Online (Budgeted) Social Choice

1. Overview of the Problem

- **High-level:** Multi-winner social choice in an online setting.
- **Given:** agents arrive in an online manner, with preferences over various items.
- **Goal:** Design algorithms for incrementally adding the "best" set of items.

2. Overview of the Process

- Agents arrive in an online manner, with \( v \) (cost of \( - Setting evenly.
- Random order of arrival: \( (W, \pi) \) determined in advance; order of arrival is uniformly random. A common model in online computation (e.g., \([2, 3]\)).
- Distributional: Every ranking is sampled i.i.d. from an unknown distribution \( D \).
- An important special case: Mallows \( \pi \)-distribution \( D((x, \pi)) \). The reference ranking \( x \) is unknown to the algorithm; dispersion parameter \( \pi \) is known.

3. Model & Definitions

- Alternative set \( A = \{a_1, \ldots, a_m\} \).
- Algorithm starts with an empty slate \( S_t = \emptyset \), capacity \( k \).
- \( n \) agents, arriving in an online manner.
- Upon arrival in step \( t = 1, \ldots, n \):
  - Agent \( t \) reveals her full preference \( v_t \) – ranking over \( A \).
  - The algorithm can add items to the slate (or leave it unchanged). \( S_t \) - state of the slate after step \( t \).
  - Agent \( t \) takes the highest ranked item in \( S_t \), based on \( v_t \).
  - No additional items can added when \( |S_t| = k \).
- ALG maps preferences sequences to slate states:
  \( (v_1, v_2, \ldots, v_n) \Rightarrow (S_1, S_2, \ldots, S_n) \)
- Initial assumption: alternative addition is *irrevocable*.
- The Buyback relaxation: allow for the removal of items at a cost of \( p \). [1]

4. Objective Function

- Use **positional scoring rule:**
  - Non-increasing score vector: \( e \in \mathbb{R}^m \)
  - Agent \( t \)'s value for \( a \in A \): \( F_t(a) = a(v_t(a)). \)
  - Agent \( t \)'s value for slate \( S_t \); \( F_t(S_t) = \max_{s \subset A} F_t(s) \).

- **Competitive ratio:** The ratio of the sum of the agents’ scores, to the score of best offline slate \( S^* \):

\[
\min_{v} \frac{\sum_{t=1}^{n} F_t(S_t)}{\max_{S \subset A, |S| = k} \sum_{t=1}^{n} F_t(S)} = \text{ALG's total score} \tag{1}
\]

5. Input Models

- We consider three models of sequential input:
  1. **Worst-case:** Adversarial construction of \( v = (v_1, \ldots, v_n) \):
     - Adaptive adversary: Determines \( v_t \) based on \( S_1, \ldots, S_{t-1} \).
     - Non-adaptive adversary: Needs to provide \( v \) in advance.
  2. Random order of arrival: Preferences \( (v_1, \ldots, v_n) \) determined in advance; order of arrival is uniformly random. A common model in online computation (e.g., \([2, 3]\)).
  3. Distributional: Every ranking is sampled i.i.d. from an unknown distribution \( D \).

6. Summary of Results

1. **Worst-case:**
   1. Adaptive adversary: a trivial upper bound of \( \Omega(k/n) \).
   2. Non-adaptive adversary: a worst-case example \( (v, k) \), with \( k = 1 \) for which no randomized algorithm obtains a \( \Omega(\log_{m} \log_{m} m) \)-competitive ratio.

2. **Random Order of arrival:** can approximate the offline \( k \)-slate optimization problem using sampling. Combined with a brute force \( k \)-optimization, can obtain \( \left( 1 - 1 - o(1) \right) \)-comp. ratio. Using a standard greedy algorithm can obtain \( \Omega \left( 1 - 1 - o(1) \right) \)-ratio.

3. **Distributional:** For Mallows dist., can obtain \( \left( 1 - o(1) \right) \)-competitive ratio using \( \Theta(\log m \cdot \log m) \) many samples.

4. **Worst-case revisited:** A buyback relaxation:
   - \( k > 1 \): Can obtain similar results using additional brute force \( k \)-slate optimization. Can get polytime algorithm under additional assumptions on scores.

7. The Worst-Case Model

- **Adaptive adversary:** Set \( \alpha = (1, 0, \ldots, 0) \) – each agent wants a specific alternative. The adversary will present agents who are interested alternatives that are not offered. At most \( k \) agents can be satisfied.
- **Non-adaptive:** \( k = 1 \) Same score vector \( \alpha \). Define a family of input sequences: \( \{v_1, v_2, \ldots, v_n\} \), for \( \ell \leq m \).
  - For \( v_\ell \): begins with \( \ell \) blocks \( B_1, \ldots, B_\ell \); \( |B_\ell| = n^\ell \cdot X^\ell \), for some \( X, \ell \).\( |B_{\ell-1}| \leq n \). Distribute remainder from \( n \) evenly.

- **Agents in block \( B_1 \):** only want alternative \( a_1 \).
- **Others:** \( X, n \) are constants. \( a_1 \) is the highest scored item in \( B_1 \).
  - For other alternative \( a \neq a_1 \), score of \( |B_{\ell-1}| \leq \frac{n}{m} (X^{\ell-1}) + 1 \). Setting \( X = \ell = \log_{m} \log_{m} m \) gives the stated ratio.

8. The Random Order Model

- **Step 1:** agents arrive in blocks of size \( B \).
- **Step 2:** between blocks \((\ell - 1), \ell \), delete \( S_{\ell-1} \) (cost of \( \ell \cdot B \)), pick \( S_\ell \) based on \( F(S_\ell) \).
- **Step 3:** Remaining agents arrive, using resulting slate.

- We can estimate the offline objective function \( F(S) \) (average score for slate \( S \)), by sampling first \( t = \Theta(\log n + k \log m) \) rankings.

- **Bound on estimation error:** Hoefding+union bound.

- **Optimizing slate \( S \) given \( F(S) \):
  - Brute force: \( S^* = \arg \max_{S \in \mathcal{S}} F(S) \).
  - Greedy: for \( \ell \leq k \), given \( S_\ell \), add \( \max_{a \in A} F(S_\ell \cup a) - F(S_\ell) \).

9. The Buyback Extension

- An alternative can be removed at a cost of \( p \).
- Partition the sequence \( v \) into blocks of size \( B \).
- Between blocks \((\ell - 1), \ell \), delete \( S_{\ell-1} \) (cost of \( B \)), pick \( S_\ell \) based on \( F(S_\ell) \).
- Use multiplicative weight update algorithm (MWU) [4] for selecting each block-slate.
- The regret (additive loss): regret of MWU + total removal cost.

10. Open Problems

- Lower bound: extension to \( k > 1 \).
- Other objective functions: Knapsack, production cost, etc.
- Other relaxations of the irrevocability assumption.

Bibliography