Mechanism Design for Learning Agents

Yingkai Li

EC4501/EC4501HM

Mechanism Design for Learning Agents

In online platforms, strategic agents use online learning algorithms for repeated interactions.

- Google / Microsoft allow advertisers to use learning algorithms to bid in Ad Auctions;
- High-frequency trading firms use reinforcement learning to adjust buy/sell decisions in real time;
- Individual re-sellers use bandit algorithms to set optimal prices by continuously adjusting and learning from buyer responses in resale markets;
- ...

Mechanism Design for Learning Agents

In online platforms, strategic agents use online learning algorithms for repeated interactions.

- Google / Microsoft allow advertisers to use learning algorithms to bid in Ad Auctions;
- High-frequency trading firms use reinforcement learning to adjust buy/sell decisions in real time;
- Individual re-sellers use bandit algorithms to set optimal prices by continuously adjusting and learning from buyer responses in resale markets;
- ...

Question: how to design optimal mechanisms for platforms when users adopt no-regret learning algorithms.

Example: Repeated Auctions

T periods, a single item for sale in each period. The buyer's value v drawn from F with support $0 \leq v_1 < \cdots < v_m \leq 1$. Value v is persistent across periods.

At any period $t \leq T$:

- seller offers K options: each bid b_i is associated with an outcome $(x_{i,t}, p_{i,t})$ where $p_{i,t} \in [0, x_{i,t} \cdot b_i]$.

Example: Repeated Auctions

T periods, a single item for sale in each period. The buyer's value v drawn from F with support $0 \leq v_1 < \cdots < v_m \leq 1$. Value v is persistent across periods.

At any period $t \leq T$:

- seller offers K options: each bid b_i is associated with an outcome $(x_{i,t}, p_{i,t})$ where $p_{i,t} \in [0, x_{i,t} \cdot b_i]$.

Classic Bayesian model: the buyer best responds to the dynamic selling algorithm.

Example: Repeated Auctions

T periods, a single item for sale in each period. The buyer's value v drawn from F with support $0 \le v_1 < \cdots < v_m \le 1$. Value v is persistent across periods.

At any period $t \le T$:

- seller offers K options: each bid b_i is associated with an outcome $(x_{i,t}, p_{i,t})$ where $p_{i,t} \in [0, x_{i,t} \cdot b_i]$.

Classic Bayesian model: the buyer best responds to the dynamic selling algorithm.

Learning model [Braverman, Mao, Schneider and Weinberg '17]: the buyer uses learning algorithms for online bidding.

Mean-based Algorithms

Definition (Mean-based Algorithms)

An algorithm is a γ-mean-based algorithm if it is the case that whenever $\hat{\mu}_{i,t} < \hat{\mu}_{j,t} - \gamma T$, the probability that the algorithm pulls arm i on round t is at most γ. We say an algorithm is mean-based if it is γ-mean-based for some $\gamma = o(1)$.

Examples of mean-based algorithms: Hedge, EXP3, etc.

Mean-based Algorithms

Definition (Mean-based Algorithms)

An algorithm is a γ-mean-based algorithm if it is the case that whenever $\hat{\mu}_{i,t} < \hat{\mu}_{j,t} - \gamma T$, the probability that the algorithm pulls arm i on round t is at most γ. We say an algorithm is mean-based if it is γ-mean-based for some $\gamma = o(1)$.

Examples of mean-based algorithms: Hedge, EXP3, etc.

Optimal mechanism design when agents use mean-based algorithms.

Full Welfare Extraction

Theorem

If the buyer uses a mean-based algorithm (e.g., EXP3), the seller can extract revenue $(1-\varepsilon)T \cdot \mathrm{Val}(F) - o(T)$.

Full Welfare Extraction

Theorem

If the buyer uses a mean-based algorithm (e.g., EXP3), the seller can extract revenue $(1-\varepsilon)T \cdot \mathrm{Val}(F) - o(T)$.

Key Idea: The seller can design an auction that "lures" the buyer into bidding high early on by offering the item for free, then charges high prices later.

- such a luring behavior is not beneficial for rational agents, since they will not be exploited given high prices.

A Simple Illustration

Consider an example where the buyer's value is $\frac{1}{4}$ with probability $\frac{1}{2}$, and is $\frac{1}{2}$ and 1 with probability $\frac{1}{4}$ each.

- optimal welfare is $\frac{1}{2}$, and the optimal revenue is $\frac{1}{4}$ for rational agents.

A Simple Illustration

Consider an example where the buyer's value is $\frac{1}{4}$ with probability $\frac{1}{2}$, and is $\frac{1}{2}$ and 1 with probability $\frac{1}{4}$ each.

- optimal welfare is $\frac{1}{2}$, and the optimal revenue is $\frac{1}{4}$ for rational agents.

A dynamic auction:

- **Arm 0**: bidding 0
  - Always charge $p_t = 0$, never give the item.
- **Arm 1**: bidding 1
  - First $T/2$ rounds: Charge $p_t = 0$, give the item for free.
  - Next $T/2$ rounds: Charge $p_t = 1$, give the item.

A Simple Illustration

Buyer Behavior:

- buyer with value 1 and $\frac{1}{2}$ chooses arm 1 until T;
- buyer with value $\frac{1}{4}$ chooses arm 1 until $\frac{2T}{3}$.

A Simple Illustration

Buyer Behavior:

- buyer with value 1 and $\frac{1}{2}$ chooses arm 1 until T;
- buyer with value $\frac{1}{4}$ chooses arm 1 until $\frac{2T}{3}$.

Revenue:

- The seller earns $\frac{T}{3}$ revenue, which is better than $\frac{T}{4}$.

A Simple Illustration

Buyer Behavior:

- buyer with value 1 and $\frac{1}{2}$ chooses arm 1 until T;
- buyer with value $\frac{1}{4}$ chooses arm 1 until $\frac{2T}{3}$.

Revenue:

- The seller earns $\frac{T}{3}$ revenue, which is better than $\frac{T}{4}$.

There exists a dynamic mechanism that achieves a revenue close to $\frac{T}{2}$.

Two Critics

In simple illustration, there are two main criticism of the result:

1. the agent with value $\frac{1}{2}$ can obtain a higher utility by mimicking the learning strategy of value $\frac{1}{4}$;
2. the auction requires the agent to overbid to extract a high revenue.

Two Critics

In simple illustration, there are two main criticism of the result:

1. the agent with value $\frac{1}{2}$ can obtain a higher utility by mimicking the learning strategy of value $\frac{1}{4}$;
2. the auction requires the agent to overbid to extract a high revenue.

Theorem

There exists learning algorithms such that the average revenue the seller can extract is at most the Myerson's optimal revenue.

Restore the incentives by allowing the learning algorithms to consider strategies of mimicking other types.

No-overbidding

Theorem

The seller can extract a revenue strictly higher than the Myerson's optimal revenue even when the agent does not overbid.

No-overbidding

Theorem

The seller can extract a revenue strictly higher than the Myerson's optimal revenue even when the agent does not overbid.

Another dynamic auction:

- **Arm 0**: bidding 0
  - Always charge $p_t = 0$, never give the item.
- **Arm 1**: bidding $\frac{1}{4}$
  - First $T/3$ rounds: Charge $p_t = 0$, never give the item.
  - Next $2T/3$ rounds: Charge $p_t = \frac{1}{4}$, give the item.
- **Arm 2**: bidding $\frac{1}{2}$
  - Always charge $p_t = \frac{1}{2}$, give the item.

No-overbidding

Buyer Behavior:

- buyer with value 1 chooses arm 2 until T;
- buyer with value $\frac{1}{2}$ chooses arm 2 until $\frac{T}{3}$, and then switch to arm 1;
- buyer with value $\frac{1}{4}$ chooses arm 1 until T.

No-overbidding

Buyer Behavior:

- buyer with value 1 chooses arm 2 until T;
- buyer with value $\frac{1}{2}$ chooses arm 2 until $\frac{T}{3}$, and then switch to arm 1;
- buyer with value $\frac{1}{4}$ chooses arm 1 until T.

Revenue:

- The seller earns $\frac{7T}{24}$ revenue, which is better than $\frac{T}{4}$.

Conclusion

Summary: The seller can extract close to the full welfare of the buyer by designing an auction that exploits the buyer's no-regret learning behavior.

Conclusion

Summary: The seller can extract close to the full welfare of the buyer by designing an auction that exploits the buyer's no-regret learning behavior.

Key Insights:

- The seller uses a combination of free and paid rounds to "lure" the buyer into overpaying.
- The buyers can protect themselves from being exploited by not overbidding, or by adopting more sophisticated algorithms.

Incentivizing Exploration

In many applications, to acquire information, the online platform need to incentivize strategic user to explore various options:

- incentivizing patients in clinical trials;
- incentivizing consumers to dine in newly opened restaurants for reviews on Yelp;
- incentivizing firms to develop in new technologies;
- ...

Incentivizing Exploration

In many applications, to acquire information, the online platform need to incentivize strategic user to explore various options:

- incentivizing patients in clinical trials;
- incentivizing consumers to dine in newly opened restaurants for reviews on Yelp;
- incentivizing firms to develop in new technologies;
- ...

The incentives of the designer and the strategic users are not aligned.

- designer benefits from collecting information for long-run decisions;
- users only benefit from short-run decisions.

Incentivizing Exploration

A platform faces a sequence of myopic agents.

- n arms, each arm i has a stochastic return drawn from distribution $F_i \in \Delta([0,1])$;
- prior belief D_i about the possible reward distributions for arm i.

Incentivizing Exploration

A platform faces a sequence of myopic agents.

- n arms, each arm i has a stochastic return drawn from distribution $F_i \in \Delta([0, 1])$;
- prior belief D_i about the possible reward distributions for arm i.

At each time $t \leq T$:

- a myopic agent arrives;
- the platform can make a recommendation to the myopic agent based on the history at time t;
- myopic agent chooses an arm to maximize his payoff at time t;
- bandit feedback: the platform only observes the payoff of the chosen arm.

Fully Revelation

A possible strategy is to fully reveal the history rewards to the myopic agent at any time t.
Fully revealing is exactly the same as follow-the-leader.

- the platform suffers from a linear regret by fully revealing.

Fully Revelation

A possible strategy is to fully reveal the history rewards to the myopic agent at any time t.
Fully revealing is exactly the same as follow-the-leader.

- the platform suffers from a linear regret by fully revealing.

Question: is it possible to improves the regret to sublinear?

- cannot directly ask the agent to explore suboptimal arms due to myopic incentives;
- incentivize via partial information revelation.

Hidden Exploration

Hidden exploration with parameter γ:

- with probability γ: randomly recommend an arm;
- with probability $1 - \gamma$: the best arm based on the history.

Hidden Exploration

Hidden exploration with parameter γ:

- with probability γ: randomly recommend an arm;
- with probability $1 - \gamma$: the best arm based on the history.

In each period $t \leq T$, the agent only sees the realized recommendation without observing the full history of rewards.

- for sufficiently small probability γ, the agent has incentives to follow the recommendation for all periods.

Hidden Exploration

Simple illustration: two arms

- arm 1: good state and bad state with equal probabilities
  - good state: reward 1 with probability $\frac{2}{3}$ and reward 0 with probability $\frac{1}{3}$;
  - bad state: reward 1 with probability $\frac{1}{3}$ and reward 0 with probability $\frac{2}{3}$.
- arm 2: fix reward $\frac{1}{2}$.

Hidden Exploration

Simple illustration: two arms

- arm 1: good state and bad state with equal probabilities
  - good state: reward 1 with probability $\frac{2}{3}$ and reward 0 with probability $\frac{1}{3}$;
  - bad state: reward 1 with probability $\frac{1}{3}$ and reward 0 with probability $\frac{2}{3}$.
- arm 2: fix reward $\frac{1}{2}$.

In period 1, let the agent choose arm 1, and the principal observes the realized reward.

Hidden Exploration

Simple illustration: two arms

- arm 1: good state and bad state with equal probabilities
  - good state: reward 1 with probability $\frac{2}{3}$ and reward 0 with probability $\frac{1}{3}$;
  - bad state: reward 1 with probability $\frac{1}{3}$ and reward 0 with probability $\frac{2}{3}$.
- arm 2: fix reward $\frac{1}{2}$.

In period 1, let the agent choose arm 1, and the principal observes the realized reward.

In period 2, hidden exploration can incentivize the agent to choose arm 1 with positive probability even if the realized reward is small for arm 1.

Hidden Exploration

Simple illustration: two arms

- arm 1: good state and bad state with equal probabilities
  - good state: reward 1 with probability $\frac{2}{3}$ and reward 0 with probability $\frac{1}{3}$;
  - bad state: reward 1 with probability $\frac{1}{3}$ and reward 0 with probability $\frac{2}{3}$.
- arm 2: fix reward $\frac{1}{2}$.

In period 1, let the agent choose arm 1, and the principal observes the realized reward.

In period 2, hidden exploration can incentivize the agent to choose arm 1 with positive probability even if the realized reward is small for arm 1.

When receiving recommendation of arm 1, the agent cannot distinguish between

1. the realization of arm 1 is high in the first period and the principal recommends the agent to exploit (with probability $1-\gamma$); and
2. the realization of arm 1 is low in the first period and the principal recommends the agent to explore (with probability γ).

Explore-then-Exploit

Replace the exploration phase with hidden exploration.

Explore-then-Exploit

Replace the exploration phase with hidden exploration.

Given parameter $T_0 \le T$:

- apply hidden exploration for each period $t \le T_0$;
- for any period $t \in [T_0, T]$, choose arm

$$i_t^* = \operatorname*{argmax}_{i \in [n]} \hat{\mu}_{i,T_0}.$$

Explore-then-Exploit

Replace the exploration phase with hidden exploration.

Given parameter $T_0 \leq T$:

- apply hidden exploration for each period $t \leq T_0$;
- for any period $t \in [T_0, T]$, choose arm

$$i_t^* = \operatorname*{argmax}_{i \in [n]} \hat{\mu}_{i,T_0}.$$

With $T_0 = O(\frac{\sqrt{nT}}{\gamma})$ periods, the estimation error is small, which ensures no regret.

- not tight for regret: exploration is not adjusted dynamically based on estimation.

Thompson Sampling

Thompson sampling algorithm is automatically incentive compatible given a sufficient number of initial samples [Sellke and Slivkins 21].

- Thompson sampling is a randomized algorithm where better arms are sampled with higher probabilities.

Thompson Sampling

Thompson sampling algorithm is automatically incentive compatible given a sufficient number of initial samples [Sellke and Slivkins 21].

- Thompson sampling is a randomized algorithm where better arms are sampled with higher probabilities.

Implication: given a sufficient number of initial samples, Thompson sampling achieves optimal regret under incentivized exploration.

Thompson Sampling

Thompson sampling algorithm is automatically incentive compatible given a sufficient number of initial samples [Sellke and Slivkins 21].

- Thompson sampling is a randomized algorithm where better arms are sampled with higher probabilities.

Implication: given a sufficient number of initial samples, Thompson sampling achieves optimal regret under incentivized exploration.

Initial samples: collected through hidden exploration.