Information Processing and Management 51 (2015) 757–772

Contents lists available at ScienceDirect

Information Processing and Management

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

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

ARTICLE INFO

Article history:
Received 1 October 2014
Revised 14 May 2015
Accepted 7 July 2015
Available online 22 August 2015

Keywords:
Learning to rank
Information retrieval
Evaluation metric

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 the absence 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.

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 web search, where commercial search engines help users find their need in the extremely large collection of the World Wide Web. Among useful applications of machine learning based ranking outside web search are automatic text summarization, machine translation, drug discovery and determining the ideal order of maintenance operations (Rudin, 2009). In addition, McNee, Riedl, and Konstan (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 & 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.

* Corresponding author at: Eindhoven University of Technology, Department of Mathematics and Computer Science, P.O. Box 513, 5600MB Eindhoven, The Netherlands. Tel.: +31 634085760.
E-mail addresses: n.tax@tue.nl (N. Tax), sander.bockting@avanade.com (S. Bockting), d.hiemstra@utwente.nl (D. Hiemstra).

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

http://dx.doi.org/10.1016/j.ipm.2015.07.002
0306-4573/

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 & Chang, 2011), the *Yandex Internet Mathematics 2009* contest,[2] the LETOR datasets (Qin, Liu, Xu, & Li, 2010), 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, Oliveira, Almeida, and Gonçalves (2013) analyzed the ranking accuracy of a set of models on both LETOR 3.0 and 4.0. Busa-Fekete, Kégl, Éltető, and Szarvas (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. 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.

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. 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, Qin, Liu, Cheng, & Li, 2011; Lin, Yeh, & Liu, 2012).
2. The study focuses on a different learning to rank task (e.g. rank aggregation or transfer ranking) (Ah-Pine, 2008; Argentini, 2012; Chen et al., 2010; Dammak, Kammoun, & Ben Hamadou, 2011; De, 2013; De & Diaz, 2011, 2012; De, Diaz, & Raghavan, 2010, 2012; Derhami, Khodadadian, Ghasemzadeh, & Zareh Bidoki, 2013; Desarkar, Joshi, & Sarkar, 2011; Duh & Kirchhoff, 2011; Hoi & Jin, 2008; Lin, Yu, & Chen, 2011; Miao & Tang, 2013; Pan, Lai, Liu, Tang, & Yan, 2013; Qin, Geng, & Liu, 2010; Volkovs & Zemel, 2012, 2013; Wang, Tang et al., 2009).
3. The study used an altered version of a benchmark that contained additional features (Bidoki & Thom, 2009; Ding, Qin, & Zhang, 2010).
4. The study provides no exact data of the evaluation results (e.g. results are only in graphical form) (Adams & Zemel, 2011; Agarwal & Collins, 2010; Alejo, Fernández-Luna, Huete, & Pérez-Vázquez, 2010; Benbouzid, Busa-Fekete, & Kégl, 2012; Chang & Zheng, 2009; Chen, Weinberger, Chapelle, Kedem, & Xu, 2012; Ciaramita, Murdock, & Plachouras, 2008; Geng, Yang, Xu, & Hua, 2012; He, Ma, & Niub, 2010; Huang & Frey, 2008; Karimzadehgan, Li, Zhang, & Mao, 2011; Kuo, Cheng, & Wang, 2009; Li, Wang, Ni, Huang, & Xie, 2008; Ni, Huang, & Xie, 2008; Pan, Luo, Tang, & Huang, 2011; Petterson, Yu, Mcauley, & Caetano, 2009; Qin, Liu, Zhang, Wang, Xiong et al., 2008; Sculley, 2009; Shivaswamy &

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

Table 1
Included learning to rank evaluation benchmark collections.

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 2
Forward references of learning to rank benchmark papers.

Benchmark	Paper	# of forward references
LETOR 1.0 & 2.0	Liu et al. (2007)	307
LETOR 3.0	Qin, Liu, et al. (2010)	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 3
Google scholar search results for learning to rank benchmarks.

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

Joachims, 2011; Stewart & Diaz, 2012; Sun, Huang, & Feng, 2011; Swersky, Tarlow, Adams, Zemel, & Frey, 2012; Wang & Xu, 2010; Wang, Kuai, Huang, Li, & Ni, 2008; Wu et al., 2011; Xia, Liu, Wang, Zhang, & Li, 2008; Xu, Chapelle, & Weinberger, 2012; Xu, Kersting, & Joachims, 2010; Zhu, Chen, et al., 2009; Zhou, Ding, You, & Xiao, 2011).

5. The study reported evaluation results in a different metric than the metrics chosen for this meta-analysis (Kersting & Xu, 2009; Mohan, Chen, & Weinberger, 2011; Pahikkala, Tsivtsivadze, Airola, Järvinen, & Boberg, 2009; Thuy, Vien, Viet, & Chung, 2009; Yu & Joachims, 2009).
6. The study reported a higher performance on baseline methods than official benchmark runs (Acharyya, Koyejo, & Ghosh, 2012; Asadi, 2013; Banerjee, Dubey, Machchhar, & Chakrabarti, 2009; Bian, 2010; Bian, Li, Li, Zheng, & Zha, 2010; Carvalho, Elsas, Cohen, & Carbonell, 2008; Dubey, Machchhar, Bhattacharyya, & Chakrabarti, 2009; Peng, Macdonald, & Ounis, 2010; Song, Ng, Leung, & Fang, 2014; Tran & Pham, 2012).
7. The study did not report any baseline performance that allowed us to check validity of the results (Buffoni, Gallinari, Usunier, & Calauzènes, 2011; Chakrabarti, Khanna, Sawant, & Bhattacharyya, 2008; Wang, Bennett, & Collins-Thompson, 2012).

3. A methodology for comparing learning to rank methods cross-benchmark

Qin, Liu, et al. (2010) 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

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

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

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

where *IWN* is the Ideal Winning Number, defined as

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

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 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.

4.1. NDCG@3

Fig. 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

Fig. 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

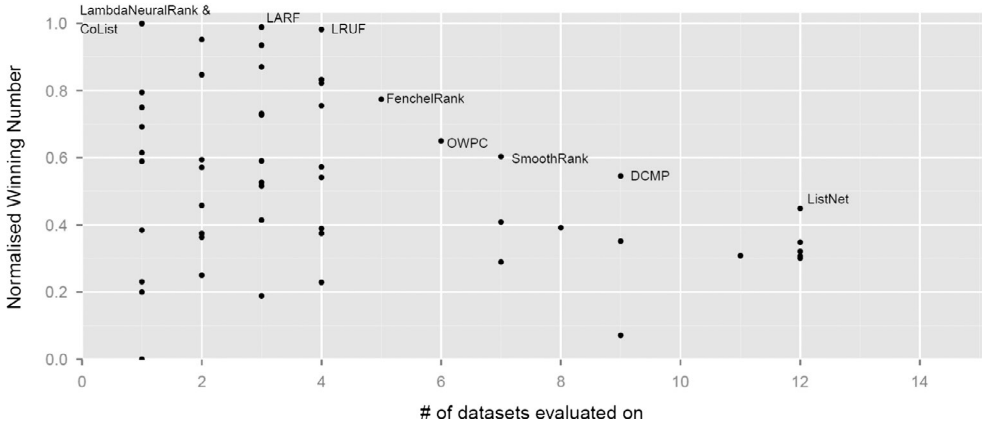

Fig. 1. NDCG@3 comparison of 87 learning to rank methods.

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 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

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

4.4. MAP

Fig. 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

Fig. 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. Fig. 5 labels the Pareto optimal algorithms, but also the Rank-2 Pareto optima, which are the labels the algorithms with exactly one 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

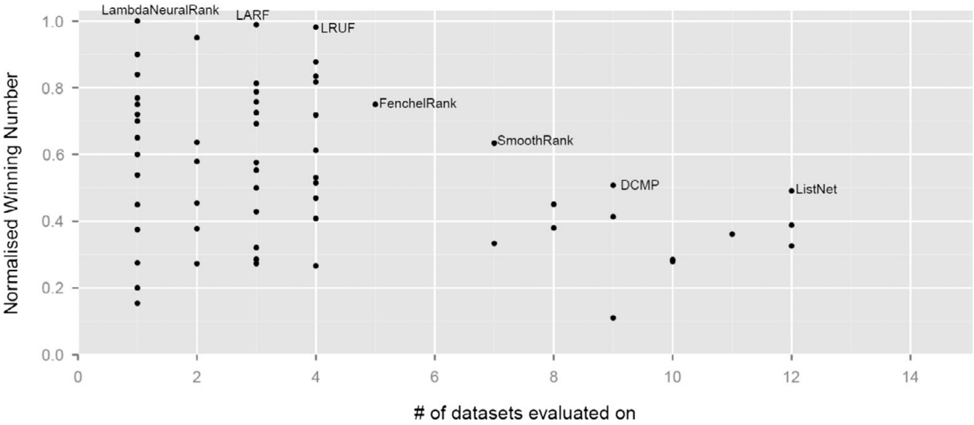

Fig. 2. NDCG@5 comparison of 87 learning to rank methods.

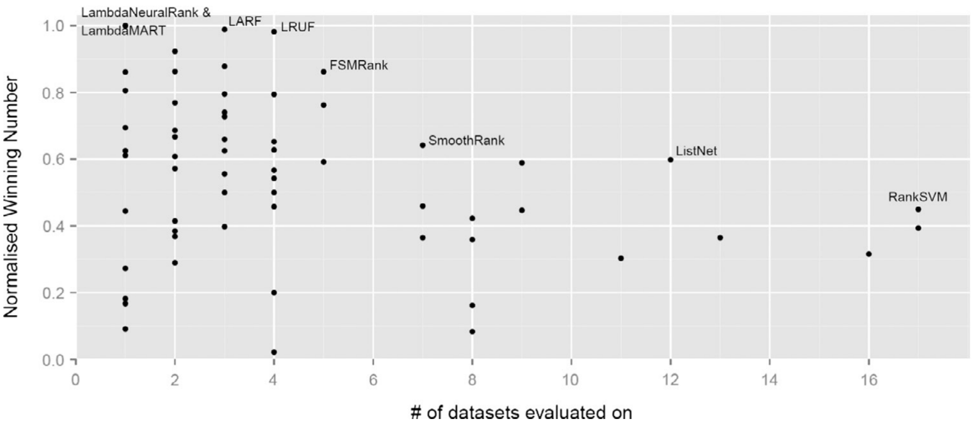

Fig. 3. NDCG@10 comparison of 87 learning to rank methods.

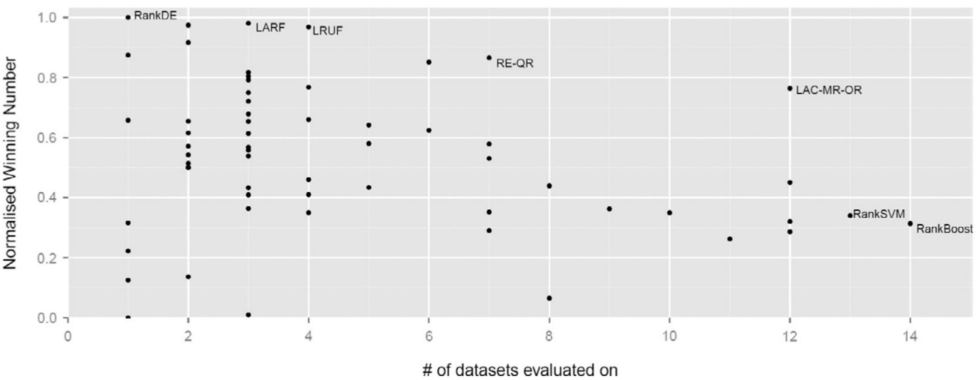

Fig. 4. MAP comparison of 87 learning to rank methods.

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 weight as the MAP comparison.

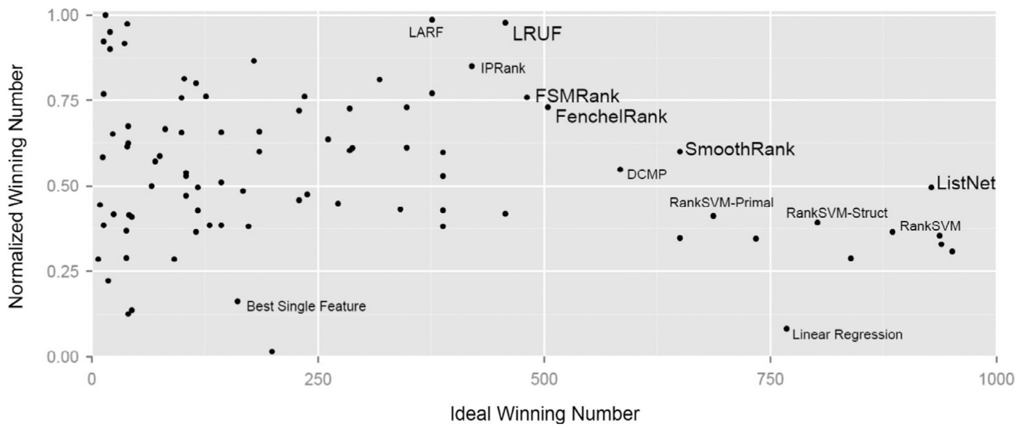

Fig. 5. Cross-benchmark comparison of 87 learning to rank methods.

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, Xia, Guo, & Wang (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 out one evaluation measure. Table 4 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 in Section 2 shows the 20 datasets used 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 Fig. 6 in a series of bagplots, which is a bivariate

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.

	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

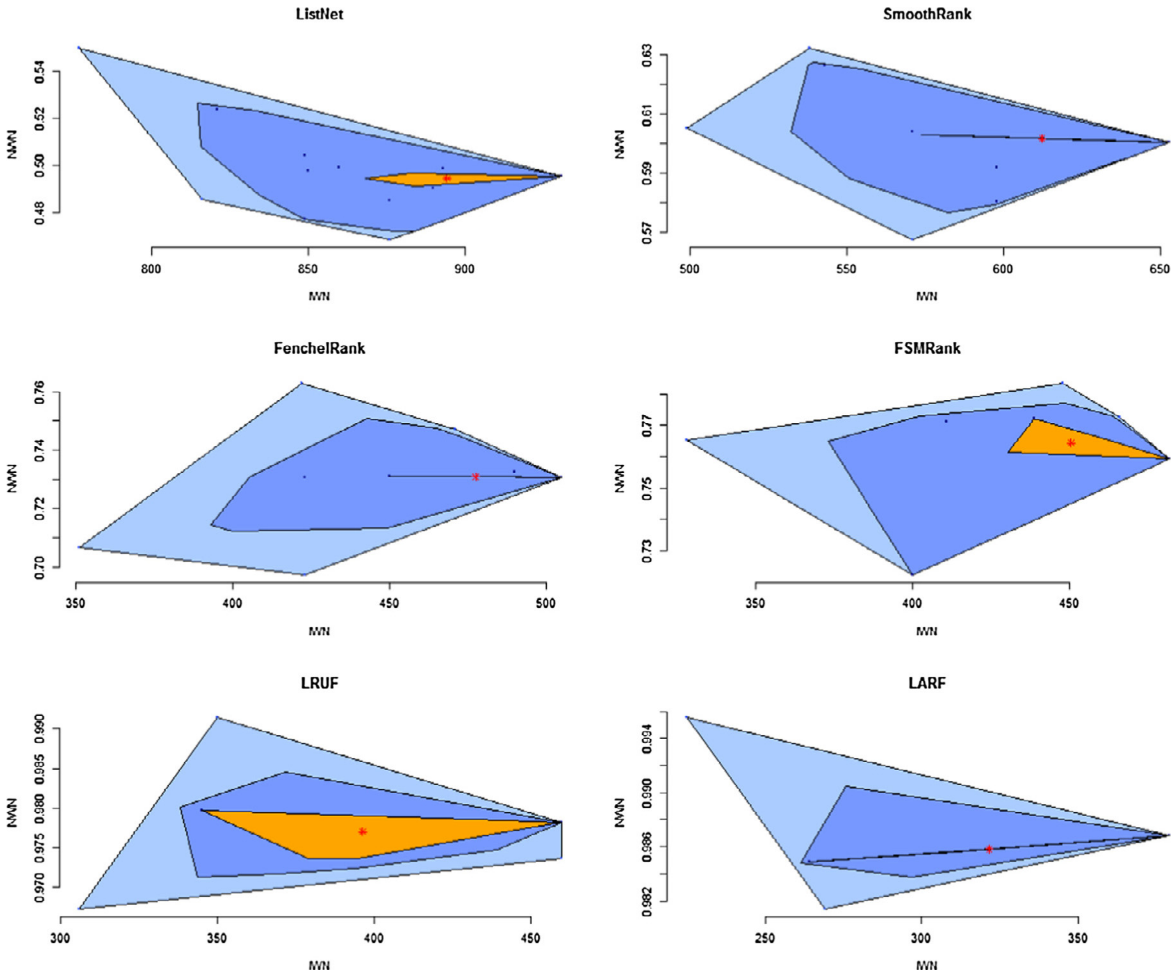

Fig. 6. Bagplots showing the variance in NWN and IWN of the Pareto optimal rankers when a dataset is left out of the comparison.

generalization of the boxplot proposed by Rousseeuw, Ruts, and Tukey (1999). Bagplot extends the univariate concept of rank as used in a boxplot to a halfspace location depth. The *depth median*, shown in orange, is the deepest location. Surrounding it is a *bag*, the dark blue area in Fig. 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.

The bagplots give insight into the degree to which the found *IWN* and *NWN* scores of the ranking methods are dependent on evaluation results on a small subset of the datasets that these ranking methods were evaluated on. A large bag and fence indicate that the *IWN* and *NWN* performance was not consistent over all the datasets on which the ranking method in question was evaluated, while a small bag and fence indicate consistent *IWN* and *NWN* performance over all the datasets on which that ranker was evaluated.

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 variance seems to be more or less consistent for all ranking methods except LRUF and LARF. As the NWN mean decreases from top-to-bottom and left-to-right, the variance-to-mean ratio increases. It is important to stress that the low NWN variance of LRUF and LARF 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 that were available for these ranking methods. As the number of evaluation results for LRUF and LARF is lower than for the other Pareto optimal rankers, the certainty of their ranking performance is considered to be lower.

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 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 Fig. 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 Fig. 5. Our work contributes to this by identifying promising learning to rank methods that researchers could focus on in performing additional evaluation runs.

Appendix A. Meta-analysis ranking methods & data sources

See Table A.5.

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

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, Liu, and Li (2010)	Qin, Liu, and Li (2010)
ApproxNDCG	Qin, Liu, and Li (2010)	Qin, Liu, and Li (2010)
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, Liu, Zhang, Wang et al. (2008)	Qin, Liu, Zhang, Wang et al. (2008)
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, Pan, Liu, et al. (2013)	Lai, Pan, Liu, et al. (2013, 2013), 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, Huang et al. (2012)
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, Lin, et al. (2011)	Lin, Lin, et al. (2011)
GroupMLE	Lin et al. (2010)	Lin, Lin, et al. (2011)
IntervalRank	Moon et al. (2010)	Moon et al. (2010), Freno et al. (2011)
IPRank	Wang, Ma et al. (2009)	Wang, Ma et al. (2009), 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, Huang et al. (2012), Volkovs et al. (2011)
ListMLE	Xia et al. (2008)	Lin et al. (2010, 2011), 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, Tang et al. (2010)	Peng, Tang et al. (2010)
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, Ma, and Wang (2010)	He, Ma, and Wang (2010)
RankDE	Bollegala et al. (2011)	Sato et al. (2013)
RankELM (pairwise)	Zong and Huang (2013)	Zong and Huang (2013)

(*continued on next page*)

Table A.5 (*continued*)

Method	Described in	Evaluated in
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, Ma, and Wang (2010), 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, Pan, Liu, et al. (2013)
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, Liu, and Li (2010)
SortNet	Rigutini et al. (2008)	Rigutini et al. (2008), Freno et al. (2011), Papini and Diligenti (2012)
SparseRank	Lai, Pan, Tang, et al. (2013)	Lai, Pan, Tang, et al. (2013)
SVM^{MAP}	Yue et al. (2007)	L3, Wang, Huang et al. (2012), Xu et al. (2008), Niu et al. (2012)
SwarmRank	Diaz-Aviles et al. (2009)	Sato et al. (2013)
TGRank	Lai, Pan, Liu, et al. (2013)	Lai, Pan, Liu, et al. (2013)
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)

Appendix B. Meta-analysis raw data

See Table B.6.

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

	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

(*continued on next page*)

Table B.6 (*continued*)

Method	NDCG@3		NDCG@5		NDG@10		MAP		CROSS		
	NWN	#ds	NWN	#ds	NWN	#ds	NWN	#ds	WN	IWN	NWN
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

References

Acharyya, S., Koyejo, O., & 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., & Zemel, R. S. (2011). Ranking via Sinkhorn Propagation. Available from: <arXiv:1106.1925>.

Agarwal, S., & Collins, M. (2010). Maximum Margin Ranking Algorithms for Information Retrieval. In *Proceedings of the 32nd European conference on information retrieval research (ECIR)* (pp. 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)* (Vol. 1, pp. 662–668).

Alcântara, O. D., Pereira, Á. R., Jr., Almeida, H. M., Gonçalves, M. A., Middleton, C., & 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., & 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)* (pp. 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., & Lin, J. (2013). Training efficient tree-based models for document ranking. In *Proceedings of the 25th European conference on advances in information retrieval* (Vol. 7814, pp. 146–157).

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

Benbouzid, D., Busa-Fekete, R., & Kégl, B. (2012). Fast classification using sparse decision DAGs. In *Proceedings of the 29th international conference on machine learning (ICML)* (pp. 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., & 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., & 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)* (Vol. 5478, pp. 538–545).

Bollegala, D., Noman, N., & 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* (pp. 1771–1778).

Buffoni, D., Gallinari, P., Usunier, N., & Calauzènes, C. (2011). Learning scoring functions with order-preserving losses and standardized supervision. In *Proceedings of the 28th international conference on machine learning (ICML)* (pp. 825–832).

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

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., et al. (2005). Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on machine learning* (pp. 89–96).

Burges, C. J. C., Ragno, R., & Le, Q. V. (2006). Learning to rank with nonsmooth cost functions. In *Advances in neural information processing systems (NIPS)* (Vol. 6, pp. 193–200).

Busa-Fekete, R., Kégl, B., Éltető, T., & 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., & 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., & Li, H. (2007). Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th international conference on machine learning* (pp. 129–136).

Carvalho, V. R., Elsas, J. L., Cohen, W. W., & Carbonell, J. G. (2008). Suppressing outliers in pairwise preference ranking. In *Proceedings of the 17th ACM conference on information and knowledge management (CIKM)* (pp. 1487–1488).

Chakrabarti, S., Khanna, R., Sawant, U., & 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)* (pp. 88–96).

Chang, X., & 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)* (Vol. III, pp. 143–146).

Chapelle, O., & Chang, Y. (2011). Yahoo! learning to rank challenge overview. *Journal of Machine Learning Research-Proceedings Track, 14*, 1–24.

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

Chen, X. W., Wang, H., & 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)* (pp. 505–512).

Chen, M., Weinberger, K. Q., Chapelle, O., Kedem, D., & Xu, Z. (2012). Classifier cascade for minimizing feature evaluation cost. In *International conference on artificial intelligence and statistics* (pp. 218–226).

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

Ciaramita, M., Murdock, V., & Plachouras, V. (2008). Online learning from click data for sponsored search. In *Proceedings of the 17th international conference on World Wide Web* (pp. 227–236).

Cossock, D., & Zhang, T. (2006). Subset ranking using regression. In *Proceedings of the 19th annual conference on learning theory (COLT)* (pp. 605–619).

Dammak, F., Kammoun, H., & 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)* (pp. 49–54).

De, A. (2013). On the role of compensatory operators in fuzzy result merging for metasearch. In *Proceedings of the fifth international conference on pattern recognition and machine intelligence (RReMI)* (Vol. 8251, pp. 551–556).

de Almeida, H. M., Gonçalves, M. A., Cristo, M., & 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* (pp. 399–406).

De, A., & 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* (pp. 17–20).

De, A., & Diaz, E. (2012). Fuzzy analytical network models for metasearch. In *Revised and selected papers of the international joint conference on computational intelligence (IJCCI)* (Vol. 399, pp. 197–210).

De, A., Diaz, E., & 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)* (Vol. 3, pp. 300–303).

De, A., Diaz, E. D., & Raghavan, V. V. (2012). Weighted fuzzy aggregation for metasearch: An application of choquet integral. In *Proceedings of the 14th international conference on information processing and management of uncertainty in knowledge-based systems (IPMU)* (pp. 501–510).

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

Desarkar, M., Joshi, R., & 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)* (Vol. 6744, pp. 268–273).

Diaz-Aviles, E., Nejdl, W., & Schmidt-Thieme, L. (2009). Swarming to rank for information retrieval. In *Proceedings of the 11th annual conference on genetic and evolutionary computation* (pp. 9–16).

Ding, W., Qin, T., & Zhang, X. D. (2010). Learning to rank with supplementary data. In *Proceedings of the sixth Asia information retrieval societies conference (AIRS)* (Vol. 6458, pp. 478–489).

Dubey, A., Machchhar, J., Bhattacharyya, C., & Chakrabarti, S. (2009). Conditional models for non-smooth ranking loss functions. In *Proceedings of the ninth IEEE international conference on data mining (ICDM)* (pp. 129–138).

Duh, K., & 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* (pp. 251–258).

Duh, K., Suzuki, J., & Nagata, M. (2011). Distributed learning-to-rank on streaming data using alternating direction method of multipliers. In *Proceedings of the NIPS big learning workshop*.

Duh, K., & Kirchhoff, K. (2011). Semi-supervised ranking for document retrieval. *Computational Speech and Language, 25*(2), 261–281.

Elsas, J. L., Carvalho, V. R., & 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)* (pp. 55–64).
Freno, N., Papini, T., & Diligenti, M. (2011). Learning to rank using Markov random fields. In *Proceedings of the 10th international conference on machine learning and applications and workshops (ICMLA)* (Vol. 2, pp. 257–262).
Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *The Journal of Machine Learning Research (JMLR), 4*, 933–969.
Ganjisaffar, Y., Caruana, R., & 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* (pp. 85–94).
Gao, W., & 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* (pp. 63–72).
Geng, X., Liu, T. Y., Qin, T., & Li, H. (2007). Feature selection for ranking. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 407–414).
Geng, X., Liu, T. Y., Qin, T., Arnold, A., Li, H., & 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* (pp. 115–122).
Geng, X., Qin, T., Liu, T. Y., Cheng, X. Q., & Li, H. (2011). Selecting optimal training data for learning to rank. *Information Processing & Management, 47*(5), 730–741.
Geng, B., Yang, Y., Xu, C., & Hua, X. S. (2012). Ranking model adaptation for domain-specific search. *IEEE Transactions On Knowledge and Data Engineering, 24*(4), 745–758.
Gomes, G., Oliveira, V. C., Almeida, J. M., & 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., & 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* (pp. 259–266).
He, Q., Ma, J., & Wang, S. (2010). 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)* (pp. 1449–1452).
He, Q., Ma, J., & Niub, X. (2010). Learning to rank for information retrieval using the clonal selection algorithm. *Journal of Information and Computational Science, 7*(1), :153–159.
Herbrich, R., Graepel, T., & Obermayer, K. (1999). Support vector learning for ordinal regression. In *Proceedings of the ninth international conference on artificial neural networks* (Vol. 1, pp. 97–102).
Hoi, S. C. H., & Jin, R. (2008). Semi-supervised ensemble ranking. In *Proceedings of the 23rd national conference on artificial intelligence* (Vol. 2, pp. 634–639).
Huang, J. C., & Frey, B. J. (2008). Structured ranking learning using cumulative distribution networks. In *Advances in neural information processing systems (NIPS)* (pp. 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* (pp. 133–142).
Kao, C. Y., & 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., & 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* (pp. 377–386).
Kersting, K. & 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* (pp. 676–691).
Kuo, J. W., Cheng, P. J., & Wang, H. M. (2009). Learning to rank from bayesian decision inference. In *Proceedings of the 18th ACM conference on information and knowledge management (CIKM)* (pp. 827–836).
Lai, H., Tang, Y., Luo, H. X., & Pan, Y. (2011). Greedy feature selection for ranking. In *Proceedings of the 15th international conference on computer supported cooperative work in design (CSCWD)* (pp. 42–46).
Lai, H., Pan, Y., Liu, C., Lin, L., & Wu, J. (2013). Sparse learning-to-rank via an efficient primal-dual algorithm. *IEEE Transactions on Computers, 62*(6), 1221–1233.
Lai, H., Pan, Y., Tang, Y., & Liu, N. (2013). Efficient gradient descent algorithm for sparse models with application in learning-to-rank. *Knowledge-Based Systems, 49*, 190–198.
Lai, H. J., Pan, Y., Tang, Y., & Yu, R. (2013). 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., Déjean, S., & Mothe, J. (2013). Nonconvex regularizations for feature selection in ranking with sparse SVM. *IEEE Transactions on Neural Networks and Learning Systems, 25*(6), 1118–1130.
Le, Q. & Smola, A. (2007). Direct optimization of ranking measures. Technical report, NICTA.
Li, D., Wang, Y., Ni, W., Huang, Y., & Xie, M. (2008). An ensemble approach to learning to rank. In *Proceedings of the fifth international conference on fuzzy systems and knowledge discovery (FSKD)* (Vol. 2, pp. 101–105).
Lin, Y., Lin, H., Yang, Z., & 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)* (pp. 330–338).
Lin, Y., Lin, H., Ye, Z., Jin, S., & Sun, X. (2010). Learning to rank with groups. In *Proceedings of the 19th ACM international conference on information and knowledge management (CIKM)* (pp. 1589–1592).
Lin, H. Y., Yu, C. H., & Chen, H. H. (2011). Query-dependent rank aggregation with local models. In *Proceedings of the seventh asia information retrieval societies conference (AIRS)*, (Vol. 7097, pp. 1–12).
Lin, Y., Lin, H., Wu, J., & Xu, K. (2011). Learning to rank with cross entropy. In *Proceedings of the 20th ACM international conference on information and knowledge management (CIKM)* (pp. 2057–2060).
Lin, J. Y., Yeh, J. Y., & 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)* (pp. 45–49).
Liu, T. Y., Xu, J., Qin, T., Xiong, W., & 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* (pp. 3–10).
McNee, S. M., Riedl, J., & 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* (pp. 1097–1101).
Metzler, D., & Croft, W. B. (2007). Linear feature-based models for information retrieval. *Information Retrieval, 10*(3), 257–274.
Miao, Z. & 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)* (pp. 376–383).
Mohan, A., Chen, Z., & 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., & 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* (pp. 151–160).
Ni, W., Huang, Y., & 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)* (pp. 262–269).
Niu, S., Guo, J., Lan, Y., & 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* (pp. 751–760).

Pahikkala, T., Airola, A., Naula, P., & 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* (pp. 11–18).

Pahikkala, T., Tsivtsivadze, E., Airola, A., Järvinen, J., & 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., & 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., & Huang, C.-Q. (2011). Learning to rank with document ranks and scores. *Knowledge-Based Systems, 24*(4), 478–483.

Papini, T. & 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., & Torgeson, C. (2006). A picture of search. In *Proceedings of the first international conference on scalable information systems (InfoScale)* (Vol. 152).

Pavlov, D. Y., Gorodilov, A., & 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)*. (pp. 1897–1900).

Peng, J., Macdonald, C., & Ounis, I. (2010). Learning to select a ranking function. In *Proceedings of the 32nd European conference on information retrieval research (ECIR)* (pp. 114–126). Berlin, Heidelberg.

Peng, Z., Tang, Y., Lin, L., & Pan, Y. (2010). Learning to rank with a weight matrix. In *Proceedings of the 14th international conference on computer supported cooperative work in design (CSCWD)* (pp. 18–21).

Petterson, J., Yu, J., Mcauley, J. J., & Caetano, T. S. (2009). Exponential family graph matching and ranking. In *Advances in neural information processing systems (NIPS)* (pp. 1455–1463).

Qin, T., Geng, X., & Liu, T. Y. (2010). A new probabilistic model for rank aggregation. In *Advances in neural information processing systems (NIPS)* (Vol. 10, pp. 1948–1956).

Qin, T., Liu, T. Y., Zhang, X. D., Wang, D. S., Xiong, W. Y., & Li, H. (2008). Learning to rank relational objects and its application to web search. In *Proceedings of the 17th international conference on World Wide Web* (pp. 407–416).

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

Qin, T., Zhang, X. D., Wang, D. S., Liu, T. Y., Lai, W., & Li, H. (2007). Ranking with multiple hyperplanes. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 279–286).

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

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

Ravikumar, P. D., Tewari, A., & Yang, E. (2011). On NDCG consistency of listwise ranking methods. In *International conference on artificial intelligence and statistics* (pp. 618–626).

Renjifo, C. & 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)* (pp. 1–6).

Rigutini, L., Papini, T., Maggini, M., & 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)* (pp. 76–79).

Robertson, S. E. & 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* (pp. 232–241).

Rousseeuw, P. J., Ruts, I., & 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., & 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)* (pp. 3371–3378).

Sculley, D. (2009). Large scale learning to rank. In *NIPS workshop on advances in ranking* (pp. 1–6).

Shivaswamy, P. K. & Joachims, T. (2011). Online learning with preference feedback. Available from: <arXiv:1111.0712>.

Silva, T. P. C., de Moura, E. S., Cavalcanti, J. a. M. B., da Silva, A. S., de Carvalho, M. G., & 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., & Ng, W. (2013). FP-rank: An effective ranking approach based on frequent pattern analysis. In *Database systems for advanced applications* (pp. 354–369).

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

Stewart, A. & Diaz, E. (2012). Epidemic intelligence: For the crowd, by the crowd. In *Proceedings of the 12th international conference on web engineering (ICWE)* (pp. 504–505). Berlin, Heidelberg.

Sun, Z., Qin, T., Tao, Q., & 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* (pp. 259–266).

Sun, H., Huang, J., & 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.

Swersky, K., Tarlow, D., Adams, R., Zemel, R., & Frey, B. (2012). Probabilistic n-choose-k models for classification and ranking. In *Advances in neural information processing systems (NIPS)* (pp. 3059–3067).

Tan, M., Xia, T., Guo, L., & 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* (pp. 856–864).

Taylor, M., Guiver, J., Robertson, S., & Minka, T. (2008). Softrank: Optimizing non-smooth rank metrics. In *Proceedings of the international conference on web search and data mining (WSDM)* (pp. 77–86).

Thuy, N. T. T., Vien, N. A., Viet, N. H., & Chung, T. C. (2009). Probabilistic ranking support vector machine. In *Proceedings of the sixth international symposium on neural networks* (Vol. 5552, pp. 345–353).

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

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

Tran, T. T. & Pham, D. S. (2012). ConeRANK: Ranking as learning generalized inequalities. Available from: <arXiv:1206.4110>.

Tsai, M. -F., Liu, T. Y., Qin, T., Chen, H. H., & 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* (pp. 383–390).

Usunier, N., Buffoni, D., & Gallinari, P. (2009). Ranking with ordered weighted pairwise classification. In *Proceedings of the 26th annual international conference on machine learning* (pp. 1057–1064).

Veloso, A. A., Almeida, H. M., Gonçalves, M. A., & Meira, W., Jr. (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* (pp. 267–274).

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

Volkovs, M. N. & Zemel, R. S. (2009). Boltzrank: Learning to maximize expected ranking gain. In *Proceedings of the 26th annual international conference on machine learning* (pp. 1089–1096).
Volkovs, M. N. & Zemel, R. S. (2012). A flexible generative model for preference aggregation. In *Proceedings of the 21st international conference on World Wide Web* (pp. 479–488).
Volkovs, M. N. & 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)* (pp. 89–98).
Volkovs, M. N., Larochelle, H., & Zemel, R. S. (2011). Loss-sensitive training of probabilistic conditional random fields. Available from: <arXiv:1107.1805>.
Wang, L., Bennett, P. N., & Collins-Thompson, K. (2012). Robust ranking models via risk-sensitive optimization. In *Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval* (pp. 761–770).
Wang, C. J., Huang, H. S., & Chen, H. H. (2012). 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)* (pp. 490–495).
Wang, S., Ma, J., & Liu, J. (2009). Learning to rank using evolutionary computation: Immune programming or genetic programming? In *Proceedings of the 18th ACM conference on information and knowledge management (CIKM)* (pp. 1879–1882).
Wang, B., Tang, J., Fan, W., Chen, S., Yang, Z., & Liu, Y. (2009). Heterogeneous Cross domain ranking in latent space. In *Proceedings of the 18th ACM conference on information and knowledge management (CIKM)* (pp. 987–996).
Wang, F. & 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)* (pp. 259–262).
Wang, Y., Kuai, Y. H., Huang, Y. L., Li, D., & Ni, W. J. (2008). Uncertainty-based active ranking for document retrieval. In *Proceedings of the international conference on machine learning and cybernetics* (Vol. 5, pp. 2629–2634).
Wang, S., Gao, B. J., Wang, K., & Lauw, H. W. (2011). CCRank: Parallel learning to rank with cooperative coevolution. In *Proceedings of the 25th AAAI conference on artificial intelligence*.
Wang, Y., Huang, Y., Pang, X., Lu, M., Xie, M., & Liu, J. (2013). Supervised rank aggregation based on query similarity for document retrieval. *Soft Computing, 17*(3), 421–429.
Wu, M., Chang, Y., Zheng, Z., & 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)* (pp. 1923–1926).
Wu, J., Yang, Z., Lin, Y., Lin, H., Ye, Z., & 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*. (pp. 1091–1092).
Xia, F., Liu, T. Y., Wang, J., Zhang, W., & Li, H. (2008). Listwise approach to learning to rank: Theory and algorithm. In *Proceedings of the 25th annual international conference on machine learning* (pp. 1192–1199).
Xu, Z., Chapelle, O., & Weinberger, K. Q. (2012). The greedy miser: Learning under test-time budgets. In *Proceedings of the 29th international conference on machine learning (ICML)* (pp. 1175–1182).
Xu, Z., Kersting, K., & 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)* (Vol. 6323, pp. 499–514).
Xu, J. & 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* (pp. 391–398).
Xu, J., Liu, T. Y., Lu, M., Li, H., & 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 information retrieval* (pp. 107–114).
Yu, C. -N. J. & Joachims, T. (2009). Learning structural SVMs with latent variables. In *Proceedings of the 26th annual international conference on machine learning (ICML)* (pp. 1169–1176). New York, NY, USA.
Yue, Y., Finley, T., Radlinski, F., & 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* (pp. 271–278).
Zhou, D., Ding, Y., You, Q., & Xiao, M. (2011). Learning to rank documents using similarity information between objects. In *Proceedings of the 18th international conference on neural information processing (ICONIP)* (Vol. Part II, pp. 374–381).
Zhou, K., Xue, G. -R., Zha, H., 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* (pp. 275–282).
Zhu, C., Chen, W., Zhu, Z. A., Wang, G., Wang, D., & 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)* (pp. 817–826).
Zong, W., & Huang, G. B. (2013). Learning to rank with extreme learning machine. *Neural Processing Letters*, 1–12.