Bandit Online Optimization over the Permutahedron

Nir Ailon\(^1\), Kohei Hatano\(^2\), and Eiji Takimoto\(^2\)

\(^1\) Department of Computer Science, Technion
\(^2\) Department of Informatics, Kyushu University

nailon@cs.technion.ac.il, \{hatano,eiji\}@inf.kyushu-u.ac.jp

Abstract. The permutahedron is the convex polytope with vertex set consisting of the vectors \((\pi(1), \ldots, \pi(n))\) for all permutations (bijective functions) \(\pi\) over \(\{1, \ldots, n\}\). We study a bandit game in which, at each step \(t\), an adversary chooses a hidden weight vector \(s_t\), a player chooses a vertex \(\pi_t\) of the permutahedron and suffers an observed instantaneous loss of \(\sum_{i=1}^{n} \pi_t(i) s_t(i)\).

We study the problem in two regimes. In the first regime, \(s_t\) is a point in the polytope dual to the permutahedron. Algorithm CombBand of Cesa-Bianchi et al. (2009) guarantees a regret of \(O(n\sqrt{T\log n})\) after \(T\) steps. Unfortunately, CombBand requires at each step an \(n\)-by-\(n\) matrix permanent computation, a \#P-hard problem. Approximating the permanent is possible in the impractical running time of \(O(n^{10})\), with an additional heavy inverse-polynomial dependence on the sought accuracy. We provide an algorithm of slightly worse regret \(O(n^{3/2}\sqrt{T})\) but with more realistic time complexity \(O(n^3)\) per step. The technical contribution is a bound on the variance of the Plackett-Luce noisy sorting process’s ‘pseudo loss’, obtained by establishing positive semi-definiteness of a family of 3-by-3 matrices of rational functions in exponents of 3 parameters.

In the second regime, \(s_t\) is in the hypercube. For this case we present and analyze an algorithm based on Bubeck et al.’s (2012) OSMD approach with a novel projection and decomposition technique for the permutahedron. The algorithm is efficient and achieves a regret of \(O(n\sqrt{T})\), but for a more restricted space of possible loss vectors.

1 Introduction

Consider a game in which, at each step, a player plays a permutation of some ground set \(V = \{1, \ldots, n\}\), and then suffers (and observes) a loss. We model the loss as a sum over the items of some latent quality of the item, weighted by its position in the permutation. The game is repeated, and the items’ quality can adversarially change over time. The game models many scenarios in which the player is an online system (say, a search/recommendation engine) presenting a ranked list of items (results/products) to a stream of users. A user’s experience is positive if she perceives the quality of the top items on the list as higher.
than those at the bottom. The goal of the system is to create a total positive experience for its users.

There is a myriad of methods for modelling ranking loss functions in the literature, especially (but not exclusively) for information retrieval. Our choice allows us to study the problem in the framework of online combinatorial optimization in the bandit setting, and to obtain highly nontrivial results improving on state of the art in either run time or regret bounds. More formally, we study online linear optimization over the the \( n \)-permutahedron action set, defined as the convex closure of all vectors in \( \mathbb{R}^n \) consisting of \( n \) distinct coordinates taking values in \( [n] := \{1, \ldots, n\} \) (permutations). At each step \( t = 1, \ldots, T \), the player outputs an action \( \pi_t \) and suffers a loss \( \pi'_t s_t = \sum_{i=1}^{n} \pi_t(i) s_t(i) \), where \( s_t \in \mathbb{R}^n \) is the vector of “item qualities” chosen by some adversary who knows the player’s strategy but doesn’t control their random coins. The performance of the player is the difference between their total loss and that of the optimal static player, who plays the best (in hindsight) single permutation \( \pi^* \) throughout. This difference is known as regret. Note that, given \( s_1, \ldots, s_T \), \( \pi^* \) can be computed by sorting the coordinates of \( \sum_{t=1}^{T} s_t \) in decreasing order. This is aligned with our practical requirement that items with higher quality should be placed first, and those with lower quality should be last.

2 Results, Techniques and Contribution

Our first of two results, stated as Theorem 1, is for the setting in which at each step the loss is uniformly bounded (by 1 for simplicity) in absolute value for all possible permutations. Equivalently, the vectors \( s_t \) belong to the polytope that is dual to the permutahedron. Our algorithm, BanditRank, plays permutations from a distribution known as the Plackett-Luce model (see [13]) which is widely used in statistics and econometrics (see eg [4]). It uses an inverse covariance matrix of the distribution in order to obtain an unbiased loss vector estimator, which is a standard technique [7]. The main technical difficulty (Lemma 2) is in bounding second moment properties of Plackett-Luce, by establishing positive semidefiniteness of a certain family of 3 by 3 matrices. The lemma is interesting in its own right as a tool for studying distributions over permutations. The expected regret of our algorithm is \( \mathcal{O}(n^{3/2} \sqrt{T}) \) for \( T \) steps, with running time of \( \mathcal{O}(n^3) \) per time step. This result should be compared to CombBand of [7], where a framework for playing bandit games over combinatorially structured sets was developed. Their techniques extend that of [8]. In each step, it draws a permutation from a distribution that assigns to each permutation \( \pi \) a probability of \( e^\eta \sum_{\tau=1}^{n-1} \pi'_t \tilde{s}_t \), where \( \tilde{s}_t \) is a pseudo-loss vector at time \( t \), an unbiased estimator of the loss vector \( s_t \). Their algorithm guarantees a regret of \( \mathcal{O}(n \sqrt{T \log n}) \), which is better than ours by a factor of \( \Theta(\sqrt{n/ \log n}) \). However, its computational requirements are much worse. In order to draw permutations, they need to compute nonnegative \( n \) by \( n \) matrix permanents. Unfortunately, nonnegative permanent computation is \#P-hard, as shown by [15]. On the other hand, a groundbreaking result of [12] presents a polynomial time approximation scheme