## Motivation

Setting: Multi-armed Bandits with Slates

**Game**: for $t=1$ to $T$
- World commits to losses (or rewards) $l(t) = l_1(t), l_2(t), \ldots, l_K(t) \in [-1,1]$
- Each of $N$ policies recommend slates
- Learner chooses slate $S = \{j_1, j_2, \ldots, j_s\}$ of $s$ actions
- Learner receives loss $l_{j_1}(t) + l_{j_2}(t) + \ldots + l_{j_s}(t)$
- Want to compete with following the best slate (or policy) in hindsight

## Some Previous Work

- Exp3 & Exp4 [Auer et al. 02] work for $s=1$ and a reduction to them gives worse regret.
- We beat bounds from specializing bandit online optimization [Abernethy et al. 08] and combinatorial bandits [Cesa-Bianchi and Lugosi 09].
- We use many ideas of [Warmuth and Kuzmin 06] and [Helmbold and Warmuth 07].

## Our Algorithm

### Bandit Algorithm for Unordered Slates

**Initialization**: Start an instance of $\hat{\omega}(P)$ with the uniform initial distribution $p(1) = \frac{1}{K}$. Set $\eta = \sqrt{\frac{(1-\gamma)\ln(K/s)}{Kt}}$, and $\gamma = \sqrt{\frac{\ln(K/s)}{Kt}}$.

For $t = 1, 2, \ldots, T$:
1. Obtain the distribution $p(t)$ from $\hat{\omega}(P)$.
2. Set $p'(t) = (1-\gamma)p(t) + \frac{\gamma}{K}1_A$.
3. Note that $p'(t) \in P$. Decompose $ap'(t)$ as a convex combination of slate vectors $1_S$.
4. Choose a slate $S$ to display with probability $q_S$, and obtain the loss $l_j(t)$ for all $j \in S$.
5. Set $l_j(t) = \hat{l}_j(t) / (ap'(t))$ if $j \in S$, and 0 otherwise.
6. Send $\hat{\ell}(t)$ as the loss vector to $\hat{\omega}(P)$.

### MW(P)

MW(P) is a multiplicative weights algorithm that projects the resulting probability distribution onto a convex set. (MW concurrently appears as "component hedge" in [Koolen et al. 10])

We have versions of this algorithm for slates with experts and also with positional factors.

## Open Problems

- How to properly model interaction effects and analyze in this framework? We model rewards of actions as independent of other actions in slate.
- How to make this more efficient for large numbers of policies?
- What are good policies to use in practice on real data?
- Experimental evaluation of these algorithms.