

Learning Payoffs in Large Symmetric Games

(Extended Abstract)

Bryce Wiedenbeck
University of Michigan
btwied@umich.edu

Michael P. Wellman
University of Michigan
wellman@umich.edu

Categories and Subject Descriptors

J.4 [Social and Behavioral Sciences]: Economics

Keywords

Equilibrium computation and analysis;
Simulation techniques, tools, and platforms.

1. INTRODUCTION

Game theory offers powerful tools for reasoning about agent behavior and incentives in multi-agent systems. Most of these reasoning tools require a game model that specifies the outcome for all possible combinations of agent behaviors in the subject environment. The requirement to describe all possible outcomes often severely limits the fidelity at which we can model agent choices, or the feasible scale in agent population. Thus, game theorists must select with extreme care the scale and detail of the system to model, balancing fidelity with tractability.

This tension comes to the fore in simulation-based approaches to game modeling [3, 4], where filling in a single cell in a game matrix may require running many large-scale agent-based simulations. It is often feasible to simulate large numbers of agents interacting, but infeasible to sample all $\binom{P+S-1}{P}$ combinations of strategies in a symmetric game with P players and S strategies. If the payoff matrix must be filled completely to perform analysis, this combinatorial growth severely restricts the size of simulation-based games.

Our alternative approach accommodates incomplete specification of outcomes, extending the game model to a larger domain through an inductive learning process. We take as input data about outcomes from selected combinations of agent strategies in symmetric games, and learn a game model over the full joint strategy space. By doing so we can scale game modeling to a large number of agents without unduly restricting the size of strategy sets considered.

Our primary aim is to identify symmetric mixed-strategy ϵ -Nash equilibria and calculate social welfare for symmetric mixed strategies. We measure the quality of approximate equilibria using regret $\epsilon(\cdot)$, the the maximum gain any player can achieve by switching to a pure strategy. We measure the accuracy of social welfare estimates by absolute error.

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*
Copyright © 2015, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

2. METHODS

The first key to learning payoffs in symmetric games is encoding strategy profiles as vectors of strategy counts. A profile assigns a strategy to every player, but in a symmetric game this can be summarized as the number of players choosing each strategy. We represent this by a non-negative integer vector $\vec{s} \in \mathbb{N}^S$ with one dimension per strategy. For each strategy s , we learn a utility function u_s mapping profile vectors to real-valued payoffs via Gaussian process regression [1].

This regression gives us a model that we can query for estimates of pure-strategy payoffs. However, identifying equilibria and computing social welfare both require estimating the expected value to a single agent playing strategy s against opponents playing a symmetric mixture $\vec{\sigma} \in [0, 1]^S$:

$$\mathbf{E}[u_s(\vec{\sigma})] = \sum_{\vec{s}} \Pr(\vec{s} | \vec{\sigma}, s) u_s(\vec{s})$$

This sum can range over nearly all profiles \vec{s} in the exponentially-large game, rendering its exact computation infeasible.

We introduce two methods for estimating $\mathbf{E}[u_s(\vec{\sigma})]$ from our learned utility models: *sampling* and *point estimation*. The sampling method draws k profiles $\vec{s}_j \sim \vec{\sigma}$ randomly according to the symmetric mixture. It then queries the utility function $u_s(\vec{s}_j)$ at each of these profiles and averages their payoffs. This method is correct in the infinite-sample limit ($k \rightarrow \infty$), and provides slightly more accurate estimates in our experiments, so we recommend sampling for computing social welfare. The point estimation method makes a single query to u_s at the point $P\vec{\sigma}$:

$$\mathbf{E}[u_s(\vec{\sigma})] \approx u_s(P\vec{\sigma}).$$

At this point, population proportions playing each strategy match mixture probabilities. This method is correct in the infinite-player limit ($P \rightarrow \infty$), and is cheaper to evaluate and more numerically stable than sampling, so we recommend point estimation for computing ϵ -Nash equilibria.

Further, we found that the quality of equilibria found was improved by learning the difference between a strategy's payoff and the mean payoff for the profile, rather than learning strategy payoffs directly. We compute equilibria using *replicator dynamics* [2], an evolutionary algorithm that relies on relative differences in expected payoffs across strategies for its update step. We believe that this difference learning allows the regression method to ignore broad trends in payoffs and hone in on the smaller but more important differences among strategies.

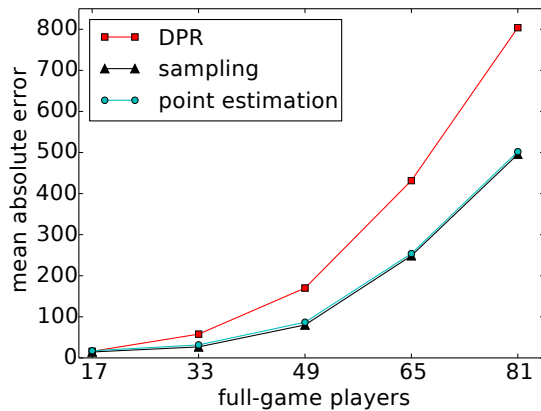


Figure 1: Social welfare: DPR and learning perform similarly on 17-player games, but diverge on larger games. Estimates from sampling are slightly more accurate than those from point estimation.

3. EXPERIMENTAL VALIDATION

We compared our methods to the best existing approach for approximating large symmetric games: deviation-preserving reduction (DPR) [5]. Our ultimate goal is to apply our learning methods to simulation-based games where we learn about payoffs only through observation data and the profile space is far too big to enumerate exhaustively. However, for our experiments, we require games with known ground-truth payoffs against which we can compare results from various approximation methods. We achieve this by generating random action-graph-game [?] instances, which have a sparse representation that allows us to generate large games and compute exact values for regret and social welfare in those true games. The largest games used in testing of related methods had 12 players and 6 strategies and therefore around 6 thousand profiles; our experiments include games up to 81 players and 8 strategies for a total of over 6 billion profiles in the full symmetric representation.

We generated random action-graph game instances with 6–8 strategies and 17–81 players. We approximated each game using both Gaussian process regression and DPR. DPR allows the analyst to select the number of players in the reduced game, trading off fidelity and tractability; we tested reduced games of several different sizes. For any given reduced-game size, DPR requires data from a specific set of full-game profiles; because of sampling noise, we drew 10 samples of each of these profiles. For each DPR model, we generated a sample set of the same size as input to the regression, but spread the samples over a larger set of profiles.

Figure 1 shows the relative accuracy of social welfare estimates, varying the size of the full game, while holding the reduced-game size constant at 5 players. Each point aggregates mean absolute error across 320 mixtures in each of 100 action-graph games. DPR and regression perform similarly on 17-player full games, but regression gives substantially better results on larger games. At the scale used in the figure, the difference between point estimation and sampling is difficult to identify, but sampling was slightly (and statistically significantly) better at all game sizes.

Figure 2 shows the regret of equilibria found by each method in 61-player full games, varying the size of the re-

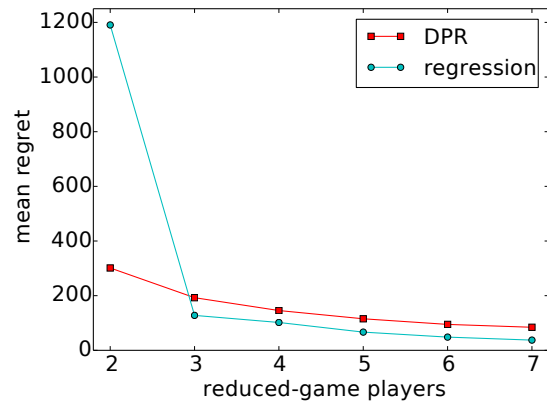


Figure 2: ϵ -Nash equilibria: Regression performs very poorly when given a tiny amount of data, but gives significantly lower regret than DPR when both have adequate data.

duced game. Each point gives the average true-game regret of all ϵ -Nash equilibria found by each method across 100 action graph games. Regression performs drastically worse on the smallest data set. We attribute this to lack of data: 360 samples were likely insufficient to move the Gaussian process far from its zero-mean prior in many regions of the profile space. However, for all larger reduced games, regression outperforms DPR by a statistically significant margin that holds roughly constant around $\epsilon = 50$.

4. DISCUSSION

The initial experiments presented here indicate that our machine learning method for approximating many-player symmetric games can accurately estimate expected values of symmetric mixed strategies, enabling us to compute social welfare and identify ϵ -Nash equilibria. Our experimental framework allowed us to test our methods on large action-graph games with structure that should be learnable, but was unknown to our learning methods. The results show that regression methods have the potential to outperform the state-of-the-art tool for approximating large simulation-based games, producing lower error in social welfare estimates and lower regret of identified equilibria.

REFERENCES

- [1] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, 2006.
- [2] P. Schuster and K. Sigmund. Replicator dynamics. *Journal of Theoretical Biology*, 100:533–538, 1983.
- [3] W. E. Walsh, R. Das, G. Tesauro, and J. O. Kephart. Analyzing complex strategic interactions in multi-agent systems. In *AAAI-02 Workshop on Game-Theoretic and Decision-Theoretic Agents*, 2002.
- [4] M. P. Wellman. Methods for empirical game-theoretic analysis (extended abstract). In *AAAI*, pages 1552–1555, 2006.
- [5] B. Wiedenbeck and M. P. Wellman. Scaling simulation-based game analysis through deviation-preserving reduction. In *AAMAS*, pages 931–938, 2012.