Online Learning of Permutations
Using Extended Formulation

Holakou Rahmanian  David P. Helmbold  S.V.N. Vishwanathan

The Prediction Game

Fix $n$ jobs to be processed by a processor. Let $n = 4$ in each trial $t = 1 \ldots T$.

1. The learner randomly predicts with a permutation $\sigma^{t-1} = [3, 2, 4, 1]$
2. The adversary reveals the processing time for all the jobs $t^t \in [0,1]^n$
3. The learner incurs a loss of $\ell^t = [3, 6, \ldots, 1, 8]$
   \[
   \sigma^{t-1} - \ell^t = 3 \cdot 3 + 2 \cdot 2 + 4 \cdot 1 + 1 \cdot 8 = 5.3
   \]

Challenge 1
There are too many permutations! Hard to apply algorithms like Hedge and Randomized Weighted Majority

Challenge 2
The permutahedron $\mathcal{V} := \text{conv}(S_n)$ has too many facets in $\mathbb{R}^n$! Hard to apply algorithms like Component Hedge

Extended Formulation

Constructing Extended Formulations Using Reflection Relations

Initialize the polytope to a corner ($P_i$), and pass the polytope through a sequence of partial reflections.

Extended Formulation

Sorting Networks as Sequence of Reflection Relations

The XF-Perm Algorithm

A representation $w = (v, x, s)$ with 3 parts:

1. Original components $v$
   Goal: Incorporate a natural loss $\ell$
2. Extended formulation $x$
   Goal: Efficiently describe the polytope
3. Slack variables $s$
   Goal: Facilitate prediction and projection

$w$ will be in a polytope with a small number of facets:

**The Augmented Formulation Space $W$**

$w = (v, x, s) \in \mathbb{R}^{n+2m}$ s.t.

1. $Ax + s = b$ (m facets)
2. $v = Mx + \sigma_M$ (n facets)

We sample with the same expectation as $v$:

**Prediction – Fast Prediction**
Sample $\sigma$ from a distribution $D$ over permutations $S_n$, which has efficient sampling and $E[D[\sigma]] = v$

Multiplicative update for components of $v$:

**Update**

For all $i \in \{1, \ldots, n\}$,

$$\tilde{v}_i^{t-1} = v_i^t \exp(-\eta t^t)$$

Going back to the polytope with relative entropy projection:

**Projection – Iterative Bregman Projections**

Goal: Find $w^* := \arg \min_{w \in W} \Delta(w||\nu^{t-1})$.

Solution: Project onto the facet associated with a particular constraint and then repeatedly cycle through the $m + n$ constraints

Fast Prediction with Distribution $D$

Sampling from $D$ Run the sorting network backwards with “probabilistic swaps”, i.e.

1. Start with $\sigma = [1, 2, \ldots, n]$
2. For the $k$th comparator $k \in \{1 \ldots m\}$
   - Swap the values of the wires w.p. $(x_k + s_k)$

Direct prediction in $O(m)$ time without decomposition

The Initialization Trick and Regret Bound

With proper tuning of $\eta$:

$$E[L_{\text{XF-Perm}}] - L_{\text{best}} \leq \sqrt{2L_{\text{best}} \Delta(w^{0})||w^{0}) + \Delta(w^{0})||w^{0})}$$

Initialization Idea: Start with an infeasible point with good distance properties, then project it onto the polytope

Using AKS sorting networks with $m = O(n \log n)$ comparators and initializing $w^{0} = \arg \min \Delta(w||\nu^n)$:

$$E[L_{\text{XF-Perm}}] - L_{\text{best}} \leq O((n)^{2}(\log n)\frac{1}{\sqrt{T}})$$

Performance Comparisons

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Regret Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>OnlineRank [Ailon, 2014]</td>
<td>$O(n^2 \sqrt{T})$</td>
</tr>
<tr>
<td>XF-Perm</td>
<td>$O(n^2 (\log n) \sqrt{T})$</td>
</tr>
<tr>
<td>PermELearn [Helmbold &amp; Warmuth, 2009]</td>
<td>$O(n^2 (\log n) \sqrt{T})$</td>
</tr>
<tr>
<td>PermutahedronLearn [Yasutake et. al, 2011]</td>
<td>$O(n^2 \sqrt{T})$</td>
</tr>
<tr>
<td>Follow the Perturbed Leader [Kalai &amp; Vempala, 2005]</td>
<td>$O(n \sqrt{T})$</td>
</tr>
<tr>
<td>Hedge [Freund &amp; Schapire, 1997]</td>
<td>$O(n (\log n) \sqrt{T})$</td>
</tr>
</tbody>
</table>