What game are we playing? Differentiably learning games from incomplete observations

Carnegie Mellon University

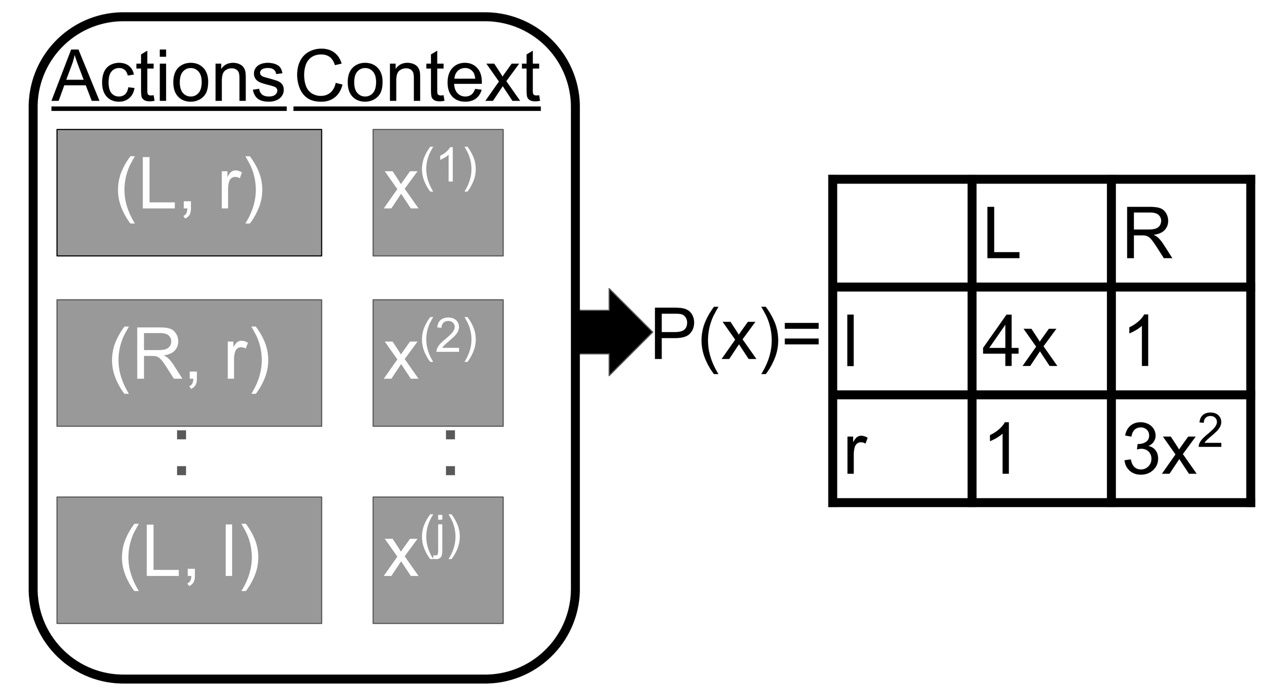
Chun Kai Ling[1], J. Zico Kolter[1], Fei Fang[2]

Department of Computer Science[1]*, Institute for Software Research*[2]*, Carnegie Mellon University*

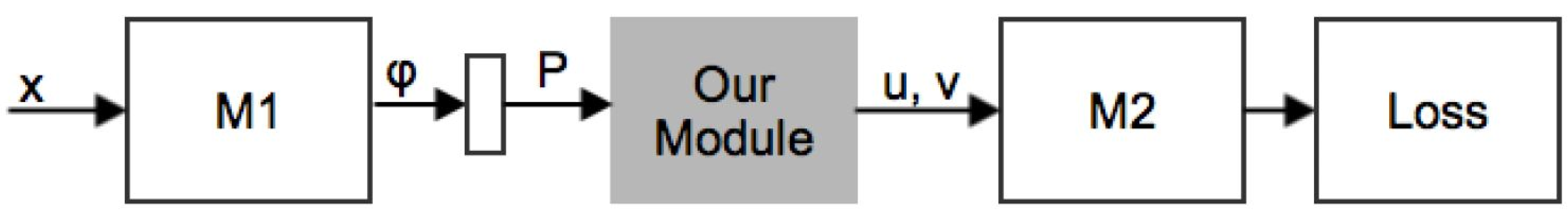
{chunkail, zkolter}@cs.cmu.edu, feifang@cmu.edu

(I) Motivation

- Our goal is to understand underlying utilities of agents in non-cooperative settings based only on observations.
- Game Theory finds optimal strategies based on known payoffs. Our setting, sometimes known as inverse game theory (e.g., Kuleshov, Waugh et al, 2011) is the reverse.
- Given a *context* x, we predict a matrix P(x), adapting to novel situations.
- Prior work either ignores context, or is restricted to special structural properties (e.g., symmetry in Vorobeychik, 2007).



(II) Contributions



- We propose a fully differentiable model which finds the Logit Quantal Response Equilibrium (QRE).
- Training may be done end-to-end by minimizing log-loss of actions observed by players.
- Our module is sufficiently flexible to learn from actions of just a single player.

(III) Approach

- Assume game to be learnt is zero-sum, normal form.
- Modelling behavior with QRE yields a unique, smooth equilibrium which is equal to regularization by entropy.
  - Results in a convex-concave problem
  - Efficient solution obtained using Newton's method
- Backpropagation performed using implicit differentiation (Dontchev & Rockafellar, 2009)
  - Gradient with respect to P is computed implicitly via Jacobian of KKT conditions (Amos & Kolter, 2017)

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

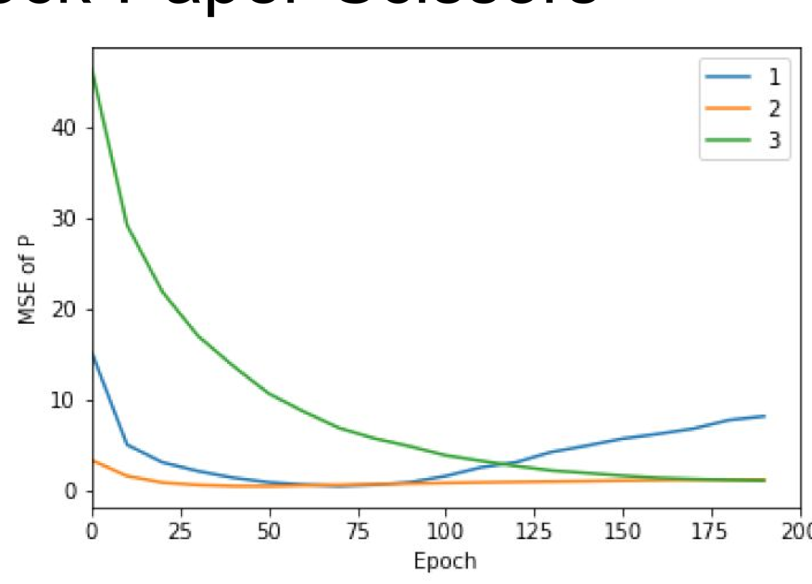
$$\nabla_P L = y_u v^T + u y_v^T, \quad \begin{bmatrix} y_u \\ y_v \\ y_\mu \\ y_\nu \end{bmatrix} = \begin{bmatrix} \text{diag}(1/u) & P & 1 & 0 \\ P^T & -\text{diag}(1/v) & 0 & 1 \\ 1^T & 0 & 0 & 0 \\ 0 & 1^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} -\nabla_u L \\ -\nabla_v L \\ 0 \\ 0 \end{bmatrix}$$

(IV) Experiments

Expt 1: Modified Rock-Paper-Scissors

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

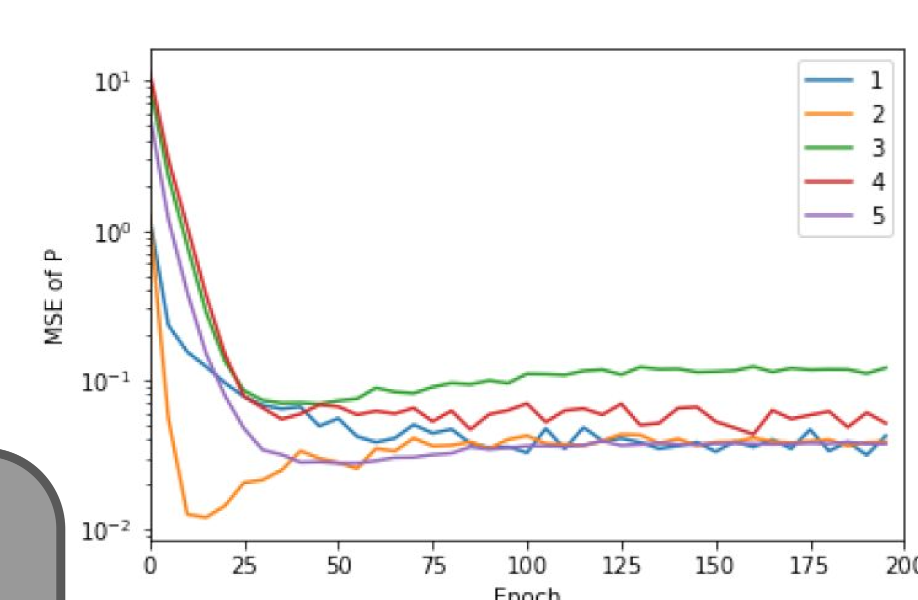
Top: P parameterized by b. Right: MSE over number of epochs

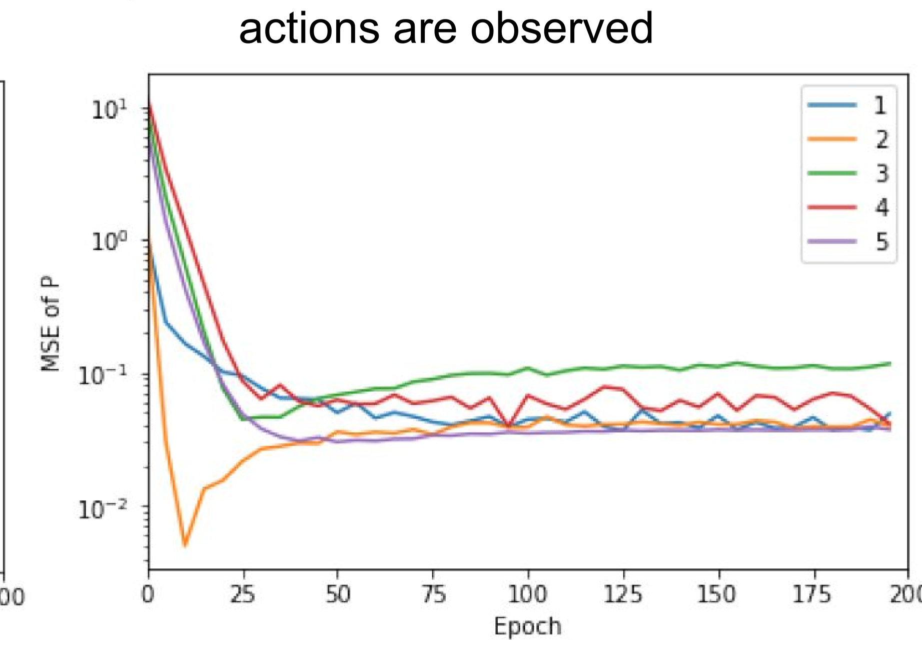


Expt 2: Resource Allocation Game

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

Left: P parameterized by R1, R2. Bottom-left: MSE when both player actions are used for training. Bottom-right: MSE when only the column player's actions are observed

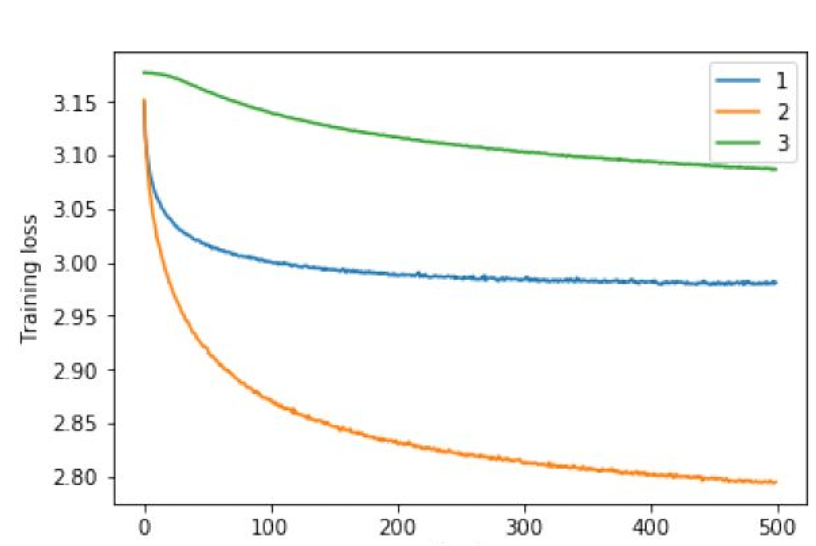


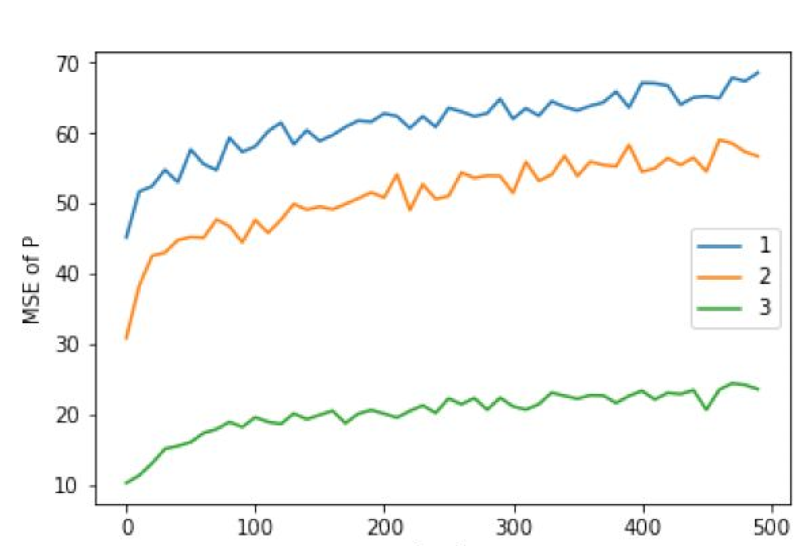


Expt 3: Compact Security Games

Bottom-left: Training log-loss of optimal actions. Bottom-right: MSE of P. Right: Validation loss compared to optimum. (Refer to the paper for details of game)

#	Val.	Optimal
1	2.993	2.841
2	2.776	2.524
3	3.083	2.957





By minimizing log-loss, most payoff matrices P were learnt accurately

(V) Discussion

- Learnt P(x) accurately from single player's action (Expt 2).
- Notable identifiability issues if P is poorly parameterized
  - Multiple P lead to same strategies (Expt 3)
  - Predicting strategies well does not imply payoffs are learnt.
  - Sensitivity of optimal actions to perturbations in P
- Future work in extensive form and general-sum games
- Applications: Security Games, Multiagent-RL