Correlated Equilibria in Graphical Games

Sham Kakade, Michael Kearns, John Langford, Luis Ortiz

What is a Game?

Players: $i \in \{1, ..., n\}$.

Action vector: $\vec{a} = \{a_1, ..., a_n\}$

Payoff matrices: $M_i(\vec{a})$

What is a Correlated Equilibria?

Joint distribution P satisfying:

For all players, i, For all actions a_i, a

$$E_{\vec{a} \sim P|a_i} M_i(\vec{a}) \geq E_{\vec{a} \sim P|a_i} M_i(\vec{a}[i:a])$$

Nash vs. Correlated Equilibria

	Nash Eq.	Correlated Eq.
Time to Verify	$O(\lvert\vec{A}\rvert n)$	$O(\lvert\vec{A}\rvert n)$
Time to Find	???	$P(\lvert\vec{A}\rvert)$
Time to App. Find	$\lvert\vec{A}\rvert n^{O\left(\frac{1}{\epsilon^2}\log n\right)}$	$P(\lvert\vec{A}\rvert)$
Space	$O(\lvert A\rvert n)$	$O(\lvert\vec{A}\rvert)$
Online Algorithm Converges?	No	Yes
Guaranteed to exist	Yes	Yes
Set Size	Smaller	Strictly Larger

Graphical Games

Graph (V, E):

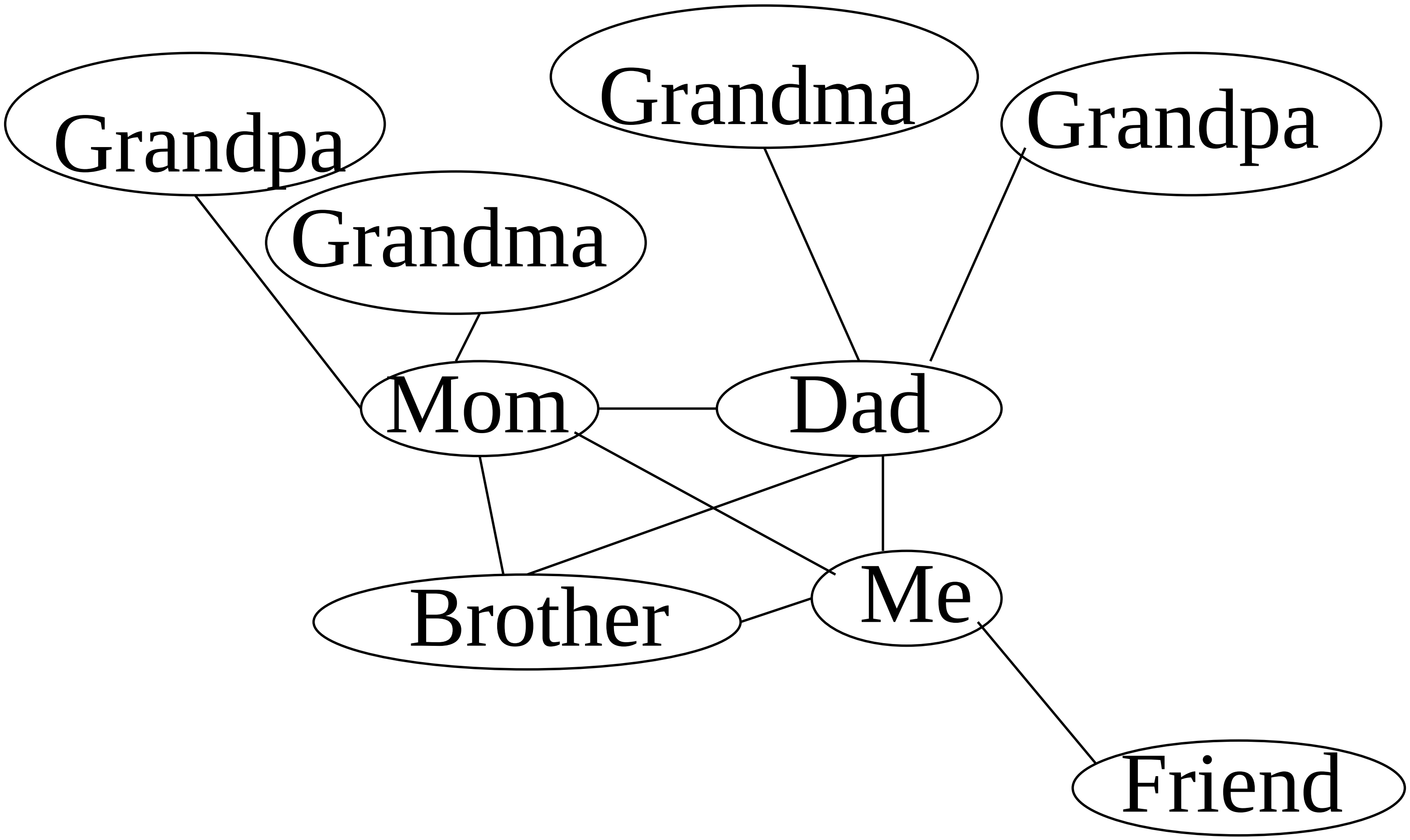
$V =$ players

$E =$ edges connecting players who affect payoffs

$\vec{a}^i =$actions in the neighborhood of player i

$M_i(\vec{a}^i) =$payoff to player i

A Graphical Game



Game Vs. Graphical Game

	Game	Graphical Game
Size	$O(\|\vec{A}\|n)$	$O(\sum_i \|\vec{A}^i\|)$
Time to Find NE	???	Tree Games: P(GG size)
Time to App. Find	$\|\vec{A}\|n^{O\left(\frac{1}{\epsilon^2}\log n\right)}$	Tree games: $P(\frac{1}{\epsilon},$ GG size)
Time to Find CE	P(size)	Tree Games: P(GG size) (here)
Space to Represent CE	O(size)	O(GG size) (here)

Outline

1. Basics

2. Statement of Results

3. Representation Notes

4. Computation Notes

Representation I: Equivalences

Cannot explicitly represent *all* CE.

Can explicitly represent all CE mod payoff equivalence.

Representation I: Equivalences

Cannot explicitly represent *all* CE.

Can explicitly represent all CE mod payoff equivalence.

Payoff equivalence:

$$E_{\vec{a}\sim P} M_i(\vec{a}^i) = E_{\vec{a}\sim Q} M_i(\vec{a}^i)$$

Local Neighborhood equivalence:

$$P(\vec{a}^i) = Q(\vec{a}^i)$$

Representation I: Equivalences

Cannot explicitly represent *all* CE.

Can explicitly represent all CE mod payoff equivalence.

Payoff equivalence:

$$E_{\vec{a} \sim P} M_i(\vec{a}^i) = E_{\vec{a} \sim Q} M_i(\vec{a}^i)$$

Local Neighborhood equivalence:

$$P(\vec{a}^i) = Q(\vec{a}^i)$$

Note: Local Neighborhood equivalence $\Rightarrow$ Payoff equivalence

Representation Theorem

Theorem: For all Graphical Games, For all CE $P(\vec{a})$, there exists Local Neighborhood equivalent $Q(\vec{a})$ representable with O(GG size).

Calculation Theorem

Theorem: For all Tree Graphical Games, there exists O(Graphical Game size) linear program whose solution is a CE.

Note: Running time $= P$(Graphical Game size)

Outline

1. Basics

2. Statement of Results

3. Representation Notes

4. Computation Notes

Representation Method

Use "Local" Markov Network $= (G, \Psi)$

1. $G = (V, E)$ = undirected graph

2. $\Psi = \{\psi_i : \vec{a}^i \to [0, \infty)\}$ = local potential functions

$$P(\vec{a}) \equiv \frac{1}{Z} \Pi_{i=1}^{n} \psi_i(\vec{a}^i)$$

Real Representation Theorem

Theorem: For all Graphical Games, For all CE $P(\vec{a})$, there exists Local Neighborhood equivalent $Q(\vec{a})$ representable with O(GG size) Local Markov Network.

Proof: By construction: solve for maximum entropy distribution subject to local neighborhood equivalence

Outline

1. Basics

2. Statement of Results

3. Representation Notes

4. Computation Notes

Computation: Basic Idea

Representation size of local neighborhoods is reasonable $\Rightarrow$ use local neighborhood versions of Correlated Equilibria constraints.

It turns out: this is sufficent when the graph is a tree.

Variables: $P_i(\vec{a}^i) =$ local neighborhood marginals

Computation: Constraints

1. Correlated Equilibria

$$\forall a, a' \quad \sum_{\vec{a}^i : a_i = a} P_i(\vec{a}^i) M_i(\vec{a}^i) \geq \sum_{\vec{a}^i : a_i = a} P_i(\vec{a}^i) M_i(\vec{a}^i[i : a'])$$

2. Distribution

$$\forall a \;\; P_i(\vec{a}^i) \geq 0 \quad \sum_{\vec{a}^i} P_i(\vec{a}^i) = 1$$

3. Intersection Consistency (let $\vec{a}^{ij}$ = actions in neighborhood of i intersect neighborhood of j)

$$\forall i, j \quad \sum_{\vec{a}^i : \vec{a}^{ij} = \vec{y}^{ij}} P_i(\vec{a}^i) = \sum_{\vec{a}^j : \vec{a}^{ij} = \vec{y}^{ij}} P_j(\vec{a}^j)$$

Computation: Precise Result

Theorem:

$$Q(\vec{a}) \equiv \frac{\Pi_{i \in V} P_i(\vec{a}^i)}{\Pi_{i,j \in E} P_i(\vec{a}^{ij})}$$

is a Correlated Equilibrium and satisfies $Q(\vec{a}^i) = P_i(\vec{a}^i)$.

Proof:

$$Q(\vec{a}) = \Pi_{i \in V} Q(a_i | \text{parents of } i)$$

$$= \Pi_{i \in V} Q(a_i | \text{parent and grandparent of } i)$$

$$= \Pi_{i \in V} \frac{P_{i \text{ parent}}(\vec{a}^{i \text{ parent}})}{P_{i \text{ parent}}(\vec{a}^{i \text{ parent}, i \text{ grandparent}})}$$

Computation: General Result

Theorem: For all tree graphical games, and all linear objective functions, linear programing computes a Correlated Equilibria in P(size) time.

Proof: Linear programming is polynomial in the number of constraints.

Future Directions

1. Extend Tree results to other graphs (already done to some extent)

2. Directed Graphs $\Rightarrow$ more concise representations?

3. Distributed Notions of Equilibria?