StochasticRank: Global Optimization of Scale-Free Discrete Functions

Aleksei Ustimenko [1] **Liudmila Prokhorenkova** [1 2 3]

Abstract

In this paper, we introduce a powerful and efficient framework for direct optimization of ranking metrics. The problem is ill-posed due to the discrete structure of the loss, and to deal with that, we introduce two important techniques: stochastic smoothing and novel gradient estimate based on partial integration. We show that classic smoothing approaches may introduce bias and present a universal solution for a proper debiasing. Importantly, we can guarantee *global convergence* of our method by adopting a recently proposed Stochastic Gradient Langevin Boosting algorithm. Our algorithm is implemented as a part of the CatBoost gradient boosting library and outperforms the existing approaches on several learning-to-rank datasets. In addition to ranking metrics, our framework applies to any scale-free discrete loss function.

1. Introduction

The quality of ranking algorithms is traditionally measured by ranking quality metrics such as Normalized Discounted Cumulative Gain (NDCG), Expected Reciprocal Rank (ERR), Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and so on (Sakai, 2013). These metrics are defined on a list of documents sorted by their predicted relevance to a query and capture the utility of that list for users of a search engine, who are more likely to scan documents starting at the top. Direct optimization of ranking metrics is an extremely challenging problem since sorting makes them piecewise constant (as functions of predicted relevances), so they are neither convex nor smooth. Many algorithms were proposed for different ranking objectives in the learning-to-rank (LTR) research field. We refer to Liu (2009) for a systematic overview of some classic methods.

[1]Yandex, Moscow, Russia [2]Moscow Institute of Physics and Technology, Dolgoprudny, Moscow Region, Russia [3]Higher School of Economics, Moscow, Russia. Correspondence to: Aleksei Ustimenko <austimenko@yandex-team.ru>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

To deal with the discrete structure of a ranking loss, one can use some smooth approximation, which is easier to optimize. This technique lies behind such well-known algorithms as SoftRank (Taylor et al., 2008), ApproxNDCG (Qin et al., 2010), RankNet (Burges, 2010), etc. The obtained smooth function can be optimized by gradient-based methods and, in particular, by Stochastic Gradient Boosting (SGB) that is known to be the learning algorithm behind most state-of-the-art LTR frameworks and is commonly preferred by major search engines (Chapelle & Chang, 2011; Yin et al., 2016). Unfortunately, all known smoothing approaches suffer from *bias* (see Sections 4.2-4.3) which prevents them from truly direct optimization. Moreover, smoothed ranking loss functions are non-convex, and existing algorithms can guarantee only local optima.

Our ultimate goal is to solve these problems and propose a *truly direct* LTR algorithm with provable guarantees of global convergence and generalization. We adopt a theoretical approach, so we start with formal definitions of the class of ranking losses and its generalization to *scale-free (SF) discrete loss functions* (Section 3.2). Our results hold for the general class of SF losses, which, in addition to all ranking metrics, includes, e.g., a recently proposed loss function for Learning-to-Select-with-Order (Vorobev et al., 2019). Then, to mitigate the discontinuity of the loss, we use stochastic smoothing. We prove that previous smoothing-based approaches are inconsistent with the underlying loss (due to the problem of ties, which we discuss in the next section) and propose a universal solution to this problem (relevance-based consistent smoothing, see Section 4.3). Next, we derive a novel stochastic gradient estimate, which can be applied to the entire class of SF losses (see Section 5). The obtained estimate has low variance and uniformly bounded error, which is crucial for our analysis. Finally, to guarantee *global* convergence of the algorithm, we adopt a recently proposed Stochastic Gradient Langevin Boosting (SGLB) algorithm (Ustimenko & Prokhorenkova, 2020). SGLB is based on a well studied Stochastic Gradient Langevin Dynamics (Gelfand et al., 1992; Raginsky et al., 2017; Erdogdu et al., 2018) and converges globally for a wide range of loss functions including non-convex ones. We adapt SGLB to our setting and obtain a gradient boosting algorithm that converges globally for the entire class of SF loss functions with provable generalization guarantees (see Section 6).

To sum up, to the best of our knowledge, the proposed StochasticRank algorithm is the first *globally converging* LTR method with provable guarantees that optimizes *exactly* the underlying ranking quality loss. StochasticRank is implemented within the official CatBoost library (Prokhorenkova et al., 2018; CatBoost, 2020). Our experiments show that StochsticRank outperforms the existing approaches on several LTR datasets.

The rest of the paper is organized as follows. In the next section, we briefly overview the related research on learning to rank. In Section 3, we formalize the problem and, in particular, define a general class of ranking loss functions. In Section 4, we formulate the problem of smoothing bias and propose an unbiased solution. Then, in Section 5, we derive a novel stochastic gradient estimate for the whole class of loss functions under consideration. In Section 6, we show how SGLB can be used to achieve global convergence. Finally, Section 7 empirically compares the proposed algorithm with existing approaches, and Section 8 concludes the paper.

2. Related Work

Usually, researches divide all LTR methods into three categories: pointwise, pairwise, and listwise (Liu, 2009).

Pointwise are the earliest and simplest methods: they approximate relevance labels based on simple or ordinal regression or classification. Such methods were shown to be ineffective for LTR, since loss functions they optimize (e.g., RMSE for relevance labels) differ significantly from the target ranking metric, e.g., NDCG@k.

Pairwise methods make a step forward and focus on pairwise preferences and thus known to outperform pointwise approaches significantly. Nevertheless, pairwise approaches still suffer from the problem of solving a different task rather than optimizing a ranking quality objective.

Listwise methods try to solve the problem directly by developing either smooth proxies of the target ranking metric like SoftRank (Taylor et al., 2008), BoltzRank (Volkovs & Zemel, 2009), ApproxNDCG (Qin et al., 2010), RankNet (Burges, 2010) or by Majorization-Minimization procedure that builds a convex upper bound on the metric on each iteration like LambdaMART (Wu et al., 2010), LambdaLoss (Wang et al., 2018), PermuRank (Xu et al., 2010), SVMRank (Cao et al., 2006), etc.

As discussed in the previous section, algorithms based on smooth approximations suffer from bias and local optima. Also, there are listwise approaches that try to optimize the target loss function without smoothing. For instance, DirectRank (Tan et al., 2013) constructs an ensemble of decision trees, where the values in the leaves are chosen to optimize the original loss. However, due to greediness, this approach can guarantee only local optima.

Finally, let us note that algorithms optimizing a convex upper bound instead of the original loss cannot be truly direct since the optimum for the upper bound can potentially be far away from the true optimum. This is nicely illustrated by Nguyen & Sanner (2013) for accuracy optimization. Let us also mention a recent approach for improving learning-to-rank algorithms by adding Gumbel noise to model predictions (Bruch et al., 2020). This is a regularization technique since it builds a convex upper bound on any given convex loss (e.g., LambdaMART).[1] Thus, from a theoretical point of view, this approach cannot be truly direct since it uses convex upper bounding.

The issue of smoothing bias mentioned in the introduction is connected to the problem of ties: if predicted relevances of some documents coincide, one has to order them somehow to compute a ranking metric. This situation may occur when two documents have equal features. More importantly, ties are always present in boosting algorithms based on discrete weak learners such as decision trees. Unfortunately, this problem is rarely addressed in LTR papers. In practice, it is reasonable to use the *worst* permutation. First, due to strong penalization, it would force an optimization algorithm to avoid ties. Second, in practice, one cannot know how a production system would rank the items, and often some attribute negatively correlated with relevance is used (e.g., sorting by a bid in online auctions). The importance of using the worst permutation is also discussed by Rudin & Wang (2018), and this ordering is adopted in some open-source libraries like CatBoost (Prokhorenkova et al., 2018). An alternative choice is to compute the expected value of a ranking metric for a *random* permutation. This choice is rarely used in practice, since it is computationally complex and gives non-trivial scores to trivial constant predictions, but is often assumed (explicitly or implicitly) by LTR algorithms (Kustarev et al., 2011).

3. Problem Formalization

3.1. Examples of Ranking Loss Functions

Before we introduce a general class of loss functions, let us define classic ranking quality functions widely used throughout the literature and in practice.[2] These loss functions depend on z, which is a vector of *scores* produced by the model, and r, which is a vector of relevance labels for a given query. The length of these vectors is denoted by n and can be different for different queries.

[1] Nesterov & Spokoiny (2017) prove this for Gaussian noise, but the same result generalizes to any centered noise.

[2] To obtain the *loss* function from the corresponding *quality* function, we multiply it by -1.

Let $s = \operatorname{argsort}(z)$, i.e., s_i is the index of a document at i-th position if documents are ordered according to their scores (if $z_i = z_j$ for $j \neq i$, then we place the less relevant one first). Let us define DCG@k, where k denotes the number of top documents we are interested in:

$$\text{DCG@k}(z, r) = \sum_{i=1}^{\min\{n,k\}} \frac{2^{r_{s_i}} - 1}{2^4 \log_2(i+1)}, \tag{1}$$

where $r_i \in [0, 4]$ are relevance labels. This quality function is called *Discounted Cumulative Gain*: for each document, the numerator corresponds to *gain* for the relevance, while the denominator *discounts* for a lower position. NDCG@k is a normalized variant of DCG@k:

$$\text{NDCG@k}(z, r) = \frac{\text{DCG@k}(z, r)}{\max_{z' \in \mathbb{R}^n} \text{DCG@k}(z', r)}. \tag{2}$$

Expected Reciprocal Rank ERR@k assumes that $r_j \in [0, 1]$:

$$\text{ERR@k}(z, r) = \sum_{i=1}^{\min\{n,k\}} \frac{r_{s_i}}{i} \prod_{j=1}^{i-1} (1 - r_{s_j}). \tag{3}$$

Mean reciprocal rank (MRR) is used for binary relevance labels $r_j \in \{0, 1\}$:

$$\text{MRR}(z, r) = \sum_{i=1}^{n} \frac{r_{s_i}}{i} \prod_{j=1}^{i-1} (1 - r_{s_j}), \tag{4}$$

which is the inverse rank of the first relevant document.

Finally, let us define a quality function for the LSO (learning to select with order) problem introduced by Vorobev et al. (2019), which is not exactly a ranking metric, but has a similar structure. The order of elements is predefined (documents are sorted by their indices), but the list of documents to be included is determined by $(\mathbb{1}_{\{z_i > 0\}})_{i=1}^n \in \{0, 1\}^n$:

$$\text{DCG-RR}(z, r) = \sum_{i=1}^{n} \frac{r_i \, \mathbb{1}_{\{z_i > 0\}}}{1 + \sum_{j<i} \mathbb{1}_{\{z_j > 0\}}}. \tag{5}$$

In the sum above, for each included document we divide its relevance by its rank.

3.2. Generalized Ranking Loss Functions

To develop a stochastic ranking theory, we first formalize the class of loss functions to which our results apply. We start with a very general class of *scale-free (SF) discrete loss functions*. Further, by ξ we denote a vector of context, which may include relevance and any other factors affecting the ranking quality value (like query type or document topic).

Definition 1. *A function $L(z, \xi) : \coprod_{n>0} \mathbb{R}^n \times \Xi_n \to \mathbb{R}$ is a* Scale-Free Discrete Loss Function *iff the following conditions hold:*

- ***Uniform boundedness**: There exists a constant $l > 0$ such that $|L(z, \xi)| \leq l$ holds $\forall n$, $\forall \xi \in \Xi_n$, $\forall z \in \mathbb{R}^n$;*
- ***Discreteness on subspaces**: For each $n \in \mathbb{N}$ and linear subspace $V \subset \mathbb{R}^n$ there exist convex open subsets $U_1, \ldots, U_k \subset V$, $k = k(n, V)$ (w.r.t. induced topology on V), mutually disjoint $U_i \cap U_j = \emptyset$ for $i \neq j$, with everywhere dense union $\overline{\cup_i U_i} = V$ ($\overline{X}$ denotes the closure of X w.r.t. the ambient topology), such that for any $\xi \in \Xi_n$ and $i \leq k$ holds $L(z, \xi)\big|_{U_i} \equiv \text{const}(i, \xi, V)$;*
- ***Jumps regularity**: By reusing U_i defined above, for any $z \notin \cup_i U_i$ either of the following conditions holds:*

$$\liminf_{z' \to z} L(z', \xi) < L(z, \xi) \leq \limsup_{z' \to z} L(z', \xi),$$

$$\liminf_{z' \to z} L(z', \xi) = L(z, \xi) = \limsup_{z' \to z} L(z', \xi),$$

where $z' \to z$ means $z' \in \cup U_i$, $z' \to z$.

- ***Scalar freeness**: For any $n > 0, \xi \in \Xi_n, z \in \mathbb{R}^n, \lambda > 0$ holds $L(\lambda z, \xi) = L(z, \xi)$.*

We denote the class of all SF discrete loss functions by $\mathcal{R}_0$. Informally speaking, $\mathcal{R}_0$ is a class of bounded discrete functions on a sphere. The jumps regularity property is needed to exclude the breaking points from $\arg\min L$. One can show that all loss functions defined in Section 3.1, including the LSO loss DCG-RR, belong to $\mathcal{R}_0$.

StochasticRank out-of-box can be applied to any SF discrete loss function. However, to guarantee global convergence, we need to use consistent smoothing (see Section 4.3), which has to be chosen based on the properties of a particular metric. We propose smoothing which is consistent for the whole class of *ranking loss functions* defined below.

Assume that $\Xi_n = \mathbb{R}^n \times \Xi'_n$ and $\xi \in \Xi_n$ is a tuple (r, ξ'), where $r \in \mathbb{R}^n$ is a vector of relevance labels. As discussed in Section 2, a particular definition of a ranking loss depends on tie resolution. When some documents have equal scores, we may either use the worst permutation (as commonly done in practice) or compute the average over all orderings of such documents (as usually assumed by LTR algorithms). The definition below assumes the worst permutation.

Definition 2. *A function $L(z, \xi) \in \mathcal{R}_0$ is a* Ranking Loss Function *iff the following properties hold:*

- ***Relevance monotonicity**: For each $n > 0$ and $z, r \in \mathbb{R}^n$, there exists $\epsilon_0 = \epsilon_0(r, z) > 0$ such that $\forall \epsilon \in (0, \epsilon_0]$ $\exists \delta = \delta(\epsilon, r, z) > 0$ such that $\forall z' : \|z' - z\| < \delta$:*

$$\limsup_{z'' \to z} L(z'', \xi) = L(z' - \epsilon r, \xi).$$

Informally, $-r$ is the worst direction for the loss function, i.e., near a breaking point with $z_i = z_j$ and $r_i > r_j$ for some i, j, it is better to have $z_i > z_j$.

- ***Strong upper semi-continuity (s.u.s.c.)**: For each $n > 0$ and $z, r \in \mathbb{R}^n$:*

$$\limsup_{z' \to z} L(z', \xi) = L(z, \xi).$$

Informally, this means that if we do not know how to rank two items (i.e., $z_i = z_j$ for $i \neq j$), then we shall rank them by placing the less relevant one first.

- ***Translation invariance**:[3] For any $n > 0, r, z \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$ holds: $L(z + \lambda \mathbb{1}_n, \xi) \equiv L(z, \xi)$, where $\mathbb{1}_n := (1, \dots, 1) \in \mathbb{R}^n$.*

- ***Pairwise decision boundary:**[4] Partition of the space for discreteness on subspaces $\{U_i\}$ for $\mathbb{R}^n$ can be obtained as connected components of $\mathbb{R}^n \setminus \cup_{i,j} \{z : z_i - z_j = 0\}$, similarly for an arbitrary subspace V.*

We denote this class of functions by $\mathcal{R}_1$. It can be shown that $\mathcal{R}_1$ includes all ranking losses defined in Section 3.1, but not the LSO loss DCG-RR which does not satisfy Relevance monotonicity.

Let us now define a class $\mathcal{R}_1^{soft}$, where instead of the worst ranking for ties, we consider the expected loss of a random ranking. For this, we replace the s.u.s.c. condition by:

- **Soft semi-continuity (s.s.c.)**: For each $n > 0$ and $r, z \in \mathbb{R}^n$ we have:

$$\lim_{\sigma \to 0_+} \mathbb{E} L(z + \sigma\varepsilon, \xi) = L(z, \xi),$$

where $\varepsilon \sim \mathcal{N}(\mathbb{0}_n, I_n)$ is a normally distributed random variable.

We will show that under some restrictive conditions (that are commonly assumed in the LTR literature), it does not matter which of the two definitions we use ($\mathcal{R}_1$ or $\mathcal{R}_1^{soft}$) as they coincide almost surely and have equal $\arg\min L$ sets. However, we will explain why these conditions do not hold in practice and in general the minimizers for $\mathcal{R}_1$ and $\mathcal{R}_1^{soft}$ do not coincide.

3.3. Model Assumptions

We assume that for each $n > 0$ and $\xi \in \Xi_n$ there is a model $f_\xi(\theta) : \mathbb{R}^m \to \mathbb{R}^n$ such that $f_\xi(\theta) = \Phi_\xi \theta$ for some matrix $\Phi_\xi \in \mathbb{R}^{n \times m}$, where $\theta \in \mathbb{R}^m$ is a vector of parameters (independent from ξ) and $m \in \mathbb{N}$ is the number of parameters. Typically, each row of Φ_ξ is a feature vector. Gradient boosting over decision trees satisfies this assumption. Indeed, let us consider all possible trees of a fixed depth formed by a finite number of binary splits obtained by binarization of the initial feature vectors. To get a linear model, we say that θ is a vector of leaf weights of these trees and Φ_ξ is a binary matrix formed by the binarized feature vectors.

We will also assume that $\langle \mathbb{1}_n, z \rangle_2 = 0$. Indeed, instead of $z = f_\xi(\theta)$ we can define the model as $z = f_\xi(\theta) - \frac{1}{n} \mathbb{1}_n^T f_\xi(\theta) \mathbb{1}_n$, which is equivalent due to the translation invariance property.

3.4. Data Distribution

Assume that we are given some distribution $\xi \sim \mathcal{D}$ on $\Xi := \coprod_{n>0} \Xi_n$ meaning that ξ also implicitly incorporates information about the number of items n, i.e., for $\xi \in \Xi$ there exists a unique number $n > 0$ so that $\xi \in \Xi_n$. $\mathcal{D}$ is some unknown distribution, e.g., the distribution of queries submitted to a search system. We are given a finite i.i.d. sample $\xi_1, \dots, \xi_N \sim \mathcal{D}$ that corresponds to the train set. Let $\mathcal{D}_N := \frac{1}{N} \sum_{i=1}^N \delta_{\xi_i}$ be the empirical distribution.

3.5. Optimization Target

The assumptions and definitions above allow us to define the expected (generalized) ranking quality for the function $L \in \mathcal{R}_0$ with respect to $\xi \sim \mathcal{D}$ and model parameters $\theta \in \mathbb{R}^m$: $\mathcal{L}(\theta) := \mathbb{E}_{\xi \sim \mathcal{D}} L(f_\xi(\theta), \xi)$. Our ultimate goal is to find $\arg\min_\theta \mathcal{L}(\theta)$. However, since the distribution $\mathcal{D}$ is unknown, we have only i.i.d. samples $\xi_1, \dots, \xi_N$ as defined above. So, we consider the expected ranking quality under the empirical distribution $\mathcal{D}_N$:

$$\mathcal{L}_N(\theta) := \mathbb{E}_{\xi \sim \mathcal{D}_N} L(f_\xi(\theta), \xi) = \frac{1}{N} \sum_{i=1}^N L(f_{\xi_i}(\theta), \xi_i).$$

We want to optimize $\mathcal{L}(\theta)$ *globally* by optimizing $\mathcal{L}_N(\theta)$. This is possible because of the stability of global minimizers even for discontinuous functions: for $N \gg 1$ an almost minimizer of $\mathcal{L}_N(\theta)$ should be an almost minimizer of $\mathcal{L}(\theta)$ (Artstein & Wets, 1995).

Thus, we need to find a global minimizer of $\mathcal{L}_N$. Due to the discrete structure, we can ignore sets of zero Lebesgue measure. Recall that essential infimum ($\operatorname{ess\,inf}$) is infimum that ignores sets of zero measure and $\operatorname{int} U$ denotes an open interior of the set U.

Definition 3. *For any function $\mathcal{L}(\theta) : \mathbb{R}^m \to \mathbb{R}$ with $\mathcal{L}_* := \operatorname{ess\,inf}_{\theta \in \mathbb{R}^m} \mathcal{L}(\theta) > -\infty$, we define*

$$\arg\min \mathcal{L}(\theta) := \operatorname{int}\big\{\theta \in \mathbb{R}^m : \mathcal{L}(\theta) = \mathcal{L}_*\big\}.$$

We need this unusual definition because of the discrete structure of our loss: we want to exclude the breaking points from $\arg\min$. One can see that despite $L(\cdot, \cdot)$ satisfies Jumps regularity, the function $\mathcal{L}_N(\theta)$ does not have to.

[3] This property is assumed only to be consistent with the learning-to-rank literature and can be omitted.

[4] This condition can also be removed, but it simplifies the analysis of smoothing bias.

Statement 1. *The set* $\arg\min_{\theta\in\mathbb{R}^m} \mathcal{L}_N(\theta)$ *is not empty.*

The proof is straightforward (see Section A of the supplementary materials).

4. Stochastic Smoothing

4.1. Smoothing of Scores

The discrete structure of ranking loss functions prevents their effective optimization. Hence, some smoothing is needed and a natural approach for this is mollification (Ermoliev et al., 1995; Dolecki et al., 1983), i.e., adding randomness to parameters. We refer to Section B.1 of the supplementary materials for the formal definition and the reasons why this approach is not applicable in our case.

Thus, instead of acting on the level of parameters θ, we act on the level of scores z: $L^\pi_\xi(z,\sigma) := \mathbb{E}L(z+\sigma\varepsilon,\xi)$, where ε is a random variable with p.d.f. $\pi(z)$. We multiply the noise by σ to preserve Scalar-freeness in a sense that $L^\pi_\xi(\lambda z,\lambda\sigma) = L^\pi_\xi(z,\sigma)$ for any $\lambda > 0$.

In the linear case $f(\theta) = \Phi\,\theta$, if $\operatorname{rk}\Phi = n$, it is not hard to show the convergence of minimizers. However, in general, we cannot assume $\operatorname{rk}\Phi = n$. In particular, this property is violated in the presence of ties that *always* occur in gradient boosting due to the discrete nature of decision trees. As a result, there is a *smoothing bias* that alters the set of minimizers.

4.2. Simple Example of Smoothing Bias

Within this section, assume for simplicity that we are dealing with one function $L(z) := L(z,\xi) : \mathbb{R}^n \to \mathbb{R}$ for some arbitrary fixed n and $\xi \in \Xi_n$. Let $\Phi = \Phi_\xi \in \mathbb{R}^{m\times n}$ and $\mathcal{L}(\theta) := L(\Phi\,\theta)$. To clearly see how a *smoothing bias* can be introduced, consider the case when $\operatorname{im}(\Phi) \subset \mathbb{R}^n \setminus \cup_{i=1}^k U_i$, where U_i are from the Discreteness on subspaces assumption for $V = \mathbb{R}^n$. Denote by $c_1,\dots,c_k \in \mathbb{R}$ the values of $L(z)$ on the corresponding subsets U_i. Consider the functions $\mathcal{L}(\theta)$ and $\mathcal{L}^\pi(\theta) := \lim_{\sigma\to 0_+} \mathbb{E}_{\varepsilon\sim\pi} L(\Phi\theta+\sigma\varepsilon)$.

The value of $\mathcal{L}^\pi(\theta)$ is fully determined by π, $c_1,\dots,c_k$ and the subsets $U_1,\dots,U_k$ in the following way: $\mathcal{L}^\pi(\theta) = \sum_i \alpha_i c_i$ with

$$\alpha_i = \alpha_i(\pi,\theta,U_1,\dots,U_k) = \lim_{\sigma\to 0_+} \mathbb{P}(\Phi\,\theta+\sigma\varepsilon\in U_i).$$

In contrast, the value $\mathcal{L}(\theta)$ depends on the values $c_1,\dots,c_k$ much weaker: for fixed θ, consider the values $c'_1,\dots,c'_{k'}$ that correspond to U_i such that $\Phi\theta \in \overline{U_i}$, then the only limitation we have is $\min c'_i < \mathcal{L}(\theta) \le \max c'_i$ (this is required by Jumps regularity), which clearly allows more flexibility than the linear combination defined above.

In LTR, the issue of smoothing bias is connected to the problems of ties: the situations when $z_i = z_j$ and $r_i \neq r_j$.

4.3. Consistent Smoothing

Definition 4. *We say that the family of distributions* $\pi_\xi(z) : \coprod_{n>0}\mathbb{R}^n \times \Xi_n \to \mathbb{R}_+$ *is a* consistent smoothing *for* $L(z,\xi) \in \mathcal{R}_0$ *and for the model* f_ξ *iff for each* $n > 0$, $\xi \in \Xi_n$ *the following limit holds almost surely locally uniform in* θ*:*

$$L(f_\xi(\theta),\xi) = \lim_{\sigma\to 0_+} L^\pi_\xi(f_\xi(\theta),\sigma).$$

If π is smooth enough and consistent, then the function $\mathcal{L}^\pi_N(\theta,\sigma) := \frac{1}{N}\sum_{i=1}^N L^\pi_{\xi_i}(f_{\xi_i}(\theta),\sigma)$ is also smooth and almost surely locally uniformly approximates the discrete loss $\mathcal{L}_N(\theta)$ as $\sigma \to 0_+$.

To optimize ranking losses, it is important to find a consistent smoothing π for functions in $\mathcal{R}_1$. Fortunately, we can do this with an arbitrary precision by shifting the normal distribution by $-\mu r$ for large enough μ. Relevance monotonicity and s.u.s.c. imply the following pointwise limit:

$$\begin{aligned}&\lim_{\mu\to\infty}\lim_{\sigma\to 0_+} \mathbb{E}_{\varepsilon\sim\mathcal{N}(-\mu r, I_n)} L(z+\sigma\varepsilon,\xi)\\ &= \lim_{\mu\to\infty}\lim_{\sigma\to 0_+} \mathbb{E}_{\varepsilon\sim\mathcal{N}(\mathbb{0}_n, I_n)} L(z-\sigma\mu r+\sigma\varepsilon,\xi) = L(z,r)\,.\end{aligned}$$

This can be strengthened to the following theorem, which is proven in Section B.2 of the supplementary materials.

Theorem 1. $\pi_{\xi,\mu} = \mathcal{N}(-\mu r, I_n)$ *is a consistent smoothing for* $\mathcal{R}_1$ *as* $\mu\to\infty$*. Formally,* $\forall\theta$ *except zero measure* $\exists\,\delta > 0\ \forall\,\epsilon > 0\ \exists\,\mu > 0\ \exists\,\sigma_0 > 0$ *such that* $\forall\sigma \in (0,\sigma_0)$ *and* $\forall\theta' : \|\theta-\theta'\| < \delta$ *holds* $|L^\pi_\xi(f_\xi(\theta'),\sigma) - L(f_\xi(\theta'),\xi)| < \epsilon$.

By similar arguments, one can show that $\mathcal{N}(\mathbb{0}, I_n)$ is a consistent smoothing for $\mathcal{R}_1^{soft}$. Note that in both cases the consistent smoothing is *universal* for the entire class ($\mathcal{R}_1$ of $\mathcal{R}_1^{soft}$), i.e., it is independent from the choice of f_ξ.

Thus, LTR problems require non-trivial smoothing to preserve consistency. However, under some restrictive assumptions on the loss and on the model, any smoothing π is consistent.

Recall that $\mathcal{L}_N(\theta) = \frac{1}{N}\sum_{i=1}^N L(\Phi_{\xi_i}\theta,\xi_i)$ and assume that $L(z,\xi) \in \mathcal{R}_0$. The following theorem is proven in Section B.3 of the supplementary materials.

Theorem 2. *Consider open and convex subsets* $U'_{ij} := U_{ij} \cap \operatorname{im}\Phi_{\xi_i}$*. If* $\forall i\ \exists j$ *s.t.* $U'_{ij} \neq \emptyset$ *and* $\overline{\cup_j U'_{ij}} = \operatorname{im}\Phi_{\xi_i}$*, then any smoothing* π *is consistent for* $\mathcal{L}_N(\theta)$*.*

In early literature on LTR, all authors used such conditions implicitly by assuming that scores for all items are different. In contrast, we do not use this assumption as it never

holds in practice (e.g., when two documents have equal features). As a result, all existing LTR approaches suffer from a smoothing bias. In contrast, for the LSO problem, any smoothing is consistent, as we discuss in Section B.4 of the supplementary materials.

4.4. Scale-Free Acceleration

It is intuitively clear that for a scale-free function it is better to have a scalar-free approximation. However, for each $\lambda > 0$ we have $L_\xi^\pi(\lambda z, \sigma) = L_\xi^\pi(z, \lambda^{-1}\sigma)$, i.e., the smoothed function is no longer scale-free. To enforce scale-freeness, we take a vector z' with $\|z'\|_2 > 0$ and define

$$L_\xi^\pi(z, \sigma | z') := L_\xi^\pi\left(z, \frac{\|z\|_2}{\|z'\|_2}\sigma\right).$$

We refer to such smoothing as *Scale-Free Acceleration* (SFA). The obtained function is indeed scale-free: $L_\xi^\pi(\lambda z, \sigma | z') \equiv L_\xi^\pi(z, \sigma | z')$ for any $\lambda > 0$.

Let $\widehat{\sigma}(z) := \frac{\|z\|_2}{\|z'\|_2}\sigma$. In our optimization, we will be interested only in the case when $z' = z_t$ is the vector of scores obtained on t-th iteration of the optimization algorithm. So, we have $\widehat{\sigma}(z_t) = \sigma$ and SFA *does not* change the scale σ.

One can imagine a sphere of radius $R = \|z'\|_2$, where we restrict $L_\xi^\pi(z, \sigma)$ and homogenize it along the rays from the origin to infinity to obtain a scalar-free function.

4.5. Smoothing Properties

Finally, let us discuss regularity assumptions for smoothing on which our optimization method relies. Consider a family of distributions with p.d.f. $\pi_\xi(z)$ with $\xi \in \Xi_n$ for some $n > 0$, $z \in \mathbb{R}^n$. We require the following properties:

- **Continuous differentiability:** $\pi_\xi(z)$ is $C^{(1)}(\mathbb{R}^n)$, i.e., is differentiable with a continuous derivative.
- **Uniformly bounded derivative:** $\forall n \in \mathbb{N}$, $\forall \xi \in \Xi_n$ we have $\|\nabla_z \pi_\xi\|_2 = \mathcal{O}(1)$ uniformly in $z \in \mathbb{R}^n$.
- **Derivative decay:** $\forall n \in \mathbb{N}$ we have $\|\nabla_z \pi_\xi\|_2 = \mathcal{O}(\|z\|_2^{-n-2})$ as $\|z\|_2 \to \infty$.
- **Tractable conditional expectations:** conditional densities $\pi_\xi^j(z_j) := \pi_\xi(z_j | z_{\setminus j})$ are easy to compute.[5]

Clearly, $\mathcal{N}(-\mu r, I_N)$ satisfies these assumptions $\forall \mu \geq 0$.

[5] We do not use the log-derivative trick, so we do not care about the ability to compute $\frac{\mathrm{d}}{\mathrm{d}z_j}\pi_\xi^j(z_j)$ and $\frac{\mathrm{d}}{\mathrm{d}z_j}\log \pi_\xi^j(z_j)$, our gradient estimates require only computation of $\pi_\xi^j(z_j)$.

5. Coordinate Conditional Sampling

5.1. Gradient Estimate

In the previous section, we required the ability to easily compute $\pi_\xi^j(z_j) = \pi_\xi(z_j | z_{\setminus j})$. This property allows us to do the following trick: we decompose $\pi_\xi(z) = \pi_\xi^j(z_j)\pi_\xi^{\setminus j}(z_{\setminus j})$ with $\pi_\xi^{\setminus j}(z_{\setminus j})$ being the marginal distribution for $z_{\setminus j}$. Then, we can represent $L_\xi^\pi(z, \sigma) = L_\xi^\pi * \pi_\xi^j * \pi_\xi^{\setminus j}$. Note that the convolution is an associative operation that commutes with differentiation and, henceforth,

$$\frac{\partial}{\partial z_j} L_\xi^\pi(z, \sigma) = \left(\frac{\partial}{\partial z_j} L_\xi^\pi * \pi_\xi^j\right) * \pi_\xi^{\setminus j}.$$

Note that we differentiate by z_j the convolution by the same z_j. So, if we want to estimate the gradient unbiasedly, we need to sample $\varepsilon_{\setminus j} \sim \pi_\xi^{\setminus j}$ and then compute exactly $\left(\frac{\partial}{\partial z_j} L_\xi^\pi * \pi_\xi^j\right)\left((z_j, z_{\setminus j} + \sigma\varepsilon_{\setminus j})\right)$. The resulting estimate would be unbiased by construction. The following lemma suggests how to deal with $\frac{\partial}{\partial z_j} L_\xi^\pi * \pi_\xi^j$.

Lemma 1. *The function $l_j(z_j) := L((z_j, z_{\setminus j}), \xi) : \mathbb{R} \to \mathbb{R}$ for all z except zero measure has at most $k' \leq k(n, \mathbb{R}^n) - 1$ (k is from the Discreteness on subspaces assumption) breaking points $b_1, \ldots, b_{k'}$ (possibly depending on $z_{\setminus j}$ and ξ) and can be represented as:*

$$l_j(z_j) = \sum_{s=1}^{k'} \Delta l_j(b_s) \mathbb{1}_{\{z_j \leq b_s\}} + \mathrm{const}(z_{\setminus j}, \xi),$$

$$\Delta l_j(b_s) := \lim_{\epsilon \to 0_+} l_j(b_s + \epsilon) - l_j(b_s - \epsilon).$$

All results of this section are proven in Section C of the supplementary materials.

Based on the above lemma, we prove the following theorem.

Theorem 3. *The derivative $\frac{\partial}{\partial z_j} L_\xi^\pi(z, \sigma)$ is equal to:*

$$-\sigma^{-1} \cdot \mathbb{E}_{\varepsilon_{\setminus j} \sim \pi_\xi^{\setminus j}} \sum_{s=1}^{k'} \Delta l_j(b_s) \pi_\xi^j(\sigma^{-1}(b_s - z_j)),$$

where k' and $b_s = b_s(z_{\setminus j} + \sigma\varepsilon_{\setminus j})$ are from Lemma 1.

Corollary 1. *For LTR losses, the above formula becomes:*

$$\frac{\partial}{\partial z_j} L_\xi^\pi(z, \sigma) = -\sigma^{-1} \cdot$$
$$\cdot \mathbb{E}_{\varepsilon_{\setminus j} \sim \pi_\xi^{\setminus j}} \sum_{s=1}^{n} \Delta l_j(z_s + \sigma\varepsilon_s) \pi_\xi^j(\sigma^{-1}(z_s - z_j) + \varepsilon_s).$$

Uniform boundedness of Δl_j and π implies the following.

Statement 2. *The estimate is uniformly bounded by $\mathcal{O}(\sigma^{-1})$.*

Proceeding analogously with each coordinate $j \in \{1, \dots, n\}$, we obtain an unbiased estimate of $\nabla_z L^\pi_\xi(z, \sigma)$ that is uniformly bounded, in contrast to the classic estimate $\sigma^{-1}(L(z + \sigma\varepsilon) - L(z))\varepsilon$ (Nesterov & Spokoiny, 2017) obtained by the log-derivative trick for the normal distribution that is also known as REINFORCE (Williams, 1992). Uniform boundedness is crucial since without it we would not be able to claim global convergence. We call such estimate *Conditional Coordinate Sampling* (CCS) and denote it by $\widehat{\nabla}_{CC} L^\pi_\xi(z, \sigma)$.

Note that for each coordinate when estimating $\widehat{\nabla}_{CC} L^\pi_\xi(z, \sigma)$ we use the shared noise vector $\varepsilon \sim \pi_\xi$, i.e., the components of the gradient can have non-trivial covariation, but due to the uniform boundness the covariation is also uniformly bounded by $\mathcal{O}(\sigma^{-1})$.

Finally, let us discuss the complexity of computing $\widehat{\nabla}_{CC} L^\pi_\xi(z, \sigma)$. The following result follows from Section D of the supplementary materials.

Statement 3. *The estimate $\widehat{\nabla}_{CC} L^\pi_\xi(z, \sigma)$ can be computed in:*

- $\mathcal{O}((k + \log n)n)$ *operations and* $\mathcal{O}(n)$ *additional memory for* (N)DCG@k *and* ERR@k.
- $\mathcal{O}(n \log n)$ *operations and* $\mathcal{O}(1)$ *memory for* MRR.

5.2. SFA Gradient Estimate

It is not hard to generalize CCS to SFA. The following theorem holds.

Theorem 4. *For $\widehat{\sigma}(z) = \left(\frac{\|z\|_2}{\|z'\|_2}\right)\sigma$ at $z' = z$ we have:*

$$\nabla_z L^\pi(z, \widehat{\sigma}(z)) = \nabla_z L^\pi_\xi - \left\langle \nabla_z L^\pi_\xi, \frac{z}{\|z\|_2} \right\rangle_2 \frac{z}{\|z\|_2}.$$

Corollary 2. *Unbiased CCS estimate for SFA can be obtained by orthogonalizing $\widehat{\nabla}_{CC} L^\pi_\xi(z, \sigma)$ and z.*

Since orthogonalization reduces the norm of the estimate, it necessarily reduces the variance, so we obtain the following corollary.

Corollary 3. *SFA CCS estimate has a lower variance than the original CCS.*

The intuition for the orthogonalization is based on Scalar-freenees: the function $L(z, \xi)$ *does not change* along z direction, so this direction in the gradient $\nabla_z L^\pi_\xi$ does not contribute to $L(z, \xi)$ optimization.

As we need to deal with possibility of $z = z' = \mathbb{0}_n$, we introduce a parameter $\nu > 0$ and replace $\|z\|_2$ by $\|z\|_2 + \nu$:

$$\widehat{\nabla}_{CC} L^\pi_\xi(z, \sigma|z', \nu)\big|_{z'=z} := \widehat{\nabla}_{CC} L^\pi_\xi(z, \sigma) - \left\langle \widehat{\nabla}_{CC} L^\pi_\xi(z, \sigma), \frac{z}{\|z\|_2 + \nu} \right\rangle_2 \frac{z}{\|z\|_2 + \nu}.$$

Lemma 2. *Bias of SFA CCS estimate is uniformly bounded:*

$$\left|\mathbb{E}\widehat{\nabla}_{CC} L^\pi_\xi(z, \sigma_0|z', \nu) - \nabla_z L^\pi_\xi(z, \widehat{\sigma})\right| = \mathcal{O}\Big(\frac{1}{\|z\| + \nu}\Big).$$

As a consequence, if $\nu \to \infty$ or $\|z\| \to \infty$, then the estimate is asymptotically unbiased.

Thus, for the convergence analysis we consider only $\widehat{\nabla}_{CC} L^\pi(z, \sigma)$ since the estimate $\widehat{\nabla}_{CC} L^\pi(z, \sigma_0|z', \nu)$ can be made unbiased by varying the parameter $\nu > 0$. In practice, we consider $\widehat{\nabla}_{CC} L^\pi_\xi(z, \sigma_0|z', \nu)$ with fixed $\nu = 10^{-2}$ as we observed that this parameter performs well enough. Moreover, SFA can be seen as a bias–variance tradeoff controlled by $\nu > 0$ for CCS estimate of $\nabla_z L^\pi_\xi(z, \sigma)$. For practical comparison of $\widehat{\nabla}_{CC} L^\pi_\xi(z, \sigma)$ and $\widehat{\nabla}_{CC} L^\pi_\xi(z, \sigma_0|z', \nu)$ we refer to Section 7, where we show that SFA gives a significant improvement.

6. Global Optimization by Diffusion

6.1. SGLB

Previously, we discussed the importance of global optimization of $\mathcal{L}_N(\theta)$. As we show in this section, this can be achieved by global optimization of smoothed $\mathcal{L}^\pi_N(\theta, \sigma)$ with $\sigma = 1$ (if smoothing is consistent) using the recently proposed Stochastic Gradient Langevin Boosting (SGLB) (Ustimenko & Prokhorenkova, 2020). SGLB is easy to apply: essentially, each iteration of standard SGB is modified via model shrinkage and adding Gaussian noise to the gradients. However, the obtained algorithm is backed by strong theoretical results, see (Ustimenko & Prokhorenkova, 2020) for the details and the supplementary materials (Section E.1) for a brief sketch. The global convergence is implied by the fact that as the number of iterations grows, the stationary distribution $p_\beta(F)$ of the predictions $F = (f_{\xi_1}(\theta), \dots, f_{\xi_N}(\theta))$ concentrates around the global optima of the implicitly regularized loss

$$\mathcal{L}^\pi_N(F, \sigma, \gamma) = \mathcal{L}^\pi_N(F, \sigma) + \frac{\gamma}{2}\|\Gamma F\|_2^2,$$

where Γ is an implicitly defined regularization matrix. More formally, $p_\beta(F) \propto \exp(-\beta \mathcal{L}^\pi_N(F, \sigma, \gamma))$.

Global convergence of SGLB requires Lipschitz smoothness and continuity (Ustimenko & Prokhorenkova, 2020). We can ensure this for the entire $\mathcal{R}_0$, which allows us to claim the following theorem (see Section E.2 for the proof).

Theorem 5. *SGLB method applied to $\mathcal{L}^\pi_N(F, \sigma)$ converges* globally *to optima of $\mathcal{L}_N(F) \equiv \mathcal{L}_N(\theta)$ when used with CCS estimate.*

The following statement ensures that we can safely fix $\sigma = 1$ and fit only γ parameter without loosing any possible solution.

Statement 4. *$\mathbb{E}_{F\sim p_\beta}\mathcal{L}_N(F) = \mathbb{E}_{F'\sim p'_\beta}\mathcal{L}_N(F')$, where p_β corresponds to (σ,γ) and p'_β to $(1,\sigma^2\gamma)$.*

Proof. Due to Scalar-freeness, we can write $\mathcal{L}^\pi_N(F,\sigma) \equiv \mathcal{L}^\pi_N(\sigma^{-1}F, 1)$ and $\frac{\gamma}{2}\|\Gamma F\|_2 \equiv \frac{\sigma^2\gamma}{2}\|\Gamma\left(\sigma^{-1}F\right)\|_2^2$. Finally, due to Scalar-freeness, the change $F' = \sigma^{-1}F$ does not change the value of $\mathcal{L}_N(F) \equiv \mathcal{L}_N(F')$ and thus the expectation does not change. □

6.2. Generalization

Ustimenko & Prokhorenkova (2020) related the generalization gap with the uniform spectral gap parameter $\lambda_* \geq 0$ for the distribution $p_\beta(\theta) := \frac{\exp(-\beta\mathcal{L}_N(\theta,\sigma,\gamma))}{\int_{\mathbb{R}^m}\exp(-\beta\mathcal{L}_N(\theta,\sigma,\gamma))\mathrm{d}\theta}$ (see Raginsky et al. (2017) for the definition of a uniform spectral gap). Here $p_\beta(\theta)$ represents the limiting (as the learning rate goes to zero) distribution of the vector of parameters θ and is induced by the distribution $p_\beta(F) \propto \exp(-\beta\mathcal{L}(F,\sigma,\gamma))$ using the relation $F = \Phi\theta$. The following theorem is proven in the supplementary materials (Section E.3).

Theorem 6. *The generalization gap $\left|\mathbb{E}_{\theta\sim p_\beta(\theta)}\mathcal{L}^\pi(\theta,\sigma) - \mathbb{E}_{\theta\sim p_\beta(\theta)}\mathcal{L}^\pi_N(\theta,\sigma)\right|$ can be bounded by:*

$$\mathcal{O}\left(\left(\beta + 2d + \frac{d^2}{\beta}\right)\frac{\exp(\mathcal{O}(\frac{\beta}{\gamma\sigma^2}))}{\gamma N}\right).$$

7. Experiments

As baseline approaches, we consider the well-known LambdaMART framework optimized for NDCG@k (Wu et al., 2010), NDCG-Loss2++ from the LambdaLoss framework (Wang et al., 2018), and SoftRank (Taylor et al., 2008). We also apply the technique proposed by Bruch et al. (2020) to the baselines, the corresponding methods are called $\mathbb{E}\lambda$-MART and $\mathbb{E}\lambda$-Loss. Similarly to Wang et al. (2018), we set the parameter μ for NDCG-Loss2++ to be equal to 5. According to our experiments, NDCG-Loss2++ performed significantly better than NDCG-Loss2, which agrees with Wang et al. (2018).

7.1. Synthetic Data

Unfortunately, in practice, we cannot verify if we have reached the global optimum as we cannot evaluate all possible ensembles of trees. But having theoretical guarantees is important as it implies the stability of the algorithm and good generalization. In this section, we describe a simple synthetic test to verify whether StochasticRank can reach the global optimum.

The following dataset is multimodal (has several local optima) for NDCG@3: the number of queries is $N = 2$, first relevance vector is $r_1 = (3,2,1)$ and the second is $r_2 = (3,2)$. We consider the following features for the first query: $x_1 = (1,0,0), x_2 = (0,1,0), x_3 = (0,0,1)$ and for the second x_3 and x_1 (in the given order).

Table 1. Experimental results on synthetic data.

Method	NDCG@3
λ-MART	0.903
λ-Loss	0.903
$\mathbb{E}\lambda$-MART	0.903
$\mathbb{E}\lambda$-Loss	0.903
SoftRank	0.903
StochasticRank	**0.917**

We consider this simple synthetic dataset for two reasons: first, it clearly shows that ranking losses are likely to be multimodal; second, it allows us to demonstrate how multimodality prevents existing approaches from reaching the global optimum.

We limited the tree depth parameter to 3, so one tree can separate all documents with different features. We set the number of iterations to 1000, learning rate to 0.1, diffusion temperature to 10^3, and model-shrink-rate to 10^{-3}.

The results are shown in Table 1. We note that the maximum achievable NDCG@3 for this dataset is 0.917, i.e., StochasticRank successfully recovers the global optimum while all other approaches converge to a local optimum 0.903.

7.2. Real Data

Datasets For our experiments, we use the following publicly available datasets. First, we use the data from YAHOO! Learning to Rank Challenge (Chapelle & Chang, 2011): there are two datasets, each is pre-divided into training, validation, and testing parts. The other datasets are WEB10K and WEB30K released by Microsoft (Qin & Liu, 2013). Following Wang et al. (2018), we use Fold 1 for these two datasets.

Quality metrics The first metric we use is NDCG@5, which is very common in LTR research. The second one is MRR, which is a well-known click-based metric. Recall that MRR requires binary labels, so we binarize each label by $\widetilde{y}_i := \mathbb{1}_{\{y_i>0\}}$. Notably, while MRR is frequently used in online evaluations, it is much less studied compared to NDCG@k and there are no effective approaches designed for it. Fortunately, our method can be easily adapted to any ranking metric via a combination of SGLB with Coordinate Conditional Sampling smoothed by Gaussian noise.

Framework We implemented all approaches in CatBoost, which is an open-source gradient boosting library outperforming the most popular alternatives like XGBoost (Chen

Table 2. Experimental results.

Method	Dataset	NDCG@5	MRR
λ-MART	Yahoo Set 1	74.53	90.21
λ-Loss	Yahoo Set 1	74.73	-
$\mathbb{E}\lambda$-MART	Yahoo Set 1	74.57	90.30
$\mathbb{E}\lambda$-Loss	Yahoo Set 1	74.75	-
SoftRank	Yahoo Set 1	71.98	90.17
SR-$\mathcal{R}_1^{soft}$	Yahoo Set 1	74.68	**91.07**
SR-$\mathcal{R}_1$	Yahoo Set 1	**74.92**	90.97
λ-MART	Yahoo Set 2	73.87	91.48
λ-Loss	Yahoo Set 2	73.89	-
$\mathbb{E}\lambda$-MART	Yahoo Set 2	73.87	91.48
$\mathbb{E}\lambda$-Loss	Yahoo Set 2	73.91	-
SoftRank	Yahoo Set 2	73.91	92.16
SR-$\mathcal{R}_1^{soft}$	Yahoo Set 2	73.95	93.16
SR-$\mathcal{R}_1$	Yahoo Set 2	**74.15**	**93.56**
λ-MART	WEB10K	48.22	81.85
λ-Loss	WEB10K	48.33	-
$\mathbb{E}\lambda$-MART	WEB10K	48.29	81.72.
$\mathbb{E}\lambda$-Loss	WEB10K	48.47	-
SoftRank	WEB10K	42.82	81.38
SR-$\mathcal{R}_1^{soft}$	WEB10K	48.19	83.08
SR-$\mathcal{R}_1$	WEB10K	**48.53**	**83.30**
λ-MART	WEB30K	49.55	83.79
λ-Loss	WEB30K	49.45	-
$\mathbb{E}\lambda$-MART	WEB30K	49.49	83.79
$\mathbb{E}\lambda$-Loss	WEB30K	49.52	-
SoftRank	WEB30K	43.46	82.73
SR-$\mathcal{R}_1^{soft}$	WEB30K	**49.67**	**85.19**
SR-$\mathcal{R}_1$	WEB30K	49.59	85.01

& Guestrin, 2016) and LightGBM (Ke et al., 2017) for several tasks (Prokhorenkova et al., 2018). LambdaMART can be easily adapted for optimizing MRR, so we implemented both versions. In contrast, LambdaLoss is specifically designed for NDCG and cannot be easily modified for MRR. For SoftRank we used CCS to estimate gradients, since the original approach is computation and memory demanding, so it is infeasible in gradient boosting which requires *all* gradients to be estimated at each iteration.

Parameter tuning For all algorithms, we set the maximum number of trees to 1000. We tune the hyperparameters using 500 iterations of random search and select the best combination using the validation set, the details are given in the supplementary materials (Section F).

Table 3. Comparison of the algorithm's features on Yahoo Set 1, where π_μ means using unbiased smoothing.

Features	NDCG@5
REINFORCE	70.74
CCS	71.89
CCS+SFA	74.55
CCS+SFA+SGLB (SR-$\mathcal{R}_1^{soft}$)	74.68
CCS+SFA+SGLB+π_μ (SR-$\mathcal{R}_1$)	**74.92**

Results The results are shown in Table 2. One can see that StochasticRank (SR-$\mathcal{R}_1$) outperforms the baseline approaches on all datasets. In all cases, the difference with the closest baseline is statistically significant with a p-value < 0.05 measured by the paired one-tailed t-test. Also, in most cases, SR-$\mathcal{R}_1$ outperforms SR-$\mathcal{R}_1^{soft}$, which clearly demonstrates the advantage of unbiased smoothing, which takes into account the tie resolution policy.

The results in Table 2 are comparable to previously reported numbers, although they cannot be compared directly, since experimental setup (e.g., the maximum number of trees) is not fully described in many cases (Wang et al., 2018). More importantly, the previously reported results can be overvalued, since many openly available libraries compute ranking metrics using neither worst (as in our case) nor "expected" permutation, but some fixed arbitrary one depending on a particular implementation of the sorting operation.

To further understand how different techniques proposed in this paper affect the quality of the algorithm, we show the improvement obtained from each feature using the Yahoo dataset and the NDCG metrics (see Table 3). We see that CCS is significantly better than REINFORCE, while SFA gives an additional significant performance boost. SGLB and consistent smoothing further improve NDCG. We note that for both REINFORCE and CCS we use one sample per gradient estimate since the most time-consuming operation for both estimates is sorting (see Section D of the supplementary materials).

8. Conclusion

In this paper, we proposed the first truly direct LTR algorithm. We formally proved that this algorithm converges *globally* to the minimizer of the target loss function. This is possible due to the combination of three techniques: unbiased smoothing for consistency between the original and smoothed losses; SGLB for global optimization via gradient boosting; and CCS gradient estimate with uniformly bounded error and low variance, which is required for SGLB to be applied. Our experiments clearly illustrate that the new algorithm outperforms state-of-the-art LTR methods.

References

Artstein, Z. and Wets, R. Consistency of minimizers and the SLLN for stochastic programs. *Journal of Convex Analysis*, 2(1-2):1–17, 1995.

Bruch, S., Han, S., Bendersky, M., and Najork, M. A stochastic treatment of learning to rank scoring functions. In *Proceedings of the 13th ACM International Conference on Web Search and Data Mining (WSDM 2020)*, pp. 61–69, 2020.

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

Cao, Y., Xu, J., Liu, T.-Y., Li, H., Huang, Y., and Hon, H.-W. Adapting ranking SVM to document retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 186–193. ACM, 2006.

CatBoost. Ranking: objectives and metrics. https://catboost.ai/docs/concepts/loss-functions-ranking.html, 2020.

Chapelle, O. and Chang, Y. Yahoo! learning to rank challenge overview. In *Proceedings of the learning to rank challenge*, pp. 1–24, 2011.

Chen, T. and Guestrin, C. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

Dolecki, S., Salinetti, G., and Wets, R. J.-B. Convergence of functions: equi-semicontinuity. *Transactions of the American Mathematical Society*, 276(1):409–429, 1983.

Erdogdu, M. A., Mackey, L., and Shamir, O. Global nonconvex optimization with discretized diffusions. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, pp. 9694–9703, 2018.

Ermoliev, Y., Norkin, V., and Wets, R. The minimization of semicontinuous functions: mollifier subgradients. *SIAM Journal on Control and Optimization*, 33, 01 1995.

Gelfand, S. B., Doerschuk, P. C., and Nahhas-Mohandes, M. Theory and application of annealing algorithms for continuous optimization. In *Proceedings of the 24th conference on Winter simulation*, pp. 494–499, 1992.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*, pp. 3146–3154, 2017.

Kustarev, A., Ustinovsky, Y., Logachev, Y., Grechnikov, E., Segalovich, I., and Serdyukov, P. Smoothing NDCG metrics using tied scores. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 2053–2056, 2011.

Liu, T.-Y. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, 2009.

Nesterov, Y. and Spokoiny, V. G. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17:527–566, 2017.

Nguyen, T. and Sanner, S. Algorithms for direct 0–1 loss optimization in binary classification. In *International Conference on Machine Learning*, pp. 1085–1093, 2013.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. Catboost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, pp. 6638–6648, 2018.

Qin, T. and Liu, T. Introducing LETOR 4.0 datasets. *CoRR*, abs/1306.2597, 2013.

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

Raginsky, M., Rakhlin, A., and Telgarsky, M. Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. *CoRR*, abs/1702.03849, 2017.

Rudin, C. and Wang, Y. Direct learning to rank and rerank. In *International Conference on Artificial Intelligence and Statistics*, pp. 775–783, 2018.

Sakai, T. Metrics, statistics, tests. In *Bridging Between Information Retrieval and Databases - PROMISE Winter School 2013, Bressanone, Italy, February 4-8, 2013. Revised Tutorial Lectures*, pp. 116–163, 2013.

Tan, M., Xia, T., Guo, L., and Wang, S. 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, 2013.

Taylor, M., Guiver, J., Robertson, S., and Minka, T. SoftRank: Optimizing non-smooth rank metrics. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pp. 77–86, 2008. ISBN 978-1-59593-927-2.

Ustimenko, A. and Prokhorenkova, L. SGLB: Stochastic Gradient Langevin Boosting. *arXiv e-prints*, art. arXiv:2001.07248, 2020.

Volkovs, M. N. and Zemel, R. S. BoltzRank: learning to maximize expected ranking gain. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1089–1096, 2009.

Vorobev, A., Ustimenko, A., Gusev, G., and Serdyukov, P. Learning to select for a predefined ranking. In *International Conference on Machine Learning*, pp. 6477–6486, 2019.

Wang, X., Li, C., Golbandi, N., Bendersky, M., and Najork, M. The lambdaloss framework for ranking metric optimization. In *Proceedings of The 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*, pp. 1313–1322, 2018.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Wu, Q., Burges, C. J., Svore, K. M., and Gao, J. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270, 2010.

Xu, J., Li, H., Liu, T.-Y., Peng, Y., Lu, M., and Ma, W.-Y. Direct optimization of evaluation measures in learning to rank. 2010.

Yin, D., Hu, Y., Tang, J., Daly, T., Zhou, M., Ouyang, H., Chen, J., Kang, C., Deng, H., Nobata, C., Langlois, J.-M., and Chang, Y. Ranking relevance in Yahoo search. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 323–332, 2016.