Continuation Methods and Curriculum Learning for Learning to Rank

Nicola Ferro
University of Padua, Padua, Italy
ferro@dei.unipd.it

Claudio Lucchese
Ca' Foscari University of Venice and ISTI-CNR, Pisa, Italy
claudio.lucchese@unive.it

Maria Maistro
University of Padua, Padua, Italy
maistro@dei.unipd.it

Raffaele Perego
ISTI-CNR, Pisa, Italy
r.perego@isti.cnr.it

ABSTRACT

In this paper we explore the use of *Continuation Methods* and *Curriculum Learning* techniques in the area of Learning to Rank. The basic idea is to design the training process as a learning path across increasingly complex training instances and objective functions. We propose to instantiate continuation methods in Learning to Rank by changing the IR measure to optimize during training, and we present two different curriculum learning strategies to identify easy training examples. Experimental results show that simple continuation methods are more promising than curriculum learning ones since they allow for slightly improving the performance of state-of-the-art λ-MART models and provide a faster convergence speed.

CCS CONCEPTS

• **Information systems → Learning to rank**; **Retrieval effectiveness**;

KEYWORDS

lambdamart; learning to rank; curriculum learning

ACM Reference Format:
Nicola Ferro, Claudio Lucchese, Maria Maistro, and Raffaele Perego. 2018. Continuation Methods and Curriculum Learning for Learning to Rank. In *2018 ACM Conference on Information and Knowledge Management (CIKM'18), October 22–26, 2018, Torino, Italy.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3269206.3269239

1 INTRODUCTION

Many proposals investigated *curriculum learning* as a general framework for several machine learning tasks, including training of deep neural networks [2]. Learning through a *curriculum* is a fascinating idea, borrowed from cognitive sciences, according to which a complex training task is designed as a multi-step *training path.* Initially, the learning algorithm is trained over simple training examples, and then it is progressively fine-tuned so as to deal with tasks of increasing complexity.

This is achieved by two main strategies. We call *Continuation Methods (CM)* [1, 7] those approaches that instead of optimizing difficult objective functions, i.e., objective functions with many local minima, transform the original function into a class of smoother or easier to minimize functions. The intuition behind this approach is that a smooth version of the target objective function can quickly and effectively drive the learning algorithm to a promising area of the search space that possibly includes the global optimum.

We call instead *Curriculum Learning (CL)* [2] those approaches where training examples of increasing complexity are gradually considered. The rationale is that the training algorithm can learn more effectively difficult examples if it was already trained on the simpler ones. In this work we explore the possibility of exploiting CM and CL approaches in the area of *Learning to Rank (LtR)*, the challenging task of training effective ranking functions from datasets of query-document pairs associated with graded relevance judgments. At the best of our knowledge this is the first attempt to investigate this research direction. Indeed, LtR is a complex task characterized by large and noisy training datasets and non-smooth objective functions. How to best fit CM and CL techniques within the LtR training process is not known. In this paper we propose and test several approaches and discuss their up- and down-sides. We show that designing a curriculum of objective functions of increasing complexity is a promising research direction since preliminary results based on CM show improved models and faster convergence of the training process. On the other hand, the same does not apply to the results achieved experimenting CL methods, which did not improve over the state-of-the-art LtR solution. Nevertheless, we believe that Curriculum Learning also may open up to new research directions in the design of effective learning algorithms, where increasingly larger sets of training examples and increasingly complex objective functions can be used dynamically to improve ranking models and minimize the overall computational cost of the training phase.

2 BACKGROUND

Continuation Methods (CM) approaches have shown to be effective in the optimization of complex objective functions [1, 7]. When the target objective function has many local minima, its direct optimization may lead to a "bad" sub-optimal result. In Continuation Methods, a multi-step optimization process is thus followed. At each step a different smooth function is optimized, i.e. a function

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3269239

easy to minimize that approximates the desired target function. The complexity and difficulty of the optimization function chosen is increased until the original target function is finally used. The intuition behind this approach is that the smooth versions of the original target function can provide a global representation of the search space which highlights the regions where the best local optima and the global one are located.

Applying a Continuation Method to a cost function C means to define a sequence of cost functions C_γ with $\gamma \in [0, 1]$, such that by increasing γ, the complexity of the function increases. Therefore, C_0 represents a highly smoothed version of the original cost function corresponding to C_1.

Curriculum Learning (CL) can be seen as a particular instantiation of Continuation Methods [2]. The basic idea is to organize the training examples in such a way that the easiest training examples are presented first and the complexity of the following ones is gradually increased. Thus, Curriculum Learning can be seen as a process exploiting a sequence of training criteria. Each training criterion corresponds to a different set of examples that can be differently weighted based on their complexity. At the subsequent steps, slightly more difficult examples are assigned with new weights. This is different from boosting approaches commonly adopted also in LtR, as in boosting the instance weights are determined during the training according to the mis-classification risk, while in Curriculum Learning weights are predetermined according to a training schedule.

Curriculum Learning methods can be formalized as follows: let z be a random variable representing a point of the dataset, that is $z = (x, y)$, where x represents the features vector and y represents the value to predict. We denote with $\mathbb{P}(z)$ the target training distribution, which represents the distribution that the *Machine Learning (ML)* algorithm aims to learn. Consider a multi-step optimization, where a parameter $\gamma \in [0, 1]$ refers to the iteration. At each step we assign a different weight W to each example z, $0 \leq W_\gamma(z) \leq 1$, therefore the target distribution will be proportional to the original distribution $Q_\gamma(z) \propto W_\gamma(z)\mathbb{P}(z)$.

The sequence of distributions $\{Q_\gamma\}_{\gamma \in [0,1]}$ is called a *Curriculum Learning (CL)* strategy if the entropy of these distributions increases:

$$H(Q_\gamma) < H(Q_{\gamma+\epsilon}) \quad \forall \epsilon > 0\ ,$$

and the sequence of weights $\{W_\gamma(z)\}_{\gamma \in [0,1]}$ is monotonically increasing:

$$W_{\gamma+\epsilon}(z) \geq W_\gamma(z) \quad \forall z\ ,\ \forall \epsilon > 0\ .$$

As an example, let $W_\gamma(z) \neq 0$ just on a finite set of easy examples, then Q_γ will concentrate on the same finite set. Increasing γ means adding some new and more complex examples, and the multi-step training corresponds to training the algorithm on an increasing sequence of subsamples of the training set, until the whole training set is considered.

Both approaches have been used successfully in several fields. Continuation Methods are applied when the function to optimize is not convex, as for example non linear optimization problems [5] in computational chemistry [7, 9], computational physics [13] and automatic control [10]. In ML, continuation methods are used with semi-supervised *Support Vector Machine (SVM)*, showing that this approach leads to lower test errors [4]. In [2], Curriculum Learning is exploited to train a language model (not used for ranking) with a deep neural network. In [12] a Curriculum Learning approach is applied to sort the training data to ease the learning of node representation in a heterogeneous star network.

In this paper we investigate how to apply CM and CL to LtR. To the best of our knowledge, neither CM nor CL have been ever exploited in combination with LtR algorithms. We will see that the integration of these methodologies is challenging and opens several questions, which we will discuss in the following.

3 CONTINUATION METHODS AND CURRICULUM LEARNING FOR LEARNING TO RANK

Learning to Rank (LtR) comprises those methods that use ML techniques to produce a ranking model [8]. As training set, LtR algorithms take a set of queries and a set of relevant and non relevant documents for these queries. Given the feature-based representation of each query-document pair and its relevance label, the goal is to learn a scoring function that induces the "ideal" ranking established by the relevance labels.

Information Retrieval (IR) measures are typically exploited to optimize LtR algorithms. However, these measures are not differentiable and thus difficult to optimize. One of the most effective algorithm addressing this challenge is λ-MART [3]. The loss function of λ-MART is based on λ-RANK gradient approximation, which considers the variation in the effectiveness after swapping two documents of a ranked list. It is defined as a product of two main components, $\lambda_{i,j} = \Delta\mathcal{M}_{ij} \cdot (1 + e^{s_i - s_j})^{-1}$, where the first term is the variation in the measure score when the documents at ranks i and j are swapped, and the second term is the derivative of the RankNet cost [3], which minimizes the number of misplaced documents. Finally, the lambda gradient of a document d_i is computed as the sum of all the pairwise gradients: $\lambda_i = \sum_j \lambda_{i,j}$. Notice that λ-MART usually optimizes *Normalized Discounted Cumulated Gain (nDCG)* as effectiveness measure $\mathcal{M}$, even if it is possible to choose any other IR measure. We investigate two different strategies to evaluate Continuation Methods and Curriculum Learning in LtR. Both of them are based on a two step training, i.e. first the algorithm is trained with an easy criterion, then it is trained with the original and more complex criterion.

Continuation Methods. Recall that continuation methods exploit increasingly complex objective functions. In order to apply a CM to λ-MART we need to smooth the λ-MART loss function. Smoother variants of nDCG have been previously proposed, e.g. SoftNDCG [14], however their performance did not prove to be significantly better than the original λ-MART loss function. Therefore, we adopted a different cost function for the first step of the proposed continuation method. We train a regression forest where the first t trees are built by minimizing *Mean Square Error (MSE)*, while the remaining are generated by optimizing nDCG, as with the standard λ-MART. Indeed, the last part of the forest trained with the λ-MART algorithm uses the previously built t regression trees as a starting point of the search space. This process allows to start the λ-MART strategy from a point in the search space that it is closer to a good solution, thanks to the preceding MSE optimization, rather than starting from scratch. We refer to this strategy as MSE-t.

Curriculum Learning. While continuation methods produce different versions of the original loss function, CL performs a sampling of the training data in order to present to the algorithm the easiest examples first. Since the concept of easiness of an example is not well defined and depends on the specific task and algorithm, we tried several possible approaches, named `l_HRel`, `q_NRel`, `q_BM25`, and `q_PageRank`.

In the first case, referred as `l_HRel`, we define the easy examples as the documents that are either highly relevant or not relevant. As discussed in Sec. 4, documents in the training set are labeled on a discrete five-steps relevance scale from not-relevant to highly relevant. The rationale is that removing fairly or partially relevant documents should simplify the discrimination between highly relevant and not relevant documents. Therefore we performed a sampling on the document labels and we initially train λ-MART on a dataset containing only highly relevant and not relevant documents.

In all the other cases, we worked at query level and we tested several ways to select easy and difficult queries. The `q_NRel` approach considers a query easy if it has many relevant documents, while hard queries are those with just a few or none relevant documents. Thus, to perform the sampling, we sorted the query by their number of relevant documents and we selected the first 25% of them, i.e., the quarter with the greatest number of relevant documents. This subset was used as initial training set for λ-MART.

Analogously, `q_BM25` and `q_PageRank` defines queries' easiness by considering their BM25 or PageRank score. This means that the documents were ranked with respect to their BM25 or PageRank feature and then nDCG was computed to determine the query score. Finally, the queries were sorted by their nDCG score and the first 25% of them was selected as initial training set.

After sampling the training set, with one of the afore mentioned approaches, we instantiate a two step curriculum strategy, with the first step producing t trained trees and the second step $T - t$. For the first step, λ-MART is trained on one of the obtained training subsets with nDCG as effectiveness measure in the objective function, and during the second step, the training is conducted on the whole dataset, again with nDCG as effectiveness measures in the objective function.

4 EXPERIMENTS

4.1 Experimental Setting

The accuracy of the different CL strategies is evaluated on the MSLR-WEB30K dataset [11]. The dataset encompasses 31,531 queries from the Microsoft Bing search engine for a total of 3,771,125 query-document pairs represented by 136 features, where each document is labeled with a relevance label in the set $\{0, 1, 2, 3, 4\}$ (from not-relevant to highly relevant). The dataset is provided as a 5-fold split. We use the state-of-the-art λ-MART as the reference LtR algorithm. Its parameters were swiped with cross-validation so as to maximize the average performance over the validation folds. Learning rate ν was evaluated in the set $\{0.01, 0.05, 0.1, 0.5\}$ and the maximum number of leaves L in the set $\{8, 16, 32, 64\}$, with best results observed with the setting $\nu = 0.05$ and $L = 64$. Notice that in the following sections, whenever we refer to the baseline we mean λ-MART trained on the whole dataset with nDCG@10 as effectiveness measure and the above settings. Due to space constraints we limit our investigation to models composed of $T = 500$ trees.

4.2 Curriculum Learning - Data Sampling

Table 1 shows the performance of λ-MART models with all the tested CL approaches together with the proposed CM approach. The performances of the different CL instantiation are compared against the baseline (top line in the table). Each row of the table reports the nDCG@10 scores of the proposed models averaged across the 5 different folds. Moreover, in each plot we report the results achieved when training a first set of t trees on a sample of the dataset and the remaining $T - t$ by exploiting the whole training set. We explored different switching points of the cost functions at t equals to 100, 200, 300, 400 trees, while the total number of trees T is always equal to 500.

Table 1 shows that, for any approach and for any possible switching point t, the CL strategy underperforms and never reaches the baseline. Among all the tested approaches, `l_Hrel` is the worst performing approach. This might be due to the sampling strategy which is too aggressive. Indeed, keeping only the highly relevant documents removes too many documents from the training set and the algorithm does not have enough positive examples.

Similarly for the `q_NRel`, `q_BM25`, and `q_PageRank` CL strategies, the sampling conducted over the training set affects the performances instead of boosting them. Recall that in this case the sampling is carried out with respect to the query and not the documents, this means that for those queries that are considered easy we keep all the documents. Although these latter CL approaches are not achieving better results than the baseline, they do perform better than the `l_Hrel` approach and their performances are much closer to the baseline score. This suggests that the sampling strategy based on the queries is less aggressive than the strategy based on the labels.

Finally, as a general trend all the CL approaches perform more and more worse when the number of trees trained on the subset increases, further suggesting that removing training instances irredeemable damages the performances. However, we do not draw definitive conclusions on the effectiveness of the data sampling strategies experimented, but we highlight that the instance weighting approach might partially with the λ-MART boosting framework.

4.3 Continuations Method - Cost Functions

Fig. 1 shows the performance of the proposed CM against the baseline. We report the results achieved when training a first set of t trees by optimizing MSE and the remaining by optimizing the target measure nDCG@10 averaged over the five folds of MSLR-WEB30K. An interesting behavior can be observed for the initial trees of the forest. Optimizing MSE provides significantly better models up to 200 tree. Instead, optimizing MSE alone leads to worse results as shown by the test when the switching point occur at $t = 400$ trees. On average, the best results were achieved with a switching point at $t = 200$ trees providing consistently better results than λ-MART.

In Tab. 2 we compare the MSE $t = 200$ models with the λ-MART baseline. In particular we show the NDCG@10 achieved by the MSE $t = 200$ model at different model sizes, and we compare it with the smallest λ-MART model providing at least the same NDCG@10.

Table 1: CL and CM nDCG@10 scores averaged across the 5 folds.

LambdaMart	0.5159			
	switch point t			
Curriculum Learning	100	200	300	400
`l_HRel`	0.5098	0.5080	0.5016	0.4856
`q_Nrel`	0.5152	0.5145	0.5133	0.5105
`q_BM25`	0.5130	0.5105	0.5070	0.4988
`q_PageRank`	0.5147	0.5137	0.5114	0.5048
Continuation Methods				
MSE-t	0.5171	**0.5187**	**0.5187**	0.5179

Table 2: Comparison of MSE t = 200 versus LambdaMart baseline when achieving at least the same NDCG@10.

MSE $t = 200$		LambdaMart		size
# trees	NDCG@10	# trees	NDCG@10	ratio
100	0.5003	170	0.5010	1.7
200	0.5080	255	0.5082	1.3
300	0.5144	400	0.5145	1.3
400	0.5174	500	–	–
500	0.5187	500	–	–

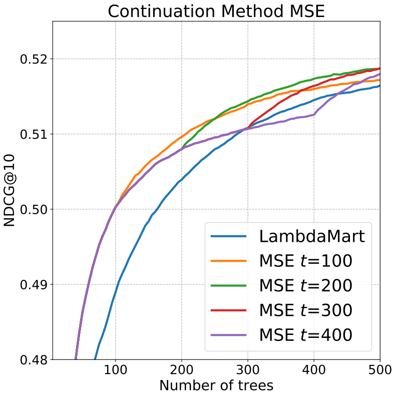

Figure 1: Continuation method by optimizing MSE during the first t trees of the ensemble.

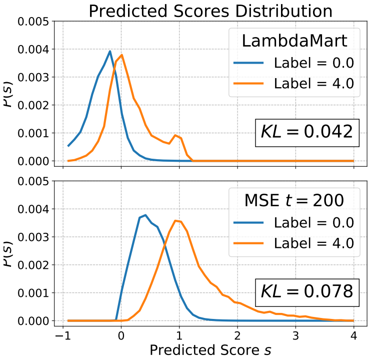

Figure 2: Discriminative power after 100 trees. Predicted score for documents with largest and smallest label.

Results show two important behaviors. First, the baseline algorithm λ-MART is never able to provide the same accuracy as the largest MSE $t = 200$ model with 500 trees. This was already clear in Fig. 1. Second, the λ-MART creates effective-equivalent models that are larger in size with 1.3 to 1.7 times more trees. This means that λ-MART reaches the same effectiveness of MSE $t = 200$ with models that are much larger and therefore more expensive to apply at scoring time. We recall that the use of larger models can be too expensive for real-world production systems, and therefore, reducing the size of the model proportionally reduces its run-time cost, thus making accurate models feasible in practice.

We further investigated the benefit provided by the CM by comparing the discriminative power of the λ-MART baseline and of MSE $t = 200$ when considering only the first 100 trees. In Fig. 2 we report the predicted score distribution only for documents with the smallest and the largest training label (note that distributions are normalized per label). Even if there is not an optimal separation, the plot highlights how the MSE allows to better discriminate between irrelevant (label=0) and highly relevant (label=4) documents, with a Kullback-Leibler divergence almost twice as big when optimizing MSE (0.078 vs. 0.042). This initial discriminative power is probably a beneficial starting point for the LtR optimization.

We conclude that optimizing MSE during the first part of the training improved the convergence speed of the learning process driving the algorithm to the most promising areas of the search space, which eventually lead to a more effective model.

5 CONCLUSION AND FUTURE WORK

The paper presented different approaches to instantiate CM and CL with λ-MART. In the context of continuation methods we train first the algorithm with MSE and then with nDCG@10. Moreover, we tested two different CL strategies to sample the training data.

Even if CL is known to be an effective approach in other areas of ML, it turned out that its application to the LtR case does not produce the same type of benefits, at least using straightforward ways of creating the curriculum. This calls for a deeper future investigation to better understand what differentiates the LtR from other ML applications in terms of CL and for the experimentation of more complex curriculum building strategies.

The results achieved by experimenting CM on increasingly difficult objective functions looks promising. As future work, we aim at investigating more complex curricula dynamically exploiting multiple objective functions and curricula with both continuation methods and data sampling.

REFERENCES

[1] E. L. Allgower, and K. Georg. Numerical Continuation Methods. An Introduction. Springer-Verlag, 1980.
[2] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, Curriculum Learning. In ICML, pages 41–48, ACM, 2009.
[3] C. J. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to Rank Using Gradient Descent. In ICML, pages 89–96, ACM, 2005.
[4] O. Chapelle, M. Chi, and A. Zien. A Continuation Method for Semi-supervised SVMs. In ICML, pages 185–192, ACM, 2006.
[5] B. Chen, and N. Xiu. A Global Linear and Local Quadratic Noninterior Continuation Method for Nonlinear Complementarity Problems Based on Chen–Mangasarian Smoothing Functions. SIAM Journal on Optimization, 9, 3, pages 605–623, SIAM, 1999.
[6] X. Chen, and A. Gupta. Webly Supervised Learning of Convolutional Networks. In ICCV, pages 1431–1439, IEEE, 2015.
[7] T. F. Coleman, and Z. Wu. Parallel Continuation-based Global Optimization for Molecular Conformation and Protein Folding. Journal of Global Optimization, 8, 1, pages 49–65, Springer, 1996.
[8] T. Liu. Learning to Rank for Information Retrieval. Foundations and Trends in Information Retrieval, 3, pages 225–331, 2009.
[9] J. J. Moré, and Z. Wu. Global Continuation for Distance Geometry Problems. SIAM Journal on Optimization, 7, 3, pages 814–836, SIAM, 1997.
[10] R. Nagamune. A Robust Solver Using a Continuation Method for Nevanlinna-Pick Interpolation with Degree Constraint. IEEE Transactions on Automatic Control, 48, 1, pages 8113–117, IEEE, 2003.
[11] T. Qin, and T. Liu. Introducing LETOR 4.0 Datasets. In CoRR, 2013.
[12] M. Qu, J. Tang, and J. Han. Curriculum Learning for Heterogeneous Star Network Embedding via Deep Reinforcement Learning. In WSDM, pages 468–476, ACM, 2018.
[13] E. Rabani, D. R. Reichman, G. Krilov, and B. J. Berne. The Calculation of Transport Properties in Quantum Liquids Using the Maximum Entropy Numerical Analytic Continuation Method: Application to Liquid Para-hydrogen. Proceedings of the National Academy of Sciences, 99, 3, pages 1129–1133, National Academy of Sciences, 2002.
[14] M. Taylor, J. Guiver, S. Robertson, and T. Minka Softrank: optimizing non-smooth rank metrics. In WSDM, pages 77–86, ACM, 2008.