Constrained Preference Elicitation

Yaron Azrieli, Chris Chambers, P.J. Healy

RUD
December 2020

Introduction

A principal wants to learn *something* about the preferences of an agent, but *not* the whole ordering

Examples:

- NYC school match: only list 12 schools
- Lab experiment on independence axiom
- Yelp survey learning how you rank a new restaurant
- Eliciting beliefs from an expert (which event they'd bet on)

A principal wants to learn *something* about the preferences of an agent, but *not* the whole ordering

Examples:

- NYC school match: only list 12 schools
- Lab experiment on independence axiom
- Yelp survey learning how you rank a new restaurant
- Eliciting beliefs from an expert (which event they'd bet on)

Which properties of preferences can be elicited in an incentive compatible way?

Comparison to mechanism design/social choice

Mechanism Design (MD) vs. Constrained Elicitation (CE):

1. Principal's goal is easier
   - MD: Implement desirable outcome. IC = constraint.
   - CE: Don't care about outcomes. $\exists$ *any* IC mechanism?
2. Incentive compatibility is harder
   - MD: Only require weak IC.
   - CE: Require strict IC.
3. Limited info is harder
   - MD: Can elicit entire $\succeq$
   - CE: Can't learn more than prescribed.

Eliciting a property

Why not elicit entire $\succeq$?

- Privacy concerns
- Future interactions
- Costs

Some related literature

- Osband (1985), Lambert-Pennock-Shoham (2008), Lambert (2018): Scoring rules to elicit properties of beliefs
- Gibbard (1977), Bahel and Sprumont (2019): Characterizing strategy-proof random mechanisms
- Carroll (2012), Saito (2013): Sufficiency of local IC constraints
- ACH (2018, 2020): Characterizing IC mechanisms in experimental environments

- X - a finite set of alternatives
  - Typical elements: $x, y, z, w, \ldots$
- O - the set of strict orders over X
  - Typical elements: $\succeq, \succeq', \ldots$

Framework

- X - a finite set of alternatives
  - Typical elements: $x, y, z, w, \ldots$
- O - the set of strict orders over X
  - Typical elements: $\succeq, \succeq', \ldots$

Definition
A *type space* $T = \{t_1, \ldots, t_k\}$ is a partition of O.

- A *type* is any $t \in T$, so $t = \{\succeq, \succeq', ..., \succeq''\}$

- Example: $t = \{\text{all } \succeq \text{ satisfying the Independence axiom}\}$

- Notation: $t(\succeq) \in T$ is the type containing $\succeq$

Examples

$X = \{x, y, z\}$

- Entire ranking:
  $T = \{\{xyz\}, \{yxz\}, \{yzx\}, \{zyx\}, \{zxy\}, \{xzy\}\}$
- First-best:
  $T = \{\{xyz, xzy\}, \{yxz, yzx\}, \{zyx, zxy\}\}$
- Top-2:
  $T = \{\{xyz, yxz\}, \{xzy, zxy\}, \{zyx, yzx\}\}$
- Best from $\{x, y\}$:
  $T = \{\{xyz, xzy, zxy\}, \{zyx, yzx, yxz\}\}$
- Ranking of x:
  $T = \{\{xyz, xzy\}, \{zxy, yxz\}, \{yzx, zyx\}\}$

Mechanisms

$\Delta(X)$ is the set of lotteries on X

Definition
A T-mechanism is any $g : T \to \Delta(X)$.

- Why random payments?
    - With deterministic mechanisms very little can be elicited

Elicitable type spaces

Recall that p strictly FOSD q relative to $\succeq$ ($p \succ^* q$) if

$$\forall x \in X \quad p(\{y : y \succeq x\}) \geq q(\{y : y \succeq x\})$$

with strict inequality for at least one x

Elicitable type spaces

Recall that p strictly FOSD q relative to $\succeq$ ($p \succ^* q$) if

$$\forall x \in X \quad p(\{y : y \succeq x\}) \geq q(\{y : y \succeq x\})$$

with strict inequality for at least one x

Definition
g is *IC* if for every $\succeq \in O$ and every $t \neq t(\succeq)$

$$g(t(\succeq)) \succ^* g(t).$$

Elicitable type spaces

Recall that p strictly FOSD q relative to $\succeq$ ($p \succ^* q$) if

$$\forall x \in X \quad p(\{y : y \succeq x\}) \geq q(\{y : y \succeq x\})$$

with strict inequality for at least one x

Definition
g is *IC* if for every $\succeq \in O$ and every $t \neq t(\succeq)$

$$g(t(\succeq)) \succ^* g(t).$$

Definition
A type space T is *elicitable* if there exists an IC T-mechanism.

Goal: Characterize elicitable type spaces (spoiler: we can't)

Top elements of menus

"What's your favorite thing from X'?"

- $\mathrm{dom}_{\succeq}(X')$ denotes the $\succeq$-maximal element in $X' \subseteq X$

- Every menu $X' \subseteq X$ defines a type space by

$$\succeq, \succeq' \in t \iff \mathrm{dom}_{\succeq}(X') = \mathrm{dom}_{\succeq'}(X')$$

Examples:

$X' = \{x, y\} \implies T = \{\{xyz, xzy, zxy\}, \{zyx, yzx, yxz\}\}$

$X' = \{x, y, z\} \implies T = \{\{xyz, xzy\}, \{yxz, yzx\}, \{zyx, zxy\}\}$

Top elements of menus

"What's your favorite thing from X'?"

- $\mathrm{dom}_{\succeq}(X')$ denotes the $\succeq$-maximal element in $X' \subseteq X$

- Every menu $X' \subseteq X$ defines a type space by

$$\succeq, \succeq' \in t \iff \mathrm{dom}_{\succeq}(X') = \mathrm{dom}_{\succeq'}(X')$$

Examples:

$X' = \{x, y\} \implies T = \{\{xyz, xzy, zxy\}, \{zyx, yzx, yxz\}\}$

$X' = \{x, y, z\} \implies T = \{\{xyz, xzy\}, \{yxz, yzx\}, \{zyx, zxy\}\}$

- The (deterministic) mechanism that pays the revealed top element in X' is IC

RPS mechanisms

- One can elicit top elements of several menus $X_1, \ldots, X_I \subseteq X$

Examples:

$$X_1 = \{x, y, z\},\ X_2 = \{x, y\}$$
$$\implies \quad T = \{\{xyz, xzy\}, \{yzx, yxz\}, \{zxy\}, \{zyx\}\}$$

$$X_1 = \{x, y\},\ X_2 = \{x, z\},\ X_3 = \{y, z\}$$
$$\implies \quad T = \{\{xyz\}, \{yxz\}, \{yzx\}, \{zyx\}, \{zxy\}, \{xzy\}\}$$

- The corresponding IC mechanism randomly chooses a menu and pays the announced top element
- This is widely used in experimental economics

RPS mechanisms

- One can elicit top elements of several menus $X_1, \ldots, X_l \subseteq X$

Examples:

$X_1 = \{x, y, z\},\ X_2 = \{x, y\}$
$\implies \quad T = \{\{xyz, xzy\}, \{yzx, yxz\}, \{zxy\}, \{zyx\}\}$

$X_1 = \{x, y\},\ X_2 = \{x, z\},\ X_3 = \{y, z\}$
$\implies \quad T = \{\{xyz\}, \{yxz\}, \{yzx\}, \{zyx\}, \{zxy\}, \{xzy\}\}$

- The corresponding IC mechanism randomly chooses a menu and pays the announced top element

- This is widely used in experimental economics

What else is elicitable?

Top sets of menus

The top-2 type space $T = \{\{xyz, yxz\}, \{xzy, zxy\}, \{zyx, yzx\}\}$ does not reveal top elements of menus but is elicitable

Top sets of menus

The top-2 type space $T = \{\{xyz, yxz\}, \{xzy, zxy\}, \{zyx, yzx\}\}$ does not reveal top elements of menus but is elicitable

- How? If they announce "x and y" pay x and y with equal probability, and z with less probability.
- Every $X' \subseteq X$ and k defines a type space by

  $$\succeq, \succeq' \in t \iff \succeq, \succeq' \text{ have the same top } k \text{ elements of } X'$$

- This is elicitable by paying the uniform lottery over the set of announced top-k elements
- Can elicit the top-k_i elements of $X_i \subseteq X$, $i = 1, \ldots, l$

Top sets of menus

The top-2 type space $T = \{\{xyz, yxz\}, \{xzy, zxy\}, \{zyx, yzx\}\}$ does not reveal top elements of menus but is elicitable

- How? If they announce "x and y" pay x and y with equal probability, and z with less probability.
- Every $X' \subseteq X$ and k defines a type space by

$$\succeq, \succeq' \in t \iff \succeq, \succeq' \text{ have the same top } k \text{ elements of } X'$$

- This is elicitable by paying the uniform lottery over the set of announced top-k elements
- Can elicit the top-k_i elements of $X_i \subseteq X$, $i = 1, \ldots, l$

Anything else??

Example (based on Shapley, 1971)

$X = \{x, y, z, w\}$

Type space:

$\{xyzw, yxzw, xywz, yxwz\}$

$\{xzyw\}$, $\{xwyz\}$, $\{xzwy, xwzy\}$

$\{ywxz\}$, $\{yzxw\}$, $\{yzwx, ywzx\}$

$\{zxyw, zyxw\}$, $\{zywx, zwyx\}$, $\{zxwy, zwxy\}$

$\{wxyz, wyxz\}$, $\{wyzx, wzyx\}$, $\{wxzy, wzxy\}$

Claim

$\exists$ IC mechanism, but type space is not generated by top sets.

There is a close connection between IC mechanisms and convex TU cooperative games...

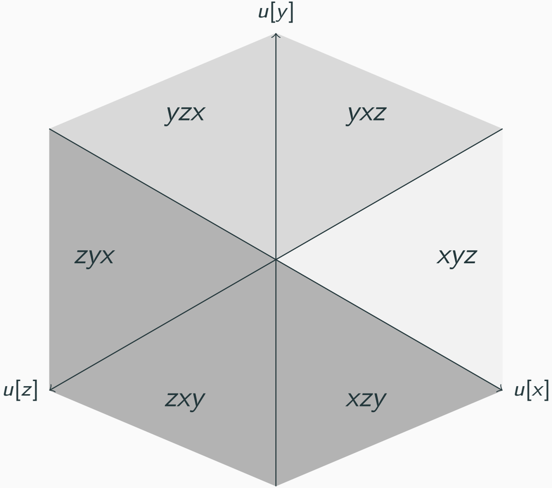
So far...

$$\{\text{all } T\}$$
$$\rotatebox{90}{$\subseteq$}$$
$$\color{red}\{T : \text{elicitable}\}$$
$$\rotatebox{90}{$\subsetneq$}$$
$$\{T : \text{generated by top sets}\}$$
$$\rotatebox{90}{$\subsetneq$}$$
$$\{T : \text{generated by top elements}\}$$

A convex type space - example

Necessary condition: **convex** type space

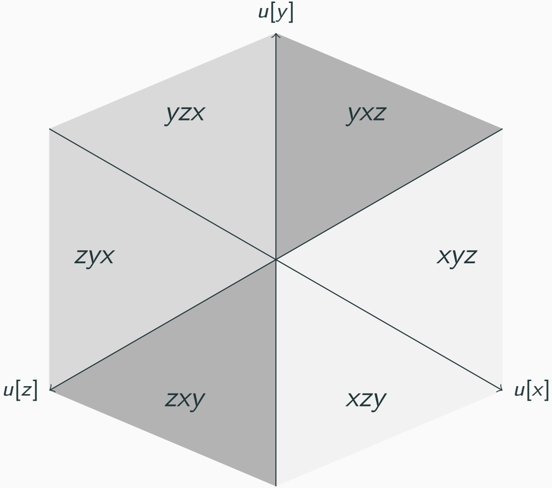
Example: $T = \{\{xyz\}, \{yxz, yzx\}, \{zyx, zxy, xzy\}\}$



A non-convex type space - example

Example of a non-convex type space:

$T = \{\{xyz, xzy\}, \{yxz, zxy\}, \{zyx, yzx\}\}$



Convexity is necessary

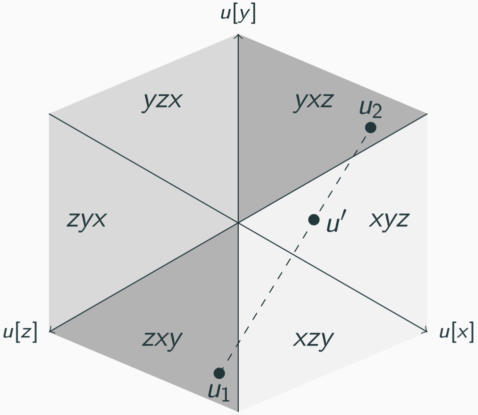
Proposition
If T is elicitable then it is convex.

$t =$ dark gray
$t' =$ off-white

IC requires:

$\mathbb{E}_{g(t)}(u_1) > \mathbb{E}_{g(t')}(u_1)$
$\mathbb{E}_{g(t)}(u_2) > \mathbb{E}_{g(t')}(u_2)$

$\implies \quad \mathbb{E}_{g(t)}(u') > \mathbb{E}_{g(t')}(u')$



Some non-convex type spaces

- The ranking of x (with $|X| \geq 3$)
- The kth ranked alternative for $1 < k < |X|$, e.g. the median
- Any binary $T = \{t_1, t_2\}$, *except* $T = \{\{x \succeq y\},\ \{y \succeq x\}\}$. In particular, tests of essentially any axiom of preferences

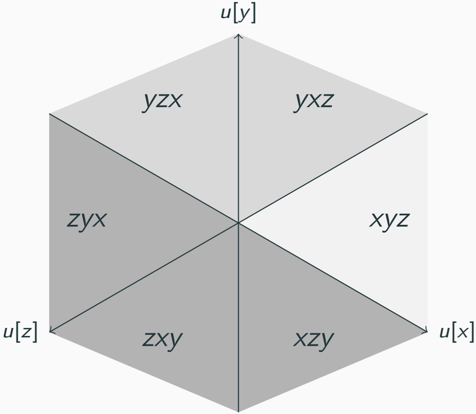
Convexity is not sufficient

$$T = \{t_1 = \{xyz\}, t_2 = \{yxz, yzx\}, t_3 = \{zyx, zxy, xzy\}\}$$

IC requires:

$g(t_1)(x) > g(t_2)(x)$
$g(t_2)(x) = g(t_3)(x)$
$g(t_3)(x) = g(t_1)(x)$

$\implies \quad g(t_1)(x) > g(t_1)(x)$



‘No bad cycles’ is necessary

- $\succeq$ and $\succeq'$ are adjacent via an x-y switch if they differ only in that $x \succeq y$ but $y \succeq' x$
- t and t' $(t \neq t')$ are adjacent via an x-y switch if there are $\succeq \in t$ and $\succeq' \in t'$ that are adjacent via an x-y switch

Proposition
Suppose T is elicitable. For any cycle of adjacent types $(t_1, t_2, \ldots, t_k, t_1)$, if t_1 and t_2 are adjacent via an x-y switch then there exist some $1 < i \leq k$ and z such that t_i and $t_{(i+1) \bmod k}$ are adjacent via a z-x switch.

Note: No bad cycles $\implies$ convexity

$$
\begin{array}{c}
\{\text{all } T\} \\
\rotatebox{-90}{$\supsetneq$} \\
\{T : \text{convex}\} \\
\rotatebox{-90}{$\supsetneq$} \\
\{T : \text{no bad cycles}\} \\
\rotatebox{-90}{$\supseteq$} \\
\{T : \text{elicitable}\} \\
\rotatebox{-90}{$\supsetneq$} \\
\{T : \text{generated by top sets}\} \\
\rotatebox{-90}{$\supsetneq$} \\
\{T : \text{generated by top elements}\}
\end{array}
$$

Neutral type spaces

- Permutation: $\pi : X \to X$
- Let πT be T, but with every $\succeq$ permuted by π

Definition
T is *neutral* if $\pi T = T$ for every π.

Neutral type spaces

- Permutation: $\pi : X \to X$
- Let πT be T, but with every $\succeq$ permuted by π

Definition
T is *neutral* if $\pi T = T$ for every π.

Proposition
Suppose T is neutral. Then the following are equivalent:
(1) T is elicitable
(2) T is convex
(3) T is generated by top sets

Neutral type spaces

$$\{\text{all } T\}$$

$$\rotatebox{90}{$\supsetneq$}$$

$$\{T : \text{convex}\}$$

$$\rotatebox{90}{$=$}$$

$$\{T : \text{elicitable}\}$$

$$\rotatebox{90}{$=$}$$

$$\{T : \text{generated by top sets}\}$$

$$\rotatebox{90}{$\supsetneq$}$$

$$\{T : \text{generated by top elements}\}$$

Robust elicitation

What if the agent has subjective beliefs about the likelihood of realizations of the randomization device (or other kinds of preferences over acts)?

Robust elicitation

What if the agent has subjective beliefs about the likelihood of realizations of the randomization device (or other kinds of preferences over acts)?

- Now payments are Savage acts, not lotteries.
- Can't pay a uniform lottery b/c can't control beliefs
- This kills our ability to elicit top sets

Robust elicitation

What if the agent has subjective beliefs about the likelihood of realizations of the randomization device (or other kinds of preferences over acts)?

- Now payments are Savage acts, not lotteries.
- Can't pay a uniform lottery b/c can't control beliefs
- This kills our ability to elicit top sets

Proposition
T is elicitable with acts iff it is generated by top elements.

Elicitation under acts

$$\{\text{all } T\}$$
$$\cup\!\!\!\!\not\,\,$$

$$\begin{gathered}
\{\text{all } T\} \\
\cup\mathrel{\not\shortparallel} \\
\{T : \text{convex}\} \\
\cup\mathrel{\not\shortparallel} \\
\{T : \text{elicitable with lotteries}\} \\
\cup\mathrel{\not\shortparallel} \\
\{T : \text{generated by top sets}\} \\
\cup\mathrel{\not\shortparallel} \\
\{T : \text{generated by top elements}\} \\
\shortparallel \\
\{T : \text{elicitable with acts}\}
\end{gathered}$$

Multiple agents

- $N = \{1, \ldots, n\}$ - agents
- T_i - agent's i type space
- $T = (T_1, \ldots, T_n)$ - a profile of type spaces
- $g : T \to \Delta(X)$ - a mechanism

Multiple agents

- $N = \{1, \ldots, n\}$ - agents
- T_i - agent's i type space
- $T = (T_1, \ldots, T_n)$ - a profile of type spaces
- $g : T \to \Delta(X)$ - a mechanism

Definition
g is *dominant-strategy IC (DIC)* if for every $i \in N$, every $\succeq_i \in O$, every $t_i \neq t_i(\succeq_i)$, and every $t_{-i} \in T_{-i}$

$$g(t_i(\succeq), t_{-i}) \succ_i^* g(t_i, t_{-i}).$$

Multiple agents

- $N = \{1, \ldots, n\}$ - agents
- T_i - agent's i type space
- $T = (T_1, \ldots, T_n)$ - a profile of type spaces
- $g : T \to \Delta(X)$ - a mechanism

Definition
g is *dominant-strategy IC (DIC)* if for every $i \in N$, every $\succeq_i \in O$, every $t_i \neq t_i(\succeq_i)$, and every $t_{-i} \in T_{-i}$

$$g(t_i(\succeq), t_{-i}) \succ_i^* g(t_i, t_{-i}).$$

Proposition
$T = (T_1, \ldots, T_n)$ *is DIC-elicitable iff each* T_i *is elicitable.*

Conclusion

- We formulate a notion of elicitability for properties of preferences
- Some necessary conditions and some sufficient conditions for elicitability, but no characterization
- We do have a characterization for neutral properties and for robust elicitation
- potential extensions: Weak orders, infinite sets of alternatives, domain restrictions,...

Thank You!