Recursive methods in Stochastic Games

Johannes Hörner[1], Satoru Takahashi[2],
Takuo Sugaya[2] and Nicolas Vieille[3]

[1]Yale

[2]Princeton

[3]HEC

Distributed Decisions via Games and Price Mechanisms,
Lund

March 13th, 2010

Outline

- Introduction
- A formal setup
- A simple example
- Two results
- Related results
- Back to the example

Introduction

Stochastic games are dynamic (discrete-time) games in which current play influences the evolution of a payoff-relevant state variable.

Introduction

Stochastic games are dynamic (discrete-time) games in which current play influences the evolution of a payoff-relevant state variable.

Very little is known on the set of equilibrium payoffs in discounted stochastic games.

Introduction

Stochastic games are dynamic (discrete-time) games in which current play influences the evolution of a payoff-relevant state variable.

Very little is known on the set of equilibrium payoffs in discounted stochastic games.

Our objective is to characterize limit set of equilibrium payoffs (as players become very patient). We do so, under some rather strong assumptions on the transitions.

Setup

We consider stochastic games with public signals.

- I is the set of players.
- S is the set of possible states.
- A^i is the action set of player i, and $A := \prod_{i \in I} A^i$.
- Y is the set of public signals.

Setup

We consider stochastic games with public signals.

- I is the set of players.
- S is the set of possible states.
- A^i is the action set of player i, and $A := \prod_{i \in I} A^i$.
- Y is the set of public signals.
- $r : S \times A \to \mathbf{R}^I$ (stage) payoff function: $r(s, a)$ is the payoff vector when playing $a \in A$ in state s.

Setup

We consider stochastic games with public signals.

- I is the set of players.
- S is the set of possible states.
- A^i is the action set of player i, and $A := \prod_{i \in I} A^i$.
- Y is the set of public signals.
- $r : S \times A \to \mathbf{R}^I$ (stage) payoff function: $r(s, a)$ is the payoff vector when playing $a \in A$ in state s.
- $p(t, y|s, a)$ is the probability of moving to $t \in S$ and of getting $y \in Y$ when playing a in state s.

All sets are finite.

Setup

We consider stochastic games with public signals.

- I is the set of players.
- S is the set of possible states.
- A^i is the action set of player i, and $A := \prod_{i \in I} A^i$.
- Y is the set of public signals.
- $r : S \times A \to \mathbf{R}^I$ (stage) payoff function: $r(s, a)$ is the payoff vector when playing $a \in A$ in state s.
- $p(t, y|s, a)$ is the probability of moving to $t \in S$ and of getting $y \in Y$ when playing a in state s.

All sets are finite.

At stage n, players choose $(a_n^i)_{i \in I}$, nature chooses the pair $(s_{n+1}, y_n) \sim p(\cdot|s_n, a_n)$, which is publicly disclosed. The game then moves to stage $n+1$.

Setup

We consider stochastic games with public signals.

- I is the set of players.
- S is the set of possible states.
- A^i is the action set of player i, and $A := \prod_{i \in I} A^i$.
- Y is the set of public signals.
- $r : S \times A \to \mathbf{R}^I$ (stage) payoff function: $r(s,a)$ is the payoff vector when playing $a \in A$ in state s.
- $p(t, y|s, a)$ is the probability of moving to $t \in S$ and of getting $y \in Y$ when playing a in state s.

All sets are finite.

At stage n, players choose $(a_n^i)_{i \in I}$, nature chooses the pair $(s_{n+1}, y_n) \sim p(\cdot|s_n, a_n)$, which is publicly disclosed. The game then moves to stage $n+1$.

Player i maximizes expectation of $(1-\delta)\sum_{n=1}^{+\infty} \delta^{n-1} r(s_n, a_n)$.

Setup (2)

We focus on subgame perfect equilibria in public strategies (PPE): these are strategies that only depend on *public information* (public signals + past and present states).

Setup (2)

We focus on subgame perfect equilibria in public strategies (PPE): these are strategies that only depend on *public information* (public signals + past and present states).

We denote by $E_\delta(s) \subset \mathbf{R}^I$ the set of PPE payoffs, when the initial state is s.

Setup (2)

We focus on subgame perfect equilibria in public strategies (PPE): these are strategies that only depend on *public information* (public signals + past and present states).

We denote by $E_\delta(s) \subset \mathbf{R}^I$ the set of PPE payoffs, when the initial state is s.

Assumption: For any $\vec{a} = (a_s) \in A^S$, the Markov chain over S with transition function $p(t|s, a_s)$ is irreducible.

Setup (2)

We focus on subgame perfect equilibria in public strategies (PPE): these are strategies that only depend on *public information* (public signals + past and present states).

We denote by $E_\delta(s) \subset \mathbf{R}^I$ the set of PPE payoffs, when the initial state is s.

Assumption: For any $\vec{a} = (a_s) \in A^S$, the Markov chain over S with transition function $p(t|s, a_s)$ is irreducible.

Then distance between $E_\delta(s)$ and $E_\delta(t)$ goes to 0.

A simple example

$1, 1$
$0, 3$

$1, 1$	$3, 0$

- Actions are denoted a and b;

A simple example

$1, 1$
$0, 3$

$1, 1$	$3, 0$

- Actions are denoted a and b;
- If a, state changes with probability $\frac{2}{3}$;
- If b, state changes with probability $\frac{1}{3}$.

A simple example

$1, 1$
$0, 3$

$1, 1$	$3, 0$

- Actions are denoted a and b;
- If a, state changes with probability $\frac{2}{3}$;
- If b, state changes with probability $\frac{1}{3}$.
- minmax payoff is 1 for each player.

A simple example

$1, 1$
$0, 3$

$1, 1$	$3, 0$

- Actions are denoted *a* and *b*;
- If *a*, state changes with probability $\frac{2}{3}$;
- If *b*, state changes with probability $\frac{1}{3}$.
- minmax payoff is 1 for each player.

$(1, 1)$ is an obvious equilibrium payoff.

A simple example

$1,1$
$0,3$

$1,1$	$3,0$

- Actions are denoted *a* and *b*;
- If *a*, state changes with probability $\frac{2}{3}$;
- If *b*, state changes with probability $\frac{1}{3}$.
- minmax payoff is 1 for each player.

$(1,1)$ is an obvious equilibrium payoff.

Are there others ?

A simple example

$1,1$
$0,3$

$1,1$	$3,0$

- Actions are denoted a and b;
- If a, state changes with probability $\frac{2}{3}$;
- If b, state changes with probability $\frac{1}{3}$.
- minmax payoff is 1 for each player.

$(1,1)$ is an obvious equilibrium payoff.

Are there others ?

Assume that states, and only states, are publicly disclosed.

A simple example

$1,1$
$0,3$

$1,1$	$3,0$

- Actions are denoted a and b;
- If a, state changes with probability $\frac{2}{3}$;
- If b, state changes with probability $\frac{1}{3}$.
- minmax payoff is 1 for each player.

$(1,1)$ is an obvious equilibrium payoff.

Are there others ?

Assume that states, and only states, are publicly disclosed.

Player 1 gets higher payoffs in state 2 than in state 1.

A simple example

$1,1$
$0,3$

$1,1$	$3,0$

- Actions are denoted a and b;
- If a, state changes with probability $\frac{2}{3}$;
- If b, state changes with probability $\frac{1}{3}$.
- minmax payoff is 1 for each player.

$(1,1)$ is an obvious equilibrium payoff.

Are there others ?

Assume that states, and only states, are publicly disclosed.

Player 1 gets higher payoffs in state 2 than in state 1.

And playing a increases the probability of moving to state 2.

A simple example

$1, 1$
$0, 3$

$1, 1$	$3, 0$

A simple example

$1, 1$
$0, 3$

$1, 1$	$3, 0$

Player 1 may be willing to play *b* when in state 1, only if provided with a *higher* continuation payoff, should the play remain in state 1.

A simple example

$1, 1$
$0, 3$

$1, 1$	$3, 0$

Player 1 may be willing to play *b* when in state 1, only if provided with a *higher* continuation payoff, should the play remain in state 1.

Equilibrium payoffs other than $(1, 1)$ thus require playing a string of *b*, then of *a*'s when in state 1, and adjusting continuation payoffs.

This is tricky...

The optimization program $\mathcal{P}(\lambda)$

The optimization program $\mathcal{P}(\lambda)$

Given weights $\lambda \in \mathbf{R}^I$, the highest equilibrium payoff in the direction λ is obtained as the value of an optimization problem.

The optimization program $\mathcal{P}(\lambda)$

Given weights $\lambda \in \mathbf{R}^I$, the highest equilibrium payoff in the direction λ is obtained as the value of an optimization problem.

A notation: Let be given $x : S \times Y \to \mathbf{R}^{S \times I}$: $x_t^i(s, y)$ is continuation payoff for player i if next state is t, when coming from s and getting y.

The optimization program $\mathcal{P}(\lambda)$

Given weights $\lambda \in \mathbf{R}^I$, the highest equilibrium payoff in the direction λ is obtained as the value of an optimization problem.

A notation: Let be given $x : S \times Y \to \mathbf{R}^{S \times I}$: $x_t^i(s,y)$ is continuation payoff for player i if next state is t, when coming from s and getting y.

We denote by $\Gamma(s,x)$ the (Shapley) one-shot game with payoffs

$$r(s,a) + \sum_{t \in S, y \in Y} p(t,y|s,a) x_t(s,y).$$

The optimization program $\mathcal{P}(\lambda)$

Given weights $\lambda \in \mathbf{R}^I$, the highest equilibrium payoff in the direction λ is obtained as the value of an optimization problem.

A notation: Let be given $x : S \times Y \to \mathbf{R}^{S \times I}$: $x_t^i(s, y)$ is continuation payoff for player i if next state is t, when coming from s and getting y.

We denote by $\Gamma(s, x)$ the (Shapley) one-shot game with payoffs

$$r(s, a) + \sum_{t \in S, y \in Y} p(t, y | s, a) x_t(s, y).$$

Define $\mathcal{P}(\lambda)$:

$$\sup \lambda \cdot v,$$

The optimization program $\mathcal{P}(\lambda)$

Given weights $\lambda \in \mathbf{R}^I$, the highest equilibrium payoff in the direction λ is obtained as the value of an optimization problem.

A notation: Let be given $x : S \times Y \to \mathbf{R}^{S \times I}$: $x_t^i(s, y)$ is continuation payoff for player i if next state is t, when coming from s and getting y.

We denote by $\Gamma(s, x)$ the (Shapley) one-shot game with payoffs

$$r(s, a) + \sum_{t \in S, y \in Y} p(t, y | s, a) x_t(s, y).$$

Define $\mathcal{P}(\lambda)$:

$$\sup \lambda \cdot v,$$

where the supremum is over all $v \in \mathbf{R}^I$, and all $x : S \times Y \to \mathbf{R}^{S \times I}$, such that:

(i) For each s, v is a N.E. payoff of $\Gamma(s, x)$.

The optimization program $\mathcal{P}(\lambda)$

Given weights $\lambda \in \mathbf{R}^I$, the highest equilibrium payoff in the direction λ is obtained as the value of an optimization problem.

A notation: Let be given $x : S \times Y \to \mathbf{R}^{S \times I}$: $x_t^i(s,y)$ is continuation payoff for player i if next state is t, when coming from s and getting y.

We denote by $\Gamma(s,x)$ the (Shapley) one-shot game with payoffs

$$r(s,a) + \sum_{t \in S, y \in Y} p(t,y|s,a) x_t(s,y).$$

Define $\mathcal{P}(\lambda)$:

$$\sup \lambda \cdot v,$$

where the supremum is over all $v \in \mathbf{R}^I$, and all $x : S \times Y \to \mathbf{R}^{S \times I}$, such that:

(i) For each s, v is a N.E. payoff of $\Gamma(s,x)$.

(ii) For every $T \subseteq S$, every permutation ϕ over T, every map $\psi : T \to Y$, one has

$$\lambda \cdot \sum x_{\phi(s)}(s, \psi(s)) \leq 0.$$

The optimization program $\mathcal{P}(\lambda)$

Given weights $\lambda \in \mathbf{R}^I$, the highest equilibrium payoff in the direction λ is obtained as the value of an optimization problem.

A notation: Let be given $x : S \times Y \to \mathbf{R}^{S \times I}$: $x_t^i(s, y)$ is continuation payoff for player i if next state is t, when coming from s and getting y.

We denote by $\Gamma(s, x)$ the (Shapley) one-shot game with payoffs

$$r(s, a) + \sum_{t \in S, y \in Y} p(t, y | s, a) x_t(s, y).$$

Define $\mathcal{P}(\lambda)$:

$$\sup \lambda \cdot v,$$

where the supremum is over all $v \in \mathbf{R}^I$, and all $x : S \times Y \to \mathbf{R}^{S \times I}$, such that:

(i) For each s, v is a N.E. payoff of $\Gamma(s, x)$.

(ii) For every $T \subseteq S$, every permutation ϕ over T, every map $\psi : T \to Y$, one has

$$\lambda \cdot \sum x_{\phi(s)}(s, \psi(s)) \leq 0.$$

The Results – A characterization

The Results – A characterization

Denote by $k(\lambda)$ the value of $\mathcal{P}(\lambda)$.

Set $\mathcal{H} = \{v : \lambda \cdot v \leq k(\lambda)$ for every $\lambda \in \mathbf{R}^I\}$.

Denote by $k(\lambda)$ the value of $\mathcal{P}(\lambda)$.

Set $\mathcal{H} = \{v : \lambda \cdot v \leq k(\lambda)$ for every $\lambda \in \mathbf{R}^I\}$.

Then $\mathcal{H} = \lim_{\delta \to 1} E_\delta(s)$.

The Results – A characterization

Denote by $k(\lambda)$ the value of $\mathcal{P}(\lambda)$.

Set $\mathcal{H} = \{v : \lambda \cdot v \leq k(\lambda) \text{ for every } \lambda \in \mathbf{R}^l\}$.

Then $\mathcal{H} = \lim_{\delta \to 1} E_\delta(s)$. Formally,

Theorem

$\limsup_{\delta \to 1} E_\delta(s) \subseteq \mathcal{H}$.

The Results – A characterization

Denote by $k(\lambda)$ the value of $\mathcal{P}(\lambda)$.

Set $\mathcal{H} = \{v : \lambda \cdot v \leq k(\lambda) \text{ for every } \lambda \in \mathbf{R}^I\}$.

Then $\mathcal{H} = \lim_{\delta \to 1} E_\delta(s)$. Formally,

Theorem

$\limsup_{\delta \to 1} E_\delta(s) \subseteq \mathcal{H}$.

Theorem

Assume that $\mathcal{H}$ has non-empty interior. Then, for every compact set W contained in the interior of $\mathcal{H}$, one has $W \subset E_\delta(s)$ for every high enough δ.

The Results – A characterization

Denote by $k(\lambda)$ the value of $\mathcal{P}(\lambda)$.

Set $\mathcal{H} = \{v : \lambda \cdot v \leq k(\lambda) \text{ for every } \lambda \in \mathbf{R}^I\}$.

Then $\mathcal{H} = \lim_{\delta \to 1} E_\delta(s)$. Formally,

Theorem

$\limsup_{\delta \to 1} E_\delta(s) \subseteq \mathcal{H}$.

Theorem

Assume that $\mathcal{H}$ has non-empty interior. Then, for every compact set W contained in the interior of $\mathcal{H}$, one has $W \subset E_\delta(s)$ for every high enough δ.

Extends to the case where some of the player are short-run players.

When does the limit set coincide with the set of feasible and individually rational payoffs ?

A Folk Theorem

When does the limit set coincide with the set of feasible and individually rational payoffs ?

- Not all feasible payoffs are equilibrium payoffs of the static game !

When does the limit set coincide with the set of feasible and individually rational payoffs ?

- Not all feasible payoffs are equilibrium payoffs of the static game !
- To implement such payoffs, players must deter deviations, and thus must condition their action choices on past play.

When does the limit set coincide with the set of feasible and individually rational payoffs ?

- Not all feasible payoffs are equilibrium payoffs of the static game !
- To implement such payoffs, players must deter deviations, and thus must condition their action choices on past play.
- Hence, one must assume public information to be sufficiently informative.

A Folk Theorem

When does the limit set coincide with the set of feasible and individually rational payoffs ?

- Not all feasible payoffs are equilibrium payoffs of the static game !
- To implement such payoffs, players must deter deviations, and thus must condition their action choices on past play.
- Hence, one must assume public information to be sufficiently informative.

Define

- $\Pi^i(s, \alpha_s)$ is the $|A^i| \times |S \times Y|$ matrix with entries $p(t, y|s, a^i, \alpha_s^{-i})$: the a^i-row of Π^i contains the (joint) distribution of the public information (next state, public signal).

When does the limit set coincide with the set of feasible and individually rational payoffs ?

- Not all feasible payoffs are equilibrium payoffs of the static game !
- To implement such payoffs, players must deter deviations, and thus must condition their action choices on past play.
- Hence, one must assume public information to be sufficiently informative.

Define

- $\Pi^i(s, \alpha_s)$ is the $|A^i| \times |S \times Y|$ matrix with entries $p(t, y|s, a^i, \alpha_s^{-i})$: the a^i-row of Π^i contains the (joint) distribution of the public information (next state, public signal).
- $\Pi^{ij}(s, \alpha_s)$ is obtained by stacking $\Pi^i(s, \alpha_s)$ and $\Pi^j(s, \alpha_s)$.

When does the limit set coincide with the set of feasible and individually rational payoffs ?

- Not all feasible payoffs are equilibrium payoffs of the static game !
- To implement such payoffs, players must deter deviations, and thus must condition their action choices on past play.
- Hence, one must assume public information to be sufficiently informative.

Define

- $\Pi^i(s, \alpha_s)$ is the $|A^i| \times |S \times Y|$ matrix with entries $p(t, y|s, a^i, \alpha_s^{-i})$: the a^i-row of Π^i contains the (joint) distribution of the public information (next state, public signal).
- $\Pi^{ij}(s, \alpha_s)$ is obtained by stacking $\Pi^i(s, \alpha_s)$ and $\Pi^j(s, \alpha_s)$.

Definition (Statistic Identifiability Conditions)

α_s has individual full rank for i at s if $\Pi^i(s, \alpha)$ has rank $|A^i|$. It has pairwise full rank for players i and j at state s if $\Pi^{ij}(s, \alpha)$ has rank $|A^i| + |A^j| - 1$.

A Folk Theorem

Definition (Statistic Identifiability Conditions)

α_s has individual full rank for i at s if $\Pi^i(s,\alpha)$ has rank $|A^i|$. It has pairwise full rank for players i and j at state s if $\Pi^{ij}(s,\alpha)$ has rank $|A^i| + |A^j| - 1$.

ifr means that public signals allow to identify (statistically) the action of player i;

pfr means moreover that players can tell which of i and j deviated.

A Folk Theorem

Definition (Statistic Identifiability Conditions)

α_s has individual full rank for i at s if $\Pi^i(s,\alpha)$ has rank $|A^i|$. It has pairwise full rank for players i and j at state s if $\Pi^{ij}(s,\alpha)$ has rank $|A^i| + |A^j| - 1$.

ifr means that public signals allow to identify (statistically) the action of player i;

pfr means moreover that players can tell which of i and j deviated.

Theorem (loose)

Under ifr and pfr, $E_\delta(s)$ converges to the set of feasible and IR payoffs (if it has non-empty interior).

Related literature

Related literature

- $|S| = 1$: *Repeated Games with Public Monitoring*.
  Characterization +FT
  Fudenberg, Levine, Maskin (Econ, 1994), Fudenberg, Levine (JET, 1994)

Related literature

- $|S| = 1$: *Repeated Games with Public Monitoring*. Characterization +FT
  Fudenberg, Levine, Maskin (Econ, 1994), Fudenberg, Levine (JET, 1994)
- $Y = A$: *Stochastic Games with Full monitoring*. A Folk Theorem
  Dutta (JET, 1995).

Related literature

- $|S| = 1$: *Repeated Games with Public Monitoring*. Characterization +FT
  Fudenberg, Levine, Maskin (Econ, 1994), Fudenberg, Levine (JET, 1994)
- $Y = A$: *Stochastic Games with Full monitoring*. A Folk Theorem
  Dutta (JET, 1995).
- $|I| = 1$: *Dynamic Programming*. ACOE.
  Hoffman-Karp (MS, 1966).

Related literature

- $|S| = 1$: *Repeated Games with Public Monitoring*. Characterization +FT
  Fudenberg, Levine, Maskin (Econ, 1994), Fudenberg, Levine (JET, 1994)
- $Y = A$: *Stochastic Games with Full monitoring*. A Folk Theorem
  Dutta (JET, 1995).
- $|I| = 1$: *Dynamic Programming*. ACOE.
  Hoffman-Karp (MS, 1966).

When specialized, our results yield exactly these existing results, and provide a unified proof.

Related literature

- $|S| = 1$: *Repeated Games with Public Monitoring*. Characterization +FT
  Fudenberg, Levine, Maskin (Econ, 1994), Fudenberg, Levine (JET, 1994)
- $Y = A$: *Stochastic Games with Full monitoring*. A Folk Theorem
  Dutta (JET, 1995).
- $|I| = 1$: *Dynamic Programming*. ACOE.
  Hoffman-Karp (MS, 1966).

When specialized, our results yield exactly these existing results, and provide a unified proof.
Also imply results by Fudenberg-Yamamoto (2009-10).

Related literature

- $|S| = 1$: *Repeated Games with Public Monitoring*. Characterization +FT
  Fudenberg, Levine, Maskin (Econ, 1994), Fudenberg, Levine (JET, 1994)
- $Y = A$: *Stochastic Games with Full monitoring*. A Folk Theorem
  Dutta (JET, 1995).
- $|I| = 1$: *Dynamic Programming*. ACOE.
  Hoffman-Karp (MS, 1966).

When specialized, our results yield exactly these existing results, and provide a unified proof.
Also imply results by Fudenberg-Yamamoto (2009-10).

The example – again

$1, 1$
$0, 3$

$1, 1$	$3, 0$

- If a, state changes with probability $\frac{2}{3}$;
- If b, state changes with probability $\frac{1}{3}$.

The example – again

$1,1$
$0,3$

$1,1$	$3,0$

- If a, state changes with probability $\frac{2}{3}$;
- If b, state changes with probability $\frac{1}{3}$.
- Feasible set is a losange with vertices $(1,1)$, $(\frac{3}{2},\frac{3}{2})$, $(\frac{1}{3},\frac{7}{3})$, $(\frac{7}{3},\frac{1}{3})$.
- Full rank assumptions are satisfied.

The example – again

$1,1$
$0,3$

$1,1$	$3,0$

- If a, state changes with probability $\frac{2}{3}$;
- If b, state changes with probability $\frac{1}{3}$.
- Feasible set is a losange with vertices $(1,1)$, $(\frac{3}{2},\frac{3}{2})$, $(\frac{1}{3},\frac{7}{3})$, $(\frac{7}{3},\frac{1}{3})$.
- Full rank assumptions are satisfied.

Hence, limit set of equilibrium payoffs is the set of all payoffs in this losange, which lie above $(1,1)$.

The example – again

$1,1$
$0,3$

$1,1$	$3,0$

- If a, state changes with probability $\frac{2}{3}$;
- If b, state changes with probability $\frac{1}{3}$.
- Feasible set is a losange with vertices $(1,1)$, $(\frac{3}{2},\frac{3}{2})$, $(\frac{1}{3},\frac{7}{3})$, $(\frac{7}{3},\frac{1}{3})$.
- Full rank assumptions are satisfied.

Hence, limit set of equilibrium payoffs is the set of all payoffs in this losange, which lie above $(1,1)$.

We still don't know how to construct equilibrium strategies...

Conclusion

We characterize the (limit) of equilibrium payoffs in stochastic games, when players get very patient.

Requires solving infinitely many linear programs.

In practice, guess and check.

Conclusion

We characterize the (limit) of equilibrium payoffs in stochastic games, when players get very patient.

Requires solving infinitely many linear programs.

In practice, guess and check.

Extensions:

- Do we need all these constraints ?
- Continuous state space: work in progress.

Where do the constraints come from ? – A reminder

Where do the constraints come from ? – A reminder

- Fix a repeated game, with payoffs $r(\cdot)$, and δ.
- Highest PPE payoff in the direction λ solves $\sup \lambda \cdot v$, subject to the constraints
  - α NE with payoff v of the Shapley game with payoff

$$(1-\delta)r(a) + \delta \sum_{y} p(y|a)w(y).$$

Where do the constraints come from ? – A reminder

- Fix a repeated game, with payoffs $r(\cdot)$, and δ.
- Highest PPE payoff in the direction λ solves $\sup \lambda \cdot v$, subject to the constraints
  - α NE with payoff v of the Shapley game with payoff

$$(1-\delta)r(a) + \delta \sum_y p(y|a)w(y).$$

  - $\lambda \cdot w(y) \leq \lambda \cdot v$ for each y

Where do the constraints come from ? – A reminder

- Fix a repeated game, with payoffs $r(\cdot)$, and δ.
- Highest PPE payoff in the direction λ solves $\sup \lambda \cdot v$, subject to the constraints
  - α NE with payoff v of the Shapley game with payoff

$$(1-\delta)r(a) + \delta \sum_y p(y|a)w(y).$$

  - $\lambda \cdot w(y) \leq \lambda \cdot v$ for each y

Setting $x(y) = \frac{\delta}{1-\delta}(w(y) - v)$, this is equivalent to the program $\sup \lambda \cdot v$, subject to

- α NE with payoff v of the Shapley game with payoff

$$r(a) + \sum_y p(y|a)x(y).$$

Where do the constraints come from ? – A reminder

- Fix a repeated game, with payoffs $r(\cdot)$, and δ.
- Highest PPE payoff in the direction λ solves $\sup \lambda \cdot v$, subject to the constraints
  - α NE with payoff v of the Shapley game with payoff

$$(1-\delta)r(a) + \delta \sum_y p(y|a)w(y).$$

  - $\lambda \cdot w(y) \leq \lambda \cdot v$ for each y

Setting $x(y) = \frac{\delta}{1-\delta}(w(y) - v)$, this is equivalent to the program $\sup \lambda \cdot v$, subject to

- α NE with payoff v of the Shapley game with payoff

$$r(a) + \sum_y p(y|a)x(y).$$

- $\lambda \cdot x(y) \leq 0$ for each y.

- Fix a repeated game, with payoffs $r(\cdot)$, and δ.
- Highest PPE payoff in the direction λ solves $\sup \lambda \cdot v$, subject to the constraints
  - α NE with payoff v of the Shapley game with payoff

$$(1-\delta)r(a) + \delta \sum_y p(y|a)w(y).$$

  - $\lambda \cdot w(y) \leq \lambda \cdot v$ for each y

Setting $x(y) = \frac{\delta}{1-\delta}(w(y) - v)$, this is equivalent to the program $\sup \lambda \cdot v$, subject to

- α NE with payoff v of the Shapley game with payoff

$$r(a) + \sum_y p(y|a)x(y).$$

- $\lambda \cdot x(y) \leq 0$ for each y.

The new program is *independent* of δ.

Where do the constraints come from ? – A relaxation

Where do the constraints come from ? – A relaxation

Natural adaptation: highest PPE payoff in direction λ solves $\sup \lambda \cdot v_s$, subject to

- α_s NE with payoff v_s of the Shapley game with payoff

$$(1-\delta)r(s,a) + \delta \sum_{t,y} p(t,y|a) w_t(s,y).$$

Where do the constraints come from ? – A relaxation

Natural adaptation: highest PPE payoff in direction λ solves $\sup \lambda \cdot v_s$, subject to

- α_s NE with payoff v_s of the Shapley game with payoff

$$(1-\delta)r(s,a) + \delta \sum_{t,y} p(t,y|a) w_t(s,y).$$

- $\lambda \cdot w_t(s,y) \leq \lambda \cdot v_t$ for each t, y.

This program is *not independent* of δ.

Where do the constraints come from ? – A relaxation

Natural adaptation: highest PPE payoff in direction λ solves $\sup \lambda \cdot v_s$, subject to

- α_s NE with payoff v_s of the Shapley game with payoff

$$(1-\delta)r(s,a) + \delta \sum_{t,y} p(t,y|a)w_t(s,y).$$

- $\lambda \cdot w_t(s,y) \leq \lambda \cdot v_t$ for each t, y.

This program is *not independent* of δ.

It becomes independent if one relaxes the last constraint to

$$\sum_{s \in T} \lambda \cdot \left(w_{\phi(s)}(s, y_s) - v_{\phi(s)}\right) \leq 0,$$

Where do the constraints come from ? – A relaxation

Natural adaptation: highest PPE payoff in direction λ solves $\sup \lambda \cdot v_s$, subject to

- α_s NE with payoff v_s of the Shapley game with payoff

$$(1-\delta)r(s,a)+\delta\sum_{t,y}p(t,y|a)w_t(s,y).$$

- $\lambda \cdot w_t(s,y) \leq \lambda \cdot v_t$ for each t,y.

This program is *not independent* of δ.

It becomes independent if one relaxes the last constraint to

$$\sum_{s\in T}\lambda\cdot\left(w_{\phi(s)}(s,y_s)-v_{\phi(s)}\right)\leq 0,$$

(for each $T\subseteq S, \phi\in\sigma(T)$ – quantifier will be omitted henceforth).

Where do the constraints come from ? – A relaxation

Natural adaptation: highest PPE payoff in direction λ solves $\sup \lambda \cdot v_s$, subject to

- α_s NE with payoff v_s of the Shapley game with payoff

$$(1-\delta)r(s,a) + \delta \sum_{t,y} p(t,y|a)w_t(s,y).$$

- $\lambda \cdot w_t(s,y) \leq \lambda \cdot v_t$ for each t,y.

This program is *not independent* of δ.

It becomes independent if one relaxes the last constraint to

$$\sum_{s\in T} \lambda \cdot \left(w_{\phi(s)}(s,y_s) - v_{\phi(s)}\right) \leq 0,$$

(for each $T \subseteq S, \phi \in \sigma(T)$ – quantifier will be omitted henceforth).

Our results show that it is the *right* relaxation.

The value of $\mathcal{P}(\lambda)$ is finite

The value of $\mathcal{P}(\lambda)$ is finite

Let q be an irreducible transition function over S, with invariant measure μ.

The value of $\mathcal{P}(\lambda)$ is finite

Let q be an irreducible transition function over S, with invariant measure μ. Builds upon Freidlin-Wenzell.

The value of $\mathcal{P}(\lambda)$ is finite

Let q be an irreducible transition function over S, with invariant measure μ. Builds upon Freidlin-Wenzell.

- For $s \in S$, a s-graph is a rooted tree over S with root s, where all states lead to s. $G(s)$ is the set of all s-graphs.

The value of $\mathcal{P}(\lambda)$ is finite

Let q be an irreducible transition function over S, with invariant measure μ. Builds upon Freidlin-Wenzell.

- For $s \in S$, a s-graph is a rooted tree over S with root s, where all states lead to s. $G(s)$ is the set of all s-graphs.
- Set $q(g) := \prod_{(t,u)\in g} q(u|t)$. Then $\mu(s) = \frac{1}{D}\sum_{g\in G(s)} q(g)$.

The value of $\mathcal{P}(\lambda)$ is finite

Let q be an irreducible transition function over S, with invariant measure μ. Builds upon Freidlin-Wenzell.

- For $s \in S$, a s-graph is a rooted tree over S with root s, where all states lead to s. $G(s)$ is the set of all s-graphs.
- Set $q(g) := \prod_{(t,u)\in g} q(u|t)$. Then $\mu(s) = \frac{1}{D} \sum_{g\in G(s)} q(g)$.

Corollary

There are $\eta_{T,\phi} \geq 0$ s.t., for each $(y_t(s)) \in \mathbf{R}^{S\times S}$, one has

$$\sum_{s\in S} \mu(s) \left(\sum_{t\in S} q(t|s) y_t(s) \right) = \sum_{T,\phi} \eta_{T,\phi} \left(\sum_{s\in T} y_{\phi(s)}(s) \right)$$

The value of $\mathcal{P}(\lambda)$ is finite

Let q be an irreducible transition function over S, with invariant measure μ. Builds upon Freidlin-Wenzell.

- For $s \in S$, a s-graph is a rooted tree over S with root s, where all states lead to s. $G(s)$ is the set of all s-graphs.
- Set $q(g) := \prod_{(t,u)\in g} q(u|t)$. Then $\mu(s) = \frac{1}{D} \sum_{g\in G(s)} q(g)$.

Corollary

There are $\eta_{T,\phi} \geq 0$ s.t., for each $(y_t(s)) \in \mathbf{R}^{S\times S}$, one has

$$\sum_{s\in S} \mu(s) \left(\sum_{t\in S} q(t|s) y_t(s) \right) = \sum_{T,\phi} \eta_{T,\phi} \left(\sum_{s\in T} y_{\phi(s)}(s) \right)$$

Corollary

If (v, x, α) is feasible in $\mathcal{P}(\lambda)$, then

$$\lambda \cdot v \leq \lambda \cdot \sum_{s\in S} \mu_\alpha(s) r(s, \alpha_s).$$

Action independent transitions

We here assume that transitions are $p(t|s)\pi(y|s,a)$.

Action independent transitions

We here assume that transitions are $p(t|s)\pi(y|s,a)$.

Proposition

Let p, q be irreducible transition functions with the same invariant measure μ. Then $\mathcal{H}(p) = \mathcal{H}(q)$.

Action independent transitions

We here assume that transitions are $p(t|s)\pi(y|s,a)$.

Proposition

Let p, q be irreducible transition functions with the same invariant measure μ. Then $\mathcal{H}(p) = \mathcal{H}(q)$.

$$\text{P1} \; : Im(I - P) = Im(I - Q)$$

Action independent transitions

We here assume that transitions are $p(t|s)\pi(y|s,a)$.

Proposition

Let p, q be irreducible transition functions with the same invariant measure μ. Then $\mathcal{H}(p) = \mathcal{H}(q)$.

P1 : $Im(I - P) = Im(I - Q)$

P2 : Let $(x_t(s))$ satisfy $\sum_{s \in T} x_{\phi(s)}(s) \leq 0$. There exists $x^* \geq x$, s.t. $\sum_{s \in T} x^*_{\phi(s)}(s) = 0$.

Action independent transitions

We here assume that transitions are $p(t|s)\pi(y|s,a)$.

Proposition

Let p,q be irreducible transition functions with the same invariant measure μ. Then $\mathcal{H}(p) = \mathcal{H}(q)$.

P1 : $Im(I-P) = Im(I-Q)$

P2 : Let $(x_t(s))$ satisfy $\sum_{s\in T} x_{\phi(s)}(s) \leq 0$. There exists $x^* \geq x$, s.t. $\sum_{s\in T} x^*_{\phi(s)}(s) = 0$.

- Fix (v,x,α) feasible in $\mathcal{P}_p(\lambda)$.

Action independent transitions

We here assume that transitions are $p(t|s)\pi(y|s,a)$.

Proposition

Let p, q be irreducible transition functions with the same invariant measure μ. Then $\mathcal{H}(p) = \mathcal{H}(q)$.

P1 : $Im(I-P) = Im(I-Q)$

P2 : Let $(x_t(s))$ satisfy $\sum_{s\in T} x_{\phi(s)}(s) \leq 0$. There exists $x^* \geq x$, s.t. $\sum_{s\in T} x^*_{\phi(s)}(s) = 0$.

- Fix (v, x, α) feasible in $\mathcal{P}_p(\lambda)$. Set $c_t(s) = \max_y \lambda \cdot c_t(s,y)$. Apply **P2** to get $c^*_t(s) = \bar{c}_t - \bar{c}_s$.

Action independent transitions

We here assume that transitions are $p(t|s)\pi(y|s,a)$.

Proposition

Let p, q be irreducible transition functions with the same invariant measure μ. Then $\mathcal{H}(p) = \mathcal{H}(q)$.

P1 : $Im(I - P) = Im(I - Q)$

P2 : Let $(x_t(s))$ satisfy $\sum_{s \in T} x_{\phi(s)}(s) \leq 0$. There exists $x^* \geq x$, s.t. $\sum_{s \in T} x^*_{\phi(s)}(s) = 0$.

- Fix (v, x, α) feasible in $\mathcal{P}_p(\lambda)$. Set $c_t(s) = \max_y \lambda \cdot c_t(s, y)$. Apply **P2** to get $c^*_t(s) = \bar{c}_t - \bar{c}_s$.
- Choose $\bar{d}$ s.t. $(I - P)\bar{c} = (I - Q)\bar{d}$. Set $d_t(s) = \bar{d}_t - \bar{d}_s$.

Action independent transitions

We here assume that transitions are $p(t|s)\pi(y|s,a)$.

Proposition

Let p, q be irreducible transition functions with the same invariant measure μ. Then $\mathcal{H}(p) = \mathcal{H}(q)$.

P1 : $Im(I - P) = Im(I - Q)$

P2 : Let $(x_t(s))$ satisfy $\sum_{s \in T} x_{\phi(s)}(s) \leq 0$. There exists $x^* \geq x$, s.t. $\sum_{s \in T} x^*_{\phi(s)}(s) = 0$.

- Fix (v, x, α) feasible in $\mathcal{P}_p(\lambda)$. Set $c_t(s) = \max_y \lambda \cdot c_t(s, y)$. Apply **P2** to get $c_t^*(s) = \bar{c}_t - \bar{c}_s$.
- Choose $\bar{d}$ s.t. $(I - P)\bar{c} = (I - Q)\bar{d}$. Set $d_t(s) = \bar{d}_t - \bar{d}_s$.
- Set

$$z_t^i(s, y) = \frac{\lambda^i}{|\lambda^i|} d_t(s) + \sum_{u \in S} \left(x_u^i(s, y) - \frac{\lambda^i}{|\lambda^i|} c_u^*(s) \right).$$

Action independent transitions

We here assume that transitions are $p(t|s)\pi(y|s,a)$.

Proposition

Let p, q be irreducible transition functions with the same invariant measure μ. Then $\mathcal{H}(p) = \mathcal{H}(q)$.

P1 : $Im(I-P) = Im(I-Q)$

P2 : Let $(x_t(s))$ satisfy $\sum_{s\in T} x_{\phi(s)}(s) \leq 0$. There exists $x^* \geq x$, s.t. $\sum_{s\in T} x^*_{\phi(s)}(s) = 0$.

- Fix (v, x, α) feasible in $\mathcal{P}_p(\lambda)$. Set $c_t(s) = \max_y \lambda \cdot c_t(s,y)$. Apply **P2** to get $c^*_t(s) = \bar{c}_t - \bar{c}_s$.
- Choose $\bar{d}$ s.t. $(I-P)\bar{c} = (I-Q)\bar{d}$. Set $d_t(s) = \bar{d}_t - \bar{d}_s$.
- Set

$$z^i_t(s,y) = \frac{\lambda^i}{|\lambda^i|} d_t(s) + \sum_{u\in S}\left(x^i_u(s,y) - \frac{\lambda^i}{|\lambda^i|} c^*_u(s)\right).$$

Then (v, z, α) is feasible in $\mathcal{P}_q(\lambda)$.

Dynamic Programming : $|I| = 1$ (1)

Dynamic Programming : $|I| = 1$ (1)

Claim $\mathcal{H}$ is a singleton, equal to $\lim_{\delta \to 1} v_\delta$.

- $\limsup\{v_\delta(s)\} \subseteq \mathcal{H}$, hence $\mathcal{H} \neq \emptyset$.

Dynamic Programming : $|I| = 1$ (1)

Claim $\mathcal{H}$ is a singleton, equal to $\lim_{\delta \to 1} v_\delta$.

- $\limsup\{v_\delta(s)\} \subseteq \mathcal{H}$, hence $\mathcal{H} \neq \emptyset$.
- $\mathcal{H} = [-k(-1), k(+1)]$, and $k(1) \geq -k(-1)$ since $\mathcal{H} \neq \emptyset$.

Dynamic Programming : $|I| = 1$ (1)

Claim $\mathcal{H}$ is a singleton, equal to $\lim_{\delta \to 1} v_\delta$.

- $\limsup\{v_\delta(s)\} \subseteq \mathcal{H}$, hence $\mathcal{H} \neq \emptyset$.
- $\mathcal{H} = [-k(-1), k(+1)]$, and $k(1) \geq -k(-1)$ since $\mathcal{H} \neq \emptyset$.
- (v, x, α) feasible in $\mathcal{P}(\lambda)$ implies (v, x, a) feasible, for each $a = (a_s)$ 'in the support' of $\alpha \Rightarrow$ pure strategies.

Dynamic Programming : $|I| = 1$ (1)

Claim $\mathcal{H}$ is a singleton, equal to $\lim_{\delta\to 1} v_\delta$.

- $\limsup\{v_\delta(s)\} \subseteq \mathcal{H}$, hence $\mathcal{H} \neq \emptyset$.
- $\mathcal{H} = [-k(-1), k(+1)]$, and $k(1) \geq -k(-1)$ since $\mathcal{H} \neq \emptyset$.
- (v, x, α) feasible in $\mathcal{P}(\lambda)$ implies (v, x, a) feasible, for each $a = (a_s)$ 'in the support' of $\alpha \Rightarrow$ pure strategies.
- If (v, x, a) feasible in $\mathcal{P}(1)$, then

$$v = \sum_{s\in S} \mu_a(s) r(s, a_s) + \sum_{T,\phi} \pi_{T,\phi} \left(\sum_{s_i n T} x_{\phi(s)}(s) \right).$$

If (w, y, a) feasible in $\mathcal{P}(-1)$, similar formula links w and y.

Dynamic Programming : $|I| = 1$ (1)

Claim $\mathcal{H}$ is a singleton, equal to $\lim_{\delta\to 1} v_\delta$.

- $\limsup\{v_\delta(s)\} \subseteq \mathcal{H}$, hence $\mathcal{H} \neq \emptyset$.
- $\mathcal{H} = [-k(-1), k(+1)]$, and $k(1) \geq -k(-1)$ since $\mathcal{H} \neq \emptyset$.
- (v, x, α) feasible in $\mathcal{P}(\lambda)$ implies (v, x, a) feasible, for each $a = (a_s)$ 'in the support' of $\alpha \Rightarrow$ pure strategies.
- If (v, x, a) feasible in $\mathcal{P}(1)$, then

$$v = \sum_{s \in S} \mu_a(s) r(s, a_s) + \sum_{T,\phi} \pi_{T,\phi} \left(\sum_{s_i n T} x_{\phi(s)}(s) \right) .$$

If (w, y, a) feasible in $\mathcal{P}(-1)$, similar formula links w and y.
Thus, $v \leq w$, hence $k(1) \leq -k(-1)$:

Dynamic Programming : $|I| = 1$ (1)

Claim $\mathcal{H}$ is a singleton, equal to $\lim_{\delta \to 1} v_\delta$.

- $\limsup\{v_\delta(s)\} \subseteq \mathcal{H}$, hence $\mathcal{H} \neq \emptyset$.
- $\mathcal{H} = [-k(-1), k(+1)]$, and $k(1) \geq -k(-1)$ since $\mathcal{H} \neq \emptyset$.
- (v, x, α) feasible in $\mathcal{P}(\lambda)$ implies (v, x, a) feasible, for each $a = (a_s)$ 'in the support' of $\alpha \Rightarrow$ pure strategies.
- If (v, x, a) feasible in $\mathcal{P}(1)$, then

$$v = \sum_{s \in S} \mu_a(s) r(s, a_s) + \sum_{T, \phi} \pi_{T,\phi} \left(\sum_{s_i n T} x_{\phi(s)}(s) \right).$$

If (w, y, a) feasible in $\mathcal{P}(-1)$, similar formula links w and y.
Thus, $v \leq w$, hence $k(1) \leq -k(-1)$: $\mathcal{H} = \{v^*\}$.

Dynamic Programming : $|I| = 1$ (1)

Claim $\mathcal{H}$ is a singleton, equal to $\lim_{\delta\to 1} v_\delta$.

- $\limsup\{v_\delta(s)\} \subseteq \mathcal{H}$, hence $\mathcal{H} \neq \emptyset$.
- $\mathcal{H} = [-k(-1), k(+1)]$, and $k(1) \geq -k(-1)$ since $\mathcal{H} \neq \emptyset$.
- (v, x, α) feasible in $\mathcal{P}(\lambda)$ implies (v, x, a) feasible, for each $a = (a_s)$ 'in the support' of $\alpha \Rightarrow$ pure strategies.
- If (v, x, a) feasible in $\mathcal{P}(1)$, then

$$v = \sum_{s\in S} \mu_a(s) r(s, a_s) + \sum_{T,\phi} \pi_{T,\phi} \left(\sum_{s_i n T} x_{\phi(s)}(s) \right).$$

If (w, y, a) feasible in $\mathcal{P}(-1)$, similar formula links w and y. Thus, $v \leq w$, hence $k(1) \leq -k(-1)$: $\mathcal{H} = \{v^*\}$.

Claim If $(x_t(s))$ is s.t. $\sum_{s\in T} x_{\phi(s)}(s) = 0$

$$v^*_s \leq r(s, a_s) + \sum_{t\in S} p(t|s) x_t(s)$$

for some $a = (a_s)$, then equality must hold.

Dynamic Programming (2)

Pick (x, a^*) such that (v^*, x, a^*) feasible in $\mathcal{P}(1)$.

Dynamic Programming (2)

Pick (x, a^*) such that (v^*, x, a^*) feasible in $\mathcal{P}(1)$.

Pick $x^* \geq x$, such that $\sum_{s \in T} x^*_{\phi(s)}(s) = 0$.

Dynamic Programming (2)

Pick (x, a^*) such that (v^*, x, a^*) feasible in $\mathcal{P}(1)$.

Pick $x^* \geq x$, such that $\sum_{s \in T} x^*_{\phi(s)}(s) = 0$.

$$\textbf{Claim}: v^* = \max_{a_s \in A} \left(r(s, a_s) + \sum_{t \in S} p(t|s, a_s) x^*_t(s) \right).$$

Dynamic Programming (2)

Pick (x, a^*) such that (v^*, x, a^*) feasible in $\mathcal{P}(1)$.

Pick $x^* \geq x$, such that $\sum_{s \in T} x^*_{\phi(s)}(s) = 0$.

$$\textbf{Claim} : v^* = \max_{a_s \in A} \left(r(s, a_s) + \sum_{t \in S} p(t|s, a_s) x^*_t(s) \right).$$

- for $a_s = a^*_s$, one has $\leq$.
- Previous claim implies $=$.

Dynamic Programming (2)

Pick (x, a^*) such that (v^*, x, a^*) feasible in $\mathcal{P}(1)$.

Pick $x^* \geq x$, such that $\sum_{s \in T} x^*_{\phi(s)}(s) = 0$.

$$\textbf{Claim} : v^* = \max_{a_s \in A} \left(r(s, a_s) + \sum_{t \in S} p(t|s, a_s) x^*_t(s) \right) .$$

- for $a_s = a^*_s$, one has $\leq$.
- Previous claim implies $=$.
- Pick $y^* \in \mathbf{R}^S$, such that $x^*_t(s) = y^*_t - y^*_s$.

Dynamic Programming (2)

Pick (x, a^*) such that (v^*, x, a^*) feasible in $\mathcal{P}(1)$.

Pick $x^* \geq x$, such that $\sum_{s \in T} x^*_{\phi(s)}(s) = 0$.

$$\textbf{Claim} : v^* = \max_{a_s \in A} \left(r(s, a_s) + \sum_{t \in S} p(t|s, a_s) x^*_t(s) \right).$$

- for $a_s = a^*_s$, one has $\leq$.
- Previous claim implies $=$.
- Pick $y^* \in \mathbf{R}^S$, such that $x^*_t(s) = y^*_t - y^*_s$.
- Then

$$v^* + y^*_s = \max_{a_s \in A} \left(r(s, a_s) + \sum_{t \in S} p(t|s, a_s) y^*_t \right).$$

This is the Average Cost Optimality Equation in DP.