What Game Are We Playing?
End-to-end Learning in Normal and Extensive Form Games

Chun Kai Ling, Fei Fang, J. Zico Kolter

School of Computer Science, Carnegie Mellon University

chunkail@cs.cmu.edu, feif@cs.cmu.edu, zkolter@cs.cmu.edu

Abstract

Although recent work in AI has made great progress in solving large, zero-sum, extensive-form games, the underlying assumption in most past work is that the parameters of the game itself are known to the agents. This paper deals with the relatively under-explored but equally important "inverse" setting, where the parameters of the underlying game are *not* known to all agents, but must be learned through observations. We propose a differentiable, end-to-end learning framework for addressing this task. In particular, we consider a regularized version of the game, equivalent to a particular form of quantal response equilibrium, and develop 1) a primal-dual Newton method for finding such equilibrium points in both normal and extensive form games; and 2) a backpropagation method that lets us analytically compute gradients of all relevant game parameters through the solution itself. This ultimately lets us learn the game by training in an end-to-end fashion, effectively by integrating a "differentiable game solver" into the loop of larger deep network architectures. We demonstrate the effectiveness of the learning method in several settings including poker and security game tasks.

1 Introduction

Recent work in artificial intelligence has led to huge advances in methods for solving large-scale, zero-sum, extensive form games, both from methodological and applied standpoints. From the algorithmic approach, methods based on Counterfactual Regret Minimization [Zinkevich *et al.*, 2008] and first-order methods for game solving [Kroer *et al.*, 2017], have enabled solutions to larger and larger games. In terms of applications, there have been a number of recent breakthroughs, including exceeding human performance in no-limit poker [Brown and Sandholm, 2017; Moravčík *et al.*, 2017], essentially weakly solving limit poker [Bowling *et al.*, 2015], work in security games with applications to infrastructure security [Pita *et al.*, 2009], and many others. However, virtually all this progress in game theoretic approaches to large games has operated under the assumption that the parameters of the game are known to the solvers, and that the main challenge is simply finding the optimal strategy. In contrast, in many real world scenarios, certain elements of the game (e.g., payoff matrices, chance node probabilities, etc), are unknown to some of the agents prior to the game. For example, in security games, we may want to understand the underlying payoffs of an adversary, rather than just their observed strategy, to better understand how aspects of the game can be manipulated or changed to get a desirable outcome.

In this paper, we propose an end-to-end framework for learning the parameters of uncertain games (both for normal-form and extensive-form games), purely by observing the actions of the agents. Although there has been a great deal of work at the intersection of game theory and reinforcement learning [Busoniu *et al.*, 2008; Bowling and Veloso, 2000], most game-theoretic analysis either assumes that the payoffs underlying the game are known (this is the standard game theory setting), or forgoes trying to learn an explicit and complete representation of the game and instead looks for merely learning agent strategies that will perform well [Letchford *et al.*, 2009; Vorobeychik *et al.*, 2007; Fearnley *et al.*, 2015]. However, in many cases when the true underlying payoffs of the agents are *not* known, our primary goal is precisely to recover or understand the payoffs. The few exceptions that focus on learning the payoffs often rely on special structures of the game (e.g., symmetry in multiplayer setting [Vorobeychik *et al.*, 2007]), or querying the best response of the agent with unknown payoffs by asking other agents to play carefully designed strategies [Blum *et al.*, 2014; Letchford *et al.*, 2009]. However, the general problem of learning game parameters by observing actions is still under-explored. One of the most closely-related works to our own is the Computational Rationalization framework [Waugh *et al.*, 2011], though 1) our approach differs in how the utilities/payoffs are modeled; and 2) we crucially focus heavily on the extensive form settings, whereas this past work considered only normal form games.

The crux of our approach is to consider the *quantal response equilibrium* (QRE), a generalization of Nash equilibrium (NE) that includes some possibility of agents acting suboptimally. We show that the solution of the QRE is a *differentiable* function of the game payoff matrix, and backpropagation can be computed analytically via implicit differentiation. We develop a solver that jointly solves the QRE for two-player zero-sum games using a primal-dual Newton
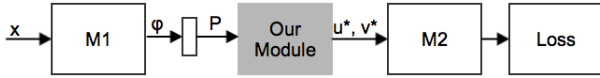
Figure 1: An example architecture utilizing our proposed module.

Method, and allows us to compute the derivatives of agent actions with respect to the underlying payoff matrix. This enables us to develop end-to-end learning approaches that can infer the payoff matrix or other parameters underlying a game merely from samples of the agents acting according to their QREs. [1] More generally, the method allows for (both normal form and extensive form) game-solving to be integrated as a module in deep learning systems, a strategy that can find use in multiple application areas.

We demonstrate the effectiveness of our approach on several domains: a toy normal-form game where payoffs depend on external context; a one-card poker game (with a small representation in strategic form, but which would already be too large to solved in normal form); and a security resource allocation game, which is an extensive-form generalization of defender-attacker game in security domain. In all settings, we show that our approach is able to learn, solely from observed actions, the relevant underlying parameters of the game, such as the payoff matrices or (agent belief over) chance node probabilities. We believe this represents a substantial step forward in understanding how game theoretic methods can be applied to uncertain settings, where the "true" parameters of the game are unknown to an agent. Due to space constraints, supplementary material and appendices are available at arXiv.[2]

2 Learning and Quantal Response in Normal Form Games

Our game-solving module provides all the elements required to perform differentiable learning through the game solution. The resulting learning approach learns a mapping from *contextual features* x to payoff matrices P and computes equilibrium strategies (u^*, v^*) under a new set of contextual features. An example architecture is presented in Figure 1. Here, P is parameterized by a domain-dependent low-dimensional vector ϕ, which is dependent on a differentiable function $M_1(x)$. Similarly, the loss function is taken after applying any differentiable $M_2(u^*, v^*)$. For the remainder of the paper, we focus on zero-sum games, which capture a wide class of adversarial environments.

We begin by considering normal form games. Although normal form games have limited real-world utility due to the fact that they can only handle relatively small-scale settings,

the game solver and learning approach in this restricted setting captures much of the intuition and basic methodology of our approach.

2.1 Zero-Sum Normal Form Games

In two-player zero-sum game with payoff matrix P, a classic min-max formulation to compute the NE is as follows

$$\min_u \max_v \quad u^T P v \tag{1}$$

$$\text{subject to} \quad 1^T u = 1, u \geq 0 \tag{2}$$

$$1^T v = 1, v \geq 0, \tag{3}$$

u and v denote the (mixed) strategies employed by the min and max player respectively. The solution (u^*, v_0) to this optimization problem and the solution (u_0, v^*) of the corresponding problem with inversed player order (i.e., $\min_v \max_u u^T P v$) forms the Nash equilibrium (u^*, v^*).

Here we present an introduction to our approach considering the case where the payoff matrix P is *not* known a priori. P could represent either a single fixed but unknown payoff matrix, or, in a more complex setting, depend on some external context x. For example, in anti-poaching games [Fang *et al.*, 2016], P depends on temperature and precipitation. In general, however, we consider the case where we observe samples of actions $a^{(i)}$, $i = 1, \ldots, N$, consisting of observed actions, from one or both players, sampled from the equilibrium strategies (u^*, v^*). The goal is to recover the true underlying payoff matrix P, or a function form $P(x)$ depending on the current context.

2.2 Quantal Response Equilibria

While extremely powerful both theoretically and as a modeling tool, the NE is poorly-suited for our purposes because:

1. NEs are overly strict. In practice, many payoff matrices result in actions never being played. This tends to be overly restrictive and does not adequately describe real-world scenarios where players are boundedly rational.

2. NEs in zero-sum games may not be unique. This leads to difficulties when resolving which NE to select.

3. NEs are discontinuous with respect to P – a small change in P can lead to jumps in u^*, v^*. This precludes integrating the technique into differentiable learning procedures.

To address these issues, in our learning setting we propose to model the player's action with the quantal response equilibria [McKelvey and Palfrey, 1995] instead. In general, QRE models situations where payoff matrices are injected with some noise. Specifically, we consider the *logit* equilibrium, where payoffs are perturbed by samples from a Gumbel distribution. The smoothness of QRE makes gradient-based approaches feasible [Amin *et al.*, 2016]. It is known that for zero-sum games, the logit equilibrium obeys the fixed point [3]

---

[1] Naturally, there are some questions here about when games are identifiable or not; in general, the answer is no, because e.g. many payoff matrices can lead to identical strategies or policies for the agent, so we do not expect to always be able to recover a true underlying payoff matrix. However, we show that for various classes of parametrized games, our approach *is* able to recover the true underlying payoffs of different agents.

[2] https://arxiv.org/abs/1805.02777

[3] In the QRE, there is an additional rationality parameter λ. In this work, we fix $\lambda = 1$ throughout.

$$u_i^* = \frac{\exp(-Pv)_i}{\sum_{q \in [n]} \exp(-Pv)_q}, \quad v_j^* = \frac{\exp(P^T u)_j}{\sum_{q \in [m]} \exp(P^T u)_q}. \tag{4}$$

It is further known that for a fixed opponent strategy, the logit equilibrium corresponds to a strategy regularized by the Gibbs entropy [Mertikopoulos and Sandholm, 2016]. Since the Gibbs entropy is strictly convex, the regularized best response is unique.

2.3 End-to-End Learning

In order to integrate zero-sum game solvers into an end-to-end learning framework, we need a method for "differentiating through" the game solution itself; that is, we need to compute the Jacobian (or more precisely, compute the Jacobian-vector products needed for backpropagation) of the quantal equilibrium solution with respect to the payoff matrix. Our method for doing so relies on techniques from differential calculus, and is a relatively straightforward extension of similar approaches to differentiating through optimization problems. However, as a prelude to the more involved extensive form solution that we will discuss shortly, we describe our method in some detail, which involves both a particular approach to solving the QRE and to differentiating through its solution.

QRE Solver
We observe that finding the fixed point in (4) is equivalent to solving the regularized min-max game

$$\min_{u \in \mathbb{R}^n} \max_{v \in \mathbb{R}^m} \quad u^T P v - H(v) + H(u)$$
$$\text{subject to} \quad 1^T u = 1, \quad 1^T v = 1, \tag{5}$$

where $H(y)$ is the Gibbs entropy $\sum_i y_i \log y_i$. Notice that the non-negative constraints are implicit from the entropy term, and that the entropy regularization renders the equilibrium continuous with respect to P. Intuitively, entropy regularization encourages players to play more randomly, and no action has probability 0. Furthermore, since the objective is strictly a convex-concave problem, it has a unique saddle point which corresponds to (u^*, v^*).

This formulation leads to a solver for the QRE for two-player zero-sum games, using a primal-dual Newton Method. To begin, the KKT conditions for the above problem are

$$Pv + \log(u) + 1 + \mu 1 = 0$$
$$P^T u - \log(v) - 1 + \nu 1 = 0 \tag{6}$$
$$1^T u = 1, \qquad 1^T v = 1,$$

where μ, ν are Lagrange multipliers for the equality constraints on u, v respectively. Following Newton's method, we get the following update rule, which provides a convergent method for computing the QRE for 2 player zero-sum games

$$Q \begin{bmatrix} \Delta u \\ \Delta v \\ \Delta \mu \\ \Delta \nu \end{bmatrix} = - \begin{bmatrix} Pv + \log u + 1 + \mu 1 \\ P^T u - \log v - 1 + \nu 1 \\ 1^T u - 1 \\ 1^v - 1 \end{bmatrix}, \tag{7}$$

where Q is the Hessian of the Lagrangian, given by

$$Q = \begin{bmatrix} \text{diag}(\frac{1}{u}) & P & 1 & 0 \\ P^T & -\text{diag}(\frac{1}{v}) & 0 & 1 \\ 1^T & 0 & 0 & 0 \\ 0 & 1^T & 0 & 0 \end{bmatrix}. \tag{8}$$

Differentiating Through QRE Solutions
The QRE solver also provides a method for computing the necessary Jacobian-vector products. The derivation follows in a similar manner to recent work in differentiating equality-constrained optimization problems [Gould *et al.*, 2016; Johnson *et al.*, 2016; Amos and Kolter, 2017] (the only difference being the min-max objective instead of a pure minimization objective, but since we compute differentials via the KKT conditions, the differences are minor). Specifically, given the solution (u^*, v^*) to the QRE, and considering some loss function $L(u^*, v^*)$ (for example, the log-likelihood of some observed data given this equilibrium probabilities), we show here how to compute the gradient of the loss with respect to the payoff P. In particular, taking differentials of the KKT conditions and rearranging leads to the following expression

$$Q \begin{bmatrix} \text{d}u & \text{d}v & \text{d}\mu & \text{d}\nu \end{bmatrix}^T = \begin{bmatrix} -\text{d}Pv & -\text{d}P^T u & 0 & 0 \end{bmatrix}^T. \tag{9}$$

For small changes denoted by $\text{d}u, \text{d}v$, we have

$$\text{d}L = \begin{bmatrix} \nabla_u L & \nabla_v L & 0 & 0 \end{bmatrix} \begin{bmatrix} \text{d}u & \text{d}v & \text{d}\mu & \text{d}\nu \end{bmatrix}^T$$
$$= \begin{bmatrix} \nabla_u L & \nabla_v L & 0 & 0 \end{bmatrix} Q^{-1} \begin{bmatrix} -\text{d}Pv & -\text{d}P^T u & 0 & 0 \end{bmatrix}^T$$
$$= \begin{bmatrix} v^T \text{d}P^T & u^T \text{d}P & 0 & 0 \end{bmatrix} Q^{-1} \begin{bmatrix} -\nabla_u L & -\nabla_v L & 0 & 0 \end{bmatrix}^T,$$

where the last step is from symmetry of Q. This expression governs how small changes in $\text{d}P$ affect L. For example, we may obtain the change in L after perturbing a single entry in P. Applying this procedure to all entries in P, simplifying and taking limits as $\text{d}P$ is small yields

$$\nabla_P L = y_u v^T + u y_v^T, \tag{10}$$

where

$$\begin{bmatrix} y_u & y_v & y_\mu & y_\nu \end{bmatrix}^T = Q^{-1} \begin{bmatrix} -\nabla_u L & -\nabla_v L & 0 & 0 \end{bmatrix}^T.$$

Hence, the forward and backward passes with our module are respectively given by: 1) Using the expression in (7), solve for the logit equilibrium given P, and 2) Using $\nabla_u L$ and $\nabla_v L$, obtain $\nabla_P L$ using (10). It is stressed that the module is sufficiently general to be included in any existing architecture where having a zero-sum game module is appropriate.

A Note on Identifiability
As mentioned in Section 1, it is natural to ask if the games are identifiable – that is, is there a unique P which under the logit QRE, generates u^*, v^*? The answer is no, in general. Assuming u^*, v^* are fixed, we can rewrite the KKT conditions in (6) as a system of linear equations in P. This system has $\mathcal{O}(nm)$ unknowns but only $\mathcal{O}(n + m)$ constraints. This implies that without a sufficiently compact parametrization, there will be infinitely many payoff matrices leading to identical equilibria. For example, one can add a constant to all entries in P without changing the QRE. When under-constrained, one cannot expect to recover P. However, as we show below, there are also many settings where it *is* possible to recover underlying parameters reliably.

3 Learning Extensive Form Games (EFG)

3.1 Sequence Form Representation

In practice, many games are more naturally and compactly represented in extensive form. Unfortunately, learning payoff matrices of their equivalent normal form representation is computationally unfeasible even for small games. For example, one-card poker has 2^{26} pure strategies per player. In order to facilitate learning of EFGs, we turn to the *sequence form* representation [Von Stengel, 1996], which is sufficiently rich to represent all strategic behaviors given perfect recall.

The sequence form replaces pure strategies by partial description of sequences specifying the player's moves over the game tree. Instead of probability vectors, we are interested in *realization plans* $u \in \mathbb{R}^n, v \in \mathbb{R}^m$, each a vector of size equal to possible actions throughout the game (thus, n and m are equal to the size of the game tree's action nodes for each player). Realization plans represent probabilities of performing a sequence of actions, in isolation from chance and other player's moves. Mathematically, this is represented by the linear constraints $Eu = e, Fv = f$. Here, E, F are matrices with entries in $\{-1, 0, 1\}$, while e, f are vectors containing $\{0, 1\}$. Together, they specify 'flow' constraints and implicitly encode parent-child relationships and information sets. These constraints may be seen as a generalization of the requirement that u, v lie in probability simplexes. The likelihood of arriving at a given node of the game tree is the product of probabilities of each sequence, multiplied by the probabilities needed by the chance player.

3.2 Dilated Entropy Regularization

Denote $\mathcal{I}_u$ and $\mathcal{I}_v$ to be all information sets for the min and max player. For an information set $i \in \mathcal{I}_u \cup \mathcal{I}_v$, $\mathcal{A}_i$ denotes the possible actions at information set i, while p_i is the action (from the same player) preceding i. Similarly, define ρ_a to be the information set immediately preceding the action a ,i.e. i where $a \in \mathcal{A}_i$. As with the normal form representation, we solve the regularized min-max problem

$$\min_u \max_v u^T P v + \sum_{i \in \mathcal{I}_u} \sum_{a \in \mathcal{A}_i} u_a \log \frac{u_a}{u_{p_i}} - \sum_{i \in \mathcal{I}_v} \sum_{a \in \mathcal{A}_i} v_a \log \frac{v_a}{v_{p_i}}. \tag{11}$$

This form of regularization is known as dilated entropy [Kroer *et al.*, 2017] or normalized entropy [Boyd and Vandenberghe, 2004], and is known to be strictly convex/concave in u and v respectively. Observe that this formulation operates in $\mathcal{O}(m + n)$ dimensions. The number of sequences is bounded by the size of the game tree and is normally much smaller than the number of pure strategies.

One of our first primary results in this paper is the fact that this particular form of regularization, applied to the sequence form, recovers the QRE as applied to the equivalent *reduced* normal form game (that is, the normal form representation of the extensive form, but with unattainable strategies omitted).

Theorem 1. *The solution to* (11) *is realization equivalent to the QRE of the game in reduced normal form.*

Proof. (Sketch) Consider the max player in isolation and his game tree, represented by alternating actions and information

states. Choose any action a_0 with parallel information sets, near the bottom of the tree (i.e. all child information sets are leaves). In sequence form, this action and its subtree may be coalesced into a set of strategies, corresponding to the Cartesian product $\prod_{\{i|p_i=a_0\}} \mathcal{A}_i$. This essentially converts a part of the sequence form into normal form. It can be shown that the solutions to (11) before and after this replacement are realization equivalent. The proof follows by repeated bottom-up application of this operation, eventually collapsing the tree to its reduced normal form, all while maintaining realization equivalence. The full proof is in the Appendix. $\square$

Theorem 1 shows dilated entropy regularization leads to a well-accepted solution concept, even if it differs slightly from the more traditional definition of the QRE for extensive form games [McKelvey and Palfrey, 1998]. A side consequence is that under certain regimes, the excessive gap technique [Kroer *et al.*, 2017; Hoda *et al.*, 2010] used to quickly solve zero-sum EFGs converges to the NE specified by QRE in reduced normal form, as the rationality-parameter tends to ∞.

3.3 Differentiable Learning in Sequence Form

Here we derive a differentiable formulation of the sequence form QRE, mirroring our derivation for the normal form case, but admittedly with significantly more complex notation due to the more involved entropy term. The KKT conditions of our optimization problem are,

$$(Pv)_a + 1 + \log(u_a) - \log(u_{p_i})$$
$$- J_a + \sum_{c \in \mathcal{C}_a} \mu_c - \mu_i = 0, \quad \forall i \in \mathcal{U}, a \in \mathcal{A}_i$$
$$(P^T u)_{a'} - 1 - \log(v_{a'}) + \log(v_{p_{i'}})$$
$$+ J_{a'} + \sum_{c \in \mathcal{C}_{a'}} \nu_c - \nu_{i'} = 0, \quad \forall i' \in \mathcal{V}, a' \in \mathcal{A}_{i'}$$
$$Eu - e = 0, \quad Fv - f = 0, \quad u \geq 0, \quad v \geq 0$$

where $\mathcal{C}_a, \mathcal{C}_{a'}$ are sets of possible information sets immediately following a or a'. $J_a, J_{a'}$ are their sizes, i.e. $|\mathcal{C}_a|, |\mathcal{C}_{a'}|$.

We write the terms on the left hand side as a vector $g(u, v, \mu, \nu)$. Taking derivatives again yields the updates for Newton's method.

$$\begin{bmatrix} -\Xi(u) & P & E^T & 0 \\ P^T & \Xi(v) & 0 & F^T \\ E & 0 & 0 & 0 \\ 0 & F & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \\ \Delta \mu \\ \Delta \nu \end{bmatrix} = -g(u, v, \mu, \nu)$$

$$\Xi(u)_{ab} = \begin{cases} -\frac{1+J_a}{u_a}, a = b \\ \frac{1}{u_b}, p_{\rho_a} = b \\ \frac{1}{u_a}, p_{\rho_b} = a \end{cases} \quad \Xi(v)_{a'b'} = \begin{cases} -\frac{1+J_{a'}}{v_{a'}}, a' = b' \\ \frac{1}{v_{b'}}, p_{\rho_{a'}} = b' \\ \frac{1}{v_{a'}}, p_{\rho_{b'}} = a' \end{cases}$$

The updates are done in exactly the same manner as (10)

$$\nabla_P L = y_u v^T + u y_v^T, \tag{12}$$

where

$$\begin{bmatrix} y_u \\ y_v \\ y_\mu \\ y_\nu \end{bmatrix} = \begin{bmatrix} -\Xi(u) & P & E^T & 0 \\ P^T & \Xi(v) & 0 & F^T \\ E & 0 & 0 & 0 \\ 0 & F & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} -\nabla_u L \\ -\nabla_v L \\ 0 \\ 0 \end{bmatrix}.$$

	R	P	S
R	0	$-b_1$	b_2
P	b_1	0	$-b_3$
S	$-b_2$	b_3	0

Figure 2: Payoff matrix of modified Rock-Paper-Scissors. Values shown are for the row player.

Implementation Notes

We can use this differentiable game solver within an automatic differentiation framework to easily obtain gradients of virtually any loss with respect to any of the game parameters. In particular, we used the PyTorch automatic differentiation library [Paszke *et al.*, 2017], and will release the full code for our solver as open source along with the release of this paper.

4 Experiments

We empirically demonstrate our module's novel aspects – learning *extensive form games* in the presence of *side information*, with *partial observations*. In the first experiment, we learn a non-symmetric variant of rock, paper, scissors with side information. We illustrate the learning of extensive form games with one-card poker, and learning with partial information with a security resource allocation game. In all cases, we minimize the log-loss of the observed sequence - that is, maximizing the likelihood of realizing observed *sequence* from the player, assuming he acts in accordance to the QRE.

Due to space constraints, hyper-parameters and details of train/test environment are deferred to the Appendix. Generally, our module works well with a medium or large batch size (e.g. 128), RMSProp [Tieleman and Hinton, 2012] or Adam [Kingma and Ba, 2014] optimizers with learning rates between $[0.0001, 0.01]$.

4.1 Rock, Paper, Scissors

Rock Paper Scissors (RPS) is among the most well-studied 2-player zero-sum game. It is well known that playing uniformly is an NE and QRE for RPS. In this experiment, we consider the following variant (Figure 2), which breaks symmetry between the 3 actions. Notice that the traditional RPS is recovered when b_1, b_2, b_3 are all 1.

We assume that each of the b's is a linear function of some features $x \in \mathbb{R}^2$, i.e., $b_y = x^T w_y, y \in \{1, 2, 3\}$, where w_y are to be learned. Features in the dataset and ground-truth weights are drawn uniformly from $[0, 1]$, and $[0, 10]$ respectively. Experiments were evaluated with a fixed test set of size 2000. The results presented in Figure 3 illustrate 2 key points. The first plot shows that dramatically improves with larger datasets, both in terms of parameters learned and predicted strategies. The second plot shows that with a reasonably sized dataset, convergence is stable and is fairly quick. However, it was observed that when the dataset is small (e.g. 200), the model may diverge from the ground truth.

4.2 One-Card Poker

We consider a simple poker game where players are dealt a single card, with ante/bets of 10, and two stages of betting for the first player. Specifically, suppose n cards labeled 1
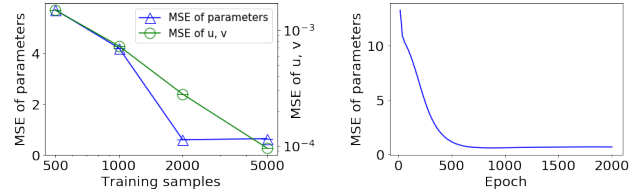

Figure 3: Left: MSE of parameters and predicted mixed strategies. Right: Convergence over 2000 epochs, using a dataset of size 2000.
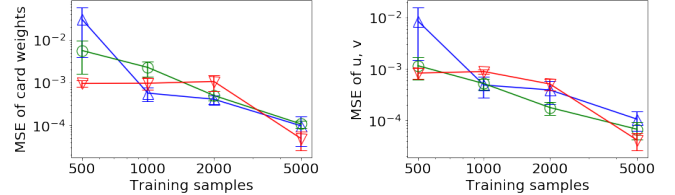

Figure 4: MSE of card weights (left) and realization plans (right).

through n are dealt uniformly. Both players begin with an ante of 10. Player 1 decides whether to bet an additional 10, followed by Player 2 (who folds if he does not call). Lastly, Player 1 may choose to bet if Player 2 raises. Both players are obliged to reveal their card at the end of each game.

While relatively simple, the game contains the key elements in extensive-form games and the strategic concepts in poker such as slow playing (e.g. not betting even if Player 1 holds the high card) and bluffing (e.g. betting even if the player holds a low card). Despite its simple structure, the game with n cards has 2^{2n} normal form pure strategies for each player. This exponential explosion of strategies is not improved by using the reduced normal form. However, in sequence form, we only need to work with realization plans of size $4n$.

Experimental Setup

We assume that the deck is stacked non-uniformly. Our goal is to learn this distribution of cards after observing many rounds of play. Let $d \in \mathbb{R}^n, d \geq 0, \sum_i d_i = 1$ be the *weights* of cards. The probability that the players are dealt cards (i, j) $(i \neq j)$ is $d_i \times \frac{d_j}{1-d_i}$. Note that this distribution is asymmetric between players. Matrices P, E, F for the case $n = 4$ are presented in the Appendix.

Remark. While counting cards seems to be a straightforward way to learn the card distribution when d does not change over time, our method is suited to learn the player's *perceived* or *believed* distribution of cards, which may be different from the distribution of cards dealt. This may even be a function of contextual features such as demographics of players.

A total of three experiments were run with $n = 4$. For each experiment, $d \sim \text{Dir}(1, 1, 1, 1)$. Each experiment comprises 5 runs of training, with same weights but different training sets. Training was for 2500 epochs, which was observed to be after convergence. The mean squared error of learned parameters are averaged over all runs and are presented in Figure 4.

$\{\#D_1, \#D_2\}$	$\{0, 3\}$	$\{1, 2\}$	$\{2, 1\}$	$\{3, 0\}$
T_1	$-R_1$	$-\frac{1}{2}R_1$	$-\frac{1}{4}R_1$	$-\frac{1}{8}R_1$
T_2	$-\frac{1}{8}R_2$	$-\frac{1}{4}R_2$	$-\frac{1}{2}R_2$	$-R_2$

Figure 5: Security resource allocation game. $n = 2, k = 3$.
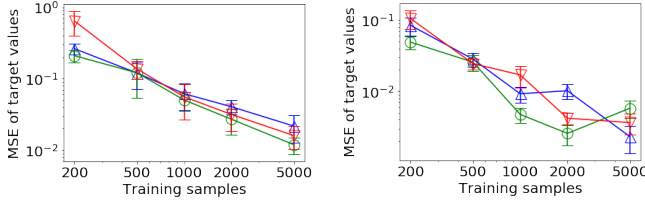
Figure 6: MSE of target values. Left: t=1, Right: t=2

4.3 Security Resource Allocation Game

In this set of experiments, we demonstrate the ability to learn from *incomplete observations* in a setting that abstracts attacks in cybersecurity domain. The defender possesses k indistinguishable and indivisible defensive resources, e.g., cyber analysts, which he splits among n targets, $\{T_1, ..., T_n\}$. In an attacking attempt, the attacker (row player) chooses one target. In the event an attack on T_i succeeds, the attacker obtains a reward of R_i (and the defender $-R_i$), otherwise, the payoffs to both parties are 0. Each defensive resource independently prevents an intrusion with probability 0.5. For example, if there are two defenders guarding T_1, the chance of a successful attack on T_1 is $\frac{1}{2^2}$. This creates a scenario where the marginal benefit of each defensive resources decreases, thus requiring the defender to strike a balance. The matrix of expected payoffs when $n = 2, k = 3$ is shown in Figure 5.

In addition to considering the case where the attacker launches a single attack, we also consider a multi-stage game where the attacker can launch t attacks, one in each stage while the defender chooses his allocation of resources in stage 1 and cannot change it in later stages. On the other hand, the attacker has the *option* of changing his target between stages. This describes a setting where analysts are deployed to specific network assets on a daily basis, while attackers are sufficiently nimble to make multiple attacks in a single day. To understand why the attacker may change target, consider that target T_i is attacked in stage 1 and the attack is unsuccessful. It may be inferred that it is more likely that T_i is better guarded, prompting the attacker to switch targets.

Three experiments are run with $n = 2, k = 5$ for games with single attack and double attack, i.e, $t = 1$ and $t = 2$. Crucially, in this set of experiments, we learn R_i only based on observations of the defender's actions. This setting yields a 10×6 sequence form payoff matrix. For each experiment, R_1 and R_2 are drawn uniformly in $[0, 2]$. Each experiment is run 10 times for at least 2000 epochs per run. The mean and standard error over each run is presented in Figure 6. The results show that our algorithm can still recover the game setting by only observing defender's actions.

4.4 Discussion

As expected, the quality of learned parameters improves as the number of data points increases. Notable exceptions occur in (i) the green plot for the security game when $t = 2$ and (ii) RPS, when comparing between training sizes of 2000 and 5000. These outliers are no longer observed when comparing MSE of u, v. For example, Figure 3 shows that predicted strategies improve significantly when going from 2000 to 5000 samples, showing that despite not converging to better parameters, the network still demonstrates a marked improvement in predicting player strategies.

5 Conclusion

In this paper, we present a fully differentiable module capable of learning payoff and other parameters in zero-sum games, given side information and partial observability. The proposed module's unique capabilities are demonstrated over a broad range of problems. Future work entails faster solvers by exploiting structure in the KKT matrix and extensions to learning general-sum games.

References

[Amin *et al.*, 2016] Kareem Amin, Satinder Singh, and Michael P Wellman. Gradient methods for stackelberg security games. In *Conference on Uncertainty in Artificial Intelligence*, pages 2–11, 2016.

[Amos and Kolter, 2017] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. *arXiv preprint arXiv:1703.00443*, 2017.

[Blum *et al.*, 2014] Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. Learning optimal commitment to overcome insecurity. In *Advances in Neural Information Processing Systems*, pages 1826–1834, 2014.

[Bowling and Veloso, 2000] Michael Bowling and Manuela Veloso. An analysis of stochastic game theory for multiagent reinforcement learning. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 2000.

[Bowling *et al.*, 2015] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit hold'em poker is solved. *Science*, 347(6218):145–149, 2015.

[Boyd and Vandenberghe, 2004] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[Brown and Sandholm, 2017] Noam Brown and Tuomas Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, page eaao1733, 2017.

[Busoniu *et al.*, 2008] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, And Cybernetics-Part C: Applications and Reviews, 38 (2), 2008*, 2008.

[Fang *et al.*, 2016] Fei Fang, Thanh Hong Nguyen, Rob Pickles, Wai Y Lam, Gopalasamy R Clements, Bo An, Amandeep Singh, Milind Tambe, and Andrew Lemieux. Deploying paws: Field optimization of the protection assistant for wildlife security. 2016.

[Fearnley *et al.*, 2015] John Fearnley, Martin Gairing, Paul W Goldberg, and Rahul Savani. Learning equilibria of games via payoff queries. *Journal of Machine Learning Research*, 16:1305–1344, 2015.

[Gould *et al.*, 2016] Stephen Gould, Basura Fernando, Anoop Cherian, Peter Anderson, Rodrigo Santa Cruz, and Edison Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, 2016.

[Hoda *et al.*, 2010] Samid Hoda, Andrew Gilpin, Javier Pena, and Tuomas Sandholm. Smoothing techniques for computing nash equilibria of sequential games. *Mathematics of Operations Research*, 35(2):494–512, 2010.

[Johnson *et al.*, 2016] Matthew Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in neural information processing systems*, pages 2946–2954, 2016.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Kroer *et al.*, 2017] Christian Kroer, Kevin Waugh, Fatma Kilinc-Karzan, and Tuomas Sandholm. Theoretical and practical advances on smoothing for extensive-form games. *arXiv preprint arXiv:1702.04849*, 2017.

[Letchford *et al.*, 2009] Joshua Letchford, Vincent Conitzer, and Kamesh Munagala. Learning and approximating the optimal strategy to commit to. In *International Symposium on Algorithmic Game Theory*, pages 250–262. Springer, 2009.

[McKelvey and Palfrey, 1995] Richard D McKelvey and Thomas R Palfrey. Quantal response equilibria for normal form games. *Games and economic behavior*, 10(1):6–38, 1995.

[McKelvey and Palfrey, 1998] Richard D McKelvey and Thomas R Palfrey. Quantal response equilibria for extensive form games. *Experimental economics*, 1(1):9–41, 1998.

[Mertikopoulos and Sandholm, 2016] Panayotis Mertikopoulos and William H Sandholm. Learning in games via reinforcement and regularization. *Mathematics of Operations Research*, 41(4):1297–1324, 2016.

[Moravčík *et al.*, 2017] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisỳ, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017.

[Paszke *et al.*, 2017] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[Pita *et al.*, 2009] James Pita, Manish Jain, Fernando Ordóñez, Christopher Portway, Milind Tambe, Craig Western, Praveen Paruchuri, and Sarit Kraus. Using game theory for los angeles airport security. *Ai Magazine*, 30(1):43, 2009.

[Tieleman and Hinton, 2012] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[Von Stengel, 1996] Bernhard Von Stengel. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14(2):220–246, 1996.

[Vorobeychik *et al.*, 2007] Yevgeniy Vorobeychik, Michael P Wellman, and Satinder Singh. Learning payoff functions in infinite games. *Mach Learn*, 67:145–168, 2007.

[Waugh *et al.*, 2011] Kevin Waugh, Brian D Ziebart, and J Andrew Bagnell. Computational rationalization: the inverse equilibrium problem. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 1169–1176. Omnipress, 2011.

[Zinkevich *et al.*, 2008] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Advances in neural information processing systems*, pages 1729–1736, 2008.