Computational Game Theory (CIS 620/OPIM 952)

Instructor: M. Kearns.

Graphical Games

Luis Ortiz, January 28, 2003

Overview

- Motivation and definitions
- Representation: Properties
- Algorithms

Motivation

- Multi-party games: large number of players.
- Traditional representation: matrix or normal form
  - every player "plays" with all others.
  - payoff matrix for each player grows exponentially with number of players!
- New representation: Graphical games
  - exploits "game structure"
  - limited interaction: each player only "plays" with a "small" subset of all other players.
  - More compact representation

[See accompanying PowerPoint presentation]

Some "Strategic Properties" of Graphical Games

- Problem still non-trivial: the eq. strategy of a player "affects" that of every other player (if G fully connected)
- Let X, Y subset of player, If X, Y disconnected in G, X, Y form <u>independent games</u>
- For every player i, if we "set" (the strategies for) the neighbors of i in G, we get 2 <u>independent subgames</u>: ① i by himself; ② all non-neighbors of i.

The (Conditional) eq. of each subgame are also <u>independent</u>.

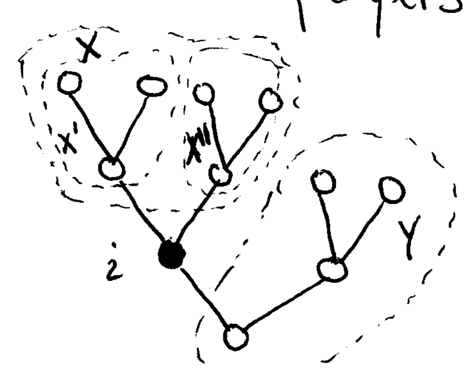
- More generally, let

  S ≡ set of players that "separates" the remaining set of players into 2 non-empty subsets X, Y.

  If we "set" the players in S, the resulting subgame (and conditional eq.) for players in X is <u>independent</u> of that for players in Y.

<u>Ex.</u>: G a tree

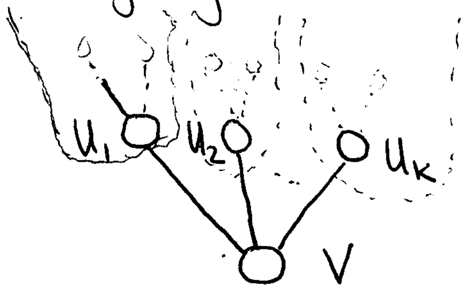
$S = \{i\}$



- (Dynamic Programing) Alg. exploits these properties.

Abstract Algorithm: Tree Case

Consider assigning a NE for root of tree



What do we need?

- "Set" $V=v$; Consider $\vec{U}=\vec{u}$, and ask
  - Is $V=v$ a best response to $\vec{U}=\vec{u}$?
  - $\forall i$, Does there exist an eq. "upstream" in which U_i plays u_i when V is "set" to v? — $T_{VU_i}(v, u_i)$
- If "yes" to all questions, $\exists$ a NE in which $V=v$ and $\vec{U}=\vec{u}$. Such a $\vec{u}$ is called a **witness** (to v)

  Otherwise, Keep trying other values for v and $\vec{u}$ until we find one!
  [NE existance $\Rightarrow$ there is at least one such setting $(v, \vec{u})$]
- For such $(v, \vec{u})$, let $V=v$ and $\vec{U}=\vec{u}$ in NE.
- Recursively, apply same "procedure" for each parent U_i

How do we get $T_{VU_i}(v, u_i)$?

Apply dynamic programming.

[See accompanying PowerPoint presentation]

Approximation Algorithm

Basic idea:

- **Discretize** mixed-strategy space (uniformly along each "dimension" $\Rightarrow$ uniform grid)

Now.

$v \in$ [grid: $0, \tau, 2\tau, \ldots, 1$]

τ-grid $\Rightarrow$ each player has $\lceil \frac{1}{\tau} \rceil$ mixed strategies to consider.

- Use approximate eq. condition:
  - replace "best-response" by "ε-best-response"
  - recall, $\vec{p}$ is ε-NE if no player can gain more than ε by unilaterally deviating from $\vec{p}$

$$\forall i, \quad \max_{a} M_i(\vec{p}[i:a]) - M_i(\vec{p}) \le \varepsilon$$

So,

- Table size: $\lceil \frac{1}{\tau} \rceil^2$
- Computation time (per player): $O\left(\lceil \frac{1}{\tau} \rceil^K\right)$

[See accompanying PowerPoint presentation for an example]

Now, How should we set τ s.t.

if $\vec{p}$ is NE, $\vec{q}$ in τ-grid, closest (in L_1) to $\vec{p}$, then $\vec{q}$ is ε-NE ?

Approximation Algorithm (Analysis)

Let $\vec{p}, \vec{q}$ joint mixed strategies ; k = (max) neighborhood size

Lemma 1: If $\forall i,\ p_i - q_i < \tau/2$, then

$$|M_i(\vec{p}) - M_i(\vec{q})| < [(1+\tau)^k - 1]/2 \leq k\tau$$

($\leq$ for $\tau < \frac{2}{k}$)

Pf: [See accompanying note]

Lemma 2: If $\vec{p}$ is NE, $\vec{q}$ in τ-grid and closest (in L_1) to $\vec{p}$, and $\tau < \frac{2}{k}$, then $\vec{q}$ is $(2k\tau)$-NE.

Pf: $\forall i$,

$$\begin{aligned} M_i(\vec{q}) &\geq M_i(\vec{p}) - k\tau && \text{(By Lemma 1)} \\ &= \max_a M_i(\vec{p}[i:a]) - k\tau && \text{(By NE defn)} \\ &\geq \max_a M_i(\vec{q}[i:a]) \underbrace{- k\tau - k\tau}_{= -2k\tau} && \text{(By Lemma 1)} \end{aligned}$$

$\therefore$ Let $\tau = \frac{\varepsilon}{2k}$. So $\lceil \frac{1}{\tau} \rceil \leq \frac{2k}{\varepsilon} + 1$

➡ Table size $\leq \left(\frac{2k}{\varepsilon} + 1\right)^2 \Rightarrow$ rep. size, poly in $\frac{1}{\varepsilon}, k, n$

$\Rightarrow$ Computation per player $\leq \left(\frac{2k}{\varepsilon} + 1\right)^k \Rightarrow$ running time poly. in $\frac{1}{\varepsilon}, n, 2^{k \log k}$

Result:
- ApproxTreeNash computes an ε-NE
- Every NE has a representative ε-NE in tables.
- Table representation size poly. in model size
- If k s.t. $k \log k = O(\log n)$, computation time also poly. in model size.

[What about multi-action games with $m > 2$?]

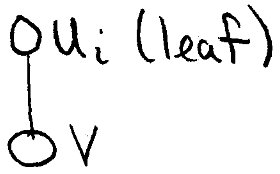
Exact Algorithm: Tree case, 2-actions [All equilibria]

Basic idea:

- Easy to compute/represent exactly the tables sent by "leaves":
  - union of "axis-parallel" "line segments"
- Use (represented) exact tables received from parents to recursively compute/represent exact tables sent to child.
  - Invariance: exact tables are finite union of "axis-parallel" "line segments"

Tables sent "down" by leaves

Consider expected payoff of leaf U_i



$$M_{U_i}(u_i, v) = u_i \underbrace{\left[M_{U_i}(1, v) - M_{U_i}(0, v) \right]}_{\Delta_{U_i}(v)} + M_{U_i}(0, v)$$

$\forall v \in [0,1]$,

$\Delta(v) > 0 \Rightarrow U_i = 1$ is best response to $V = v$

$\Delta(v) < 0 \Rightarrow U_i = 0$ "

$\Delta(v) = 0 \Rightarrow U_i = u'$ " , $\forall u' \in [0,1]$

["U_i is indifferent to $V = v$"]

How can we find "indifference" value v'?
[if it exists...]

Exact Alg. (Continued)

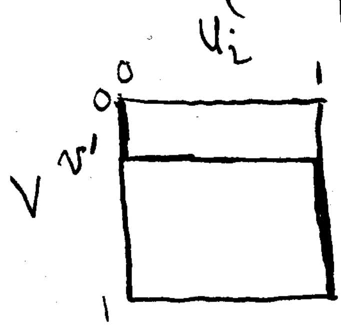
- Finding "indifference" value v' [Recall, $M_{u_i}(u_i, V)$]

$$\Delta(v) = v\underbrace{\left[M_{u_i}(1,1) - M_{u_i}(1,0) - \left(M_{u_i}(0,1) - M_{u_i}(0,0)\right)\right]}_{b} + \underbrace{M_{u_i}(1,0) + M_{u_i}(0,0)}_{c}$$

$\Delta(v') = 0$ iff either $b = 0 = c$ or

$$v' = \frac{-c}{b}, \quad b \neq 0.$$

(only care about $v \in [0,1]$!)



Let $u' : [0,1] \to \{0,1\}$ be indicator function $u'(v) \equiv I(\Delta(v) > 0)$
$\forall (v, u_i) \in [0,1]^2$,

$T_{Vu_i}(v, u_i) = 1$ iff

- $v \in [0, v']$ and $u_i \in [u'(v), u'(v)]$, or
- $v \in [v', v']$ and $u_i \in [0,1]$ or
- $v \in [v', 1]$ and $u_i \in [u'(v), u'(v)]$

u_i

V: $0 = v_0 \le v' = v_1 \le v' = v_2 \le 1 = v_3$, with $[v_0, v_1] = I_0$, $[v_1, v_2] = I_1$, $[v_2, v_3] = I_2$

v-list representation of T_{Vu_i}

In general, a sequence of points in $[0,1]$

$$0 = v_0 \le v_1 \le \ldots \le v_m = 1$$

and $\forall \ell = 0, \ldots, m$, $[v_\ell, v_{\ell+1}]$ ↦ union of intervals in $[0,1]$

$$I_1^{i,\ell} \cup \ldots \cup I_t^{i,\ell}$$

[Remark: At leaves, $t = 1$]

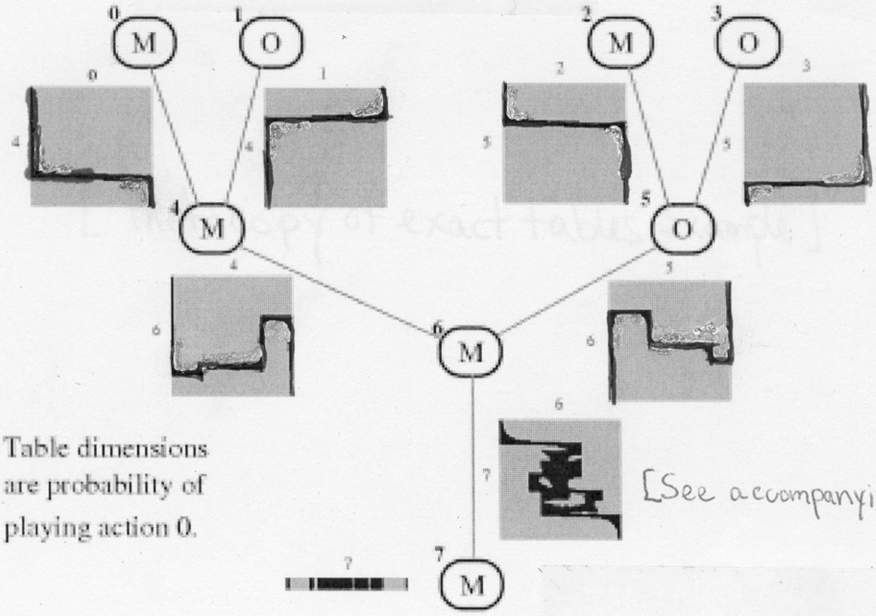
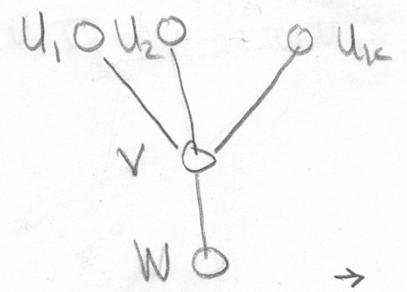


Table dimensions are probability of playing action 0.

[See accompanying PowerPoint]

- Consider



- Merge v-lists from all parents.
- Consider $v \in [v_\ell, v_{\ell+1}]$

$\vec{u} \in I_1 \times \cdots \times I_k$, $I_i \in \{ I^{i,\ell}_{j}, j = 1, \ldots t_i \}$

How do we find values for w s.t. V is <u>indifferent</u> $\forall v \in [v_\ell, v_{\ell+1}]$?

<u>Same</u> <u>idea</u>: $M_v(v, w, \vec{u}) = v\left[M_v(1, w, \vec{u}) - M_v(0, w, \vec{u})\right] + M_v(0, w, \vec{u})$

$$\begin{aligned}\Delta_v(w, \vec{u}) &= M_v(1, w, \vec{u}) - M_v(0, w, \vec{u}) \\ &= w\left[M_v(1,1,\vec{u}) - M_v(1,0,\vec{u}) - \left(M_v(0,1,\vec{u}) - M_v(0,0,\vec{u})\right)\right] \\ &\quad + M_v(1,0,\vec{u}) - M_v(0,0,\vec{u})\end{aligned}$$

So we want

$$w \in W = \{ w \in [0,1] : \exists \vec{u} \in I_1 \times \cdots \times I_k \text{ s.t. } \Delta(w, \vec{u}) = 0 \}$$

[See accompanying paper by Kearns et. al., 2001]

- Only need to check extremal points of $I_1 \times \cdots \times I_k$!

Exact Alg.

Summary:

- Can show size of tables grow exponentially with number of players [See Kearns et.al. 2001]
- Exact alg. computes a representation of all exact NE in a tree graphical game in time exponential in model size.
- Possible to generate NE from the resulting tables.

Exact Algorithm : Tree case, 2-action, single NE.

- [See acompanying paper by Littman et. al. 2002]
- Alg. computes single exact NE in 2-action, tree graphical games in time poly in model size.
- Basic idea:
  - Pick only one "path" in table $T(w,v)$ s.t. $\forall w, \exists v$ s.t. $T(w,v)=1$.
    [ignore others]
  - Which "path" should select?
    The one with minimum number of "turns"