MIT 6.S890 Topics in Multiagent Learning Fall 2024

Homework 2

Released on: Oct. 8, 2024 **Due on: Oct. 22, 2024, 11:59pm ET**

Instructions This homework contains four problems (each split into smaller tasks), for a total of 100 points. Please attach a readable printout of your code for Problem 3 at the end of your submission.

Collaboration policy You are welcome to discuss the problems with other students. However you should write up the solutions by yourself.

Recall: RVU regret bounds

Recall in class, we discussed different variants of external regret minimizers:

- Predictive FTRL:

$$x^{(t)} \leftarrow \arg\max_{x\in\mathcal{X}} \left\{ \left\langle m^{(t)} + \sum_{\tau=1}^{t-1} g^{(\tau)}, x \right\rangle - \frac{1}{\eta}\psi(x) \right\}. \tag{1}$$

- Predictive OMD (non-reflected):

$$\begin{aligned} z^{(t)} &\leftarrow \arg\max_{z\in\mathcal{X}} \left\{ \langle g^{(t-1)}, z\rangle - \frac{1}{\eta} D_\psi(z\|z^{(t-1)}) \right\}, \\ x^{(t)} &\leftarrow \arg\max_{x\in\mathcal{X}} \left\{ \langle m^{(t)}, x\rangle - \frac{1}{\eta} D_\psi(x\|z^{(t)}) \right\}. \end{aligned} \tag{2}$$

- Predictive OMD (reflected):

$$x^{(t)} \leftarrow \arg\max_{x\in\mathcal{X}} \left\{ \langle g^{(t-1)} + m^{(t)} - m^{(t-1)}, x\rangle - \frac{1}{\eta} D_\psi(x\|x^{(t-1)}) \right\}. \tag{3}$$

It is known that all the above algorithms enjoy the following regret bound:

Proposition 1 (RVU bound [Syrgkanis et al., 2015]). When $\nabla\psi(x^{(1)}) = \mathbf{0}$, all variants of Predictive FTRL and OMD described above satisfy the following regret bound:

$$\mathrm{Reg}^{(T)} \coloneqq \sum_{t=1}^{T} \left(\langle x^*, g^{(t)}\rangle - \langle x^{(t)}, g^{(t)}\rangle \right) \le \max_{\widehat{x}\in\mathcal{X}} \frac{\psi(\widehat{x}) - \psi(x^{(1)})}{\eta} + \eta \sum_{t=1}^{T} \|g^{(t)} - m^{(t)}\|_*^2 - \frac{1}{8\eta} \sum_{t=2}^{T} \|x^{(t)} - x^{(t-1)}\|^2,$$

where $\|\cdot\|$ is any norm with respect to which ψ is 1-strongly convex, and $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$.

1 Optimistic Multiplicative Weights Update (30 points)

As we discussed in class, *multiplicative weights update (MWU)* and its predictive variant *optimistic multiplicative weights update (OMWU)* are popular regret minimization algorithms for the probability simplex

$$\Delta^n \coloneqq \{(x_1, \dots, x_n) \in \mathbb{R}^n_{\geq 0} : x_1 + \cdots + x_n = 1\}. \tag{4}$$

They enjoy many strong theoretical properties, and were involved in a series of important papers in game theory. In this question, you will derive and analyze MWU and OMWU from first principles.

1.1 The negative entropy regularizer (8 points)

Recall that OMWU is a special name for the predictive FTRL algorithm when the regularizer $\psi : \Delta^n \to \mathbb{R}$ is chosen to be the *negative entropy regularizer*

$$\psi(x) \coloneqq \sum_{i=1}^{n} x_i \log(x_i).$$

To avoid annoying issues with the logarithm of 0, we will only ever evaluate and differentiate ψ in the relative interior of Δ^n, that is the set

$$\operatorname{relint} \Delta^n = \{(x_1, \dots, x_n) \in \mathbb{R}^n_{>0} : x_1 + \cdots + x_n = 1\}$$

(note the strict inequality $\mathbb{R}_{>0}$, as opposed to $\mathbb{R}_{\geq 0}$ in (4)).

For ψ to be a valid choice of regularizer in predictive FTRL, we need to check that ψ is 1-strongly convex with respect to some norm. In particular, it turns out that ψ is 1-strongly convex both with respect to the Euclidean norm

$$\|x\|_2 \coloneqq \sqrt{\sum_{i=1}^{n} x_i^2} \qquad \forall\, x \in \mathbb{R}^n$$

and with respect to the ℓ_1 norm

$$\|x\|_1 \coloneqq \sum_{i=1}^{n} |x_i| \qquad \forall\, x \in \mathbb{R}^n.$$

The easiest way to verify strong convexity in this case passes through the following well-known characterization.

Lemma 1. Let $\mathcal{X} \subseteq \mathbb{R}^n$ be a convex set, $f : \mathcal{X} \to \mathbb{R}$ be a twice-differentiable function with Hessian matrix $\nabla^2 f(x)$ at every $x \in \mathcal{X}$, and $\|\cdot\|$ be a norm. If

$$\|u\|^2_{\nabla^2 f(x)} \coloneqq u^\top \nabla^2 f(x)\, u \geq \|u\|^2 \qquad \forall\, u \in \mathbb{R}^n, x \in \mathcal{X},$$

then f is 1-strongly convex on $\mathcal{X}$ with respect to norm $\|\cdot\|$.

In the next two exercises you will use Lemma 1 to verify that ψ is 1-strongly convex on $\operatorname{relint} \Delta^n$ with respect to $\|\cdot\|_2$ and $\|\cdot\|_1$.

Problem 1.1 (4 points). Apply Lemma 1 for $\mathcal{X} = \operatorname{relint} \Delta^n$, $f = \psi$, and $\|\cdot\| = \|\cdot\|_2$ and conclude that ψ is 1-strongly convex with respect to the Euclidean norm on $\operatorname{relint} \Delta^n$.

★ Hint: the Hessian matrix of ψ is particularly nice. Start by working that out first.
★ Hint: at some point, it might be useful to argue that $1/x_i \geq 1$ for any $i \in \{1, \dots, n\}$ whenever $x \in \operatorname{relint} \Delta^n$.

Problem 1.2 (4 points). Apply Lemma 1 for $\mathcal{X} = \operatorname{relint} \Delta^n$, $f = \psi$, and $\|\cdot\| = \|\cdot\|_1$ and conclude that ψ is 1-strongly convex with respect to the ℓ_1 norm on $\operatorname{relint} \Delta^n$.

★ Hint: The Cauchy-Schwarz inequality asserts that for any pair of vectors $a, b \in \mathbb{R}^n$,

$$\left(\sum_{i=1}^n a_i b_i\right)^2 \leq \left(\sum_{i=1}^n a_i^2\right)\left(\sum_{i=1}^n b_i^2\right). \tag{5}$$

Now, let $x \in \operatorname{relint} \Delta^n$ and $u \in \mathbb{R}^n$, and consider the vectors $a \coloneqq (u_1/\sqrt{x_1}, \dots, u_n/\sqrt{x_n})$ and $b \coloneqq (\sqrt{x_1}, \dots, \sqrt{x_n})$. What happens if you plug them into (5)? Don't forget that $x_1 + \cdots + x_n = 1$ since $x \in \operatorname{relint} \Delta^n$.

1.2 Gradient of ψ and of its conjugate (9 points)

In this subsection, you will derive a formula for the gradient of ψ and for the gradient of its convex conjugate. We start by formulate the gradient of the regularizer.

Problem 1.3 (3 points). Give an expression for the gradient of ψ at any point $x \in \operatorname{relint} \Delta^n$.

Now, let's focus on the gradient of the convex conjugate of ψ, that is, the solution to the optimization problem

$$\nabla \psi^*(g) \coloneqq \operatorname*{arg\,max}_{\widehat{x} \in \operatorname{relint} \Delta^n} \{\langle g, \widehat{x} \rangle - \psi(\widehat{x})\}. \tag{6}$$

Problem (6) is a *constrained* optimization problem, since the optimization variable $\widehat{x}$ is constrained to satisfy $\widehat{x} \in \operatorname{relint} \Delta^n$. Call x^* the optimal solution to (6). As a result of an important theorem in optimization theory (the Lagrange multiplier theorem), there exists a constant (called *Lagrange multiplier*) $\alpha \in \mathbb{R}$ such that

$$g - \nabla \psi(x^*) = \alpha \mathbf{1}, \tag{7}$$

where $\mathbf{1} \in \mathbb{R}^n$ is the vector of all ones.

Problem 1.4 (3 points). Plug in the expression for the gradient of ψ that you developed in Problem 1.3 into (7). Note that (7) is a vector equation, and therefore it is equivalent to a system of n scalar equations. Isolate and solve for x_i^* for every $i \in \{1, \dots, n\}$, and show that

$$x_i^* = e^{-1-\alpha} \cdot e^{g_i} \qquad \forall\, i \in \{1, \dots, n\}. \tag{8}$$

Problem 1.5 (3 points). Use Equation (8) together with the fact that the sum of the entries of $x^* \in \operatorname{relint} \Delta^n$ must be 1 to solve for the value of the Lagrange multiplier α. Then, plug in the value of α to conclude that for any $g \in \mathbb{R}^n$, $\nabla \psi^*(g)$—that is, the solution to the argmax in (6)—satisfies

$$x_i^* = \frac{e^{g_i}}{\sum_{j=1}^n e^{g_j}} \qquad \forall\, i \in \{1, \dots, n\}.$$

1.3 OMWU as predictive FTRL (9 points)

Now that we verified that ψ is 1-strongly convex, we can safely run predictive FTRL with ψ as a regularizer. As a reminder, we list the update rule of predictive FTRL in (1). In our case, $\mathcal{X}$ will be the relative interior $\operatorname{relint}\Delta^n$ of the probability simplex Δ^n, the regularizer ψ will be the negative entropy function ψ, and $\eta > 0$ will be a generic step size. The resulting algorithm is called OMWU.

Problem 1.6 (3 points). Use the characterization of $\nabla\psi^*(g)$ given in the statement of Problem 1.5 to prove that at times $t = 2, 3, \dots$, for all $i \in \{1, \dots, n\}$, the strategies $x^{(t)} \in \Delta^n$ produced by OMWU satisfy

$$x_i^{(t)} = \frac{x_i^{(t-1)} \exp\left(\eta(g_i^{(t-1)} + m_i^{(t)} - m_i^{(t-1)})\right)}{\sum_{j=1}^n x_j^{(t-1)} \exp\left(\eta(g_j^{(t-1)} + m_j^{(t)} - m_j^{(t-1)})\right)},$$

Since OMWU is just predictive FTRL, we can use the known regret bound for predictive FTRL we saw in class to give a regret guarantee for OMWU. In the particular case of OMWU, the negative entropy function ψ was proven to be 1-strongly convex with respect to both the Euclidean norm (Problem 1.1) and the ℓ_1 norm (Problem 1.2). So, in principle, either norm can be used in Proposition 1. However, one choice dominates the other.

Problem 1.7 (3 points). The negative entropy function ψ is 1-strongly convex with respect to both the Euclidean norm (Problem 1.1) and the ℓ_1 norm (Problem 1.2). So, in principle, either norm can be used when invoking Proposition 1. Which norm do you think leads to a stronger regret bound, and why?

Problem 1.8 (3 points). Prove that the range satisfies

$$\max_{x^* \in \operatorname{relint}\Delta^n} \psi(x^*) - \psi(x^{(1)}) \le \log n.$$

Then, use Proposition 1—which was stated in general for any instantiation of FTRL—to argue that OMWU for the simplex Δ^n satisfies the regret bound

$$\mathrm{Reg}^{(T)} \le \frac{\log n}{\eta} + \eta \sum_{t=1}^T \|g^{(t)} - m^{(t)}\|_\infty^2 - \frac{1}{8\eta} \sum_{t=2}^T \|x^{(t)} - x^{(t-1)}\|_1^2.$$

★ Hint: The minimizer x^* of ψ over $\operatorname{relint}\Delta^n$ is $\nabla\psi^*(0)$. You already know how to compute this from Problem 1.5.
★ Hint: The supremum of ψ over $\operatorname{relint}\Delta^n$ is 0 (you should prove this).
★ Hint: You can take for granted the fact that $\|\cdot\|_\infty$ is the dual norm of $\|\cdot\|_1$.

1.4 OMWU as predictive OMD (4 points)

It turns out that OMWU—which was defined as the instance of predictive FTRL in which the regularizer is set the negative entropy function—is equivalent to predictive OMD with negative entropy function, in the sense that the two algorithms produce the same iterates at every time t. As a reminder, the Bregman divergence $D_\psi(\cdot \,\|\, \cdot)$ is defined with respect to any regularizer ψ and any two points x, c as

$$D_\psi(x \,\|\, c) \coloneqq \psi(x) - \psi(c) - \langle \nabla\psi(c), x - c\rangle.$$

Problem 1.9 (4 points). Consider the predictive OMD algorithm with updates in (2), where $\mathcal{X}$ is set to be the relative interior relint Δ^n of the n-simplex, the regularizer ψ is set to be the negative entropy function ψ, and $\eta > 0$ is a generic stepsize. Prove that the iterates produced by that algorithm coincide with those produced by OMWU as defined in Section 1.3.

2 Best Iterate Convergence of OGDA (20 points)

Consider the following max-min optimization problem:

$$\max_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} x^\top A y,$$

where $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}^n$ are the *convex* feasible domains. This problem can be seen as a two-player zero-sum game, where the max player controls x, and the min player controls y.

A natural approach to solving this problem is to run gradient descent/ascent for each player simultaneously. This leads to the Gradient Descent Ascent (GDA) algorithm, where x and y are optimized independently via proximal gradient descent/ascent:

$$\begin{cases} x^{(t)} \leftarrow \Pi_{\mathcal{X}}\big(x^{(t-1)} + \eta A y^{(t-1)}\big), \\ y^{(t)} \leftarrow \Pi_{\mathcal{Y}}\big(y^{(t-1)} - \eta A^\top x^{(t-1)}\big), \end{cases}$$

where $\Pi_{\mathcal{X}}(z) := \arg\min_{x \in \mathcal{X}} \|x - z\|_2^2$ is the projection of z onto $\mathcal{X}$ in Euclidean space.

However, the algorithm is not guaranteed to converge for any small constant step size $\eta > 0$, even when A is a 2×2 matrix (although the averages of $x^{(t)}$ and $y^{(t)}$ do converge).

It turns out that a slight modification to the algorithm can lead to significant improvements. The idea is to introduce optimism to the algorithm. Specifically, the Optimistic Gradient Descent Ascent (OGDA) algorithm follows these dynamics:

$$\begin{cases} x^{(t)} \leftarrow \Pi_{\mathcal{X}}\big(x^{(t-1)} + 2\eta A y^{(t-1)} - \eta A y^{(t-2)}\big), \\ y^{(t)} \leftarrow \Pi_{\mathcal{Y}}\big(y^{(t-1)} - 2\eta A^\top x^{(t-1)} + \eta A^\top x^{(t-2)}\big), \end{cases}$$

The algorithm starts by setting $x^{(1)} \in \mathcal{X}$ and $y^{(1)} \in \mathcal{Y}$ as arbitrary initial points in the feasible set. The goal of this problem is to prove OGDA is guaranteed to converge to the saddle point in the best-iterate sense, i.e.,

$$\max_{x^* \in \mathcal{X}} \min_{y^* \in \mathcal{Y}} \liminf_{t \to \infty} \big((x^*)^\top A y^{(t)} - (x^{(t)})^\top A y^*\big) = 0.$$

Since optimistic gradient descent is a specific instance of reflected predictive OMD, it is possible to derive an RVU regret bound for both players:

Problem 2.1 (3 points). Recall the cumulative regret of the max player is defined by

$$\mathrm{Reg}_1^{(T)} := \max_{x^* \in \mathcal{X}} \sum_{t=1}^{T} (x^*)^\top A y^{(t)} - \sum_{t=1}^{T} (x^{(t)})^\top A y^{(t)}.$$

Using Proposition 1, show that when we run OGDA for both players,

$$\mathrm{Reg}_1^{(T)} \le \frac{\|\mathcal{X}\|_2^2}{2\eta} + \eta \sum_{t=2}^{T} \|A y^{(t)} - A y^{(t-1)}\|_2^2 - \frac{1}{8\eta} \sum_{t=2}^{T} \|x^{(t)} - x^{(t-1)}\|_2^2,$$

where $\|\mathcal{X}\|_2 := \max_{x \in \mathcal{X}} \|x\|_2$ is the radius of the feasible set. Then, argue that the min player also enjoys

a similar regret bound.

★ Hint: Gradient descent can be captured by OMD with Euclidean norm.

According to the regret bound, we are then able to show the dynamic of OGDA is converging, when the learning rate is sufficiently small:

Problem 2.2 (7 points). Denote by $\|A\|_2 := \max_{y\in\mathbb{R}^n} \|Ay\|_2/\|y\|_2$ the operator norm of matrix A. Show that when the learning rate is sufficiently small, i.e., $\eta \le 1/(4\|A\|_2)$, the total distance of policy changes is sublinear:

$$\sum_{t=2}^{T} \left(\|x^{(t)} - x^{(t-1)}\|_2 + \|y^{(t)} - y^{(t-1)}\|_2\right) \le 4\sqrt{T}(\|\mathcal{X}\|_2 + \|\mathcal{Y}\|_2).$$

★ Hint: The sum of the cumulative regret of both players is non-negative since it is the sum of Nash gaps.
★ Hint: Apply the Cauchy-Schwarz inequality to convert the upper bound of $\sum \|\cdot\|_2^2$ into an upper bound of $\sum \|\cdot\|_2$.

In the context of proximal gradient descent, it is known that for the projection step $w \leftarrow \Pi_{\mathcal{X}}(v)$, the following inequality holds for any $x \in \mathcal{X}$:

$$\langle v - w, x - w\rangle \le 0.$$

Intuitively, this inequality implies that the vector $v - w$ points outward from the convex set $\mathcal{X}$ at the point w, and any vector $x - w$ pointing inside $\mathcal{X}$ forms an obtuse angle with $v - w$; geometrically, this means that moving from w towards any $x \in \mathcal{X}$ cannot have a positive inner product with $v - w$, which allows us to utilize this property to demonstrate the desired result:

Problem 2.3 (10 points). Show that when the learning rate satisfies $\eta \le 1/(4\|A\|_2)$, the max player achieves a sublinear cumulative regret:

$$\max_{x^*\in\mathcal{X}} \sum_{t=1}^{T} (x^*)^\top A y^{(t)} - \sum_{t=1}^{T} (x^{(t)})^\top A y^{(t)} \le \mathcal{O}\Big(\frac{\sqrt{T}}{\eta}(\|\mathcal{X}\|_2 + \|\mathcal{Y}\|_2)^2\Big).$$

Then, conclude that OGDA achieves best-iterate convergence. In specific, show that with proper selection η, for all T, there exists some $t \le T$, such that

$$(x^*)^\top A y^{(t)} - (x^{(t)})^\top A y^* \le \mathcal{O}\Big(\frac{1}{\sqrt{T}}(\|\mathcal{X}\|_2 + \|\mathcal{Y}\|_2)^2\|A\|_2\Big).$$

★ Hint: Start by applying the inequality $\langle v-w, x-w\rangle \le 0$ with update rule $x^{(t)} \leftarrow \Pi_{\mathcal{X}}\left(x^{(t-1)} + 2\eta A y^{(t-1)} - \eta A y^{(t-2)}\right.$ and $x = x^*$. From this, extract an upper bound for $(x^*)^\top A y^{(t-1)} - (x^{(t-1)})^\top A y^{(t-1)}$.
★ Hint: The additional inner product terms can be upper bounded individually using the Cauchy-Schwarz inequality.

3 Implementation of Learning in Extensive-Form Games (40 points)

In this problem, you will implement variants of CFR to solve extensive-form games, specifically focusing on two small poker variants: Kuhn poker [Kuhn, 1950] and Leduc poker [Southey et al., 2005], the latter being the larger of the two. Unlike the first homework, we provide the extensive-form representation of the game in the attached zip file, as listed below:

3.1 Game file format

Each input file contains a description of the game tree across multiple lines, encoding all the information needed to reconstruct the game. The information lists all game states line by line, followed by all the relevant details. Each line describes either a game state or an information set.

3.1.1 Game state

We identify different game states using the sequence of actions taken by all players and the nature. The game history is formatted as:

`/<P1>:<A1>/<P2>:<A2>/.../<Pn>:<An>/`

where `<Pi>:<Ai>` represents an action taken, with:

- `<Pi>` identifying the acting entity: `C` for the chance (dealer), `P1` for Player 1, and `P2` for Player 2.
- `<Ai>` representing the action taken or the card dealt. Specifically:
  - When the action is taken by either player, `<Ai>` denotes an action such as check or call (`c`), raise (`r`), or fold (`f`).
  - When the action is taken by the environment, `<Ai>` represents a card.

There are three types of nodes: decision nodes, chance nodes, and terminal nodes.

Decision nodes Decision nodes represent points where a player must choose an action:

`node <HISTORY> player <X> actions <A1> ... <An>`

where:

- `<HISTORY>` represents the sequence of actions leading to this node from the root.
- `<X>` denotes the index of the player (`1` or `2`).
- `<A1>, ..., <An>` are the available actions for Player `<X>` at this node.

Chance nodes Chance nodes represent randomness in the outcome of the environment. In the specific case of poker, these nodes represent the stochastic outcome of dealing cards:

`node <HISTORY> chance actions <A1>=<P1> ... <An>=<Pn>`

where:

- `<HISTORY>` represents the sequence of actions leading to this node from the root.
- `<A1>, ..., <An>` are the possible actions taken by the chance player.
- `<P1>, ..., <Pn>` are the probabilities associated with each action, satisfying `P1 + ... + Pn` $= 1$.

Terminal nodes Terminal nodes represent the end of a game sequence, where payoffs are assigned to the players:

`node <HISTORY> terminal payoffs 1=<Q1> 2=<Q2>`

where:

- `<HISTORY>` represents the sequence of actions leading to this node from the root.
- `<Q1>` and `<Q2>` are the payoffs for Player 1 and Player 2, respectively.

3.1.2 Information sets

Information sets group nodes that are indistinguishable to a player due to incomplete information. Each information set is defined by:

```
infoset <NAME> nodes <N1> ... <Nn>
```

where:

- `<NAME>` is a unique identifier for the information set.
- `<N1>, ..., <Nn>` are the histories of the nodes within the information set.

3.2 Find the best response

When implementing learning algorithms, it is always a good start to implement the best-response oracle. This allows you to verify the performance of your algorithm.

Problem 3.1 (10 points)**.** Implement a function that computes the best response for a given strategy. In both games, find the best response for Player 1 against a uniform strategy for Player 2, who plays uniformly over all valid actions at each decision node. Report the expected utility for Player 1. Then, compute the Nash equilibrium gap of the strategy profile in which both players play the uniform strategy. Recall that the Nash equilibrium gap of a strategy profile (x, y) is given by

$$\gamma(x, y) := \max_{x^* \in \mathcal{X}} u_1(x^*, y) - \min_{y^* \in \mathcal{Y}} u_1(x, y^*)$$

where $u_1(x, y)$ is the expected payoff for Player 1 when both players play strategies x and y, respectively.

3.3 Learning to best response

After implementing CFR, a good way to verify its functionality is to check if it can learn the best response. In particular, you will verify that your implementation, applied to Player 1, learns a best response against Player 2 when Player 2 plays the *uniform* strategy at each information set.

Problem 3.2 (15 points)**.** Implement CFR for Player 1 with regret matching at each decision node against a uniform strategy for Player 2. Compute the average strategies $(\bar{x}^T, \bar{y}^T)$ by averaging the *reach probability* in sequence form:

$$\bar{x}^{(T)} := \frac{1}{T} \sum_{t=1}^{T} x^{(t)}, \qquad \bar{y}^{(T)} := \frac{1}{T} \sum_{t=1}^{T} y^{(t)}. \tag{9}$$

Plot the expected utility (for Player 1) of the average strategies $(\bar{x}^T, \bar{y}^T)$ as a function of the number of iterations $T = 1, \ldots, 1000$. You will find the expected utility finally converge to the value reported in the previous problem.

Your solution should include two plots (one for each game). Don't forget to turn in your implementation.

3.4 Learning to Nash equilibrium

Now, it is time to find the Nash equilibrium in both games! The easiest way to do this is to run CFR for both players.

Problem 3.3 (15 points)**.** Instantiate your implementation of CFR for both players to run against each other. Compute the average strategy for both players. Plot the Nash equilibrium gap $\gamma(\bar{x}^T, \bar{y}^T)$ and the expected utility $u_1(\bar{x}^T, \bar{y}^T)$ for Player 1 of the average strategies as a function of the number of iterations $T = 1, \ldots, 1000$.

Your solution should include four plots (two for each game—one for the Nash equilibrium gap and one for the utility). Don't forget to turn in your implementation.

4 Project Proposal Submission (10 points)

For this assignment, you are required to submit a concise proposal for your course project. Your proposal should clearly define the problem you aim to address, and must contain the following elements:

- Challenge type or project title
- Names of team members
- Main objectives of the project
- Summary of the potential approach
- How will you spend the project breaks

You are encouraged to choose any project that captures your interest. For those interested in empirical work, we recommend considering the "Phantom Tic-Tac-Toe Challenge". Detailed descriptions of these projects will be made available in the course modules.

References

Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E Schapire. Fast convergence of regularized learning in games. In *Advances in Neural Information Processing Systems*, 2015.

H. W. Kuhn. A simplified two-person poker. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games*, volume 1 of *Annals of Mathematics Studies, 24*, pages 97–103. Princeton University Press, Princeton, New Jersey, 1950.

Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes' bluff: opponent modelling in poker. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pages 550–558, 2005.

6.S890 Topics in Multiagent Learning

Fall 2024