Mechanism Design; Quasilinear Utility

CPSC 532A Lecture 17

November 9, 2006

Lecture Overview

Mechanism Design

- Extend the social choice setting to a new setting where agents can't be relied upon to disclose their preferences honestly.

Definition (Mechanism)

A mechanism (over a set of agents N and a set of outcomes O) is a pair (A, M), where

- $A = A_1 \times \cdots \times A_n$, where A_i is the set of actions available to agent $i \in N$, and
- $M : A \rightarrow \Pi(O)$ maps each action profile to a distribution over outcomes.

Thus, the designer gets to specify

- the action sets for the agents (though they may be constrained by the environment)
- the mapping to outcomes, over which agents have utility
- can't change agents' preferences for outcomes or type spaces

Implementation in Dominant Strategies

Definition (Implementation in dominant strategies)

A mechanism (A, M) (over N and O) is an implementation in dominant strategies of a social choice function C over (N and O) if for any vector of utility functions u, the game (N, A, O, M, u) has an equilibrium in dominant strategies, and in any such equilibrium a^* we have $M(a^*) = C(u)$.

Implementation in Bayes-Nash equilibrium

Definition (Bayes-Nash implementation)

We begin with a mechanism (A, M) over N and O. Let $\Theta = \Theta_1 \times \cdots \times \Theta_n$ denote the set of all possible type vectors $\theta = (\theta_1, \ldots, \theta_n)$, and denote agent i's utility as $u_i : O \times \Theta \rightarrow \mathbb{R}$. Let p be a (common prior) probability distribution on Θ (and hence on u). Then (A, M) is a Bayes-Nash implementation of a social choice function C, with respect to Θ and p, if there exists a Bayes-Nash equilibrium of the game of incomplete information (N, A, Θ, p, u) such that for every $\theta \in \Theta$ and every action profile $a \in A$ that can arise given type profile θ in this equilibrium, we have that $M(a) = C(u(\cdot, \theta))$.

Properties

Forms of implementation

- Direct Implementation: agents each simultaneously send a single message to the center
- Indirect Implementation: agents may send a sequence of messages; in between, information may be (partially) revealed about the messages that were sent previously like extensive form

We can also insist that our mechanism satisfy properties like the following:

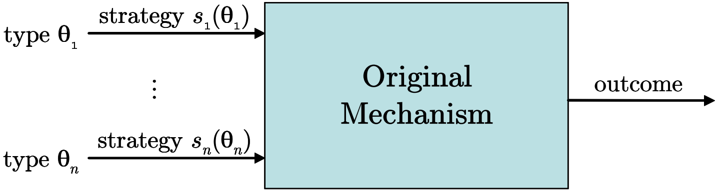
- individual rationality: agents are better off playing than not playing
- budget balance: the mechanism gives away and collects the same amounts of money
- truthfulness: agents honestly report their types

Lecture Overview

Revelation Principle

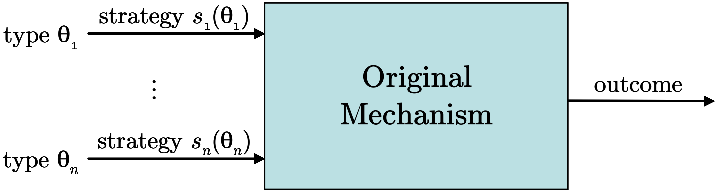
- It turns out that truthfulness can always be achieved!
- Consider an arbitrary, non-truthful mechanism (e.g., may be indirect)

Revelation Principle



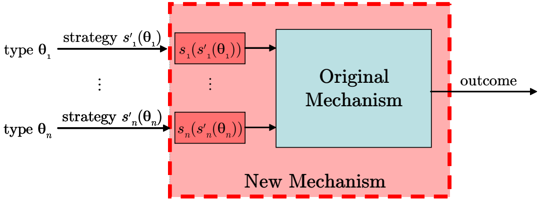
- It turns out that truthfulness can always be achieved!
- Consider an arbitrary, non-truthful mechanism (e.g., may be indirect)

Revelation Principle



- It turns out that truthfulness can always be achieved!
- Consider an arbitrary, non-truthful mechanism (e.g., may be indirect)
- Recall that a mechanism defines a game, and consider an equilibrium $s = (s_1, \ldots, s_n)$

Revelation Principle



- We can construct a new direct mechanism, as shown above
- This mechanism is truthful by exactly the same argument that s was an equilibrium in the original mechanism
- "The agents don't have to lie, because the mechanism already lies for them."

Computational Criticism of the Revelation Principle

- computation is pushed onto the center
  - often, agents' strategies will be computationally expensive
    - e.g., in the shortest path problem, agents may need to compute shortest paths, cutsets in the graph, etc.
  - since the center plays equilibrium strategies for the agents, the center now incurs this cost
- if computation is intractable, so that it cannot be performed by agents, then in a sense the revelation principle doesn't hold
  - agents can't play the equilibrium strategy in the original mechanism
  - however, in this case it's unclear what the agents will do

Discussion of the Revelation Principle

- The set of equilibria is not always the same in the original mechanism and revelation mechanism
  - of course, we've shown that the revelation mechanism does have the original equilibrium of interest
  - however, in the case of indirect mechanisms, even if the indirect mechanism had a unique equilibrium, the revelation mechanism can also have new, bad equilibria
- So what is the revelation principle good for?
  - recognition that truthfulness is not a restrictive assumption
  - for analysis purposes, we can consider only truthful mechanisms, and be assured that such a mechanism exists
  - recognition that indirect mechanisms can't do (inherently) better than direct mechanisms

Lecture Overview

Impossibility Result

Theorem (Gibbard-Satterthwaite)

Consider any social choice function C of N and O. If:

1. $|O| \geq 3$ *(there are at least three outcomes);*
2. C *is onto; that is, for every* $o \in O$ *there is a preference vector* $\succ$ *such that* $C(\succ) = o$ *(this property is sometimes also called citizen sovereignty); and*
3. C *is dominant-strategy truthful,*

then C *is dictatorial.*

What does this mean?

- We should be discouraged about the possibility of implementing arbitrary social-choice functions in mechanisms.
- However, in practice we can circumvent the Gibbard-Satterthwaite theorem in two ways:
  - use a weaker form of implementation
    - note: the result only holds for dominant strategy implementation, not e.g., Bayes-Nash implementation
  - relax the onto condition and the (implicit) assumption that agents are allowed to hold arbitrary preferences

Lecture Overview

Quasilinear Utility

Definition (Quasilinear preferences)

Agents have quasilinear preferences in an n-player Bayesian game when the set of outcomes is $O = X \times \mathbb{R}^n$ for a finite set X, and the utility of an agent i with type θ_i is given by $u_i(o, \theta_i) = u_i(x, \theta_i) - f_i(p_i)$, where $o = (x, p_i)$ is an element of O, $u_i(x, \theta_i)$ is an arbitrary function and $f_i : \mathbb{R} \rightarrow \mathbb{R}$ is a strictly monotonically increasing function.

Quasilinear utility

- $u_i(o, \theta_i) = u_i(x, \theta_i) - f_i(p_i)$
- We split the mechanism into a choice rule and a payment rule:
  - $x \in X$ is a discrete, non-monetary outcome
  - $p_i \in \mathbb{R}$ is a monetary payment (possibly negative) that agent i must make to the mechanism
- Implications:

Quasilinear utility

- $u_i(o, \theta_i) = u_i(x, \theta_i) - f_i(p_i)$
- We split the mechanism into a choice rule and a payment rule:
  - $x \in X$ is a discrete, non-monetary outcome
  - $p_i \in \mathbb{R}$ is a monetary payment (possibly negative) that agent i must make to the mechanism
- Implications:
  - $u_i(x, \theta_i)$ is not influenced by the amount of money an agent has
  - agents don't care how much others are made to pay (though they *can* care about how the choice affects others.)

Quasilinear utility

- $u_i(o, \theta_i) = u_i(x, \theta_i) - f_i(p_i)$
- We split the mechanism into a choice rule and a payment rule:
  - $x \in X$ is a discrete, non-monetary outcome
  - $p_i \in \mathbb{R}$ is a monetary payment (possibly negative) that agent i must make to the mechanism
- Implications:
  - $u_i(x, \theta_i)$ is not influenced by the amount of money an agent has
  - agents don't care how much others are made to pay (though they *can* care about how the choice affects others.)
- What is $f_i(p_i)$?