Learning to Rank in theory and practice

From Gradient Boosting to Neural Networks and Unbiased Learning

Claudio Lucchese
Franco Maria Nardini
Ca' Foscari University, Venice, Italy,
ISTI-CNR, Pisa, Italy
claudio.lucchese@unive.it
f.nardini@isti.cnr.it

Rama Kumar Pasumarthi
Sebastian Bruch
Michael Bendersky
Xuanhui Wang
Google AI
{ramakumar,bruch,bemike,
xuanhui}@google.com

Harrie Oosterhuis
Rolf Jagerman
Maarten de Rijke
University of Amsterdam
{oosterhuis,rolf.jagerman,derijke}@
uva.nl

ABSTRACT

This tutorial aims to weave together diverse strands of modern learning-to-rank (LtR) research, and present them in a unified full-day tutorial. First, we will introduce the fundamentals of LtR, and an overview of its various sub-fields. Then, we will discuss some recent advances in gradient boosting methods such as LambdaMART by focusing on their efficiency/effectiveness trade-offs and optimizations. We will then present TF-Ranking, a new open source TensorFlow package for neural LtR models, and how it can be used for modeling sparse textual features. We will conclude the tutorial by covering unbiased LtR – a new research field aiming at learning from biased implicit user feedback.

The tutorial will consist of three two-hour sessions, each focusing on one of the topics described above. It will provide a mix of theoretical and hands-on sessions, and should benefit both academics interested in learning more about the current state-of-the-art in LtR, as well as practitioners who want to use LtR techniques in their applications.

KEYWORDS

Learning To Rank, Efficiency/Effectiveness trade-offs in Learning to Rank, Neural Learning to Rank, Unbiased Learning to Rank.

ACM Reference format:
Claudio Lucchese, Franco Maria Nardini, Rama Kumar Pasumarthi, Sebastian Bruch, Michael Bendersky, Xuanhui Wang, Harrie Oosterhuis, Rolf Jagerman and Maarten de Rijke. 2019. Learning to Rank in theory and practice . In *Proceedings of SIGIR'19, Paris, France., July July 21–24, 2019,* 4 pages.
DOI:

1 DESCRIPTION AND SCHEDULE

This tutorial will be presented as a full-day tutorial at SIGIR 2019, Paris, France. The tutorial is organized in three sessions lasting two hours each.

Session I: Efficiency/Effectiveness Trade-offs

- *Introduction to LtR and aims of the tutorial.* (30 min.)
  - Introduction on LtR [39], its historical evolution and main results [34, 48, 73] and the illustration of the goals of the tutorial.
  - The role of LtR in modern Web search engines. Review of the main approaches of LtR: focus on tree-based models [10, 10, 25, 46] and artificial neural networks [5, 19, 23, 29, 54, 63, 79]. Discussion of the quality vs. efficiency trade-off in the use of LtR models [13, 46, 65]. Brief description of multi-stage ranking architectures [16, 21, 47, 77].
- *Efficiency in Learning to Rank* (60 min.)
  Detailed analysis of state-of-the-art solutions for improving the efficiency of LtR models along different dimensions.
  - *Feature analysis*:
    * by removing features to speed up both training and model evaluation [28].
    * by introducing meta-features for list-aware query-document representation [43].
    * by reducing feature evaluation cost [74, 75].
  - *Pruning forests of regression trees*:
    * by using drop-out from artificial neural networks [67].
    * by removing trees at learning time [41].
    * by removing trees at post-learning [40, 42].
  - *Optimizing efficiency within the model learning process*:
    * by jointly optimizing efficiency and effectiveness in linear ranking models [68].
    * by learning compact and fast trees [7].
    * by employing oblivious trees for boosting efficiency and generalization power [60].
    * by introducing a novel cascade ranking model that simultaneously improve ranking effectiveness and retrieval efficiency. [69].
    * by learning *temporally constrained* ranking functions [70].
    * by learning efficient approximations of tree-based models through artificial neural networks [19].
  - *Approximate score computation and dynamic trade-off prediction*:

        * by using optimization strategies to allow short-circuiting score computations in additive LtR models [11].
        * by dynamically predicting the result set size to optimize the performance of the entire retrieval system [20].
        * by learning how to best balance feature importance and feature costs in multi-stage cascade ranking models [16].
        * by learning an end-to-end cascade of rankers using backpropagation [26].
    - *Efficient traversal of tree-based LtR models and efficient approximation*:
        * by employing standard approaches: Conditional Operators, If-Then-Else [22, 44].
        * by using vectorized traversal of trees [8].
        * by employing novel parallel traversal strategies: QuickScorer (CPU-based, SIMD, Multi-thread, and GPU versions) [22, 38, 44, 45], and RapidScorer [76] for compressed representations of trees when employing large numbers of leafs.
        * by defining cache-conscious optimization strategies for tree-based models [33, 64].
- *Hands-on Session* (30 min.)
  We show how to develop state-of-the-art strategies to gain a more efficient ranking model without losing effectiveness. Given a model learnt with a state-of-the-art algorithm such as LambdaMART, we will show how to reduce its runtime cost by a factor larger than 18×.
    - Publicly Available Datasets ([15, 22, 56]) and implementations (TensorFlow Ranking [55], XGBoost [17], LightGBM[2], CatBoost[3], QuickRank [12], jForests [27], RankLib[1], pGBRT[66])
    - In-depth analysis of several state-of-the-art strategies for scoring documents with forests of regression trees. We share the source code of several state-of-the-art solutions, including QuickScorer [44] (under NDA), and discuss CPU and cache profiling. We show how these algorithms allow to reduce the scoring time of a ranking model by a factor up to 275×.

Session II: Neural Learning to Rank using TensorFlow

- Session 1 (30 mins)
    - Introduction to Neural Ranking
      Neural learning-to-rank primer [49]
      Groupwise scoring methods [6]
    - Introduction to TensorFlow Ranking
      TensorFlow and Estimator framework overview [18]
      TensorFlow Ranking: components and APIs [55]
- Coffee Break
- Session 2 (90 mins)
    - Introduction to data formats and datasets
    - Colaboratory demo setup
    - Demo: TensorFlow Ranking for Search using the MSLR-Web30k dataset
      Dealing with numerical features
      Exploring various losses, scoring functions and metrics
    - Demo: TensorFlow Ranking for Passage Retrieval using the MSMARCO dataset
      Learning embeddings to model sparse textual features
      Incorporating pre-trained embeddings, e.g., BERT [24]
    - Discussion and questions.

Session III: Unbiased Learning to Rank

- ***Introduction to Learning from User Interactions*** (10 min)
  Limitations of the supervised approach
  The limitations of using annotated datasets [15, 39, 58, 71].
  Learning from user interactions
  User behavior indicates *true* user preferences [34, 57] but contain biases [78], i.e. *position bias* and *selection bias*.
- ***Counterfactual Learning to Rank*** (50 min)
  Counterfactual evaluation
  Inverse Propensity Scoring (IPS) and how it produces an unbiased estimate of online metrics.
  Propensity-weighted Learning to Rank (LTR)
  The recent propensity-weighted LTR methods [9, 37, 71].
  Estimating position bias
  Position bias estimation techniques [72], both online estimation [72] and offline estimation [4, 14].
  Practical considerations
  Some of the practical difficulties and their solutions, such as propensity overfitting [36, 62] and high variance [61].
- ***Online Learning to Rank*** (45 min)
  Online evaluation
  Interleaving and how it deals with position bias [32, 35].
  Dueling Bandit Gradient Descent
  Describe Dueling Bandit Gradient Descent (DBGD) the method that defined a decade of Online Learning to Rank (OLTR) algorithms.
  5 min – Extensions of DBGD and their limitations
  The extensions of DBGD do not provide long-term improvements in performance. [30, 31, 50, 53, 59, 80].
  Regret bounds of DBGD and their problems
  Empirical [51, 59] and theoretical problems [52] with DBGD.
  Pairwise Differentiable Gradient Descent
  Latest OLTR method [51] that does not rely on DBGD.
  Comparison of PDGD and DBGD
  An empirical and theoretical comparison between Pairwise Differentiable Gradient Descent (PDGD) and DBGD [51, 52].
- Conclusion (15 min)
  Summarize and contrast the two methodologies
  Reflect on the two approaches to unbiased LTR, contrast their properties and applicability.
  Future directions for unbiased learning to rank
  We finish by describing the promising directions that future LTR work could investigate.

2 SUPPORTING MATERIALS

You can find more materials related to this tutorial on our website http://ltr-tutorial-sigir19.isti.cnr.it/.

REFERENCES

[1] 2012. RankLib. (2012). https://sourceforge.net/p/lemur/wiki/RankLib/

[2] 2017. Microsoft LightGBM. (2017). https://github.com/Microsoft/LightGBM

[3] 2017. Yandex CatBoost. (2017). https://catboost.yandex/

[4] Aman Agarwal, Ivan Zaitsev, and Thorsten Joachims. 2018. Consistent position bias estimation without online interventions for learning-to-rank. *arXiv preprint arXiv:1806.03555* (2018).

[5] Qingyao Ai, Keping Bi, Jiafeng Guo, and W Bruce Croft. 2018. Learning a deep listwise context model for ranking refinement. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 135–144.

[6] Qingyao Ai, Xuanhui Wang, Nadav Golbandi, Michael Bendersky, and Marc Najork. 2018. Learning Groupwise Scoring Functions Using Deep Neural Networks. (2018). arXiv:cs.IR/1811.04415

[7] Nima Asadi and Jimmy Lin. 2013. Training Efficient Tree-based Models for Document Ranking. In *Proc. ECIR*. Springer-Verlag, 146–157. http://dx.doi.org/10.1007/978-3-642-36973-5_13

[8] Nima Asadi, Jimmy J. Lin, and Arjen P. de Vries. 2014. Runtime Optimizations for Tree-Based Machine Learning Models. *IEEE Trans. Knowl. Data Eng.* 26, 9 (2014), 2281–2292.

[9] Mike Bendersky, Xuanhui Wang, Marc Najork, and Don Metzler. 2018. Learning with sparse and biased feedback for personal search. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*. 5219–5223.

[10] Christopher J.C. Burges. 2010. *From RankNet to LambdaRank to LambdaMART: An Overview*. Technical Report Technical Report MSR-TR-2010-82. Microsoft Research.

[11] B. Barla Cambazoglu, Hugo Zaragoza, Olivier Chapelle, Jiang Chen, Ciya Liao, Zhaohui Zheng, and Jon Degenhardt. 2010. Early Exit Optimizations for Additive Machine Learned Ranking Systems. In *Proc. WSDM*. ACM, 411–420.

[12] Gabriele Capannini, Domenico Dato, Claudio Lucchese, Monica Mori, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, and Nicola Tonellotto. 2015. QuickRank: a C++ Suite of Learning to Rank Algorithms. In *Proc. IIR*.

[13] Gabriele Capannini, Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, and Nicola Tonellotto. 2016. Quality versus efficiency in document scoring with learning-to-rank models. *IP&M* 52, 6 (2016), 1161 – 1177.

[14] Ben Carterette and Praveen Chandar. 2018. Offline comparative evaluation with incremental, minimally-invasive online feedback. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 705–714.

[15] Olivier Chapelle and Yi Chang. 2011. Yahoo! learning to rank challenge overview.. In *Yahoo! Learning to Rank Challenge*. 1–24.

[16] Ruey-Cheng Chen, Luke Gallagher, Roi Blanco, and J Shane Culpepper. 2017. Efficient cost-aware cascade ranking in multi-stage retrieval. In *Proc. ACM SIGIR*. ACM, 445–454.

[17] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 785–794.

[18] Heng-Tze Cheng, Zakaria Haque, Lichan Hong, Mustafa Ispir, Clemens Mewald, Illia Polosukhin, Georgios Roumpos, D Sculley, Jamie Smith, David Soergel, and others. 2017. Tensorflow estimators: Managing simplicity vs. flexibility in high-level machine learning frameworks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1763–1771.

[19] Daniel Cohen, John Foley, Hamed Zamani, James Allan, and W Bruce Croft. 2018. Universal approximation functions for fast learning to rank: Replacing expensive regression forests with simple feed-forward networks. In *Proc. ACM SIGIR*. ACM, 1017–1020.

[20] J. Shane Culpepper, Charles L. A. Clarke, and Jimmy J. Lin. 2016. Dynamic Trade-Off Prediction in Multi-Stage Retrieval Systems. *CoRR* abs/1610.02502 (2016). http://arxiv.org/abs/1610.02502

[21] Van Dang, Michael Bendersky, and W Bruce Croft. 2013. Two-Stage learning to rank for information retrieval. In *Advances in Information Retrieval*. Springer, 423–434.

[22] Domenico Dato, Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Nicola Tonellotto, and Rossano Venturini. 2016. Fast Ranking with Additive Ensembles of Oblivious and Non-Oblivious Regression Trees. *ACM Trans. Inf. Syst.* 35, 2, Article 15 (Dec. 2016), 31 pages.

[23] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft. 2017. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 65–74.

[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[25] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of Statistics* 29, 5 (2001), 1189–1232.

[26] Luke Gallagher, Ruey-Cheng Chen, Roi Blanco, and J. Shane Culpepper. 2019. Joint Optimization of Cascade Ranking Models. In *WSDM*. ACM, 15–23.

[27] Yasser Ganjisaffar, Rich Caruana, and Cristina Lopes. 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 (SIGIR '11)*. ACM, New York, NY, USA, 85–94. DOI: http://dx.doi.org/10.1145/2009916.2009932

[28] Andrea Gigli, Claudio Lucchese, Franco Maria Nardini, and Raffaele Perego. 2016. Fast Feature Selection for Learning to Rank. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval (ICTIR '16)*. ACM, New York, NY, USA, 167–170. DOI: http://dx.doi.org/10.1145/2970398.2970433

[29] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 55–64.

[30] Katja Hofmann, Anne Schuth, Shimon Whiteson, and Maarten de Rijke. 2013. Reusing historical interaction data for faster online learning to rank for IR. In *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 183–192.

[31] Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. 2013. Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval. *Information Retrieval* 16, 1 (2013), 63–90.

[32] Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. 2013. Fidelity, soundness, and efficiency of interleaved comparison methods. *ACM Transactions on Information Systems (TOIS)* 31, 4 (2013), 17.

[33] Xin Jin, Tao Yang, and Xun Tang. 2016. A Comparison of Cache Blocking Methods for Fast Execution of Ensemble-based Score Computation. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '16)*. ACM, New York, NY, USA, 629–638. DOI: http://dx.doi.org/10.1145/2911451.2911520

[34] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 133–142.

[35] Thorsten Joachims. 2003. Evaluating retrieval performance using clickthrough data. In *Text Mining*, J. Franke, G. Nakhaeizadeh, and I. Renz (Eds.). Physica/Springer Verlag, 79–96.

[36] Thorsten Joachims, Adith Swaminathan, and Maarten de Rijke. 2018. Deep learning with logged bandit feedback. In *ICLR*.

[37] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *Proc. of WSDM*. 781–789.

[38] Francesco Lettich, Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Nicola Tonellotto, and Rossano Venturini. 2018. Parallel Traversal of Large Ensembles of Decision Trees. *IEEE Transactions on Parallel and Distributed Systems* (2018).

[39] Tie-Yan Liu and others. 2009. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval* 3, 3 (2009), 225–331.

[40] Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Fabrizio Silvestri, and Salvatore Trani. 2016. Post-Learning Optimization of Tree Ensembles for Efficient Ranking. In *Proc. ACM SIGIR*. ACM, 949–952.

[41] Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Fabrizio Silvestri, and Salvatore Trani. 2017. X-DART: Blending Dropouts and Pruning for Efficient Learning To Rank. In *Proc. ACM SIGIR*. ACM.

[42] Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Fabrizio Silvestri, and Salvatore Trani. 2018. X-CLEaVER: Learning Ranking Ensembles by Growing and Pruning Trees. *ACM Transactions on Intelligent Systems and Technology (TIST)* 9, 6 (2018), 62.

[43] Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, and Nicola Tonellotto. 2015. Speeding Up Document Ranking with Rank-based Features. In *Proc. ACM SIGIR*. ACM, 895–898.

[44] Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Nicola Tonellotto, and Rossano Venturini. 2015. QuickScorer: A Fast Algorithm to Rank Documents with Additive Ensembles of Regression Trees. In *Proc. ACM SIGIR*. ACM, 73–82.

[45] Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Nicola Tonellotto, and Rossano Venturini. 2016. Exploiting CPU SIMD Extensions to Speed-up Document Scoring with Tree Ensembles. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '16)*. ACM, New York, NY, USA, 833–836. DOI: http://dx.doi.org/10.1145/2911451.2914758

[46] Claudio Lucchese, Franco Maria Nardini, Raffaele Perego, Salvatore Orlando, and Salvatore Trani. 2018. Selective Gradient Boosting for Effective Learning to Rank. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 155–164.

[47] Craig Macdonald, Rodrygo LT Santos, and Iadh Ounis. 2013. The whens and hows of learning to rank for web search. *Information Retrieval* 16, 5 (2013), 584–628.

[48] Donald Metzler and W Bruce Croft. 2007. Linear feature-based models for information retrieval. *Information Retrieval* 10, 3 (2007), 257–274.

[49] Bhaskar Mitra and Nick Craswell. 2017. Neural Models for Information Retrieval. *arXiv preprint arXiv:1705.01509* (2017).

[50] Harrie Oosterhuis and Maarten de Rijke. 2017. Balancing speed and quality in online learning to rank for information retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 277–286.

[51] Harrie Oosterhuis and Maarten de Rijke. 2018. Differentiable unbiased online learning to rank. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1293–1302.

[52] Harrie Oosterhuis and Maarten de Rijke. 2019. Optimizing ranking models in an online setting. In *European Conference on Information Retrieval*. Springer.

[53] Harrie Oosterhuis, Anne Schuth, and Maarten de Rijke. 2016. Probabilistic multileave gradient descent. In *European Conference on Information Retrieval*. Springer, 661–668.

[54] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. 2017. Deeprank: A new deep architecture for relevance ranking in information retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 257–266.

[55] Rama Kumar Pasumarthi, Xuanhui Wang, Cheng Li, Sebastian Bruch, Michael Bendersky, Marc Najork, Jan Pfeifer, Nadav Golbandi, Rohan Anil, and Stephan Wolf. 2018. TF-Ranking: Scalable TensorFlow Library for Learning-to-Rank. (2018). arXiv:cs.IR/1812.00073

[56] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 Datasets. *CoRR* abs/1306.2597 (2013). http://arxiv.org/abs/1306.2597

[57] Filip Radlinski, Madhu Kurup, and Thorsten Joachims. 2008. How does click-through data reflect retrieval quality?. In *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 43–52.

[58] Mark Sanderson. 2010. Test collection based evaluation of information retrieval systems. *Foundations and Trends in Information Retrieval* 4, 4 (2010), 247–375.

[59] Anne Schuth, Harrie Oosterhuis, Shimon Whiteson, and Maarten de Rijke. 2016. Multileave gradient descent for fast online learning to rank. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 457–466.

[60] Ilya Segalovich. 2010. Machine learning in search quality at Yandex. *Invited Talk, SIGIR* 114 (2010).

[61] Adith Swaminathan and Thorsten Joachims. 2015. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning*. 814–823.

[62] Adith Swaminathan and Thorsten Joachims. 2015. The self-normalized estimator for counterfactual learning. In *advances in neural information processing systems*. 3231–3239.

[63] Jiaxi Tang and Ke Wang. 2018. Ranking Distillation: Learning Compact Ranking Models With High Performance for Recommender System. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2289–2298.

[64] Xun Tang, Xin Jin, and Tao Yang. 2014. Cache-conscious Runtime Optimization for Ranking Ensembles. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '14)*. ACM, New York, NY, USA, 1123–1126. DOI: http://dx.doi.org/10.1145/2600428.2609525

[65] Niek Tax, Sander Bockting, and Djoerd Hiemstra. 2015. A cross-benchmark comparison of 87 learning to rank methods. *Information Processing & Management* 51, 6 (2015), 757 – 772.

[66] Stephen Tyree, Kilian Q Weinberger, Kunal Agrawal, and Jennifer Paykin. 2011. Parallel boosted regression trees for web search ranking. In *Proceedings of the 20th international conference on World wide web*. ACM, 387–396.

[67] Rashmi Korlakai Vinayak and Ran Gilad-Bachrach. 2015. DART: Dropouts meet Multiple Additive Regression Trees. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research)*, Guy Lebanon and S. V. N. Vishwanathan (Eds.), Vol. 38. PMLR, San Diego, California, USA, 489–497. http://proceedings.mlr.press/v38/korlakaivinayak15.html

[68] Lidan Wang, Jimmy J. Lin, and Donald Metzler. 2010. Learning to efficiently rank. In *SIGIR*, Fabio Crestani, Stéphane Marchand-Maillet, Hsin-Hsi Chen, Efthimis N. Efthimiadis, and Jacques Savoy (Eds.). ACM, 138–145.

[69] Lidan Wang, Jimmy J. Lin, and Donald Metzler. 2011. A cascade ranking model for efficient ranked retrieval. In *SIGIR*, Wei-Ying Ma, Jian-Yun Nie, Ricardo A. Baeza-Yates, Tat-Seng Chua, and W. Bruce Croft (Eds.). ACM, 105–114.

[70] Lidan Wang, Donald Metzler, and Jimmy J. Lin. 2010. Ranking under temporal constraints. In *CIKM*, Jimmy Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An (Eds.). ACM, 79–88.

[71] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to Rank with Selection Bias in Personal Search. In *Proc. of SIGIR*. 115–124.

[72] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position Bias Estimation for Unbiased Learning to Rank in Personal Search. In *Proc. of WSDM*. 610 –618.

[73] Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. *Information Retrieval* 13, 3 (2010), 254–270.

[74] Zhixiang Xu, Olivier Chapelle, and Kilian Q Weinberger. 2012. The Greedy Miser: Learning under Test-time Budgets. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*. 1175–1182.

[75] Zhixiang Eddie Xu, Matt J Kusner, Kilian Q Weinberger, Minmin Chen, and Olivier Chapelle. 2014. Classifier cascades and trees for minimizing feature evaluation cost. *Journal of Machine Learning Research* 15, 1 (2014), 2113–2144.

[76] Ting Ye, Hucheng Zhou, Will Y. Zou, Bin Gao, and Ruofei Zhang. 2018. RapidScorer: Fast Tree Ensemble Evaluation by Maximizing Compactness in Data Level Parallelization. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. ACM, New York, NY, USA, 941–950. DOI: http://dx.doi.org/10.1145/3219819.3219857

[77] Dawei Yin, Yuening Hu, Jiliang Tang, Tim Daly, Mianwei Zhou, Hua Ouyang, Jianhui Chen, Changsung Kang, Hongbo Deng, Chikashi Nobata, Jean-Marc Langlois, and Yi Chang. 2016. Ranking Relevance in Yahoo Search. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM, New York, NY, USA, 323–332. DOI: http://dx.doi.org/10.1145/2939672.2939677

[78] Yisong Yue, Rajan Patel, and Hein Roehrig. 2010. Beyond Position Bias: Examining Result Attractiveness As a Source of Presentation Bias in Clickthrough Data. In *Proc. of WWW*. 1011–1018.

[79] Hamed Zamani, Bhaskar Mitra, Xia Song, Nick Craswell, and Saurabh Tiwary. 2018. Neural ranking models with multiple document fields. In *Proceedings of the eleventh ACM international conference on web search and data mining*. ACM, 700–708.

[80] Tong Zhao and Irwin King. 2016. Constructing Reliable Gradient Exploration for Online Learning to Rank. In *CIKM*. ACM, 1643–1652.