Algorithmic Game Theory

Gonzalo Díaz

Universidad Católica de Chile

27 March 2012

Introducción

La *teoría algorítmica de juegos* es un área de investigación que combina el *diseño de algoritmos* y la *teoría de juegos*.

Juegos

- Un *juego* consiste en un conjunto de n *jugadores*, $J = \{1, 2, \ldots, n\}$.
- Cada jugador i tiene su propio conjunto de *estrategias posibles* S_i.
- Para jugar el juego, cada jugador i elige una estrategia $s_i \in S_i$.
- $s = (s_1, \ldots, s_n)$ es el *vector de estrategias* elegidas por los jugadores.
- $s \in S = S_1 \times \cdots \times S_n$. Este último conjunto contiene las combinaciones posibles de elección de estrategias de todos los jugadores.

- Cada jugador tendrá un *orden de preferencias* sobre los vectores $s \in S$. La forma usual de lograr esto es que cada jugador tenga una *función de utilidad* $u_i : S \to \mathbb{R}$ que asigna a cada vector s un número que representa su *utilidad* después de jugada la ronda.
- Para algunos juegos es más conveniente definir la función de costo $c_i : S \to \mathbb{R}$, donde un costo menor es más deseable.
- Es crucial que la función de utilidad (costo) u_i (c_i) dependa de la elección de estrategia de *todos* los jugadores (S), y no sólo del jugador i.

Ejemplo: Dilema del Prisionero

DILEMA DEL PRISIONERO es un juego donde existen dos estrategias posibles: **Confesar** y **No Confesar**.
Las funciones de costo de ambos jugadores se pueden resumir en la siguiente tabla:

	j_2 **Confiesa**	j_2 **Calla**
j_1 **Confiesa**	4, 4	5, 1
j_1 **Calla**	1, 5	2, 2

Aquí, cada entrada muestra los costos incurridos por los jugadores: (c_1, c_2).

Solución de la Estrategia Dominante

Notación: Para un vector de estrategias $s \in S$, donde s_i es la estrategia jugada por el jugador i, denotamos con s_{-i} al vector de dimensión $(n-1)$ que contiene las estrategias elegidas por los demás jugadores.
Así, como alternativa a la notación $u_i(s)$, usaremos $u_i(s_i, s_{-i})$, cuando sea conveniente.

Definición

Un vector de estrategias $s \in S$ es una *solución de estrategia dominante* ssi $\forall i, s' \in S$:

$$u_i(s_i, s'_{-i}) \;\geq\; u_i(s'_i, s'_{-i})$$

Teorema

El juego DILEMA DEL PRISIONERO tiene una solución de estrategia dominante: $s = (\textbf{Confesar}, \textbf{Confesar})$.

Demostración: Para j_i:

$$c_1(\textbf{Co}, \textbf{Co}) = 4 < c_1(\textbf{Ca}, \textbf{Co}) = 5,$$
$$c_1(\textbf{Co}, \textbf{Ca}) = 1 < c_1(\textbf{Ca}, \textbf{Ca}) = 2.$$

Para j_2:

$$c_2(\textbf{Co}, \textbf{Co}) = 4 < c_2(\textbf{Ca}, \textbf{Co}) = 5,$$
$$c_2(\textbf{Co}, \textbf{Ca}) = 1 < c_2(\textbf{Ca}, \textbf{Ca}) = 2.$$

Ejemplo: Juego de Contaminación

JUEGO DE CONTAMINACIÓN es una generalización de DILEMA DEL PRISIONERO, con múltiples jugadores. Asumimos n países. Cada país j_i puede elegir $s_i \in \{\textbf{Legislar}, \textbf{No legislar}\}$. El costo de legislar es 3, pero cada país que no legisla agrega un costo 1 a cada país.

Si en una ronda k países no legislan:

- Habrá k puntos de costo asociados a la contaminación.
- Un país que no legisló: $c = k$.
- Un país que sí legisló: $c = k + 3$.

Teorema

Juego de Contaminación tiene la siguiente solución de estrategia dominante: $s_i =$ **No legislar** $\forall i$.

Si en una ronda k países no legislan:

- Los k países que no legislan tienen costo $c_i = k$.
- Los $n - k$ países que sí legislan tienen costo $c_i = k + 3$.

En este escenario, un país que sí legisla puede mejorar su costo cambiando de estrategia. En la solución propuesta, ningún país puede cambiar su estrategia para mejorar su costo.

Nota: En la práctica, muy pocos juegos tienen una solución de estrategia dominante. El área de *market design* busca diseñar juegos con soluciones de estrategias dominantes.

Ejemplo

Para modelar una subasta:

- v_i: Precio que el jugador i le asigna (privadamente) al objeto subastado.
- $u_i = v_i - p$ si lo logra comprar a un precio p, si no, $u_i = 0$.
- s_i: Precio que ofrece el jugador i.

Cada jugador coloca su oferta en un sobre sellado y lo entrega.

A continuación, se necesita un *mecanismo* para elegir el ganador de la subasta.

Mecanismo

Definición

Un mecanismo $\mathcal{M}$ es un algortimo que, dadas las estrategias de los jugadores, calcula el resultado (utilidades).

Ejemplo

En una subasta simple, el mecanismo es el siguiente:

- j `<- Jugador que entregó oferta más grande`
- p `<-` s_j
- `Vender el objeto al jugador` j `con un precio` p

Si los jugadores no conocen las valuaciones de los demás, este juego no tiene una solución de estrategia dominante. Si los jugadores conocen tales valuaciones, la mejor estrategia del ganador es ofrecer el segundo precio más alto.

Ejemplo: Subasta de Vickrey

SUBASTA DE VICKREY (o SUBASTA DE SEGUNDO PRECIO varía el mecanismo de una subasta simple:

- j `<-` `Jugador con oferta más grande`
- k `<-` `Jugador con segunda oferta más grande`
- p `<-` s_k
- `Vender el objeto al jugador` j `con un precio` p

Incluso si los jugadores no conocen las valuaciones de los demás, la estrategia óptima para cada jugador es ofrecer su valuación real.

Muchos juegos no tienen solución de estrategia dominante. En este caso se busca un concepto de equilibrio menos restrictivo.

Equilibrio de Nash

Definición

Un vector de estrategias $s \in S$ es un *equilibrio de Nash* ssi $\forall j_i, s_i' \in S_i$:

$$u_i(s_i, s_{-i}) \geq u_i(s_i', s_{-i}).$$

Aquí, si las estrategias de los demás se mantienen constantes, ningún jugador puede cambiar de estrategia y mejorar su utilidad.

Teorema

Una solución de estrategia dominante es un equilibrio de Nash.

Equilibrio de Nash

Corolario

Los equilibrios de Nash no necesariamente son soluciones óptimas para los jugadores.

Nota: Los equilibrios de Nash no son únicos.

Ejemplo

Batalla de los Generos es un juego donde un hombre y la mujer deben decidir la actividad a realizar juntos. El hombre prefiere ver fútbol, mientras que la mujer prefiere ver tenis. Este juego tiene la siguiente tabla de utilidades:

	h **ve Fútbol**	h **ve Tenis**
m **ve Fútbol**	5, 6	1, 1
m **ve Tenis**	1, 1	6, 5

Este juego tiene dos equilibrios de Nash y cero soluciones de estrategia dominante.

Ejemplo

COMPARANDO MONEDAS Cada jugador elige si mostrar cara o sello en una moneda. Si las dos monedas son iguales, gana j_1. Si son diferentes, gana j_2:

	j_1 **Cara**	j_1 **Sello**
j_2 **Cara**	1, -1	-1, 1
j_2 **Sello**	-1, 1	1, -1

Este juego no tiene equilibrios de Nash ni estrategias dominantes.

Estrategias mixtas

Introducimos una distribución de probabilidad sobre la elección de estrategias (para cada jugador).

Ejemplo

COMPARANDO MONEDAS CON ESTRATEGIAS MIXTAS A diferencia del caso anterior, ahora j_1 eligirá **C** con probabilidad p y **S** con probabilidad $1 - p$, y similarmente para j_2, con probabilidad q. La tabla de utilidades es la misma.

Aquí se asume que el jugador buscará maximizar la ganancia esperada.

Ejemplo

Si j_1 elige $p = 1$ (juega **C** ciertamente) y j_2 elige q, las ganancias esperadas para j_1 son $(-1)(q) + (1)(1) = 1 - 2q$. Si ahora j_1 elige $p = 0$ sus ganacias esperadas serán $(1)(q) + (-1)(1 - q) = 2q - 1$.

Ejemplo

El equilibrio de Nash será para $p = q = 1/2$.

Teorema

En un juego con estrategias mixtas, siempre existe un equilibrio de Nash.

"We consider algorithmic problems in a distributed setting where the participants cannot be assumed to follow the algorithm but rather their own self-interest. As such participants, termed agents, are capable of manipulating the algorithm, the algorithm designer should ensure in advance that the agents' interests are best served by behaving correctly.[1]"

[1] Ronen, Amir (1999), "Algorithmic mechanism design", *Proceedings of the 31st ACM Symposium on Theory of Computing (STOC '99)*.

Matching Markets

Asignación de médicos y hospitales

- Jugadores: médicos y hospitales.
- Cada doctor d_i tiene una *lista de preferencias* $P(d_i)$ (ej: $P(d_1) = (h_2, h_1, \ldots)$).
- Cada hospital tiene una lista de preferencias $P(h_i)$.

Definición

Un *matching* μ es un conjunto de pares jugador-hospital, donde cada jugador y cada hospital aparecen a lo más una vez.

Notación

Para un doctor d, si el hospital h_i está antes que el hospital h_j en la lista de prefencias: $P(d) = (\ldots, h_i, \ldots, h_j, \ldots)$, se escribe:

$$h_i >_d h_j.$$

Notación

En un matching μ, si $(d, h) \in \mu$ se escribe $\mu(d) = h$ y $\mu(h) = d$.

Definición

Una pareja (d, h) es una *pareja bloqueante* de un matching μ ssi:

$$(d, h) \notin \mu \quad \wedge \quad h >_d \mu(d) \quad \wedge \quad d >_h \mu(h).$$

Definición

Un matching μ es *estable* si no existen parejas bloqueantes.

Ejemplo

Dadas las siguientes preferencias:

$$P(d_1) = (h_1, h_2), \qquad P(h_1) = (d_2, d_1),$$
$$P(d_2) = (h_1, h_2), \qquad P(h_2) = (d_1, d_2),$$

y el matching:

$$\mu = \{(d_1, h_1),\ (d_2, h_2)\},$$

la pareja (d_2, h_1) es bloqueante, por lo que μ no es un matching estable.

Truthful Matching Mechanisms

Definición

Un mecanismo $\mathcal{M}$ es *estable* si produce matchings estables.

Definición

Sea $Q(a)$ una lista de preferencias *alterada* para el jugador a (su lista de preferencias real es $P(a)$). Un mecanismo $\mathcal{M}$ es *verídico* (*truthful*) ssi $\forall a, P(a), Q(a), Q_{-a}$:

$$\mathcal{M}(a; P(a), Q_{-a}) \;\geq_a\; \mathcal{M}(a; Q(a), Q_{-a}).$$

- En un mercado real, ¿qué tenderá a suceder si se aplica un mecanismo que no es estable?

  Mercado negro.

- ¿Y si se aplica un mecanismo que no es verídico?

  Manipulación.

Definición

Definimos el siguiente mecanismo para un mercado de matchings, con hombres $M = \{m_1, \ldots, m_n\}$, mujeres W, y preferencias declaradas P:

```
Initializar W' = W
Para i = 1 hasta n
  Sea w en W' la mujer favorita del hombre m_i,
    entre las mujeres restantes W'.
  Sea la pareja del hombre m_i: mu(m_i) = w.
  W' = W' - {w}.
```

Teorema

El mecanismo anterior es verídico, y no es estable.

Definición

Para el problema de MATCHING, definimos el mecanismo **GS** (Gale, Shapley):

```
mu <- matching vacia.
while existe un hombre libre.
  h <- hombre libre.
  m <- mujer mas deseada por h que no lo ha rechazado.
  if m esta libre
    agregar (h,m) a mu.
  if m no esta libre pero prefiere h a mu(m)
    eliminar (mu(m), m) de mu.
    marcar que mu(m) fue rechazado por m.
    agregar (h,m) a mu.
  else
    marcar que h fue rechazado por m.
return mu
```

Teorema

GS es estable y no es verídico.

Ejemplo

Consideremos el siguiente mercado:

$$P(m_1) = w_1, w_2 \quad P(w_1) = m_2, m_1,$$
$$P(m_2) = w_2, w_1 \quad P(w_2) = m_1, m_2.$$

En este caso, el mecanismo **GS** entrega los siguientes resultados:

$$\mu^M \to M = \{(m_1, w_1), (m_2, w_2)\},$$
$$\mu^W \to M = \{(m_1, w_2), (m_2, w_1)\}.$$

Definición

Un matching μ se dice *débilmente Pareto-optimal* ssi:

$$\nexists \nu \left[\forall m \in M(\nu(m) >_m \mu(m)) \right].$$

Definición

Un matching μ se dice *fuertemente Pareto-optimal* ssi:

$$\nexists \nu \left[(\forall m \in M(\nu(m) \geq_m \mu(m))) \ \wedge \ (\exists m \in M(\nu(m) >_m \mu(m))) \right].$$