Learning to Rank in Theory and Practice

From Gradient Boosting to Neural Networks and Unbiased Learning

Tutorial @ ACM SIGIR 2019

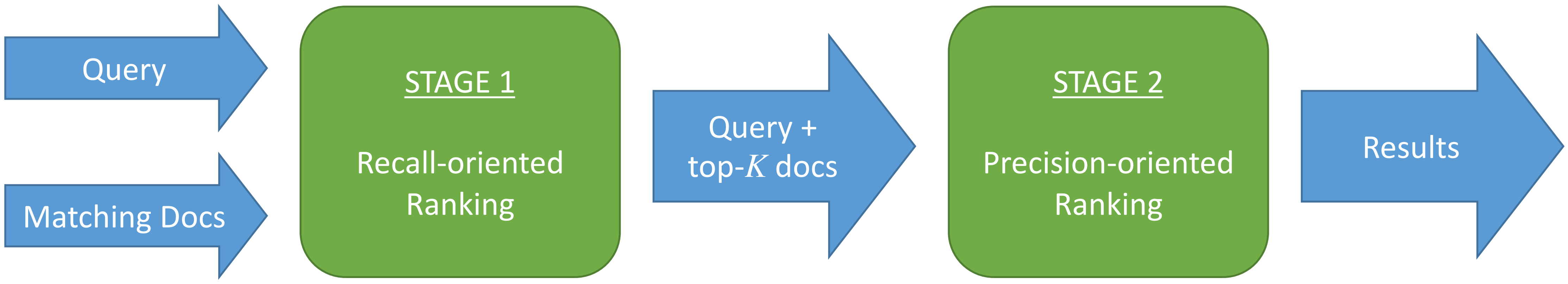
http://ltr-tutorial-sigir19.isti.cnr.it/

Session I: Efficiency/Effectiveness Trade-offs

Claudio Lucchese
Ca’ Foscari University of Venice
Venice, Italy

Franco Maria Nardini
HPC Lab, ISTI-CNR
Pisa, Italy

Two-stage (or more) Ranking Architecture

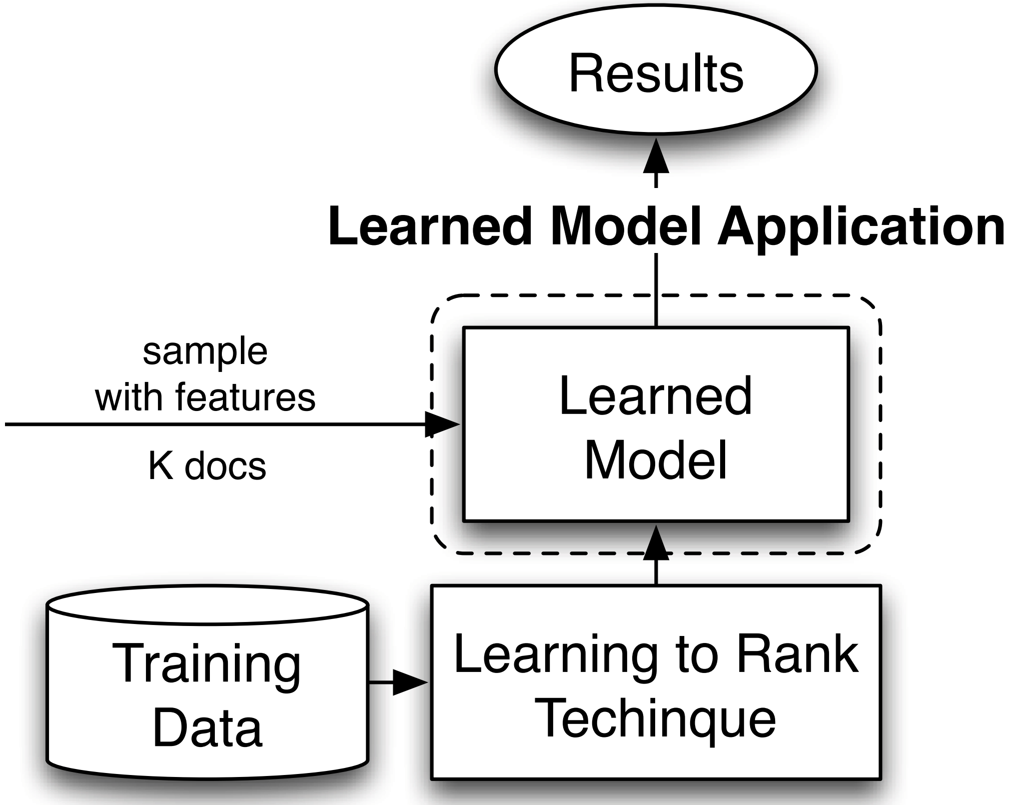


Efficiency/Effectiveness Trade-offs

- Efficiency in Learning to Rank (LtR) has been addressed in different ways

- Main research lines
  - Feature selection
  - Optimizing efficiency within the learning process
  - Approximate score computation and efficient cascades
  - Efficient traversal of tree-based models

- Different impact on the architecture



Feature Selection

Feature Selection

- A reduced set of highly discriminative and non redundant features results in a **reduced feature extraction cost** and in a **faster learning/ranking**
- Classification of feature selection methods [GE03]
  - **Filter** methods: feature selection is defined as a preprocessing step and can be independent from learning
  - **Wrapper** methods: utilizes a learning system as a black box to score subsets of features
  - **Embedded** methods: perform feature selection within the training process
- Wrapper or embedded methods: higher computational cost / algorithm dependent
  - not suitable for a LtR scenario involving hundreds of continuous or categorical features
- Focus on **filter** methods
  - Allow for a fast pre-processing of the dataset
  - Totally independent from the learning process

[GE03] Isabelle Guyon and Andre Elisseff. *An introduction to variable and feature selection*. The Journal of Machine Learning Research, 3:1157–1182, 2003.

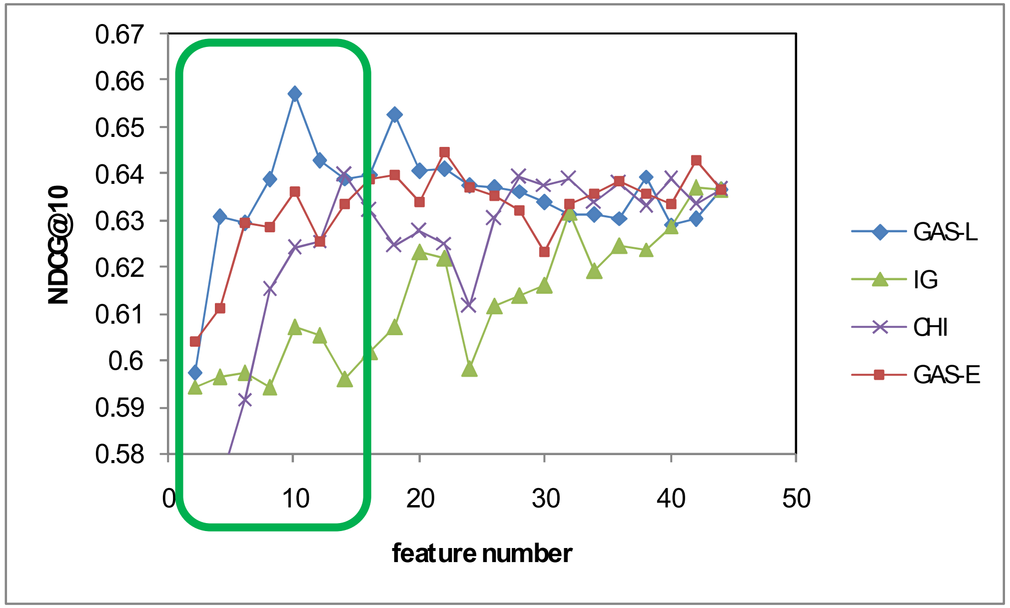
GAS [GLQL07]

- Geng *et al.* are the first proposing feature selection methods for ranking
- Authors propose to exploit ranking information for selecting features
  - They use IR metrics to measure the **importance** of each feature
    - MAP, NDCG: rank instances by feature, evaluate and take the result as importance score
  - They use **similarities** between features to avoid selecting redundant ones
    - By using ranking results of each feature: Kendall's tau, averaged over all queries
- Feature selection as a **multi-objective optimization problem**: maximum importance and minimum similarity
- Greedy Search Algorithm (GAS) performs feature selection **iteratively**
  - Update phase needs the tuning of an hyper-parameter *c* weighting the impact of the update

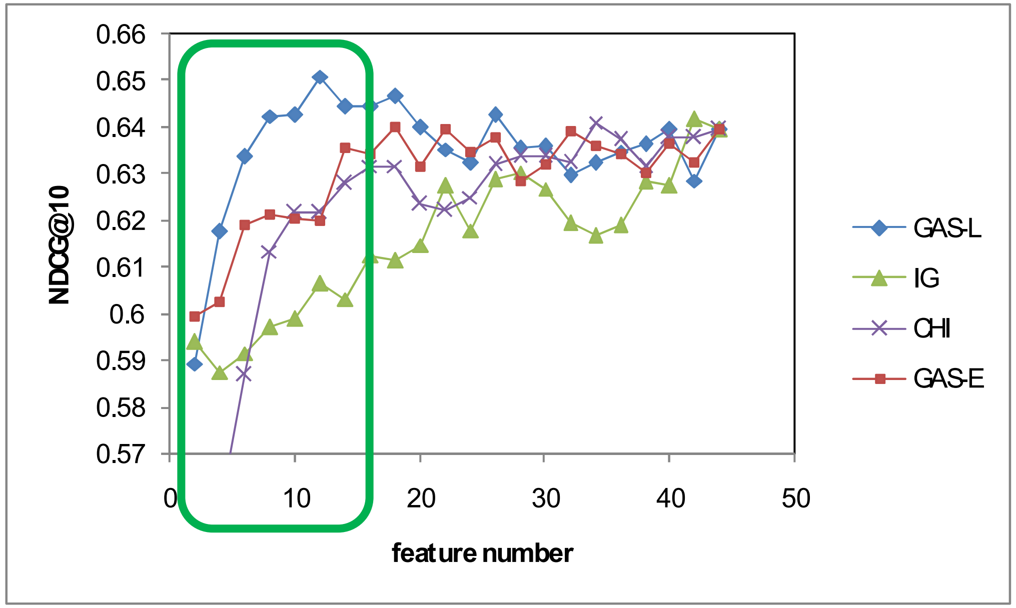
[GLQL07] X. Geng, T. Liu, T. Qin, and H. Li. *Feature selection for ranking*. In Proc. ACM SIGIR, 2007.

GAS [GLQL07]

- Experiments
  - .gov and TREC 2004 Web Track
  - BM25 as first stage
  - 44 features per doc
- Evaluation Measures
  - MAP
  - NDCG
- Applied to second stage ranker
  - Ranking SVM
  - RankNet



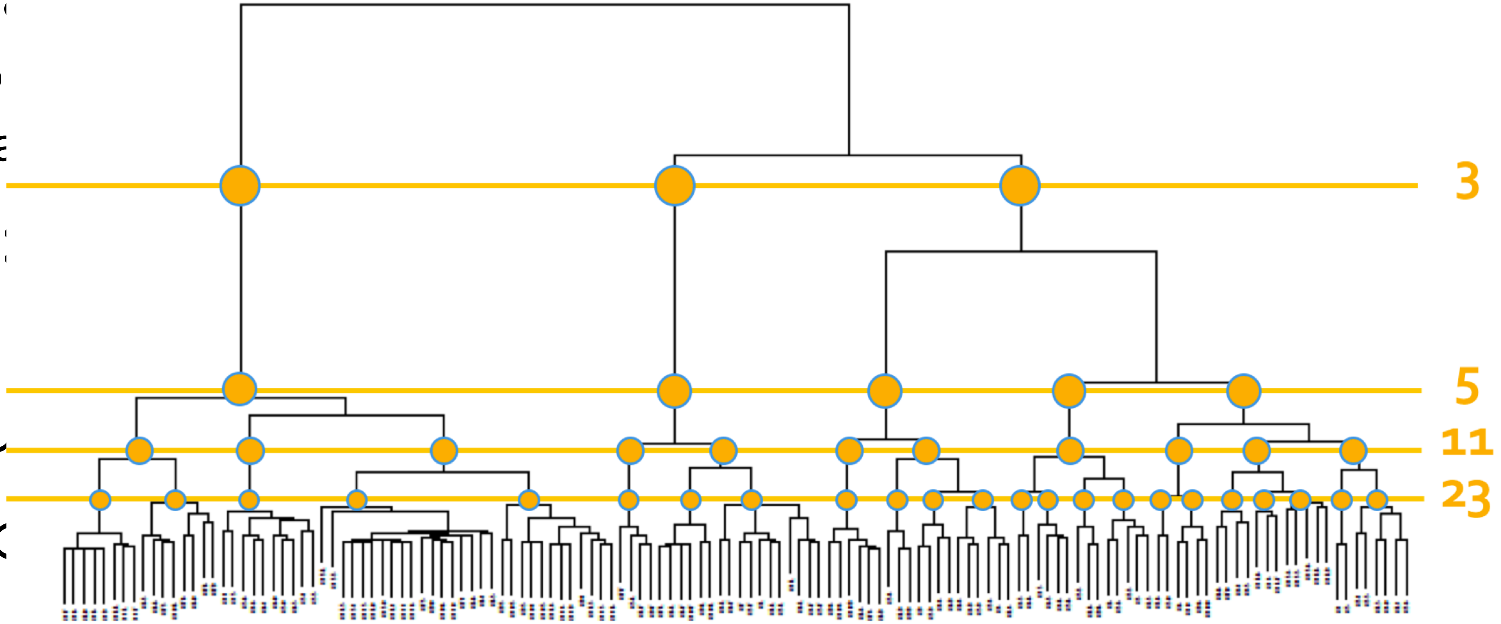
(b) NDCG@10 of Ranking SVM



(b) NDCG@10 of RankNet

Fast Feature Selection for LtR [GLNP16]

- Gigli *et al*. propose three novel filter methods providing flexible and model-free feature selection
  - Two parameter-free variations of GAS: NGAS and XGAS
  - HCAS exploits hierarchica
    - Only one feature per gro
    - Two variants: Single-linka
- Importance of a feature: single feature
- Similarity between featu
- No need to tune hyper-p



[GLNP16] A. Gigli, C. Lucchese, F. M. Nardini, and R. Perego. *Fast feature selection for learning to rank*. In Proc. ACM ICTIR, 2016.

Fast Feature Selection for Learning to Rank

- Experiments
  - MSLR-Web10K (Fold1) and Yahoo LETOR
  - By varying the subset sampled
  - Results confirms Geng *et al.* [GLQL07]
- Evaluation Measures
  - NDCG@10
- For small subsets (5%, 10%, 20%):
  - Best performance by HCAS with “Single Linkage”.
  - Statistically significant w.r.t. GAS
  - Performance against the full model

MSN-1

Subset %	**NGAS**	**XGAS p = 0.05**	**HCAS “single”**	**HCAS “ward”**	**GAS c = 0.01**
5%	0.4011▼	0.4376▲	**0.4423**▲	0.4289	0.4294
10%	0.4459	0.4528	**0.4643**▲	0.4434▼	0.4515
20%	0.4710	0.4577▼	**0.4870**▲	0.4820	0.4758
30%	0.4739▼	0.4825	0.4854	**0.4879**	0.4848
40%	0.4813	0.4834	0.4848	0.4853	**0.4863**
Full	0.4863	0.4863	0.4863	0.4863	0.4863

Further Reading

- Pan *et al.* use boosted regression trees to investigate greedy and randomized wrapper methods [PCA+09].
- Dang and Croft propose a wrapper method that uses best first search and coordinate ascent to greedily partition a set of features into subsets to be selected [DC10].
- Hua *et al.* propose a feature selection method based on clustering: *k*-means is first used to aggregate similar features, then the most relevant feature in each cluster is chosen to form the final set [HZL+10].
- Laporte *et al.* [LFC+12] and Lai *et al.* [LPTY13] use embedded methods for selecting features and building the ranking model at the same step, by solving a convex optimization problem.
- Naini and Altingovde use greedy diversification methods to solve the feature selection problem [NA14].
- Xu *et al.* solve the feature selection task by modifying the gradient boosting algorithm used to learn forests of regression trees [XHW+14].

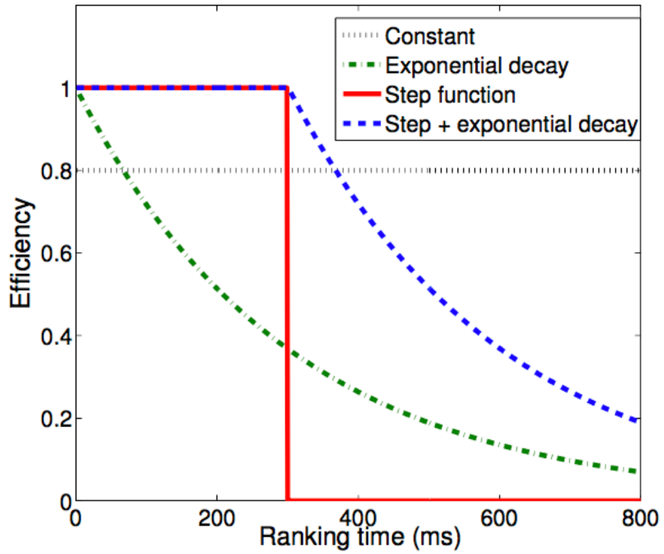
Optimizing Efficiency within the Learning Process

Learning to Efficiently Rank [WLM10]

- Wang *et al.* propose a new cost function for learning models that directly optimize the tradeoff metric: Efficiency-Effectiveness Tradeoff Metric (EET)

$$\mathrm{EET}(Q) = \frac{(1+\beta^2)\cdot \gamma(Q)\,\sigma(Q)}{\beta^2 \cdot \sigma(Q) + \gamma(Q)} \longrightarrow \mathrm{MEET}(R) = \frac{1}{N}\sum \mathrm{EET}(Q)$$

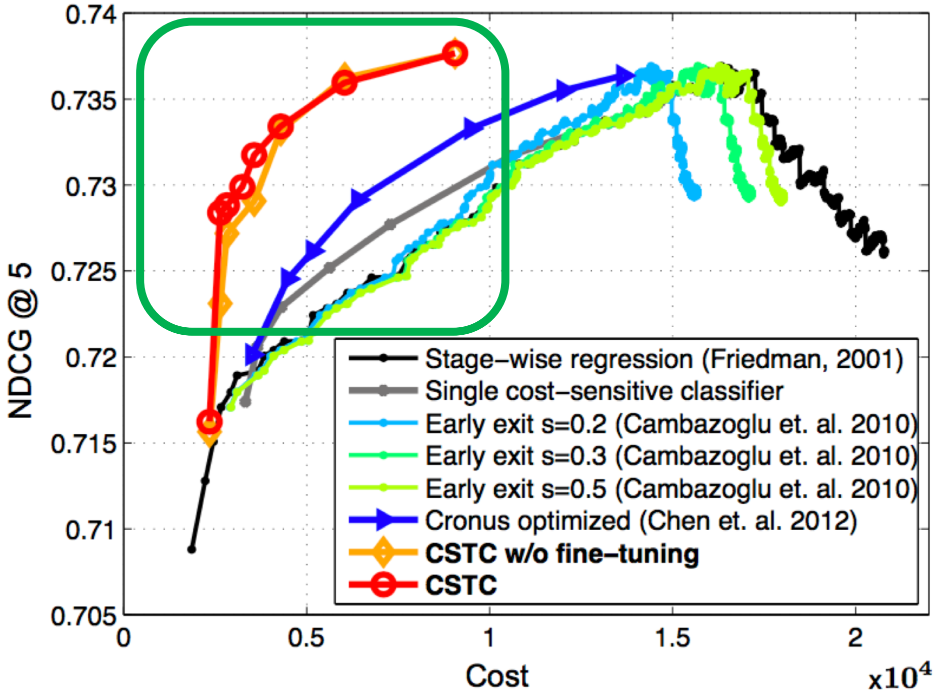
- New efficiency metrics: constant, step, exponential
- Focus on linear feature-based ranking functions
- Learned functions show significant decreased average query execution times



L. Wang, J. Lin, and D. Metzler. *Learning to efficiently rank*. In Proc. SIGIR 2010.

Cost-Sensitive Tree of Classifiers [XKWC13]

- Xu *et al.* observe that the test-time cost of a classifier is often dominated by the computation required for feature extraction
- Cost-Sensitive Tree of Classifiers: each path extracts different features and it is optimized for a specific sub-partit
  - Reduction of the average test-time complexit
  - Input-dependent feature selection
  - Dynamic allocation of time budgets: higher b
- Experiments
  - Yahoo LETOR dataset
  - Quality vs Cost



Z. Xu, M. J. Kusner, K. Q. Weinberger, and M. Chen. *Cost-sensitive tree of classifiers*. In Proc. ICML, 2013.

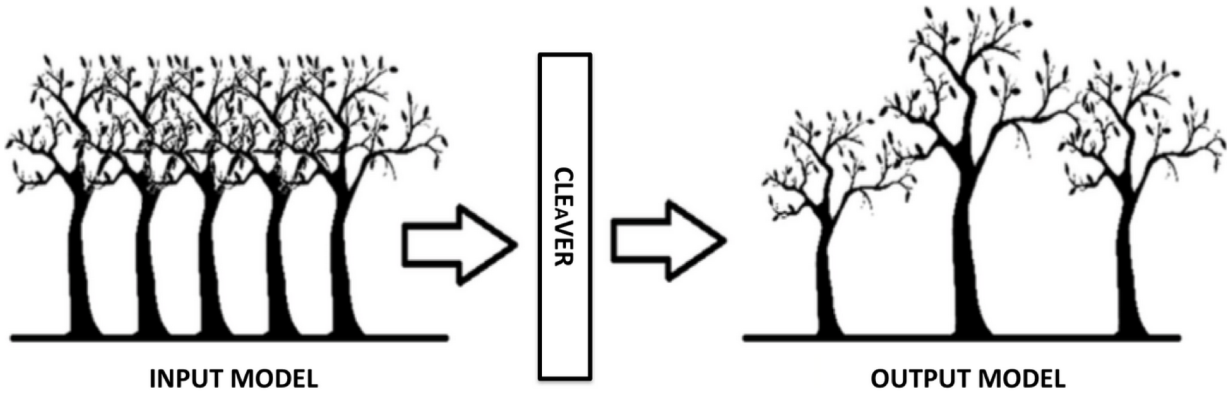
Training Efficient Tree-Based Models for Document Ranking [AL13]

- Asadi and Lin propose techniques for training GBRTs that have efficient runtime characteristics.
  - **compact**, **shallow**, and **balanced** trees yield faster predictions
- Cost-sensitive Tree Induction: jointly minimize the loss and the evaluation cost
- Two strategies
  - By directly modifying the node splitting criterion during tree induction
    - Allow split with maximum gain if it does not increase the maximum depth of the tree
    - Find a node closer to the root which, if split, result in a gain larger than the discounted maximum gain
  - Pruning while boosting with focus on **tree depth** and **density**
    - Additional stages compensate for the loss in effectiveness
    - Collapse terminal nodes until the number of internal nodes reach a balanced tree
- Experiments on MSLR-WEB10K show that the **pruning** approach is superior.
  - 40% decrease in prediction latency with minimal reduction in final NDCG.

N. Asadi and J. Lin. *Training efficient tree-based models for document ranking*. In Proc. ECIR, 2013.

CLEAVER [LNO+16a]

- Lucchese et al. propose a pruning & re-weighting post-processing methodology
- Several pruning strategies
  - random, last
  - skip, low weights
  - score loss
  - quality loss

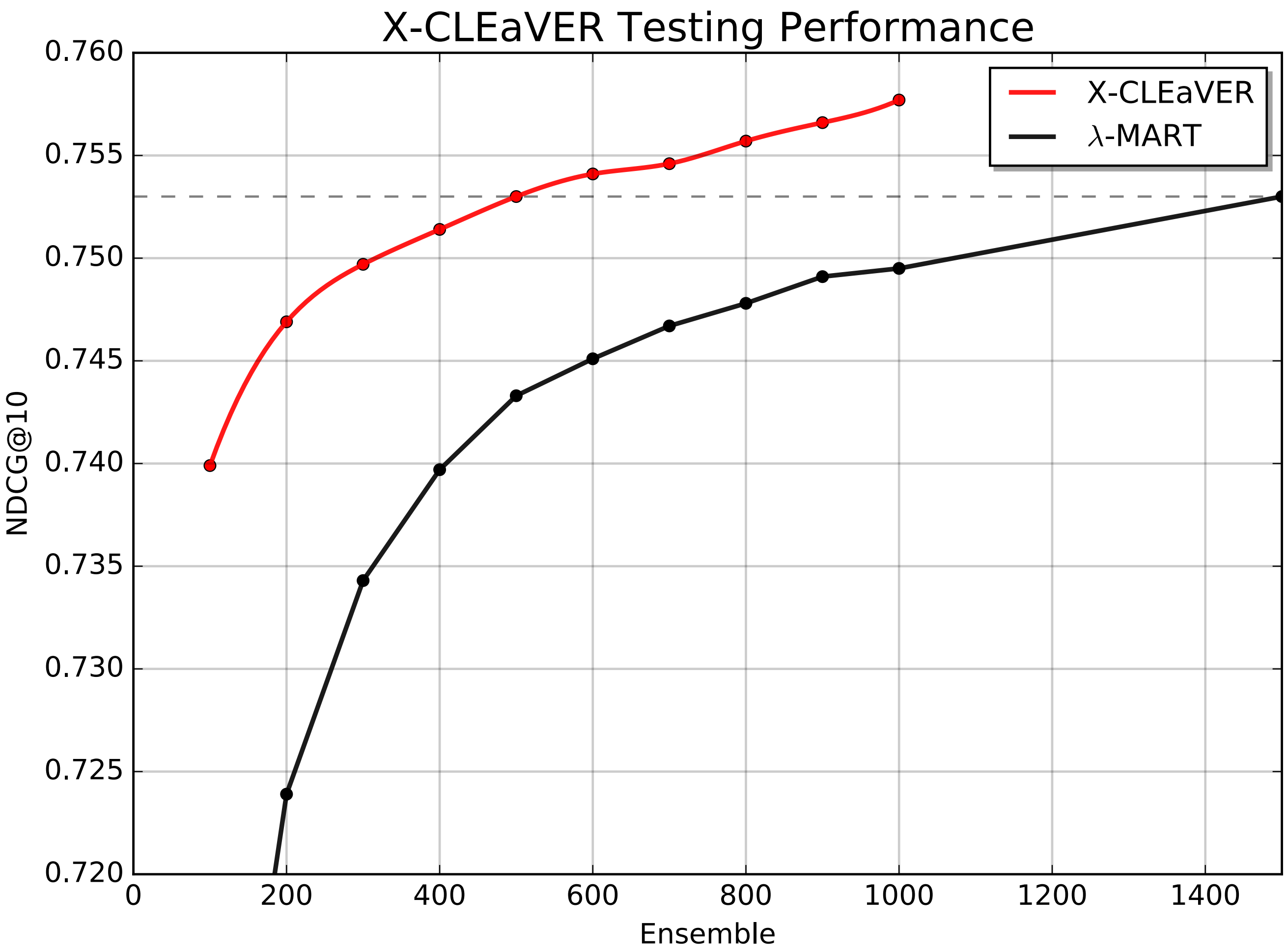


- Greedy line search strategy applied to tree weights
- Experiments on MART and LambdaMART
  - MSLR-Web30K and Istella-S LETOR
- Quality loss: same effectiveness of the original model with up to 20% of the trees

C. Lucchese, F. M. Nardini, S. Orlando, R. Perego, F. Silvestri, and S. Trani. *Post-learning optimization of tree ensembles for efficient ranking*. In Proc. ACM SIGIR, 2016.

X-CLEAVER [LNO+18]

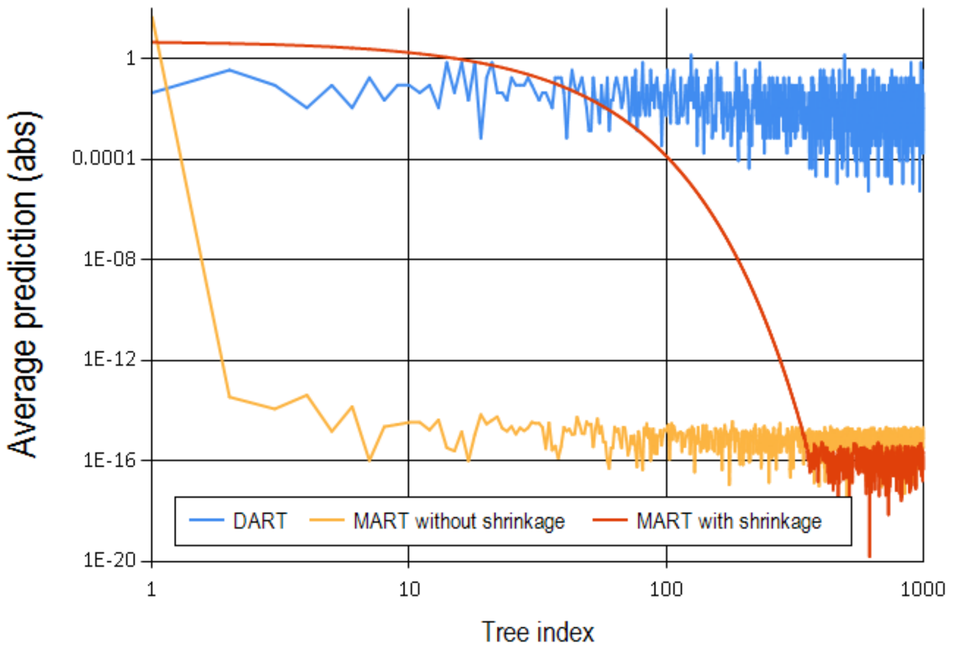
- Pruning and
  1. Redunda
  2. Weights … the desired ranking
- Same prunii
- Experiment … -1, Istella-S
  - Pruning ar … an applying a single pos
  - X-CLEaVEF … oss in performar



[LNO+18] C. Lucchese, F. M. Nardini, S. Orlando, R. Perego, F. Silvestri, and S. Trani. *X-CLEAVER: Learning Ranking Ensembles by Growing and Pruning Trees*. ACM TIST, 2018.

DART [VGB15]

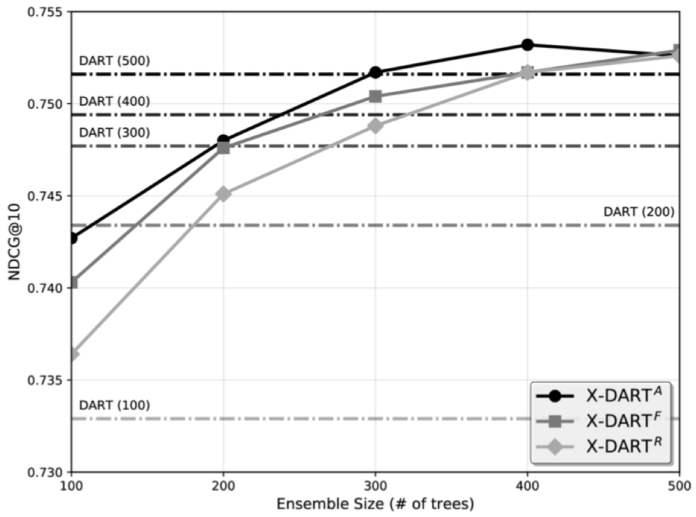
- Rashmi and Gilad-Bachrach propose to employ *dropouts* from NN while learning a MART: DART
  - Dropouts as a way to fight *over-specialization*
  - Shrinkage helps but does not solve
- DART differs from MART
  - When learning a new tree, a subset of the model is muted (random)
  - Normalization step when adding a new tree to avoid *overshooting*
- Experiments on MSLR-Web10K, NDCG@3
  - Improvement over LambdaMART



K. V. Rashmi and R. Gilad-Bachrach. *DART: Dropouts meet Multiple Additive Regression Trees*. In PMLR, 2015

X-DART [LNO+17]

- Lucchese *et al.* merge DART with pruning while training
  - like DART, some trees are muted and this set is removed after fitting if needed
- Two good news
  - X-DART builds even more compact models than DART
  - Smaller models are less prone to overfitting: potential for higher effectiveness
- Three strategies for pruning trees
  - Ratio, Fixed, Adaptive
- Experiments on MSLR-Web30K and Istella-S
  - X-DART (adaptive) provide statistically significant improvements w.r.t. DART with up to 20% less trees
  - Same effectiveness of DART with up to 40% less trees



[LNO+17] C. Lucchese, F. M. Nardini, S. Orlando, R. Perego, and S. Trani. *X-DART: Blending dropout and pruning for efficient learning to rank*. In Proc. ACM SIGIR, 2017.

Further Reading

- Xu *et al.* take into account the feature extraction cost during training to explicitly minimize the cpu-time during testing [XWC12]
  - Greedy Miser: extension of the stage-wise regression
  - Evaluation on Yahoo LETOR
  - Better efficiency-effectiveness trade-off w.r.t. stage-wise regression
- Peter *et al.* introduce the cost effective gradient boosting (GEGB) by taking into account both the feature extraction and the node evaluation costs
  - Trees are grown in best-first order: splits are evaluated for all current leaves and the one with the best objection reduction is chosen.
  - Experiments on Yahoo LETOR
  - GEGB outperforms Greedy Miser

[XWC12] Z. Xu, K. Weinberger, O. Chapelle. The Greedy Miser: Learning under Test-time Budgets. In Proc. ICML, 2012.
[PDHN17] S. Peter, F. Diego, F. Hamprect, B. Nadler. Cost efficient Gradient Boosting. In Proc. NIPS, 2017.