Prior Free Mechanism Design

Yingkai Li

EC4501/EC4501HM

Robust Mechanism Design Frameworks

Let $\mathbb{M}$ be the set of possible mechanisms (e.g., DSIC mechanisms). For any $M \in \mathbb{M}$

- $M(v)$ is the performance of mechanism M given realized value profile v;
- $M(F) \triangleq \mathbf{E}_{v \sim F}[M(F)]$.

Robust Mechanism Design Frameworks

Let $\mathbb{M}$ be the set of possible mechanisms (e.g., DSIC mechanisms). For any $M \in \mathbb{M}$

- $M(v)$ is the performance of mechanism M given realized value profile v;
- $M(F) \triangleq \mathbf{E}_{v\sim F}[M(F)]$.

Bayesian Optimal

$$\mathrm{OPT}_F = \operatorname*{argmax}_{M \in \mathbb{M}} M(F).$$

Robust Mechanism Design Frameworks

Let $\mathbb{M}$ be the set of possible mechanisms (e.g., DSIC mechanisms). For any $M \in \mathbb{M}$

- $M(v)$ is the performance of mechanism M given realized value profile v;
- $M(F) \triangleq \mathbf{E}_{v\sim F}[M(F)]$.

Bayesian Optimal

$$\mathrm{OPT}_F = \operatorname*{argmax}_{M\in\mathbb{M}} M(F).$$

Prior-independent Optimal

$$\beta = \min_{M\in\mathbb{M}} \max_{F\in\mathbb{F}} \frac{\mathrm{OPT}_F(F)}{M(F)}$$

Robust Mechanism Design Frameworks

Let $\mathbb{M}$ be the set of possible mechanisms (e.g., DSIC mechanisms). For any $M \in \mathbb{M}$

- $M(v)$ is the performance of mechanism M given realized value profile v;
- $M(F) \triangleq \mathbf{E}_{v\sim F}[M(F)]$.

Bayesian Optimal

$$\mathrm{OPT}_F = \operatorname*{argmax}_{M\in\mathbb{M}} M(F).$$

Prior-independent Optimal

$$\beta = \min_{M\in\mathbb{M}} \max_{F\in\mathbb{F}} \frac{\mathrm{OPT}_F(F)}{M(F)}$$

Prior-free Optimal (for Benchmark B)

$$\gamma_B = \min_{M\in\mathbb{M}} \max_{v\in\mathbb{V}} \frac{B(v)}{M(v)}$$

Prior Free Mechanism Design

Prior-free Optimal (for Benchmark B)

$$\gamma_B = \min_{M \in \mathbb{M}} \max_{v \in \mathbb{V}} \frac{B(v)}{M(v)}$$

- relates to worst-case competitive analysis [Sleator and Tarjan '85; Littlestone and Warmuth '94].

Prior Free Mechanism Design

Prior-free Optimal (for Benchmark B)

$$\gamma_B = \min_{M \in \mathbb{M}} \max_{v \in \mathbb{V}} \frac{B(v)}{M(v)}$$

Optimal mechanism depends on the ad hoc benchmark.

Prior Free Mechanism Design

Prior-free Optimal (for Benchmark B)

$$\gamma_B = \min_{M \in \mathbb{M}} \max_{v \in \mathbb{V}} \frac{B(v)}{M(v)}$$

Optimal mechanism depends on the ad hoc benchmark.

Example: single-item, two agents, $v_{(1)} \geq v_{(2)}$.

- $B(v) = v_{(1)}$: no mechanism can approximate this benchmark.
- $B(v) = v_{(2)}$: second-price auction is prior-free optimal.
- $B(v) = k \cdot v_{(2)}$: randomly markup the second highest value by $\sqrt{k}$ with probability $\frac{1}{2}$ achieves a $2\sqrt{k}$ approximation.

Digital Auctions

n agents, unlimited supply of items.

Digital Auctions

n agents, unlimited supply of items.

Optimal single price omniscient auction:

$$B^*(v) = \max_{1 \leq i \leq n} i \cdot v_{(i)}$$

where $v_{(i)}$ is the ith highest value.

Digital Auctions

n agents, unlimited supply of items.

Optimal single price omniscient auction:

$$B^*(v) = \max_{1 \leq i \leq n} i \cdot v_{(i)}$$

where $v_{(i)}$ is the ith highest value.

No mechanism can approximate the benchmark of B^*.

- consider the instance where the highest revenue only comes from the highest value.

Digital Auctions

Optimal single price omniscient auction with at least two sales: [Goldberg, Hartline, Karlin, Saks and Wright '06]

$$B^{(2)}(v) = \max_{2 \leq i \leq n} i \cdot v_{(i)}.$$

Digital Auctions

Optimal single price omniscient auction with at least two sales: [Goldberg, Hartline, Karlin, Saks and Wright '06]

$$B^{(2)}(v) = \max_{2 \le i \le n} i \cdot v_{(i)}.$$

Exclude the possibility of all revenue contribution comes from a single agent.

Random Sampling

ProfitExtract_R

The Profit Extraction auction (ProfitExtract_R), given target profit R, is defined as follows:

1. Find the largest k such that the highest k bidders' bids are at lease R/k.
2. Charge these k bidders R/k and reject all others.

ProfitExtract_R is incentive compatible given any $R \geq 0$.

Random Sampling

$\mathrm{ProfitExtract}_R$

The Profit Extraction auction ($\mathrm{ProfitExtract}_R$), given target profit R, is defined as follows:

1. Find the largest k such that the highest k bidders' bids are at lease R/k.
2. Charge these k bidders R/k and reject all others.

$\mathrm{ProfitExtract}_R$ is incentive compatible given any $R \geq 0$.

RSPE

The Random Sampling Profit Extraction auction (RSPE) is works as follows:

1. Partition bids uniformly at random into two sets: b' and b''.
2. Compute R' and R'' as the optimal single price profits for b' and b'', respectively.
3. Compute the auction results by running $\mathrm{ProfitExtract}_{R''}$ on b' and $\mathrm{ProfitExtract}_{R'}$ on b''.

RSPE is incentive compatible.

Random Sampling

Theorem (Goldberg, Hartline, Karlin, Saks and Wright '06)

The Random Sampling Profit Extraction auction (RSPE) is a 4-approximation to $B^{(2)}$.

Random Sampling

Theorem (Goldberg, Hartline, Karlin, Saks and Wright '06)

The Random Sampling Profit Extraction auction (RSPE) *is a* 4*-approximation to* $B^{(2)}$.

Revenue from RSPE is at least $\min\{R', R''\}$.

Random Sampling

Theorem (Goldberg, Hartline, Karlin, Saks and Wright '06)

The Random Sampling Profit Extraction auction (RSPE) *is a* 4*-approximation to* $B^{(2)}$.

Revenue from RSPE is at least $\min\{R', R''\}$.

Let $p \geq 0, k \geq 2$ be the optimal price and number of winners in benchmark $B^{(2)}$.
Let k' be the number of those k winners in b' and k'' be the number of those k winners in b''.

$$R' \geq p \cdot k' \text{ and } R'' \geq p \cdot k''.$$

Random Sampling

Theorem (Goldberg, Hartline, Karlin, Saks and Wright '06)

The Random Sampling Profit Extraction auction (RSPE) *is a* 4*-approximation to* $B^{(2)}$.

Revenue from RSPE is at least $\min\{R', R''\}$.

Let $p \geq 0, k \geq 2$ be the optimal price and number of winners in benchmark $B^{(2)}$.
Let k' be the number of those k winners in b' and k'' be the number of those k winners in b''.

$$R' \geq p \cdot k' \text{ and } R'' \geq p \cdot k''.$$

Therefore, for any v,

$$\frac{\mathrm{RSPE}(v)}{B^{(2)}(v)} \geq \mathbf{E}\left[\frac{\min\{k', k''\}}{k}\right] \geq \frac{1}{4}.$$

Prior-free Optimal Mechanism

Given the benchmark of $B^{(2)}$, is RSPE the optimal mechanism?

Prior-free Optimal Mechanism

Given the benchmark of $B^{(2)}$, is RSPE the optimal mechanism? No.

There exists a mechanism that guarantees a 2.42-approximation to the optimal [Chen, Gravin and Lu '14].

Prior-free Optimal Mechanism

Given the benchmark of $B^{(2)}$, is RSPE the optimal mechanism? No.

There exists a mechanism that guarantees a 2.42-approximation to the optimal [Chen, Gravin and Lu '14].

- non-constructive proof.

Prior-free Optimal Mechanism

Given the benchmark of $B^{(2)}$, is RSPE the optimal mechanism? No.

There exists a mechanism that guarantees a 2.42-approximation to the optimal [Chen, Gravin and Lu '14].

- non-constructive proof.

Question: is $B^{(2)}$ the right benchmark for the prior-free model? Is the corresponding optimal mechanism for $B^{(2)}$ the appropriate mechanism to adopt for such robust environments?

Benchmark Optimization

Question: What makes a good benchmark?

Benchmark Optimization

Question: What makes a good benchmark?

Goal: discriminate between good and bad mechanisms [Hartline and Roughgarden '08].

Benchmark Optimization

Question: What makes a good benchmark?

Goal: discriminate between good and bad mechanisms [Hartline and Roughgarden '08].

- too small: many mechanisms can approx, doesn't discriminate.
- too large: no mechanism can approx, doesn't discriminate.

Benchmark Optimization

Question: What makes a good benchmark?

Goal: discriminate between good and bad mechanisms [Hartline and Roughgarden '08].

- too small: many mechanisms can approx, doesn't discriminate.
- too large: no mechanism can approx, doesn't discriminate.

Informal Goal: formalize "too small", optimize "too large".

Normalized Benchmarks

Intuition: benchmark must not be too small.

Normalized Benchmarks

Intuition: benchmark must not be too small.

Definition (Normalized Benchmark)

For distributions $\mathbb{F}$, benchmark B is normalized if $B(F) \geq \mathrm{OPT}_F(F)$ for all $F \in \mathbb{F}$. $\mathbb{B}(\mathbb{F})$ is family of normalized benchmarks.

E.g., welfare is a normalized benchmark for revenue.

Normalized Benchmarks

Intuition: benchmark must not be too small.

Definition (Normalized Benchmark)

For distributions $\mathbb{F}$, benchmark B is normalized if $B(F) \geq \mathrm{OPT}_F(F)$ for all $F \in \mathbb{F}$. $\mathbb{B}(\mathbb{F})$ is family of normalized benchmarks.

E.g., welfare is a normalized benchmark for revenue.

Theorem (Hartline, Roughgarden '08)

Prior-free γ-approximation of normalized benchmark for $\mathbb{F}$
$\Rightarrow$ *prior-independent γ-approximation for $\mathbb{F}$.*

Normalized Benchmarks

Intuition: benchmark must not be too small.

Definition (Normalized Benchmark)

For distributions $\mathbb{F}$, benchmark B is normalized if $B(F) \geq \mathrm{OPT}_F(F)$ for all $F \in \mathbb{F}$. $\mathbb{B}(\mathbb{F})$ is family of normalized benchmarks.

E.g., welfare is a normalized benchmark for revenue.

Theorem (Hartline, Roughgarden '08)

Prior-free γ-approximation of normalized benchmark for $\mathbb{F}$
$\Rightarrow$ *prior-independent γ-approximation for $\mathbb{F}$.*

For any $F \in \mathbb{F}$,

$$M(F) \geq \frac{1}{\gamma} \cdot B(F) \geq \frac{1}{\gamma} \cdot \mathrm{OPT}_F(F).$$

Benchmark Resolution

Intuition: benchmark must not be too big.

Benchmark Resolution

Intuition: benchmark must not be too big.

Definition (Benchmark Resolution)

$$\gamma_B = \min_{M \in \mathbb{M}} \max_{v \in \mathbb{V}} \frac{B(v)}{M(v)}.$$

Benchmark Resolution

Intuition: benchmark must not be too big.

Definition (Benchmark Resolution)

$$\gamma_B = \min_{M \in \mathbb{M}} \max_{v \in \mathbb{V}} \frac{B(v)}{M(v)}.$$

- The resolution of welfare is ∞.
- The resolution of $B^{(2)}$ in digital auction is 2.42.

Benchmark Resolution

Intuition: benchmark must not be too big.

Definition (Benchmark Resolution)

$$\gamma_B = \min_{M \in \mathbb{M}} \max_{v \in \mathbb{V}} \frac{B(v)}{M(v)}.$$

- The resolution of welfare is ∞.
- The resolution of $B^{(2)}$ in digital auction is 2.42.

Benchmark Optimization

Find normalized benchmark with finest resolution:

$$\gamma = \min_{B \in \mathbb{B}(\mathbb{F})} \min_{M \in \mathbb{M}} \max_{v \in \mathbb{V}} \frac{B(v)}{M(v)}.$$

Equivalence Between Prior Independent and Prior Free

Benchmark Optimization

$$\gamma = \min_{B \in \mathbb{B}(\mathbb{F})} \min_{M \in \mathbb{M}} \max_{v \in \mathbb{V}} \frac{B(v)}{M(v)}$$

Prior-independent Optimization

$$\beta = \min_{M \in \mathbb{M}} \max_{F \in \mathbb{F}} \frac{\mathrm{OPT}_F(F)}{M(F)}$$

Equivalence Between Prior Independent and Prior Free

Benchmark Optimization

$$\gamma = \min_{B \in \mathbb{B}(\mathbb{F})} \min_{M \in \mathbb{M}} \max_{v \in \mathbb{V}} \frac{B(v)}{M(v)}$$

Prior-independent Optimization

$$\beta = \min_{M \in \mathbb{M}} \max_{F \in \mathbb{F}} \frac{\mathrm{OPT}_F(F)}{M(F)}$$

Theorem (Hartline, Johnson and Li '20)

Benchmark optimization = prior-independent optimization, i.e., $\gamma = \beta$

Equivalence Between Prior Independent and Prior Free

Benchmark Optimization

$$\gamma = \min_{B \in \mathbb{B}(\mathbb{F})} \min_{M \in \mathbb{M}} \max_{v \in \mathbb{V}} \frac{B(v)}{M(v)}$$

Prior-independent Optimization

$$\beta = \min_{M \in \mathbb{M}} \max_{F \in \mathbb{F}} \frac{\mathrm{OPT}_F(F)}{M(F)}$$

Theorem (Hartline, Johnson and Li '20)

Benchmark optimization = prior-independent optimization, i.e., $\gamma = \beta$

Sketch:

1. prior-independent opt. program suggests a benchmark and algorithm, best benchmark and algorithm is no worse.
2. benchmark program suggests a algorithm that is a prior-independent approximation, optimal prior-independent approximation is no worse.

Proof of (1)

Benchmark Optimization

$$\gamma = \min_{B \in \mathbb{B}(\mathbb{F})} \min_{M \in \mathbb{M}} \max_{v \in \mathbb{V}} \frac{B(v)}{M(v)}$$

Prior-independent Optimization

$$\beta = \min_{M \in \mathbb{M}} \max_{F \in \mathbb{F}} \frac{\mathrm{OPT}_F(F)}{M(F)}$$

$\gamma \leq \beta$: prior-independent opt. program suggests a benchmark and algorithm, best benchmark and algorithm is no worse.

- let M be solution to prior-independent program with objective value β
- define scaled-up benchmark: $B(v) = \beta \cdot M(v)$
- (M, B) are valid for benchmark program with objective β
- optimal solution to benchmark program has $\gamma \leq \beta$.

Proof of (2)

Benchmark Optimization

$$\gamma = \min_{B \in \mathbb{B}(\mathbb{F})} \min_{M \in \mathbb{M}} \max_{v \in \mathbb{V}} \frac{B(v)}{M(v)}$$

Prior-independent Optimization

$$\beta = \min_{M \in \mathbb{M}} \max_{F \in \mathbb{F}} \frac{\mathrm{OPT}_F(F)}{M(F)}$$

$\beta \leq \gamma$: benchmark program suggests a mechanism that is a prior-independent approximation, optimal prior-independent approximation is no worse.

- let (M, B) be solution to benchmark program with objective value γ
- normalization of benchmark $\Rightarrow$ M is prior-independent γ-approx.
- M is valid solution for prior-independent program with objective γ
- optimal solution to prior-independent program has $\beta \leq \gamma$.

Open Questions

Benchmark Optimization

Find normalized benchmark with finest resolution:

$$\gamma = \min_{B \in \mathbb{B}(\mathbb{F})} \min_{M \in \mathbb{M}} \max_{v \in \mathbb{V}} \frac{B(v)}{M(v)}.$$

Question: is this the right way for benchmark optimization?

Open Questions

Benchmark Optimization

Find normalized benchmark with finest resolution:

$$\gamma = \min_{B\in\mathbb{B}(\mathbb{F})} \min_{M\in\mathbb{M}} \max_{v\in\mathbb{V}} \frac{B(v)}{M(v)}.$$

Question: is this the right way for benchmark optimization?

Its connection to prior-independent analysis indicates that it loses some of desirable robustness properties when adopting the "optimal" benchmark.

- e.g., online learning [Hartline, Johnson and Li '20].