Notes on algorithms in LeToR

Harsh Thakkar

January 4, 2015

Abstract

In this note I try to list a possible number of algorithms and approaches put into practice in learning to rank based schemes. The survey is carried out form the work of author phophalia. This note is a part of an exercise carried out amongst the new introductions of habits for the year 2015. I hope this continues, and in a long term turns out to be an integral atomic part of the overall process of my research paradigm, *i.e.* note making!

1 Approaches in letor

1. **Pointwise approaches**
   (a) Regression The conventional and simple idea of ordinal regression is to map the ordinal scales into numeric values, and then solve the problem as a standard regression problem. In this approach, a document - query pair is considered in the training phase.
   (b) McRank
   (c) RankProp
2. **Pairwise approaches**
   (a) AdaBoost
   (b) RankBoost - based on adaboost algorithm
   (c) **Neural network based approaches**
      i. RankNet
      ii. LambdaRank - based on RankNet
   (d) **SVM based approaches**
      i. RankSVM [14]
3. **Listwise approaches**
   (a) ListNet (neural network based, see: 2c) - similar to RankNet

(b) LambdaMART - [10] combines MART and LambdaRank. MART is a boosted tree model in which output of the model is linear combination of the outputs of a set of regression trees. Since MART models derivatives and LambdaRank works by specifying the derivatives at any point during training.

(c) BoltzRank - It uses Boltzman distribution, [11] is to define a probability distribution over document permutations, and consider the expectation of the target performance measure under this distribution.

(d) BayesRank - It directly optimizes the Bayes Risk related to the ranking accuracy in terms of the IR evaluation measures. It uses Plackett-Luce Model as probability model of permutations. A multilayer perceptron ***neural network*** is designed for learning BayesRank with NDCG related permutation loss. 2c

(e) FRank - FRank algorithm is proposed in [13] which is based on the concept of fidelity from physics.

(f) Rank Cosine - To find the similarity between estimated output and available ground truth result, Rank Cosine approach is proposed in [14].

2 Tools and resources

2.1 Available libraries

2.2 Proposed library to be used

RankLib is a library of learning to rank algorithms. Currently eight popular algorithms have been implemented:

1. MART (Multiple Additive Regression Trees, a.k.a. Gradient boosted regression tree)[15]
2. RankNet[16]
3. RankBoost[17]
4. AdaRank[18]
5. Coordinate Ascent[19]
6. LambdaMART[20]
7. ListNet[21]
8. Random Forests[22]

It also implements many retrieval metrics as well as provides many ways to carry out evaluation.

References

[1] Shashua A. and Levin A. Taxonomy of large margin principle algorithms for ordinal regression problems, In Proceedings of NIPS 2002.

[2] P. Li, C. Burges and Qiang Wu, McRank: Learning to Rank using multiple classification and gradient boosting, In Proc of NIPS 2007.

[3] Y. Freund, R. Iyer, R. E. Schapire and Y. Singer, An Efficient Boosting algorithm for combining preferences, Journal of Machine Learning Research(JMLR), vol. 4, pp. 933-969, 2003

[4] Y. Freund and R. E. Schapire, A short introduction to boosting, Journal of Japanese society for Artificial Intelligence, vol. 14, no. 5, pp. 771-780, Sept., 1999.

[5] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton and G. Hullender, Learning to Rank using Gradient Descent, In Proc. Of 22nd ICML, 2002.

[6] R. Caruana, S. Baluja, and T. Mitchell. Using the feature to sort out the present: Rankprop and Multitask learning for medical risk evaluation, Advances on Neural Information Processing System, 1996.

[7] O. Chapelle and Y. Chang, Yahoo! Learning to Rank Challenge Overview, Journal of Machine Learning Research, vol. 14, pp. 1-24, 2011.

[8] T. Y. Liu, J. Xu, T. Qin, W. Xiong and H. Li, LETOR: Benchmark dataset for research on Learning to Rank for Information Retrieval, SIGIR 2007 Workshop on Learning To Rank for Information Retrieval (LR4IR 2007), 2007.

[9] C. J. Burges, From RankNet to LambdaRank to LambdaMART: An Overview, Microsoft Research Technical ReportMSRTR201082.

[10] M. N. Volkovs and R. S. Zemel, BoltzRank: Learning to maximize expected ranking gain, In Proc. of 26th ICML, 2009.

[11] J. W. Kuo, Pu-Jen Cheng and H. M. Wang, Learning to rank from Bayesian Decision Inference, In Proc. of CIKM, 2009.

[12] M. F. Tasi, T. Y. Liu, T. Qin, H. H. Chen and W. Y. Ma, FRank: a ranking method with fidelity loss, In Proc. of 30th SIGIR, 2007.

[13] T. Qin, X. D. Zhang, M. F. Tsai, D. S. Wang, T. Y. Lin and H. Li, Querylevel loss functions for Information Retrieval, Information Processing and Management, vol. 44, pp. 838-855, 2008.

[14] T. Joachims, Optimizing search engines using click through data, In Proc. of eighth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD), 2002.

[15] J.H. Friedman. Greedy function approximation: A gradient boosting machine. Technical Report, IMS Reitz Lecture, Stanford, 1999; see also Annals of Statistics, 2001.

[16] C.J.C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton and G. Hullender. Learning to rank using gradient descent. In Proc. of ICML, pages 89-96, 2005.

[17] Y. Freund, R. Iyer, R. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. The Journal of Machine Learning Research, 4: 933-969, 2003.

[18] J. Xu and H. Li. AdaRank: a boosting algorithm for information retrieval. In Proc. of SIGIR, pages 391-398, 2007.

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

[20] Q. Wu, C.J.C. Burges, K. Svore and J. Gao. Adapting Boosting for Information Retrieval Measures. Journal of Information Retrieval, 2007.

[21] Z. Cao, T. Qin, T.Y. Liu, M. Tsai and H. Li. Learning to Rank: From Pairwise Approach to Listwise Approach. ICML 2007.

[22] L. Breiman. Random Forests. Machine Learning 45 (1): 532, 2001.