost points of the bags seems to be more or less equal for all ranking methods while the NWN means are decreasing from top-to-bottom and and left-to-right. Therefore, the variance-to-mean ratio increases from top-top-bottom and from left-to-right. On the IWN dimension it is notable that LRUF and LARF has very low variance. It is important to stress that this does not imply high certainty about the level of ranking performance of these ranking methods, it solely shows the low variance in the evaluation results available for the ranking methods.

6. Limitations

In the NWN calculation, the weight of each benchmark on the total score is determined by the number of evaluation measurements on this benchmark. By calculating it in this way, we implicitly make the assumption that the learning to rank methods are (approximately) distributed uniformly over the benchmarks, such that the average learning to rank method tested are approximately equally hard for each dataset. It could be the case however that this assumption is false and that the accurateness of the learning to rank methods on a dataset is not dataset independent.

A second limitation is that the datasets on which learning to rank methods have been evaluated cannot always be regarded a random choice. It might be the case that some researchers chose to publish results for exactly those benchmark datasets that showed the most positive results for their learning to rank method.

Another limitation is that our comparison methodology relies on the correctness of the evaluation results found in the literature search step. This brings up

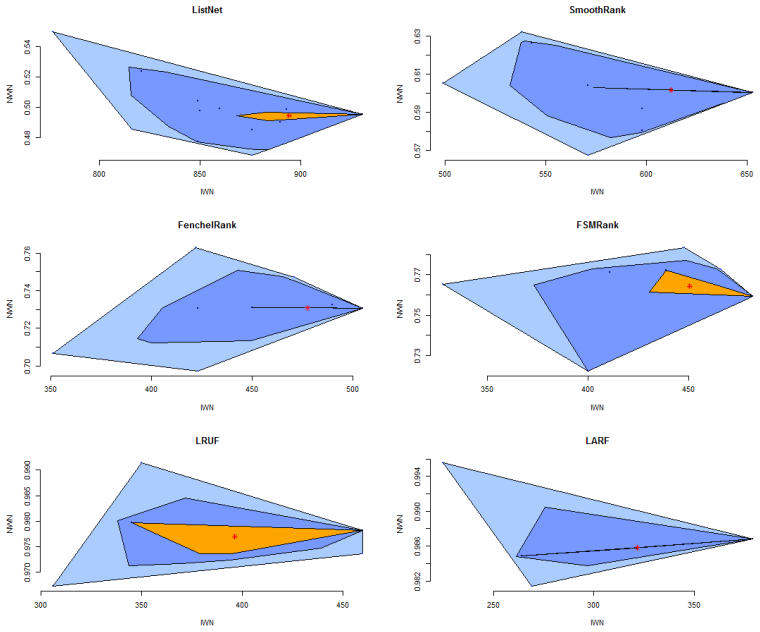

Figure 6: Bagplots showing the variance in NWN and IWN of the Pareto optimal rankers when a dataset is left out of the comparison

a risk of overly optimistic evaluation results affecting our NWN results. Limiting the meta-analysis to those studies that report comparable results on one of the baseline methods of a benchmark set reduces this limitation but does not solve it completely. By taking IWN into account in Figure 5 we further mitigate this limitation, as IWN is loosely related with the number of studies that reported evaluation results for an algorithm.

Our comparison regarded evaluation results on NDCG@$\{3, 5, 10\}$ and MAP. By making the decision to include NDCG at three cut-off points and only a single MAP entry, we implicitly attain a higher weight for NDCG compared to MAP on an analysis that combines all measurements on the four metrics. This implicit weighting could be regarded as arbitrary, but the number of algorithm evaluation results gained by this makes it a pragmatic approach. Note that another implicit weighting lies in the paper dimension. Hence, the higher number of evaluation results specified in a paper, the higher the influence of this paper on the outcome of the analysis. This implicit weighting is not harmful to the validity of our comparison, as papers with a large number of evaluation results are more valuable than papers with a few evaluation results. In addition, papers with a high number of evaluation results are not expected to be less reliable than papers with fewer evaluation results.

7. Contributions

We proposed a new way of comparing learning to rank methods based on sparse evaluation results data on a set of benchmark datasets. Our comparison methodology comprises of two components: 1) NWN, which provides insight in the ranking accuracy of the learning to rank method, and 2) IWN, which gives insight in the degree of certainty concerning the performance of the ranking accuracy.

Based on our literature search for evaluation results on well-known benchmarks collections, a lot of insight has been gained with the cross-benchmark comparison on which methods tend to perform better than others. However, no closing arguments can be formulated on which learning to rank methods are most accurate. LRUF, FSMRank, FenchelRank, SmoothRank and ListNet were found to be the Pareto optimal learning to rank algorithms in the NWN and IWN dimensions: for these ranking algorithm it holds that no other algorithm produced both more accurate rankings (NWN) and a higher degree of certainty of ranking accuracy (IWN). From left to right, the ranking accuracy of these methods decreases while the certainty of the ranking accuracy increases.

More evaluation runs are needed for the methods on the left side of Figure 5. Our work contributes to this by identifying promising learning to rank methods that researchers could focus on in performing additional evaluation runs.

References

Acharyya, S., Koyejo, O., and Ghosh, J. (2012). Learning to rank with Bregman divergences and monotone retargeting. In *Proceedings of the 28th Conference on Uncertainty in artificial intelligence (UAI)*.

Adams, R. P. and Zemel, R. S. (2011). Ranking via Sinkhorn Propagation. *arXiv preprint arXiv:1106.1925*.

Agarwal, S. and Collins, M. (2010). Maximum Margin Ranking Algorithms for Information Retrieval. In *Proceedings of the 32nd European Conference on Information Retrieval Research (ECIR)*, pages 332–343.

Ah-Pine, J. (2008). Data Fusion in Information Retrieval Using Consensus Aggregation Operators. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 1, pages 662–668.

Alcântara, O. D., Pereira Jr, Á. R., Almeida, H. M., Gonçalves, M. A., Middleton, C., and Baeza-Yates, R. (2010). Wcl2r: A benchmark collection for learning to rank research with clickthrough data. *Journal of Information and Data Management*, 1(3):551.

Alejo, O., Fernández-Luna, J. M., Huete, J. F., and Pérez-Vázquez, R. (2010). Direct Optimization of Evaluation Measures in Learning to Rank Using Particle Swarm. In *Proceedings of the Workshop on Database and Expert Systems Applications (DEXA)*, pages 42–46.

Argentini, A. (2012). *Ranking Aggregation Based on Belief Function Theory*. PhD thesis, University of Trento.

Asadi, N. (2013). *Multi-Stage Search Architectures for Streaming Documents*. PhD thesis, University of Maryland.

Asadi, N. and Lin, J. (2013). Training Efficient Tree-Based Models for Document Ranking. In *Proceedings of the 25th European Conference on Advances in Information Retrieval*, volume 7814, pages 146–157.

Banerjee, S., Dubey, A., Machchhar, J., and Chakrabarti, S. (2009). Efficient and accurate local learning for ranking. In *SIGIR workshop on Learning to rank for information retrieval*, pages 1–8.

Benbouzid, D., Busa-Fekete, R., and Kégl, B. (2012). Fast classification using sparse decision DAGs. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 951–958.

Bian, J. (2010). *Contextualized Web Search: Query-dependent Ranking and Social Media Search*. PhD thesis, Georgia Institute of Technology.

Bian, J., Li, X., Li, F., Zheng, Z., and Zha, H. (2010). Ranking Specialization for Web Search: A Divide-and-conquer Approach by Using Topical RankSVM. In *Proceedings of the 19th International Conference on World Wide Web*.

Bidoki, A. M. Z. and Thom, J. (2009). Combination of Documents Features Based on Simulated Click-through Data. In *Proceedings of the 31st European Conference on Information Retrieval Research (ECIR)*, volume 5478, pages 538–545.

Bollegala, D., Noman, N., and Iba, H. (2011). RankDE: Learning a Ranking Function for Information Retrieval using Differential Evolution. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, pages 1771–1778.

Buffoni, D., Gallinari, P., Usunier, N., and Calauzènes, C. (2011). Learning scoring functions with order-preserving losses and standardized supervision. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 825–832.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. (2005). Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 89–96.

Burges, C. J. C. (2010). From RankNet to LambdaRank to LambdaMART: An overview. Technical report, Microsoft Research.

Burges, C. J. C., Ragno, R., and Le, Q. V. (2006). Learning to rank with nonsmooth cost functions. In *Advances in Neural Information Processing Systems (NIPS)*, volume 6, pages 193–200.

Busa-Fekete, R., Kégl, B., Éltető, T., and Szarvas, G. (2013). Tune and mix: learning to rank using ensembles of calibrated multi-class classifiers. *Machine learning*, 93(2-3):261–292.

Cai, F., Guo, D., Chen, H., and Shu, Z. (2012). Your Relevance Feedback Is Essential: Enhancing the Learning to Rank Using the Virtual Feature Based Logistic Regression. *PloS one*, 7(12):e50112.

Cao, Z., Qin, T., Liu, T. Y., Tsai, M. F., and Li, H. (2007). Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, pages 129–136.

Carvalho, V. R., Elsas, J. L., Cohen, W. W., and Carbonell, J. G. (2008). Suppressing Outliers in Pairwise Preference Ranking. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM)*, pages 1487–1488.

Chakrabarti, S., Khanna, R., Sawant, U., and Bhattacharyya, C. (2008). Structured Learning for Non-smooth Ranking Losses. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 88–96.

Chang, X. and Zheng, Q. (2009). Preference Learning to Rank with Sparse Bayesian. In *Proceedings of the IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT) - Volume III*, pages 143–146.

Chapelle, O. and Chang, Y. (2011). Yahoo! Learning to Rank Challenge Overview. *Journal of Machine Learning Research-Proceedings Track*, 14:1–24.

Chapelle, O. and Wu, M. (2010). Gradient descent optimization of smoothed information retrieval metrics. *Information retrieval*, 13(3):216–235.

Chen, D., Xiong, Y., Yan, J., Xue, G. R., Wang, G., and Chen, Z. (2010). Knowledge transfer for cross domain learning to rank. *Information Retrieval*, 13(3):236–253.

Chen, M., Weinberger, K. Q., Chapelle, O., Kedem, D., and Xu, Z. (2012). Classifier cascade for minimizing feature evaluation cost. In *International Conference on Artificial Intelligence and Statistics*, pages 218–226.

Chen, X. W., Wang, H., and Lin, X. (2009). Learning to rank with a novel kernel perceptron method. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, pages 505–512.

Ciaramita, M., Murdock, V., and Plachouras, V. (2008). Online Learning from Click Data for Sponsored Search. In *Proceedings of the 17th International Conference on World Wide Web*, pages 227–236.

Cossock, D. and Zhang, T. (2006). Subset ranking using regression. In *Proceedings of the 19th Annual Conference on Learning Theory (COLT)*, pages 605–619.

Dammak, F., Kammoun, H., and Ben Hamadou, A. (2011). An Extension of RankBoost for semi-supervised Learning of Ranking Functions. In *Proceedings of the Fifth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM*, pages 49–54.

De, A. (2013). On the Role of Compensatory Operators in Fuzzy Result Merging for Metasearch. In *Proceedings of the Fifth Internation Conference on Pattern Recognition and Machine Intelligence (RReMI)*, volume 8251, pages 551–556.

De, A. and Diaz, E. (2012). Fuzzy Analytical Network Models for Metasearch. In *Revised and Selected Papers of the International Joint Conference on Computational Intelligence (IJCCI)*, volume 399, pages 197–210.

De, A., Diaz, E., and Raghavan, V. V. (2010). Search Engine Result Aggregation using Analytical Hierarchy Process. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 3, pages 300–303.

De, A. and Diaz, E. D. (2011). A Fuzzy Ordered Weighted Average (OWA) Approach to Result Merging for Metasearch Using the Analytical Network Process. In *Proceedings of the Second International Conference on Emerging Applications of Information Technology*, pages 17–20.

De, A., Diaz, E. D., and Raghavan, V. V. (2012). Weighted Fuzzy Aggregation for Metasearch: An Application of Choquet Integral. In *Proceeings of the 14th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, pages 501–510.

de Almeida, H. M., Gonçalves, M. A., Cristo, M., and Calado, P. (2007). A combined component approach for finding collection-adapted ranking functions based on genetic programming. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 399–406.

Derhami, V., Khodadadian, E., Ghasemzadeh, M., and Zareh Bidoki, A. M. (2013). Applying reinforcement learning for web pages ranking algorithms. *Applied Soft Computing*, 13(4):1686–1692.

Desarkar, M., Joshi, R., and Sarkar, S. (2011). Displacement Based Unsupervised Metric for Evaluating Rank Aggregation. In *Proceedings of the Fourth International Conference on Pattern Recognition and Machine Intelligence (PReMI)*, volume 6744, pages 268–273.

Diaz-Aviles, E., Nejdl, W., and Schmidt-Thieme, L. (2009). Swarming to rank for information retrieval. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, pages 9–16.

Ding, W., Qin, T., and Zhang, X. D. (2010). Learning to Rank with Supplementary Data. In *Proceedings of the Sixth Asia Information Retrieval Societies Conference (AIRS)*, volume 6458, pages 478–489.

Dubey, A., Machchhar, J., Bhattacharyya, C., and Chakrabarti, S. (2009). Conditional Models for Non-smooth Ranking Loss Functions. In *Proceedings of the Ninth IEEE International Conference on Data Mining (ICDM)*, pages 129–138.

Duh, K. and Kirchhoff, K. (2008). Learning to rank with partially-labeled data. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 251–258.

Duh, K. and Kirchhoff, K. (2011). Semi-supervised Ranking for Document Retrieval. *Computational Speech and Language*, 25(2):261–281.

Duh, K., Suzuki, J., and Nagata, M. (2011). Distributed Learning-to-Rank on Streaming Data using Alternating Direction Method of Multipliers. In *Proceedings of the NIPS Big Learning Workshop*.

Elsas, J. L., Carvalho, V. R., and Carbonell, J. G. (2008). Fast learning of document ranking functions with the committee perceptron. In *Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM)*, pages 55–64.

Freno, N., Papini, T., and Diligenti, M. (2011). Learning to Rank using Markov Random Fields. In *Proceeings of the 10th International Conference on Machine Learning and Applications and Workshops (ICMLA)*, volume 2, pages 257–262.

Freund, Y., Iyer, R., Schapire, R. E., and Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *The Journal of Machine Learning Research (JMLR)*, 4:933–969.

Ganjisaffar, Y., Caruana, R., and Lopes, C. V. (2011). Bagging gradient-boosted trees for high precision, low variance ranking models. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 85–94.

Gao, W. and Yang, P. (2014). Democracy is good for ranking: Towards multi-view rank learning and adaptation in web search. In *Proceedings of the 7th ACM international Conference on Web Search and Data Mining*, pages 63–72.

Geng, B., Yang, Y., Xu, C., and Hua, X. S. (2012). Ranking Model Adaptation for Domain-Specific Search. 24(4):745–758.

Geng, X., Liu, T. Y., Qin, T., Arnold, A., Li, H., and Shum, H.-Y. (2008). Query dependent ranking using k-nearest neighbor. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–122.

Geng, X., Liu, T. Y., Qin, T., and Li, H. (2007). Feature selection for ranking. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 407–414.

Geng, X., Qin, T., Liu, T. Y., Cheng, X. Q., and Li, H. (2011). Selecting optimal training data for learning to rank. *Information Processing & Management*, 47(5):730–741.

Gomes, G., Oliveira, V. C., Almeida, J. M., and Gonçalves, M. A. (2013). Is Learning to Rank Worth it? A Statistical Analysis of Learning to Rank Methods in the LETOR Benchmarks. *Journal of Information and Data Management*, 4(1):57.

Guiver, J. and Snelson, E. (2008). Learning to rank with softrank and gaussian processes. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 259–266.

He, Q., Ma, J., and Niub, X. (2010a). Learning to Rank for Information Retrieval Using the Clonal Selection Algorithm. *Journal of Information and Computational Science*, 7(1):153–159.

He, Q., Ma, J., and Wang, S. (2010b). Directly optimizing evaluation measures in learning to rank based on the clonal selection algorithm. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1449–1452.

Herbrich, R., Graepel, T., and Obermayer, K. (1999). Support vector learning for ordinal regression. In *Proceedings of the Ninth International Conference on Artificial Neural Networks*, volume 1, pages 97–102 vol.1.

Hoi, S. C. H. and Jin, R. (2008). Semi-supervised Ensemble Ranking. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, pages 634–639.

Huang, J. C. and Frey, B. J. (2008). Structured ranking learning using cumulative distribution networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 697–704.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142.

Kao, C. Y. and Fahn, C. S. (2013). A multi-stage learning framework for intelligent system. *Expert Systems with Applications*, 40(9):3378–3388.

Karimzadehgan, M., Li, W., Zhang, R., and Mao, J. (2011). A stochastic learning-to-rank algorithm and its application to contextual advertising. In *Proceedings of the 20th International conference on World Wide web*, pages 377–386.

Kersting, K. and Xu, Z. (2009). Learning Preferences with Hidden Common Cause Relations. In *Proceedings of the European Conference on Machine Learning (ECML)and Knowledge Discovery in Databases (KDD): Part I*, pages 676–691.

Kuo, J. W., Cheng, P. J., and Wang, H. M. (2009). Learning to Rank from Bayesian Decision Inference. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, pages 827–836.

Lai, H., Pan, Y., Liu, C., Lin, L., and Wu, J. (2013a). Sparse learning-to-rank via an efficient primal-dual algorithm. *IEEE Transactions on Computers*, 62(6):1221–1233.

Lai, H., Pan, Y., Tang, Y., and Liu, N. (2013b). Efficient gradient descent algorithm for sparse models with application in learning-to-rank. *Knowledge-Based Systems*, 49:190–198.

Lai, H., Tang, Y., Luo, H. X., and Pan, Y. (2011). Greedy feature selection for ranking. In *Proceedings of the 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 42–46.

Lai, H. J., Pan, Y., Tang, Y., and Yu, R. (2013c). FSMRank: Feature Selection Algorithm for Learning to Rank. *IEEE transactions on Neural Networks and Learning Systems*, 24(6):940–952.

Laporte, L., Flamary, R., Canu, S., Dejean, S., and Mothe, J. (2013). Nonconvex Regularizations for Feature Selection in Ranking With Sparse SVM. PP(99):1.

Le, Q. and Smola, A. (2007). Direct optimization of ranking measures. Technical report, NICTA.

Li, D., Wang, Y., Ni, W., Huang, Y., and Xie, M. (2008). An Ensemble Approach to Learning to Rank. In *Proceedings of the Fifth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, volume 2, pages 101–105.

Lin, H. Y., Yu, C. H., and Chen, H. H. (2011a). Query-Dependent Rank Aggregation with Local Models. In *Proceedings of the Seventh Asia Information Retrieval Societies Conference (AIRS)*, volume 7097, pages 1–12.

Lin, J. Y., Yeh, J. Y., and Liu, C. C. (2012). Learning to rank for information retrieval using layered multi-population genetic programming. In *Proceedings of the 2012 IEEE International Conference on Computational Intelligence and Cybernetics (CyberneticsCom)*, pages 45–49.

Lin, Y., Lin, H., Wu, J., and Xu, K. (2011b). Learning to rank with cross entropy. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2057–2060.

Lin, Y., Lin, H., Yang, Z., and Su, S. (2009). A Boosting Approach for Learning to Rank Using SVD with Partially Labeled Data. In *Proceedings of the fifth Asia Information Retrieval Symposium (AIRS)*, pages 330–338.

Lin, Y., Lin, H., Ye, Z., Jin, S., and Sun, X. (2010). Learning to rank with groups. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1589–1592.

Liu, T. Y., Xu, J., Qin, T., Xiong, W., and Li, H. (2007). LETOR: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of the SIGIR 2007 workshop on learning to rank for information retrieval*, pages 3–10.

McNee, S. M., Riedl, J., and Konstan, J. A. (2006). Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing systems*, pages 1097–1101.

Metzler, D. and Croft, W. B. (2007). Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274.

Miao, Z. and Tang, K. (2013). Semi-supervised Ranking via List-Wise Approach. In *Proceedings of the 14th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)*, pages 376–383.

Mohan, A., Chen, Z., and Weinberger, K. Q. (2011). Web-Search Ranking with Initialized Gradient Boosted Regression Trees. *Journal of Machine Learning Research (JMLR)*, 14:77–89.

Moon, T., Smola, A., Chang, Y., and Zheng, Z. (2010). IntervalRank: isotonic regression with listwise and pairwise constraints. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 151–160.

Ni, W., Huang, Y., and Xie, M. (2008). A Query Dependent Approach to Learning to Rank for Information Retrieval. In *Proceedings of the 2008 The Ninth International Conference on Web-Age Information Management (WAIM)*, pages 262–269.

Niu, S., Guo, J., Lan, Y., and Cheng, X. (2012). Top-k learning to rank: labeling, ranking and evaluation. In *Proceedings of the 35th international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 751–760.

Pahikkala, T., Airola, A., Naula, P., and Salakoski, T. (2010). Greedy RankRLS: a Linear Time Algorithm for Learning Sparse Ranking Models. In *SIGIR 2010 Workshop on Feature Generation and Selection for Information Retrieval*, pages 11–18.

Pahikkala, T., Tsivtsivadze, E., Airola, A., Järvinen, J., and Boberg, J. (2009). An efficient algorithm for learning to rank from preference graphs. *Machine Learning*, 75(1):129–165.

Pan, Y., Lai, H., Liu, C., Tang, Y., and Yan, S. (2013). Rank Aggregation via Low-Rank and Structured-Sparse Decomposition. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*.

Pan, Y., Luo, H.-X., Tang, Y., and Huang, C.-Q. (2011). Learning to rank with document ranks and scores. *Knowledge-Based Systems*, 24(4):478–483.

Papini, T. and Diligenti, M. (2012). Learning-to-rank with Prior Knowledge as Global Constraints. In *Proceedings of the First International Workshop on Combining Constraint Solving with Mining and Learning (CoCoMiLe)*.

Pass, G., Chowdhury, A., and Torgeson, C. (2006). A picture of search. In *Proceedings of the First International Conference on Scalable Information Systems (InfoScale)*, volume 152.

Pavlov, D. Y., Gorodilov, A., and Brunk, C. A. (2010). BagBoo: a scalable hybrid bagging-the-boosting model. In *Proceedings of the 19th ACM International conference on Information and Knowledge Management (CIKM)*, pages 1897–1900.

Peng, J., Macdonald, C., and Ounis, I. (2010a). Learning to Select a Ranking Function. In *Proceedings of the 32nd European Conference on Information Retrieval Research (ECIR)*, pages 114–126, Berlin, Heidelberg.

Peng, Z., Tang, Y., Lin, L., and Pan, Y. (2010b). Learning to rank with a Weight Matrix. In *Proceedings of the 14th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 18–21.

Petterson, J., Yu, J., Mcauley, J. J., and Caetano, T. S. (2009). Exponential Family Graph Matching and Ranking. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1455–1463.

Qin, T., Geng, X., and Liu, T. Y. (2010a). A New Probabilistic Model for Rank Aggregation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 10, pages 1948–1956.

Qin, T., Liu, T., Xu, J., and Li, H. (2010b). LETOR: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13(4):346–374.

Qin, T., Liu, T. Y., and Li, H. (2010c). A general approximation framework for direct optimization of information retrieval measures. *Information retrieval*, 13(4):375–397.

Qin, T., Liu, T. Y., Zhang, X. D., Wang, D. S., and Li, H. (2008a). Global ranking using continuous conditional random fields. In *Advances in neural information processing systems (NIPS)*, pages 1281–1288.

Qin, T., Liu, T. Y., Zhang, X. D., Wang, D. S., Xiong, W. Y., and Li, H. (2008b). Learning to Rank Relational Objects and Its Application to Web Search. In *Proceedings of the 17th International Conference on World Wide Web*, pages 407–416.

Qin, T., Zhang, X. D., Wang, D. S., Liu, T. Y., Lai, W., and Li, H. (2007). Ranking with multiple hyperplanes. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 279–286.

Ravikumar, P. D., Tewari, A., and Yang, E. (2011). On NDCG consistency of listwise ranking methods. In *International Conference on Artificial Intelligence and Statistics*, pages 618–626.

Renjifo, C. and Carmen, C. (2012). The discounted cumulative margin penalty: Rank-learning with a list-wise loss and pair-wise margins. In *Proceedings of the 2012 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6.

Rigutini, L., Papini, T., Maggini, M., and Scarselli, F. (2008). Sortnet: Learning to rank by a neural-based sorting algorithm. In *Proceedings of the SIGIR Workshop on Learning to Rank for Information Retrieval (LR4IR)*, pages 76–79.

Robertson, S. E. and Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 232–241.

Rousseeuw, P. J., Ruts, I., and Tukey, J. W. (1999). The bagplot: a bivariate boxplot. *The American Statistician*, 53(4):382–387.

Rudin, C. (2009). The P-Norm Push: A simple convex ranking algorithm that concentrates at the top of the list. *The Journal of Machine Learning Research*, 10:2233–2271.

Sato, H., Bollegala, D., Hasegawa, Y., and Iba, H. (2013). Learning non-linear ranking functions for web search using probabilistic model building GP. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 3371–3378.

Sculley, D. (2009). Large scale learning to rank. In *NIPS Workshop on Advances in Ranking*, pages 1–6.

Shivaswamy, P. K. and Joachims, T. (2011). Online learning with preference feedback. *arXiv preprint arXiv:1111.0712*.

Silva, T. P. C., de Moura, E. S., Cavalcanti, J. a. M. B., da Silva, A. S., de Carvalho, M. G., and Gonçalves, M. A. (2009). An evolutionary approach for combining different sources of evidence in search engines. *Information Systems*, 34(2):276–289.

Song, Y., Leung, K., Fang, Q., and Ng, W. (2013). FP-Rank: An Effective Ranking Approach Based on Frequent Pattern Analysis. In *Database Systems for Advanced Applications*, pages 354–369.

Song, Y., Ng, W., Leung, K. W. T., and Fang, Q. (2014). SFP-Rank: significant frequent pattern analysis for effective ranking. *Knowledge and Information Systems*, pages 1–25.

Stewart, A. and Diaz, E. (2012). Epidemic Intelligence: For the Crowd, by the Crowd. In *Proceedings of the 12th International Conference on Web Engineering (ICWE)*, pages 504–505, Berlin, Heidelberg.

Sun, H., Huang, J., and Feng, B. (2011). QoRank: A query-dependent ranking model using LSE-based weighted multiple hyperplanes aggregation for information retrieval. *International Journal of Intelligent Systems*, 26(1):73–97.

Sun, Z., Qin, T., Tao, Q., and Wang, J. (2009). Robust sparse rank learning for non-smooth ranking measures. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 259–266.

Swersky, K., Tarlow, D., Adams, R., Zemel, R., and Frey, B. (2012). Probabilistic n-Choose-k Models for Classification and Ranking. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3059–3067.

Tan, M., Xia, T., Guo, L., and Wang, S. (2013). Direct optimization of ranking measures for learning to rank models. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 856–864.

Taylor, M., Guiver, J., Robertson, S., and Minka, T. (2008). Softrank: optimizing non-smooth rank metrics. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*, pages 77–86.

Thuy, N. T. T., Vien, N. A., Viet, N. H., and Chung, T. C. (2009). Probabilistic Ranking Support Vector Machine. In *Proceedings of the Sixth International Symposium on Neural Networks*, volume 5552, pages 345–353.

Torkestani, J. A. (2012a). An adaptive learning automata-based ranking function discovery algorithm. *Journal of Intelligent Information Systems*, 39(2):441–459.

Torkestani, J. A. (2012b). An adaptive learning to rank algorithm: Learning automata approach. *Decision Support Systems*, 54(1):574–583.

Tran, T. T. and Pham, D. S. (2012). ConeRANK: Ranking as Learning Generalized Inequalities. *arXiv preprint arXiv:1206.4110*.

Tsai, M.-F., Liu, T. Y., Qin, T., Chen, H. H., and Ma, W. Y. (2007). FRank: a ranking method with fidelity loss. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 383–390.

Usunier, N., Buffoni, D., and Gallinari, P. (2009). Ranking with ordered weighted pairwise classification. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1057–1064.

Veloso, A., Gonçalves, M. A., Meira Jr, W., and Mossri, H. (2010). Learning to rank using query-level rules. *Journal of Information and Data Management*, 1(3):567.

Veloso, A. A., Almeida, H. M., Gonçalves, M. A., and Meira Jr, W. (2008). Learning to rank at query-time using association rules. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 267–274.

Volkovs, M. N., Larochelle, H., and Zemel, R. S. (2011). Loss-sensitive Training of Probabilistic Conditional Random Fields. *arXiv preprint arXiv:1107.1805*.

Volkovs, M. N. and Zemel, R. S. (2009). Boltzrank: learning to maximize expected ranking gain. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1089–1096.

Volkovs, M. N. and Zemel, R. S. (2012). A Flexible Generative Model for Preference Aggregation. In *Proceedings of the 21st International Conference on World Wide Web*, pages 479–488.

Volkovs, M. N. and Zemel, R. S. (2013). CRF Framework for Supervised Preference Aggregation. In *Proceedings of the 22Nd ACM International Conference on Conference on Information and Knowledge Management (CIKM)*, pages 89–98.

Wang, B., Tang, J., Fan, W., Chen, S., Yang, Z., and Liu, Y. (2009a). Heterogeneous Cross Domain Ranking in Latent Space. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, pages 987–996.

Wang, C. J., Huang, H. S., and Chen, H. H. (2012a). Automatic construction of an evaluation dataset from wisdom of the crowds for information retrieval applications. In *Proceedings of the 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 490–495.

Wang, F. and Xu, X. (2010). AdaGP-Rank: Applying boosting technique to genetic programming for learning to rank. In *Proceedings of the IEEE Youth Conference on Information Computing and Telecommunications (YC-ICT)*, pages 259–262.

Wang, L., Bennett, P. N., and Collins-Thompson, K. (2012b). Robust Ranking Models via Risk-sensitive Optimization. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 761–770.

Wang, S., Gao, B. J., Wang, K., and Lauw, H. W. (2011). CCRank: Parallel Learning to Rank with Cooperative Coevolution. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*.

Wang, S., Ma, J., and Liu, J. (2009b). Learning to Rank using Evolutionary Computation: Immune Programming or Genetic Programming? In *Proceedings of the 18th ACM conference on Information and knowledge management (CIKM)*, pages 1879–1882.

Wang, Y., Huang, Y., Pang, X., Lu, M., Xie, M., and Liu, J. (2013). Supervised rank aggregation based on query similarity for document retrieval. *Soft Computing*, 17(3):421–429.

Wang, Y., Kuai, Y. H., Huang, Y. L., Li, D., and Ni, W. J. (2008). Uncertainty-based active ranking for document retrieval. In *Proceedings of the International Conference on Machine Learning and Cybernetics*, volume 5, pages 2629–2634.

Wu, J., Yang, Z., Lin, Y., Lin, H., Ye, Z., and Xu, K. (2011). Learning to rank using query-level regression. In *Proceedings of the 34th international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1091–1092.

Wu, M., Chang, Y., Zheng, Z., and Zha, H. (2009). Smoothing DCG for learning to rank: A novel approach using smoothed hinge functions. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, pages 1923–1926.

Xia, F., Liu, T. Y., Wang, J., Zhang, W., and Li, H. (2008). Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th Annual International Conference on Machine Learning*, pages 1192–1199.

Xu, J. and Li, H. (2007). Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 391–398.

Xu, J., Liu, T. Y., Lu, M., Li, H., and Ma, W. Y. (2008). Directly Optimizing Evaluation Measures in Learning to Rank. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Rnformation Retrieval*, pages 107–114.

Xu, Z., Chapelle, O., and Weinberger, K. Q. (2012). The Greedy Miser: Learning under Test-time Budgets. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 1175–1182.

Xu, Z., Kersting, K., and Joachims, T. (2010). Fast Active Exploration for Link-Based Preference Learning Using Gaussian Processes. In *Proceedings of the European Conference on Machine Learning (ECML) and Principles and Practice of Knowledge Discovery in Databases (PKDD)*, volume 6323, pages 499–514.

Yu, C.-N. J. and Joachims, T. (2009). Learning Structural SVMs with Latent Variables. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 1169–1176, New York, NY, USA.

Yue, Y., Finley, T., Radlinski, F., and Joachims, T. (2007). A support vector method for optimizing average precision. In *Proceedings of the 30th Annual Unternational ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 271–278.

Zhou, D., Ding, Y., You, Q., and Xiao, M. (2011). Learning to Rank Documents Using Similarity Information Between Objects. In *Proceedings of the 18th International Conference on Neural Information Processing (ICONIP) - Volume Part II*, pages 374–381.

Zhou, K., Xue, G.-R., Zha, H., and Yu, Y. (2008). Learning to rank with ties. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–282.

Zhu, C., Chen, W., Zhu, Z. A., Wang, G., Wang, D., and Chen, Z. (2009). A General Magnitude-preserving Boosting Algorithm for Search Ranking. In

Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM), pages 817–826.

Zong, W. and Huang, G. B. (2013). Learning to Rank with Extreme Learning Machine. *Neural Processing Letters*, pages 1–12.

Appendix A. Meta-analysis Ranking Methods & Data Sources

Method	Described in	Evaluated in
AdaRank-MAP	Xu and Li (2007)	L2, L3, L4
AdaRank-NDCG	Xu and Li (2007)	L2, L3, L4, Busa-Fekete et al. (2013); Tan et al. (2013)
ADMM	Duh et al. (2011)	Duh et al. (2011)
ApproxAP	Qin et al. (2010c)	Qin et al. (2010c)
ApproxNDCG	Qin et al. (2010c)	Qin et al. (2010c)
BagBoo	Pavlov et al. (2010)	Ganjisaffar et al. (2011)
Best Single Feature		Gomes et al. (2013)
BL-MART	Ganjisaffar et al. (2011)	Ganjisaffar et al. (2011)
BoltzRank-Single	Volkovs and Zemel (2009)	Volkovs and Zemel (2009, 2013)
BoltzRank-Pair	Volkovs and Zemel (2009)	Volkovs and Zemel (2009); Ganjisaffar et al. (2011); Volkovs and Zemel (2013)
BT	Zhou et al. (2008)	Zhou et al. (2008)
C-CRF	Qin et al. (2008a)	Qin et al. (2008a)
CA	Metzler and Croft (2007)	Busa-Fekete et al. (2013); Tan et al. (2013)
CCRank	Wang et al. (2011)	Wang et al. (2011)
CoList	Gao and Yang (2014)	Gao and Yang (2014)
Consistent-RankCosine	Ravikumar et al. (2011)	Tan et al. (2013)
DCMP	Renjifo and Carmen (2012)	Renjifo and Carmen (2012)
DirectRank	Tan et al. (2013)	Tan et al. (2013)
EnergyNDCG	Freno et al. (2011)	Freno et al. (2011)
FBPCRank	Lai et al. (2011)	Lai et al. (2011)
FenchelRank	Lai et al. (2013a)	Lai et al. (2013a,b); Laporte et al. (2013)
FocusedBoost	Niu et al. (2012)	Niu et al. (2012)
FocusedNet	Niu et al. (2012)	Niu et al. (2012)
FocusedSVM	Niu et al. (2012)	Niu et al. (2012)
FP-Rank	Song et al. (2013)	Song et al. (2013)
FRank	Tsai et al. (2007)	L2, L3, Wang et al. (2012a)
FSMRank	Lai et al. (2013c)	Lai et al. (2013c); Laporte et al. (2013)
FSM^{SVM}	Lai et al. (2013c)	Lai et al. (2013c)
GAS-E	Geng et al. (2007)	Lai et al. (2013c)
GP	de Almeida et al. (2007)	Alcântara et al. (2010)
GPRank	Silva et al. (2009)	Torkestani (2012a)
GRankRLS	Pahikkala et al. (2010)	Pahikkala et al. (2010)
GroupCE	Lin et al. (2011b)	Lin et al. (2011b)
GroupMLE	Lin et al. (2010)	Lin et al. (2011b)
IntervalRank	Moon et al. (2010)	Moon et al. (2010); Freno et al. (2011)
IPRank	Wang et al. (2009b)	Wang et al. (2009b); Torkestani (2012a)
KeepRank	Chen et al. (2009)	Chen et al. (2009)
KL-CRF	Volkovs et al. (2011)	Volkovs et al. (2011)
LAC-MR-OR	Veloso et al. (2008)	Veloso et al. (2008); Alcântara et al. (2010)
LambdaMART	Burges (2010)	Asadi and Lin (2013); Ganjisaffar et al. (2011)
LambdaNeuralRank	Papini and Diligenti (2012)	Papini and Diligenti (2012)
LambdaRank	Burges et al. (2006)	Papini and Diligenti (2012); Tan et al. (2013)
LARF	Torkestani (2012a)	Torkestani (2012a)
Linear Regression	Cossock and Zhang (2006)	L3, Wang et al. (2012a); Volkovs et al. (2011)
ListMLE	Xia et al. (2008)	Lin et al. (2010, 2011b); Gao and Yang (2014)
ListNet	Cao et al. (2007)	L2, L3, L4
ListReg	Wu et al. (2011)	Wu et al. (2011)
LRUF	Torkestani (2012b)	Torkestani (2012b)
MCP	Laporte et al. (2013)	Laporte et al. (2013)
MHR	Qin et al. (2007)	L2
MultiStageBoost	Kao and Fahn (2013)	Kao and Fahn (2013)
NewLoss	Peng et al. (2010b)	Peng et al. (2010b)
OWPC	Usunier et al. (2009)	Usunier et al. (2009)
PERF-MAP	Pan et al. (2011)	Torkestani (2012b)
PermuRank	Xu et al. (2008)	Xu et al. (2008)
Q.D.KNN	Geng et al. (2008)	Wang et al. (2013)
RandomForest	Gomes et al. (2013)	Gomes et al. (2013)
Rank-PMBGP	Sato et al. (2013)	Sato et al. (2013)
RankAggNDCG	Wang et al. (2013)	Wang et al. (2013)
RankBoost	Freund et al. (2003)	L2, L3, L4, Busa-Fekete et al. (2013); Alcântara et al. (2010); Sato et al. (2013)
RankBoost (Kernel-PCA)	Duh and Kirchhoff (2008)	Duh and Kirchhoff (2008); Sato et al. (2013)
RankBoost (SVD)	Lin et al. (2009)	Lin et al. (2009)
RankCSA	He et al. (2010b)	He et al. (2010b)
RankDE	Bollegala et al. (2011)	Sato et al. (2013)
RankELM (pairwise)	Zong and Huang (2013)	Zong and Huang (2013)
RankELM (pointwise)	Zong and Huang (2013)	Zong and Huang (2013)
RankMGP	Lin et al. (2012)	Lin et al. (2012)
RankNet	Burges et al. (2005)	Busa-Fekete et al. (2013); Papini and Diligenti (2012); Niu et al. (2012)
RankRLS	Pahikkala et al. (2009)	Pahikkala et al. (2010)
RankSVM	Herbrich et al. (1999); Joachims (2002)	L2, L3, Busa-Fekete et al. (2013); Freno et al. (2011); He et al. (2010b); Alcântara et al. (2010); Papini and Diligenti (2012)
RankSVM-Struct		L3, L4
RankSVM-Primal		L3, Lai et al. (2011)
RCP	Elsas et al. (2008)	Elsas et al. (2008)
RE-QR	Veloso et al. (2010)	Veloso et al. (2010)
REG-SHF-SDCG	Wu et al. (2009)	Wu et al. (2009)
Ridge Regression	Cossock and Zhang (2006)	L3
RSRank	Sun et al. (2009)	Lai et al. (2013a)
SmoothGrad	Le and Smola (2007)	Tan et al. (2013)
SmoothRank	Chapelle and Wu (2010)	L3, Chapelle and Wu (2010)
SoftRank	Taylor et al. (2008); Guiver and Snelson (2008)	Qin et al. (2010c)
SortNet	Rigutini et al. (2008)	Rigutini et al. (2008); Freno et al. (2011); Papini and Diligenti (2012)
SparseRank	Lai et al. (2013b)	Lai et al. (2013b)
SVM^{MAP}	Yue et al. (2007)	L3, Wang et al. (2012a); Xu et al. (2008); Niu et al. (2012)
SwarmRank	Diaz-Aviles et al. (2009)	Sato et al. (2013)
TGRank	Lai et al. (2013a)	Lai et al. (2013a)
TM	Zhou et al. (2008)	Zhou et al. (2008); Papini and Diligenti (2012); Tan et al. (2013)
VFLR	Cai et al. (2012)	Cai et al. (2012)

Table A.5: Learning to rank algorithms with measurements on benchmark datasets

Appendix B. Meta-analysis Raw Data

	NDCG@3		NDCG@5		NDG@10		MAP		CROSS		
Method	NWN	#ds	NWN	#ds	NWN	#ds	NWN	#ds	WN	IWN	NWN
AdaRank-MAP	0.3529	12	0.3884	12	0.3648	13	0.3206	12	334	940	0.3553
AdaRank-NDCG	0.3122	12	0.3259	12	0.3158	16	0.2863	12	295	954	0.3092
ADMM	-	-	-	-	0.4444	1	-	-	4	9	0.4444
ApproxAP	-	-	-	-	-	-	0.5000	2	33	66	0.5000
ApproxNDCG	0.8000	1	0.7500	1	0.8611	1	-	-	93	116	0.8017
BagBoo	0.8333	2	0.8400	1	-	-	0.6545	2	97	128	0.7578
Best Single Feature	-	-	-	-	0.1615	8	-	-	26	161	0.1615
BL-MART	0.8776	3	0.7200	1	-	-	0.8036	3	106	130	0.8154
BoltzRank-Pair	0.8286	4	0.8350	4	-	-	0.5804	5	256	351	0.7293
BoltzRank-Single	0.7524	4	0.7184	4	-	-	0.4336	5	215	351	0.6125
BT	0.7273	3	0.7879	3	-	-	0.7500	3	75	99	0.7576
C-CRF	-	-	0.9500	2	-	-	-	-	19	20	0.9500
CA	-	-	-	-	0.6522	4	-	-	15	23	0.6522
CCRank	-	-	-	-	-	-	0.6154	2	24	39	0.6154
CoList	1.0000	1	1.0000	1	0.1667	1	-	-	3	8	0.3750
Consistent-RankCosine	-	-	-	-	0.7692	2	-	-	10	13	0.7692
DCMP	0.5477	9	0.5079	9	0.5888	9	-	-	322	587	0.5486
DirectRank	-	-	-	-	0.9231	2	-	-	12	13	0.9231
EnergyNDCG	0.3778	2	0.3778	2	0.4146	2	-	-	51	131	0.3893
FBPCRank	0.4235	3	0.5529	3	-	-	-	-	83	170	0.4882
FenchelRank	0.7760	5	0.7500	5	0.7623	5	0.6418	5	369	505	0.7307
FocusedBoost	0.3753	2	0.4545	2	0.6863	2	-	-	73	143	0.5105
FocusedNet	0.4583	2	0.6364	2	0.8627	2	-	-	94	143	0.6573
FocusedSVM	0.2371	2	0.2727	2	0.6078	2	-	-	55	143	0.3846
FP-Rank	-	-	0.9000	1	-	-	-	-	18	20	0.9000
FRank	0.3137	11	0.2849	10	0.3029	11	0.2623	11	244	842	0.2898
FSMRank	0.8351	4	0.8776	4	0.8621	5	0.5789	7	366	482	0.7593
FSM^{SVM}	0.2292	2	0.4082	4	0.5426	4	0.3500	4	149	389	0.3830
GAS-E	0.3814	4	0.4694	4	0.4574	4	0.4100	4	167	389	0.4293
GP	-	-	-	-	0.6667	2	0.5000	2	7	12	0.5833
GPRank	0.8750	3	0.7253	3	0.6591	3	0.8173	3	293	379	0.7731
GRankRLS	-	-	-	-	0.2895	2	-	-	11	38	0.2895
GroupCE	0.7292	3	-	-	0.7273	3	0.7212	3	209	288	0.7257
GroupMLE	0.5208	3	-	-	0.6250	3	0.6538	3	173	288	0.6007
IntervalRank	0.6000	1	0.3750	1	-	-	0.3158	1	51	118	0.4322
IPRank	0.9375	3	0.8132	3	0.7955	3	0.8514	6	360	423	0.8511
KeepRank	-	-	-	-	-	-	0.5385	3	56	104	0.5385
KL-CRF	0.5946	2	0.5789	2	-	-	-	-	44	75	0.5867
LAC-MR-OR	-	-	-	-	0.6667	2	0.7642	12	179	235	0.7617
LambdaMART	0.4082	3	-	-	1.0000	1	0.6786	3	62	109	0.5688
LambdaNeuralRank	1.0000	1	1.0000	1	1.0000	1	-	-	15	15	1.0000
LambdaRank	0.2000	1	0.2000	1	0.5714	2	-	-	10	24	0.4167
LARF	0.9896	3	0.9890	3	0.9886	3	0.9808	3	374	379	0.9868
Linear Regression	0.0754	9	0.1099	9	0.0829	8	0.0650	8	64	771	0.0830
ListMLE	0.0000	2	0.0000	1	0.0213	4	0.00962	3	3	240	0.0125
ListNet	0.4480	12	0.4911	12	0.5982	12	0.4504	12	461	931	0.4952
ListReg	0.7292	3	0.6923	3	-	-	0.4327	3	178	291	0.6117
LRUF	0.9828	4	0.9817	4	0.9818	4	0.9680	4	450	460	0.9783
MCP	-	-	-	-	-	-	0.5714	2	40	70	0.5714
MHR	0.7500	1	0.6000	1	0.6250	1	0.0000	1	17	41	0.5714
MultiStageBoost	-	-	-	-	-	-	0.1364	2	6	44	0.1364
NewLoss	0.5208	3	0.4286	3	0.3977	3	-	-	124	275	0.4509
OWPC	0.6475	6	-	-	-	-	0.6241	6	167	263	0.6350
PERF-MAP	0.3966	4	0.2661	4	0.2000	4	0.7680	4	193	460	0.4196
PermuRank	-	-	-	-	-	-	0.4091	3	18	44	0.4091
Q.D.KNN	-	-	0.3205	3	0.5000	3	0.5584	3	105	229	0.4585
RandomForest	-	-	-	-	0.4224	8	0.4389	8	147	341	0.4311
Rank-PMBGP	-	-	0.7692	1	0.2727	1	0.8750	1	27	40	0.6750
RankAggNDCG	-	-	0.5000	3	0.8784	3	0.7922	3	165	229	0.7205
RankBoost	0.3303	12	0.2794	10	0.3936	17	0.3134	14	312	942	0.3312
RankBoost (Kernel-PCA)	-	-	0.2857	3	-	-	-	-	26	91	0.2857
RankBoost (SVD)	-	-	0.2727	3	0.5556	3	0.5682	3	49	104	0.4712
RankCSA	-	-	-	-	-	-	0.9167	2	33	36	0.9167
RankDE	-	-	0.5385	1	0.1818	1	1.0000	1	25	40	0.6250
RankELM (pairwise)	0.6475	1	0.6500	1	0.6944	1	0.5143	2	112	186	0.6022
RankELM (pointwise)	0.7000	1	0.7000	1	0.8056	1	0.5429	2	123	186	0.6613
RankMGP	-	-	-	-	-	-	0.2222	1	4	18	0.2222
RankNet	0.1887	3	0.2857	3	0.5915	5	-	-	66	173	0.3815
RankRLS	-	-	-	-	0.3684	2	-	-	14	38	0.3684
RankSVM	0.3014	12	0.3613	11	0.4496	17	0.3400	13	324	888	0.3649
RankSVM-Primal	0.3911	8	0.4509	8	0.4591	7	0.3520	7	284	690	0.4116
RankSVM-Struct	0.3518	9	0.4136	9	0.4467	9	0.3624	9	316	805	0.3925
RCP	-	-	0.5758	3	0.7407	3	0.3636	3	55	104	0.5288
RE-QR	-	-	-	-	-	-	0.8659	7	155	179	0.8659
REG-SHG-SDCG	0.4000	1	0.4500	1	-	-	0.6579	1	59	118	0.5000
Ridge Regression	0.4074	7	0.3333	7	0.3648	7	0.2905	7	227	653	0.3476
RSRank	0.5773	4	0.5306	4	0.6277	4	0.6600	4	233	389	0.5990
SmoothGrad	-	-	-	-	0.3846	2	-	-	5	13	0.3846
SmoothRank	0.6049	7	0.6340	7	0.6415	7	0.5307	7	392	653	0.6003
SoftRank	0.2500	1	0.2750	1	0.6111	1	-	-	43	116	0.3707
SortNet	0.2667	2	0.5147	4	0.5667	4	0.5000	2	114	239	0.4770
SparseRank	0.8241	4	0.8173	4	0.7944	4	-	-	259	319	0.8119
SVM^{MAP}	0.2901	7	0.3801	8	0.3591	8	0.3498	10	255	737	0.3460
SwarmRank	-	-	0.1538	1	0.0909	1	0.1250	1	5	40	0.1250
TGRank	0.5464	4	0.6122	4	0.5000	4	0.4600	4	206	389	0.5296
TM	0.5909	3	0.7576	3	-	-	0.6136	3	65	99	0.6566
VFLR	-	-	-	-	-	-	0.9744	2	38	39	0.9744

Table B.6: Raw data of cross-benchmark comparison