Structure Learning for Approximate Solution of Many-Player Games

Zun Li, Michael P. Wellman
University of Michigan, Ann Arbor
{lizun, wellman}@umich.edu

COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF MICHIGAN

SRG Strategic Reasoning Group

Normal Form Games: Limitations

- In an N-player normal form game $\mathscr{G}$, *agent* $n \in \{1, \ldots, N\}$ chooses its action $a_n \in \{1, \ldots, M\}$, and receives payoff $u_n(\boldsymbol{a})$ as a function of the agents' joint action $\boldsymbol{a}$.
- The payoff for n under a joint mixed strategy $\boldsymbol{\sigma}$ is $u_n(\boldsymbol{\sigma}) \triangleq \mathbb{E}_{\boldsymbol{a}\sim\boldsymbol{\sigma}}[u_n(a_n, \boldsymbol{a}_{-n})]$, the deviation payoff of n to m under $\boldsymbol{\sigma}$ is $u_n(a_n, \boldsymbol{\sigma}_{-n}) \triangleq \mathbb{E}_{\boldsymbol{a}_{-n}\sim\boldsymbol{\sigma}_{-n}}[u_n(m, \boldsymbol{a}_{-n})]$
- Solution concept: $\boldsymbol{\sigma}$ is an ϵ-Nash Equilibrium if $\max_{n,a_n} u_n(a_n, \boldsymbol{\sigma}_{-n}) - u_n(\boldsymbol{\sigma}) \le \epsilon$
- The representational complexity is $O(NM^N)$, which is prohibitive when N is large. Need more succinct representation!
- The computational complexity of solving a Nash is PPAD complete. Need more advanced computational tools!

Empirical Game Models [1]

- Empirical Game Theoretical Analysis (EGTA) employs simulation or sampling to induce a game model.
- Formally, in EGTA the multiagent environment is represented by a *game oracle* $\mathcal{O}$ (*e.g.*,, a simulator)
- A dataset $\mathcal{D}$ of action-payoff tuples $(\boldsymbol{a}, \boldsymbol{u})$ could be queried to the oracle, where $\boldsymbol{u}$ is the (noisy) payoff vector associated with action profile $\boldsymbol{a}$.
- A normal-form game model induced from $\mathcal{D}$ is called an *empirical* game.
- In EGTA, the game analyst does not need to store the information of the whole game matrix to compute an approximate Nash.

Succinct Game Models

- *Games with Symmetry* [3]:
  - *Anonymous game*: agent n's payoff depends only on its action and how many agents choose each action: $u_n(a_n, \boldsymbol{a}_{-n}) = u_n(a_n, f_1, \ldots, f_M)$.
  - *Symmetric game*: $\forall n.\ u_n = u$.
  - *Role-symmetric game*: Let $\mathcal{R}(n) \in \{1, \ldots, K\}$ denote the role for agent n. Then the payoff for agent n depends on its action and the action distribution within each role: $u_n(a_n, \boldsymbol{a}_{-n}) = u_n^{\mathcal{R}}(a_n, f_{1,1}, \ldots, f_{1,M}, \ldots, f_{K,M})$.
- *Games with Sparsity*: in a *graphical games* [2], agent n's payoff depends only on the joint action profile over its neighborhood $\mathcal{N}(n)$ on an interaction graph, $u_n(a_n, \boldsymbol{a}_{-n}) = u_n(a_n, \boldsymbol{a}_{\mathcal{N}(n)})$.

Game Model Learning [4] & Iterative Structure Learning Framework

- Game model learning: Solving a complex unknown game by learning a succinct representation of it in a *hypothesis game space* whose structure can be exploited for equilibrium computation, the solution of which can be served as an approximate solution of the origin game

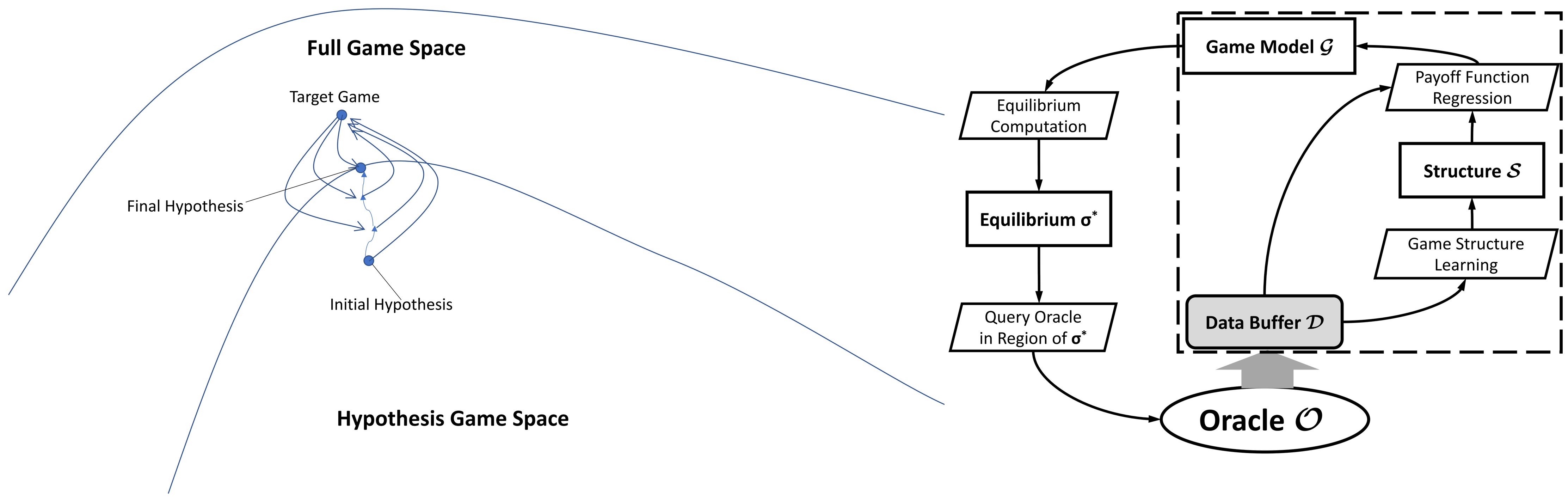

Figure 1: Game Model Learning & Iterative Structure Learning Framework

- Iterative structure learning framework: The only explicit game descriptors are the sets of agents and actions. Starting with an arbitrary guess solution $\boldsymbol{\sigma}^*$, on each iteration,
  - Queries oracle $\mathcal{O}$ in the region of $\boldsymbol{\sigma}^*$, obtaining by this online sampling process a new dataset , which is added to the data buffer $\mathcal{D}$.
  - Through offline interaction with $\mathcal{D}$, we then learn a game model using function approximators, and solve it to reach the next $\boldsymbol{\sigma}^*$.

K-Roles: Learning Role Symmetry

- Hyperparameter $\hat{K}$: the number of roles
- Idea: Represent each agent as their deviation payoffs and use unsupervised learning on the vector embeddings
- Can be regarded as *feature extraction.*

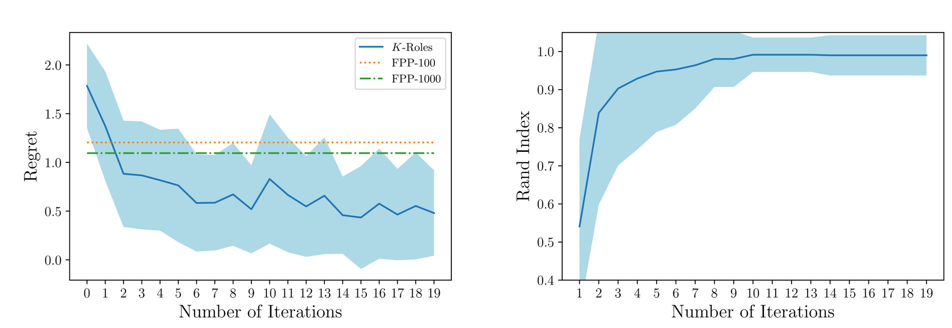

Figure 2: Performance of K-Roles on a 300-agent, 3-action, 3-role role-symmetric game. Left and right figures respectively measures the equilibrium and structure quality, w.r.p. the true game model

G3L: Learning Graphical Structure

- Hyperparameter $\hat{\kappa}$: the maximum size of neighborhood
- Idea: Greedily learn a graphical model guided by payoff training loss.
- Can be regarded as *feature selection.*

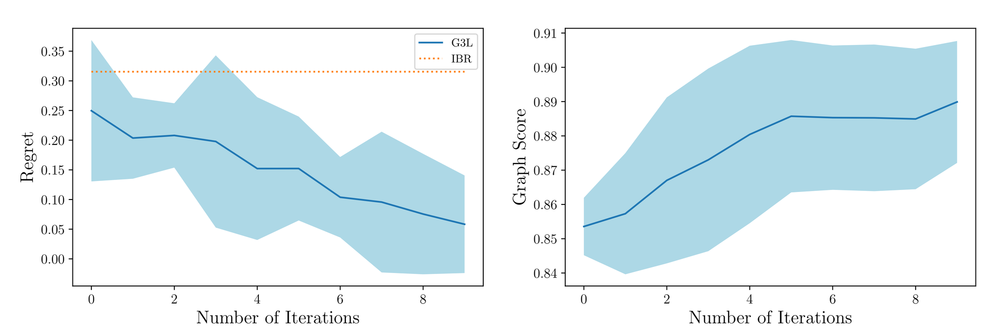

Figure 3: Performance of G3L on a 100-agent, 2-action graphical game. Left and right figures respectively measures the equilibrium and structure quality, w.r.p. the true game model

Symmetry Can Arise from Sparsity

- $u_n = y_n - \zeta \cdot x_n$. y_n is a symmetric game term while x_n is a graphical game term, $\zeta \ge 0$ a structure parameter defining a spectrum of game between perfect symmetry and perfect sparsity.

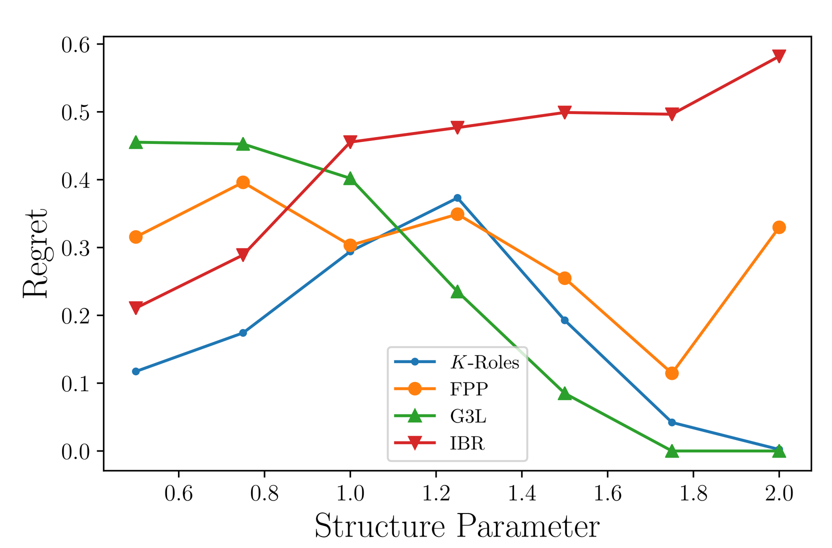

Figure 4: Performance of all methods on an approximately structured game class.

References

[1] Wellman, Michael P. "Methods for empirical game-theoretic analysis." *AAAI.* 2006.

[2] Kearns, Michael, Michael L. Littman, and Satinder Singh. "Graphical models for game theory." *UAI.* 2001.

[3] Jiang, Albert Xin, Kevin Leyton-Brown, and Navin AR Bhat. "Action-graph games." *Games and Economic Behavior* 71.1 (2011): 141-173.

[4] Vorobeychik, Y. and Wellman, M. and Singh, S., "Learning payoff functions in infinite games" in *Machine Learning*, 67(1-2) 145–168, 2007.

SCAN ME

Take a picture to download the full paper