Selective Sampling with Almost Optimal Guarantees for Learning to Rank from Pairwise Preferences

Nir Ailon
Department of Computer Science
Technion - Israel Institute of Technology
nailon@cs.technion.ac.il

Ron Begleiter
Department of Computer Science
Technion - Israel Institute of Technology
ronbeg@cs.technion.ac.il

Esther Ezra
NYU Courant Institute of Mathematical Science
esther@cims.nyu.edu

Abstract

One of the practical obstacles of *learning to rank from pairwise preference labels* is in its (apparent) quadric sample complexity. Some heuristics have been tested for overriding this obstacle. In this workshop we will present new provable method for reducing this sample-complexity, almost reaching the informational lower bound, while suffering only negligible sacrifice of optimality. Our main results: (1) We define a novel structural property of function spaces endowed with a metric loss function that allows searching at an exponential rate. (2) We demonstrate that this structural property applies to the search space arising in the problem of learning to rank from pairwise preferences. As a result, we show that $O(n\,\mathrm{poly}(\log n, \varepsilon^{-1}))$ adaptively sampled preferences suffice in order to obtain a ranking of $\varepsilon \cdot OPT$ regret, where OPT is loss of the optimal ranking. This is near-optimal in terms of *information theory*. (3) We show how our algorithm can be used to construct a query-efficient version of the famous SVM-rank relaxation, when the set of points are endowed with feature vectors and we are searching in the restricted space of linear permutations. (4) For feature spaces with a fixed dimension, we show an additional slight improvement in the query complexity using advanced tools from computational geometry.

1 Introduction

Ranking items according to their comparison in pairs dates back to the classical work of Thurstone [12]. We focus on the combinatorial aspect of this problem, studying *learning to rank with pairwise preferences*, a close relative of *Minimum Feedback Arc-Set in Tournaments* (MFAST) from the world of combinatorial optimization.

In MFAST, the goal is to find a full linear order of V given all quadratically many pairwise comparisons of elements. It turns that MFAST is NP-hard [4], though Kenyon-Mathieu and Schudy [10] show a PTAS for this problem.

From a learning theoretical point of view, the need to acquire quadratically many pairwise preferences is unacceptable even for moderately large sets V which arise in applications. On the other hand, very simple examples demonstrate that uniform subsampling of pairs works poorly. In this work we investigate an active sampling method instead.

Very recently Ailon [1] has shown a query efficient version of Kenyon-Mathieu and Schudy's work, with comparable (though slightly worse) bounds compared to our work, which will be presented in the workshop. His algorithm is limited in the following significant way: It cannot be used for searching in restricted hypothesis spaces. This is a significant drawback, because practitioners often wish to rank data using a restricted set of permutations, e.g., those induced by the space of linear score functions applied to an embedding of the set V in some real vector (feature) space.

Our proposed method is not only amenable to this practical setting, but it also lends itself to yet an additional improvement in query complexity using powerful techniques known as ε-relative approximations [11, 7]. Other important recent work on approximating the minimum feedback arc-set problem, much of which inspired this work (yet none comparable with it) can be found., e.g., in [2, 5, 6].

Our results are implied by a general active learning algorithm, which is interesting in its own right and may find other applications. A working version of this work can be found in [3].

2 A Powerful Structural Property: Query Efficient ε-Smooth Approximations of Relative Regret Functions

Consider the following general setting. We are given a finite space $\mathcal{X}$ endowed with a measure μ.[1] Let $\mathcal{F}$ denote some class of functions from $\mathcal{X}$ to $\mathbb{R}$, and $h : \mathcal{X} \to \mathbb{R}$ some function not necessarily in $\mathcal{F}$. Let $\mathbf{d}(f, g)$ be a pseudo-metric over pairs of functions $f, g \in F \cup \{h\}$, defined as $\mathbf{d}(f, g) = \int_{\mathcal{X}} L(f(x), g(x))d\mu$, where L is a pseudodometric on reals. For each $f \in \mathcal{F}$ we say that $\mathbf{d}(f, h)$ is the *cost* of f. We seek a function $f^* = \mathrm{argmin}_{f\in\mathcal{F}}\, \mathbf{d}(f, h)$.

We are interested in the particular case where h is unknown, but we have access to an h-oracle returning a value $h(x)$ of a chosen $x \in \mathcal{X}$ per unit cost. The ε-query complexity of the problem is defined as the number of queries (h-oracle calls) required in order to find a function $f \in \mathcal{F}$ satisfying $\mathbf{d}(f, h) \leq (1 + \varepsilon)\mathbf{d}(f^*, h)$.

Definition 2.1. Let f be an arbitrary function in $\mathcal{F}$. Define the *f-relative regret* function $\Delta_f : \mathcal{F} \mapsto \mathbb{R}$ as:

$$\Delta_f(g) = \mathbf{d}(g, h) - \mathbf{d}(f, h) \,.$$

For a nonnegative parameter $\varepsilon < 1$ we say that a function $\widehat{\Delta}_f : \mathcal{F} \mapsto \mathbb{R}$ is an ε-smooth approximation of Δ_f if for all $g \in \mathcal{F}$,

$$|\widehat{\Delta}_f(g) - \Delta_f(g)| \leq \varepsilon\mathbf{d}(f, g). \tag{2.1}$$

This definition suggests the iterative search over $\mathcal{F}$, depicted in Algorithm 2.1. It finds a function $\hat{f} \in \mathcal{F}$ satisfying $\mathbf{d}(\hat{f}, h) \leq \big(1 + O(\epsilon)\big)\mathbf{d}(f^*, h)$, as long as we can optimize over ε-smooth approximations of f-relative regret functions Δ_f. The choice of the number of iterations k can be derived from Theorem 2.2 below, showing that the excess cost above a $(1 + O(\varepsilon))\mathbf{d}(f^*, h)$ target decays exponentially.

Algorithm 2.1 Finding an $\big(1 + O(\epsilon)\big)$-competitive estimation

---

Input: an initial solution $\hat{f}_0 \in \mathcal{F}$, an estimation parameter $\epsilon \in (0, 1/5)$, and number of iterations k

  $i \leftarrow 0$
  repeat
    $\hat{f}_{i+1} \leftarrow \mathrm{argmin}_{g\in\mathcal{F}},\widehat{\Delta}_{\hat{f}_i}(g)$, where $\widehat{\Delta}_{\hat{f}_i}(g)$ is an ε-smooth approximation of $\Delta_{\hat{f}_i}$
    $i \leftarrow i + 1$
  until i equals k
  return $\hat{f}_k$

Theorem 2.2. *Let $\varepsilon \in (0, 1/2)$. Let OPT denote $\min_{f\in\mathcal{F}} \mathbf{d}(f, h)$, and $\hat{f}_0$ be an arbitrary function. Then the following holds for $\hat{f}_i$ obtained in Algorithm 2.1 for any $1 \leq i \leq k$:*

$$\mathbf{d}(\hat{f}_i, h) \leq (1 + O(\epsilon))\, \mathit{OPT} + O(\epsilon^i)\mathbf{d}(\hat{f}_0, h). \tag{2.2}$$

[1]We will either consider a probability measure or a counting measure in what follows.

The goal is now to construct an ε-smooth approximation $\widehat{\Delta}_{\hat{f}}$ of a relative regret function $\Delta_{\hat{f}}$ for a given $\hat{f}$ using few queries into h. In the next section, we show how this can be done for the problem of ranking from preferences.

3 ε-Smooth Approximations for Ranking

We consider the following problem of ranking from preference labels. We are given a ground set V of size n. The space $\mathcal{X}$ consists of all pairs $(u,v) \in V \times V$. The space $\mathcal{F}$ consists of permutations on V, viewed as functions from $\mathcal{X}$ to $\{0,1\}$ satisfying pairwise consistency $\pi((u,v)) = 1 - \pi((v,u))$ and transitivity: $\pi((u,w)) \leq \pi((u,v)) + \pi((v,w))$. The unknown input function h is assumed to be pairwise consistent, but not necessarily transitive.[2] The pseudometric $\mathbf{d}$ is defined as $\mathbf{d}(\pi,\sigma) = \frac{1}{2}\sum_{u,v\in V} \mathbf{1}_{\pi((u,v))\neq\sigma((u,v))}$. If $\pi,\sigma \in \mathcal{F}$ then this is known as the Kendall-τ metric. Minimizing $\mathbf{d}(\cdot,h)$ over $\mathcal{F}$ given h in its entirety is known as the NP-Hard problem of minimum feedback arc-set in tournaments in combinatorial optimization.

Consider the following idea for creating an ε-smooth approximation of the relative regret function Δ_π, for a fixed $\pi \in \mathcal{F}$. Start by creating a random sample $S \subseteq \mathcal{X}$ of $O(n\,\mathrm{poly}(\log n, \varepsilon^{-1}))$ pairs of elements in V. The sample should not be selected uniformly. Instead, we use π to create a bias. For each pair of distinct elements u,v, we will independently include (u,v) in S with probability $p_\pi(u,v) := \min\{1, \mathrm{poly}(\log n, \varepsilon^{-1})/\delta_\pi(u,v)\}$, where $\delta_\pi(u,v)$ is the distance between the location of u and v in the ordering induced by π. For example, if u is adjacent to v then $\delta_\pi(u,v)=1$, and if u and v are first and last (respectively), then $\delta_\pi(u,v) = n-1$.[3]

For distinct $u,v \in V$ and a permutation σ, let $C_{u,v}(\sigma)$ denote the contribution of the pair u,v to $\mathbf{d}(\sigma,h)$, namely: $C_{u,v}(\sigma) := \mathbf{1}_{\sigma(u,v)\neq h(u,v)}$. (Note that clearly $C_{u,v} \equiv C_{v,u}$.) Our estimator $\widehat{\Delta}_\pi(\sigma)$ can be finally defined as follows:

$$\widehat{\Delta}_\pi(\sigma) = \frac{1}{2}\sum_{u,v\in V} p_\pi(u,v)^{-1}\big(C_{u,v}(\sigma) - C_{u,v}(\pi)\big)\ . \tag{3.1}$$

Theorem 3.1. *For any permutation $\pi \in \Pi$ the function $\widehat{\Delta}_\pi : \mathcal{F} \to \mathbb{R}$ defined in (3.1) is, with high probability (at least $1 - n^\alpha$ for some $\alpha > 0$) an ε-smooth approximation of the relative regret function Δ_π. Additionally, the number of h-queries required in order to construct $\widehat{\Delta}_\pi$ is $O(n\,\mathrm{poly}(\log n, \varepsilon^{-1}))$ with high probability.*

Using our construction (3.1) and Algorithm 2.1 and the fact that $\mathbf{d}(\cdot,\cdot)$ is both integer and bounded by n^2, we conclude from Theorem 2.2:

Corollary 3.2. *There exists an active learning algorithm for the problem of minimizing $\mathbf{d}(\cdot,h)$ over the set $\mathcal{F}$ of permutations to within an $(1+O(\varepsilon))$ approximation with a total query complexity of $O(n\,\mathrm{poly}(\log n, \varepsilon^{-1}))$, achieved by executing $O(\log n)$ iterations of Algorithm 2.1.*

4 Convex relaxations

Instead of optimizing $\mathbf{d}(f,h)$ over $f \in \mathcal{X}$, assume we are interested in optimizing $\tilde{\mathbf{d}}(f,h)$, where

$$\tilde{\mathbf{d}}(f,h) = \int_{\mathcal{X}} \tilde{L}(f(x),h(x))d\mu$$

and $\tilde{L}$ is such that for all a,b:

$$L(a,b) \leq \tilde{L}(a,b)\ . \tag{4.1}$$

Slightly abusing notation, we redefine for all $f \in \mathcal{F}$ the relative regret function $\Delta_f : \mathcal{F} \to \mathbb{R}$ to be

$$\Delta_f(g) = \tilde{\mathbf{d}}(g,h) - \tilde{\mathbf{d}}(f,h)\ .$$

Consider again Algorithm 2.1 with respect to this definition of Δ_f. Theorem 2.2 is now replaced with the following:

[2] Note that now, greek letters eg π,σ denote functions in $\mathcal{F}$, and not f,g... as before.

[3] We omit the exact polynomial $\mathrm{poly}(\log n, \varepsilon^{-1})$ from this workshop abstract for simplicity.

Theorem 4.1. *Assume Equation (2.1) holds for some $\varepsilon \in (0, 1/2)$. Let $\texttt{OPT}$ denote the minimal possible cost $\tilde{\mathbf{d}}(f, h)$ over $f \in \mathcal{F}$. Then the following holds for the estimation $\hat{f}_i$ computed in the i'th iteration of Algorithm 2.1, for $1 \le i \le k$:*

$$\tilde{\mathbf{d}}(\hat{f}_i, h) \le (1 + O(\varepsilon))\,\texttt{OPT} + O(\varepsilon^i)\tilde{L}(\hat{f}_0, h)\ . \tag{4.2}$$

4.1 The SVM hinge-loss convex relaxation

The following is a typical setting in learning to rank applciations [see, e.g., 8, 9], which serves as an example for the usefulness of the construction in the previous discussion. Consider the ranking problem in Section 3 where now each point $u \in V$ is endowed with a (feature) vector $\mathbf{u} \in \mathbb{R}^d$.

The set of feasible solutions will now be the subset of permutations induced by linear functionals. Each possible vector $\mathbf{w} \in \mathbb{R}^d$ defines a corresponding permutation $\pi_{\mathbf{w}}$ over V by

$$u \prec_{\pi_{\mathbf{w}}} v \iff f_{\mathbf{w}}(\mathbf{u}, \mathbf{v}) \triangleq \langle \mathbf{w}, \mathbf{v} - \mathbf{u} \rangle \le 0\ .$$

Intuitively, u "beats" v if its utility $\langle \mathbf{w}, \mathbf{u} \rangle$ is higher than the utility $\langle \mathbf{w}, \mathbf{u} \rangle$ of v. The goal is to fit a utility function to the (noisy) information h.

Our search space $\mathcal{F}$ will now be $\{f_{\mathbf{w}} \in \mathbb{R}^d : \|\mathbf{w}\|_2 \le C_{\text{reg}}\}$ for some constant $C_{\text{reg}} > 0$.[4] The function $L(x, y)$ is taken to be

$$L(x, y) = \frac{1}{2}\mathbf{1}_{\text{sgn}(x) \ne \text{sgn}(y)}\ .$$

The metric $\mathbf{d}$ over $\mathcal{F}$ becomes exactly the Kendall-τ distance applied to the corresponding two permutations induced by its arguments.

The relaxed function $\tilde{L}$ is taken to be $\tilde{L}(x, y) = \max\{0, 1 - xy\}$, known as the *hinge loss* function. It is not hard to see that with these definitions, (4.1) holds. Theorem 4.1 is now applicable.

Finally, it can be easily shown that a sampling scheme similar to the one presented in Section 3 ensures that we can compute an ε-smooth approximation of $\Delta_{f_{\mathbf{w}}}$ for any $f_{\mathbf{w}} \in \mathcal{F}$ using $O(n\,\text{poly}(\log n, \varepsilon^{-1}))$ h-queries. The net result is as follows: The problem of minimizing $\mathbf{d}(\cdot, h)$ over $\mathcal{F}$ is a certain regularized SVM, and in fact, the well known rank-SVM. Minimizing the ε-smooth approximation $\widehat{\Delta}_{f_{\mathbf{w}}}(\cdot)$ that we construct is, equivalently, a *subsampled* SVM, and as such, can be solved in polynomial time. Algorithm 2.1 converges to a $(1 + O(\varepsilon))$-approximate solution to rank-SVM, where in each iteration we solve a subsampled SVM. The convergence rate is exponentially fast.

5 ε-Smooth Approximations for Ranking: The Fixed Dimensional Geometric Version

If time permits, we will show how to slightly strengthen the main result Theorem 3.1 of Section 3 in the case of permutations induced by linear functions (as considered in Section 4.1), when the feature space dimension d is considered to be constant. This requires an interesting application of a notion of ε-relative approximations in bounded VC dimension spaces [7, 11].

References

[1] Nir Ailon. An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity. In *NIPS (to appear)*, 2011.

[2] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM*, 55(5):23:1–23:27, October 2008.

[3] Nir Ailon, Ron Begleiter, and Esther Ezra. A new active learning scheme with applications to learning to rank from pairwise preferences. *arxiv:1110.2136*, 2011.

[4] Noga Alon. Ranking tournaments. *SIAM Journal on Discrete Mathematics*, 20, 2006.

[4] This constant serves the role of a *regularizer* and will affect the constants hiding in the $O-$ notation of the sample complexity of computing ε-smooth approximations.

[5] Mark Braverman and Elchanan Mossel. Noisy sorting without resampling. In *SODA*, pages 268–276, 2008.

[6] Don Coppersmith, Lisa K. Fleischer, and Atri Rurda. Ordering by weighted number of wins gives a good ranking for weighted tournaments. *ACM Trans. Algorithms*, 6:55:1–55:13, July 2010.

[7] Sariel Har-Peled and Micha Sharir. Relative (ε, p)-approximations in geometry. *Discrete & Computational Geometry*, 45(3):462–496, 2011.

[8] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Large margin ranking boundaries for ordinal regression. In *Advances in Large Margin Classifiers*, chapter 7, pages 115–132. 2000.

[9] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-02)*, 2002.

[10] Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, STOC '07, pages 95–103, 2007.

[11] Yi Li, Philip M. Long, and Aravind Srinivasan. Improved bounds on the sample complexity of learning. *Journal of Computer and System Sciences*, 62:2001, 2000.

[12] Louis Leon Thurstone. A law of comparative judgment. *Psychological Review*, 34(4):273–286, July 1927.