Decision-Making in Multi-Agent Systems

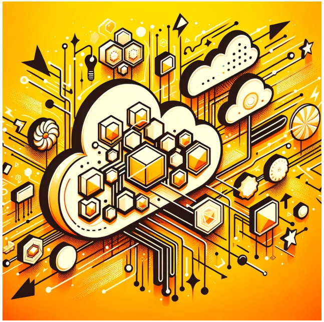
Delays, Adaptivity, and Learning in Games

Yu-Guan Hsieh

Illustrating Examples: Generated Images

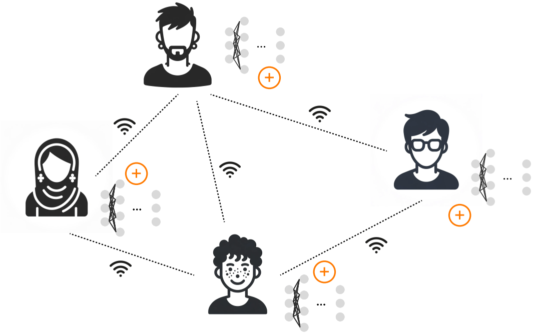
Collective improvement of a model by a group of users

Market of cloud GPU platforms

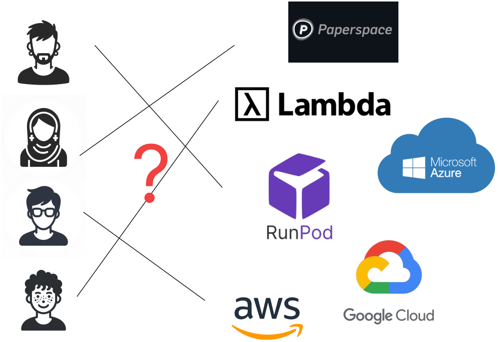


Illustrating Examples: Manually Created Images

Collective improvement of a model by a group of users

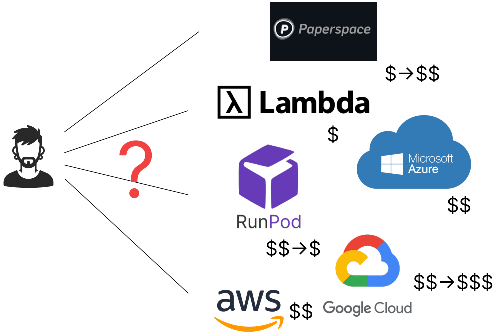


Market of cloud GPU platforms



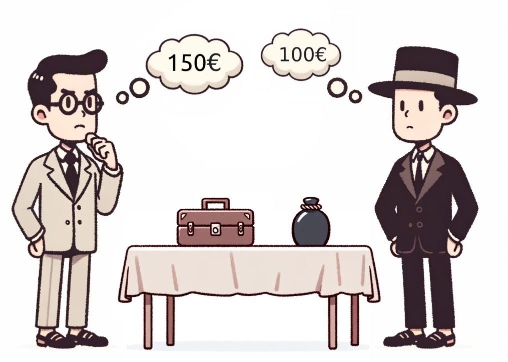
Decision-Making in Multi-Agent Systems: Challenges

- Non-stationary environment [Online learning]



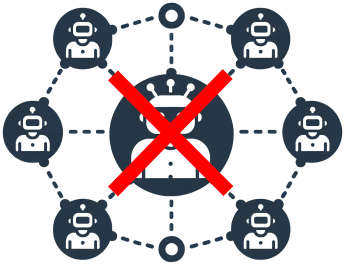
Decision-Making in Multi-Agent Systems: Challenges

- Non-stationary environment [Online learning]
- Conflicting interests [Game theory]



Decision-Making in Multi-Agent Systems: Challenges

- Non-stationary environment [Online learning]
- Conflicting interests [Game theory]
- Lack of coordination



Decision-Making in Multi-Agent Systems: Challenges

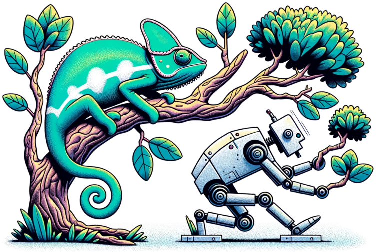
- Non-stationary environment [Online learning]
- Conflicting interests [Game theory]
- Lack of coordination
- Asynchronicity and delays

Decision-Making in Multi-Agent Systems: Challenges

- Non-stationary environment [Online learning]
- Conflicting interests [Game theory]
- Lack of coordination
- Asynchronicity and delays
- Uncertainty

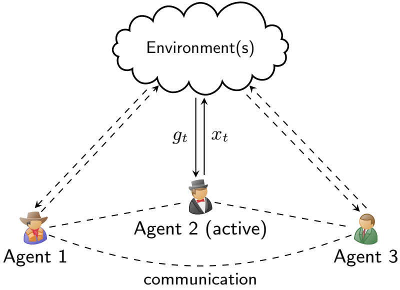
Decision-Making in Multi-Agent Systems: Challenges

- Non-stationary environment [Online learning]
- Conflicting interests [Game theory]
- Lack of coordination
- Asynchronicity and delays
- Uncertainty
- Need for adaptive methods

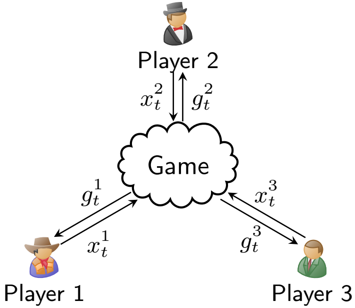


Plan

Part I: Learning in the Presence of Delays & Asynchronicities



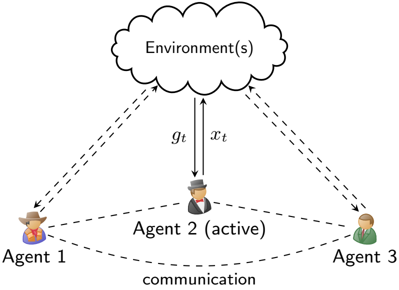
Part II: Adaptive Learning in Continuous Games with Noisy Feedback



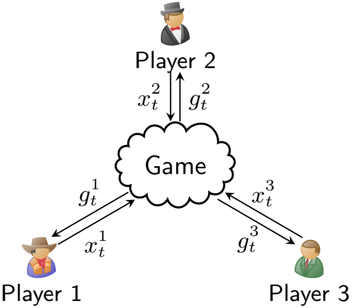
Common challenges: adaptive learning, lack of coordination, non-stationarity

Plan

Part I: Learning in the Presence of Delays & Asynchronicities



Part II: Adaptive Learning in Continuous Games with Noisy Feedback

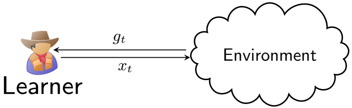


Common challenges: adaptive learning, lack of coordination, non-stationarity

Online Learning: A Framework For Sequential Decision Making

At each round $t = 1, 2, \ldots$, the learner

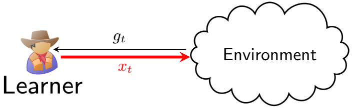
- Plays an action $x_t \in \mathcal{X}$
- Suffers loss $\ell_t(x_t)$ and receives feedback g_t



Online Learning: A Framework For Sequential Decision Making

At each round $t = 1, 2, \ldots,$ the learner

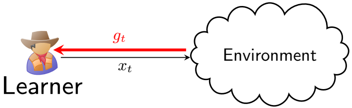
- Plays an action $x_t \in \mathcal{X}$
- Suffers loss $\ell_t(x_t)$ and receives feedback g_t



Online Learning: A Framework For Sequential Decision Making

At each round $t = 1, 2, \ldots,$ the learner

- Plays an action $x_t \in \mathcal{X}$
- Suffers loss $\ell_t(x_t)$ and receives feedback g_t



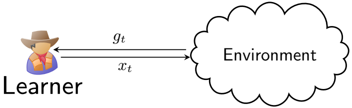
Online Learning: A Framework For Sequential Decision Making

At each round $t = 1, 2, \ldots$, the learner

- Plays an action $x_t \in \mathcal{X}$
- Suffers loss $\ell_t(x_t)$ and receives feedback g_t

- Regret of the learner with respect to $p \in \mathcal{X}$ is

$$\mathrm{Reg}_T(p) = \sum_{t=1}^{T} \Big(\underbrace{\ell_t(x_t) - \ell_t(p)}_{\text{cost of not playing } p \text{ in round } t} \Big)$$



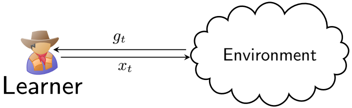
Online Learning: A Framework For Sequential Decision Making

At each round $t = 1, 2, \ldots$, the learner

- Plays an action $x_t \in \mathcal{X}$
- Suffers loss $\ell_t(x_t)$ and receives feedback g_t

- Regret of the learner with respect to $p \in \mathcal{X}$ is

$$\mathrm{Reg}_T(p) = \sum_{t=1}^{T} \Big(\underbrace{\ell_t(x_t) - \ell_t(p)}_{\text{cost of not playing } p \text{ in round } t} \Big)$$



- Online convex optimization: ℓ_t is convex with $\nabla \ell_t(x_t)$ a (sub)gradient

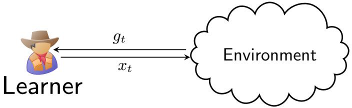
Online Learning: A Framework For Sequential Decision Making

At each round $t = 1, 2, \ldots$, the learner

- Plays an action $x_t \in \mathcal{X}$
- Suffers loss $\ell_t(x_t)$ and receives feedback g_t

- Regret of the learner with respect to $p \in \mathcal{X}$ is

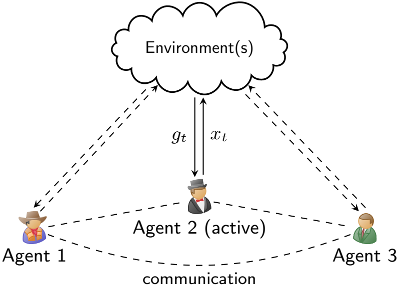
$$\mathrm{Reg}_T(p) = \sum_{t=1}^{T} \Big(\underbrace{\ell_t(x_t) - \ell_t(p)}_{\text{cost of not playing } p \text{ in round } t} \Big)$$



- Online convex optimization: ℓ_t is convex with $\nabla \ell_t(x_t)$ a (sub)gradient
- Online learning with first-order feedback: $g_t \approx \nabla \ell_t(x_t)$

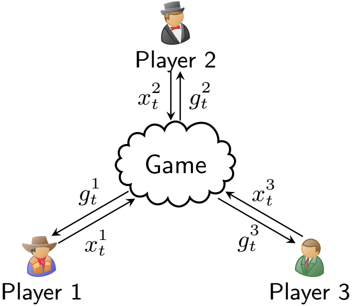
Source of Non-Stationarity

Part I: Learning in the Presence of Delays & Asynchronicities



The loss ℓ_t is given by the external environment

Part II: Adaptive Learning in Continuous Games with Noisy Feedback



The loss $\ell_t^i = \ell^i(\cdot, \mathbf{x}_t^{-i})$ comes from the interaction with other players

Part I: Learning in the Presence of Delays & Asynchronicities

Contributions for Part I

- A framework for asynchronous decentralized online learning
- Delayed dual averaging
- Template regret bound
- Adaptive learning rate with bounded delay assumption
- Adaptive learning rate without bounded delay assumption in single-agent setup
- Relation to distributed online learning
- Application to open network
- Optimistic variant

$\Big\}$ In this defense

---

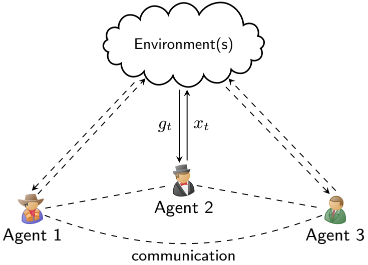
1. H., Iutzeler, Malick, and Mertikopoulos. *Multi-agent online optimization with delays: Asynchronicity, adaptivity, and optimism.* JMLR, 2022.

2. H., Iutzeler, Malick, and Mertikopoulos. *Optimization in Open Networks via Dual Averaging.* CDC, 2021.

A Framework for Asynchronous Decentralized Online Learning

At each round $t = 1, 2, \ldots$, an agent $i(t)$

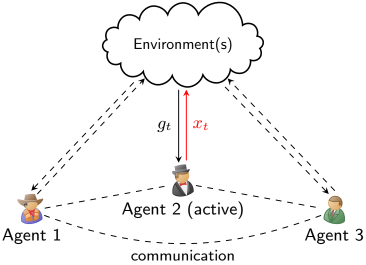
- Becomes active and plays an action $x_t \in \mathcal{X}$
- Suffers loss $\ell_t(x_t)$ and receives feedback $g_t = \nabla \ell_t(x_t)$
- Communicates with other agents



A Framework for Asynchronous Decentralized Online Learning

At each round $t = 1, 2, \ldots$, an agent $i(t)$

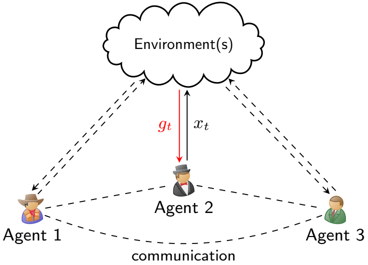
- Becomes active and plays an action $x_t \in \mathcal{X}$
- Suffers loss $\ell_t(x_t)$ and receives feedback $g_t = \nabla \ell_t(x_t)$
- Communicates with other agents



A Framework for Asynchronous Decentralized Online Learning

At each round $t = 1, 2, \ldots$, an agent $i(t)$

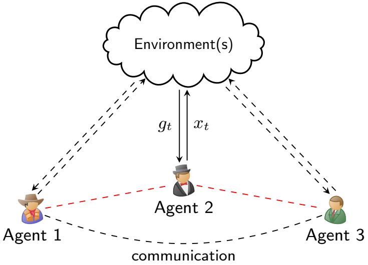
- Becomes active and plays an action $x_t \in \mathcal{X}$
- Suffers loss $\ell_t(x_t)$ and receives feedback $g_t = \nabla \ell_t(x_t)$
- Communicates with other agents



A Framework for Asynchronous Decentralized Online Learning

At each round $t = 1, 2, \ldots$, an agent $i(t)$

- Becomes active and plays an action $x_t \in \mathcal{X}$
- Suffers loss $\ell_t(x_t)$ and receives feedback $g_t = \nabla \ell_t(x_t)$
- Communicates with other agents



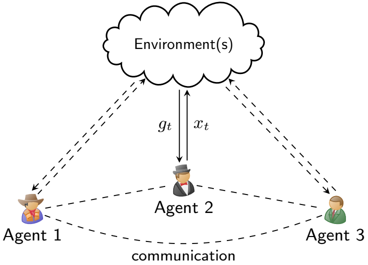
A Framework for Asynchronous Decentralized Online Learning

At each round $t = 1, 2, \ldots,$ an agent $i(t)$

- Becomes active and plays an action $x_t \in \mathcal{X}$
- Suffers loss $\ell_t(x_t)$ and receives feedback $g_t = \nabla \ell_t(x_t)$
- Communicates with other agents

- Regret of the system with respect to $p \in \mathcal{X}$ is

$$\mathrm{Reg}_T(p) = \sum_{t=1}^{T} \Big(\underbrace{\ell_t(x_t) - \ell_t(p)}_{\text{cost of not playing } p \text{ in round } t} \Big)$$



A Framework for Asynchronous Decentralized Online Learning

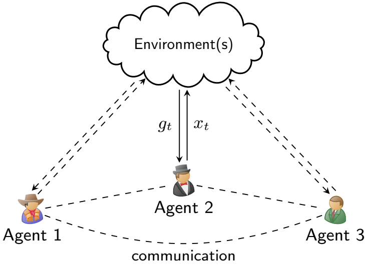
At each round $t = 1, 2, \ldots$, an agent $i(t)$

- Becomes active and plays an action $x_t \in \mathcal{X}$
- Suffers loss $\ell_t(x_t)$ and receives feedback $g_t = \nabla \ell_t(x_t)$
- Communicates with other agents

- Regret of the system with respect to $p \in \mathcal{X}$ is

$$\mathrm{Reg}_T(p) = \sum_{t=1}^{T} \Big(\underbrace{\ell_t(x_t) - \ell_t(p)}_{\text{cost of not playing } p \text{ in round } t} \Big)$$

- Communication: transmission of g_t



A Framework for Asynchronous Decentralized Online Learning

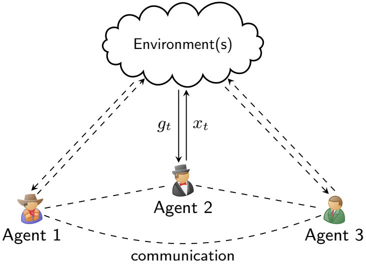
At each round $t = 1, 2, \ldots$, an agent $i(t)$

- Becomes active and plays an action $x_t \in \mathcal{X}$
- Suffers loss $\ell_t(x_t)$ and $\underline{\text{receives feedback } g_t = \nabla \ell_t(x_t)}$ Potentially with delay
- $\underline{\text{Communicates with other agents}}$ Asynchronous communication

- Regret of the system with respect to $p \in \mathcal{X}$ is

$$\mathrm{Reg}_T(p) = \sum_{t=1}^{T} \Big(\underbrace{\ell_t(x_t) - \ell_t(p)}_{\text{cost of not playing } p \text{ in round } t} \Big)$$

- Communication: transmission of g_t



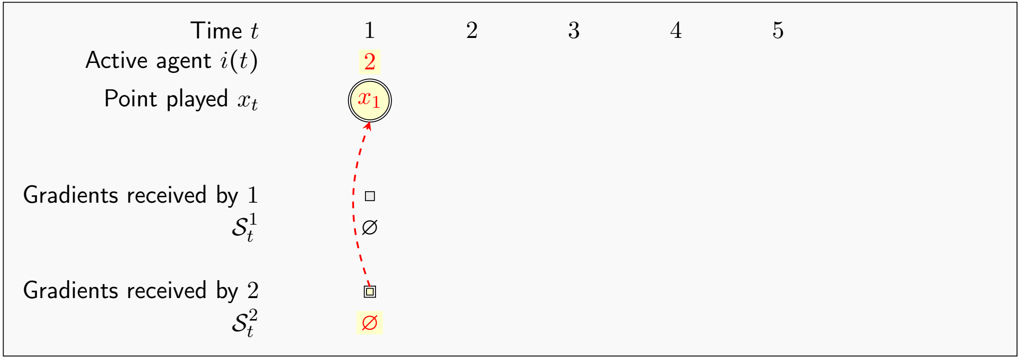
An Example With Two Agents

Time t	1	2	3	4	5
Active agent $i(t)$					
Point played x_t					
Gradients received by 1 $\mathcal{S}_t^1$					
Gradients received by 2 $\mathcal{S}_t^2$					

An Example With Two Agents

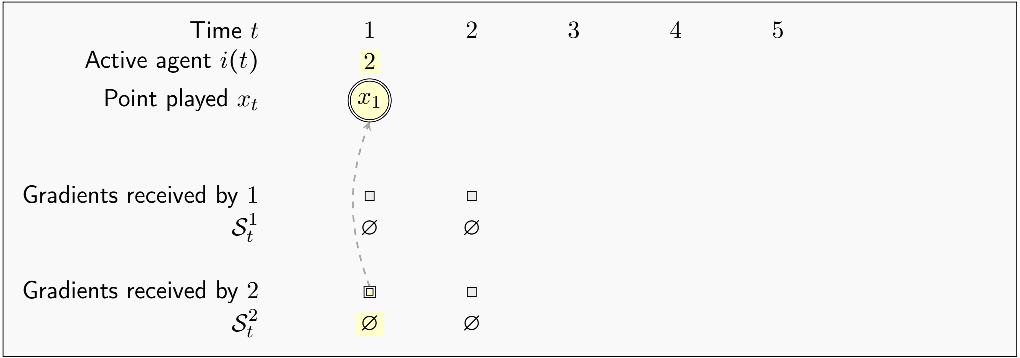
Time t	1	2	3	4	5
Active agent $i(t)$					
Point played x_t					
Gradients received by 1	$\square$				
$\mathcal{S}_t^1$	$\varnothing$				
Gradients received by 2	$\square$				
$\mathcal{S}_t^2$	$\varnothing$				

An Example With Two Agents



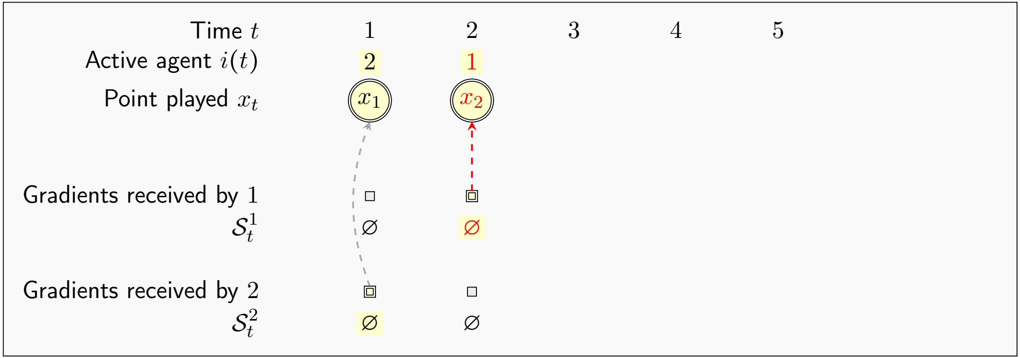
Time t	1	2	3	4	5
Active agent $i(t)$	2				
Point played x_t	x_1				
Gradients received by 1	□				
$\mathcal{S}_t^1$	$\varnothing$				
Gradients received by 2	▣				
$\mathcal{S}_t^2$	$\varnothing$				

An Example With Two Agents



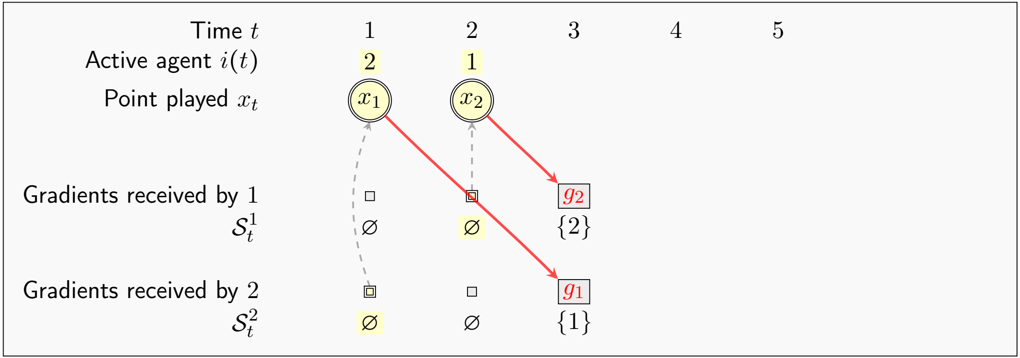
Time t	1	2	3	4	5
Active agent $i(t)$	2				
Point played x_t	x_1				
Gradients received by 1	□	□			
$\mathcal{S}_t^1$	$\varnothing$	$\varnothing$			
Gradients received by 2	▣	□			
$\mathcal{S}_t^2$	$\varnothing$	$\varnothing$			

An Example With Two Agents



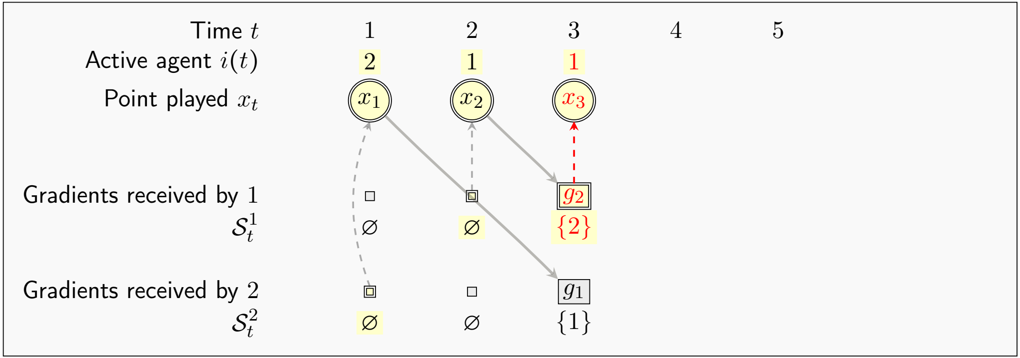
Time t	1	2	3	4	5
Active agent $i(t)$	2	1			
Point played x_t	x_1	x_2			
Gradients received by 1	□	▣			
$\mathcal{S}_t^1$	$\varnothing$	$\varnothing$			
Gradients received by 2	▣	□			
$\mathcal{S}_t^2$	$\varnothing$	$\varnothing$			

An Example With Two Agents



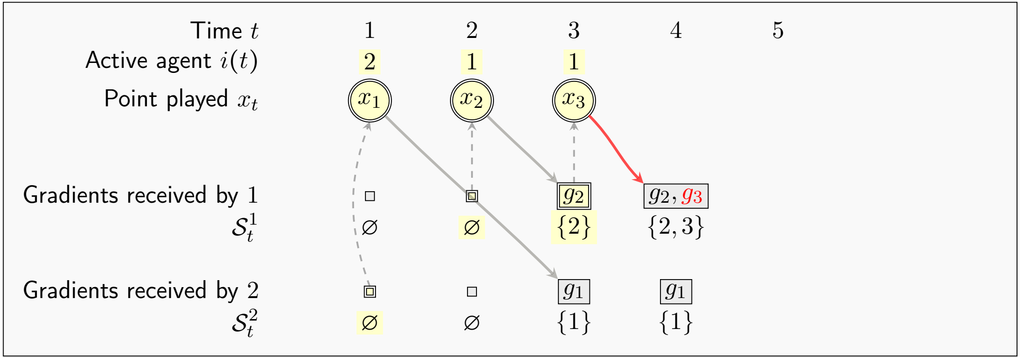
Time t	1	2	3	4	5
Active agent $i(t)$	2	1			
Point played x_t	x_1	x_2			
Gradients received by 1	□	□	g_2		
$\mathcal{S}_t^1$	$\varnothing$	$\varnothing$	$\{2\}$		
Gradients received by 2	□	□	g_1		
$\mathcal{S}_t^2$	$\varnothing$	$\varnothing$	$\{1\}$		

An Example With Two Agents



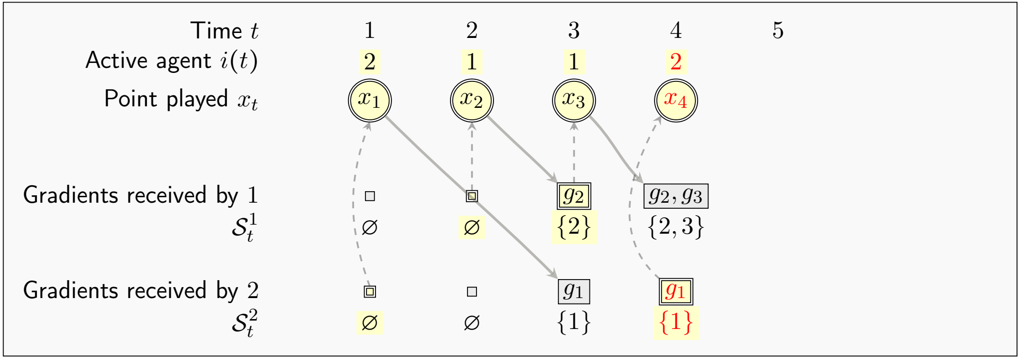
Time t	1	2	3	4	5
Active agent $i(t)$	2	1	1		
Point played x_t	x_1	x_2	x_3		
Gradients received by 1	☐	☐	g_2		
$\mathcal{S}_t^1$	$\varnothing$	$\varnothing$	$\{2\}$		
Gradients received by 2	☐	☐	g_1		
$\mathcal{S}_t^2$	$\varnothing$	$\varnothing$	$\{1\}$		

An Example With Two Agents



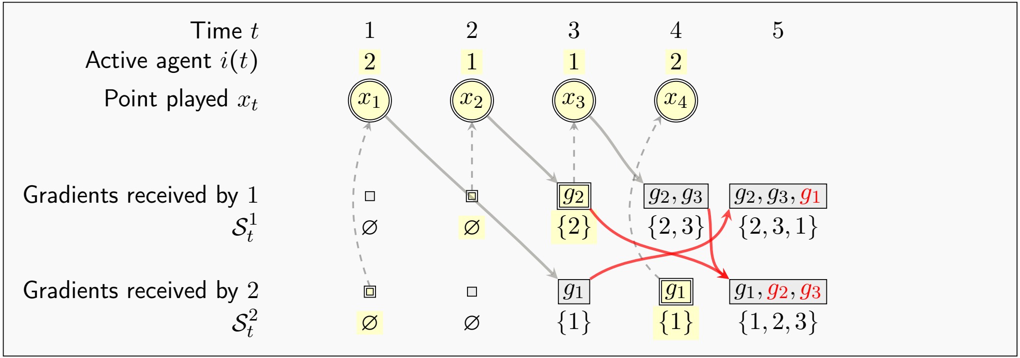
Time t	1	2	3	4	5
Active agent $i(t)$	2	1	1		
Point played x_t	x_1	x_2	x_3		
Gradients received by 1	☐	☐	g_2	g_2, g_3	
$\mathcal{S}_t^1$	$\varnothing$	$\varnothing$	$\{2\}$	$\{2,3\}$	
Gradients received by 2	☐	☐	g_1	g_1	
$\mathcal{S}_t^2$	$\varnothing$	$\varnothing$	$\{1\}$	$\{1\}$	

An Example With Two Agents



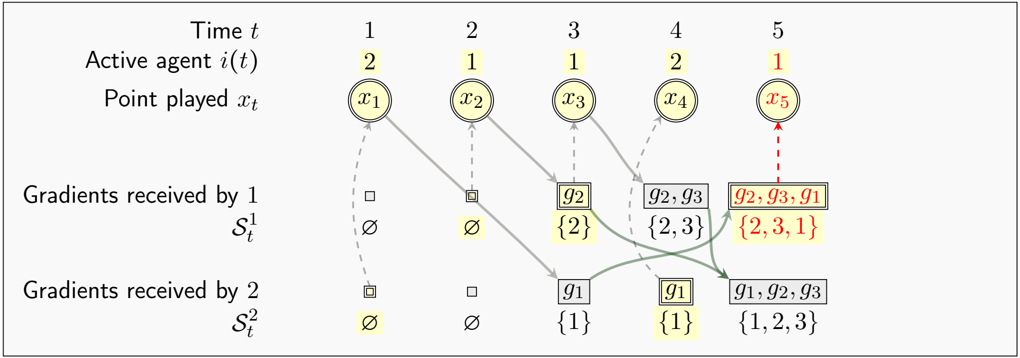
Time t	1	2	3	4	5
Active agent $i(t)$	2	1	1	2	
Point played x_t	x_1	x_2	x_3	x_4	
Gradients received by 1	☐	▣	g_2	g_2, g_3	
$\mathcal{S}_t^1$	$\varnothing$	$\varnothing$	$\{2\}$	$\{2, 3\}$	
Gradients received by 2	▣	☐	g_1	g_1	
$\mathcal{S}_t^2$	$\varnothing$	$\varnothing$	$\{1\}$	$\{1\}$	

An Example With Two Agents



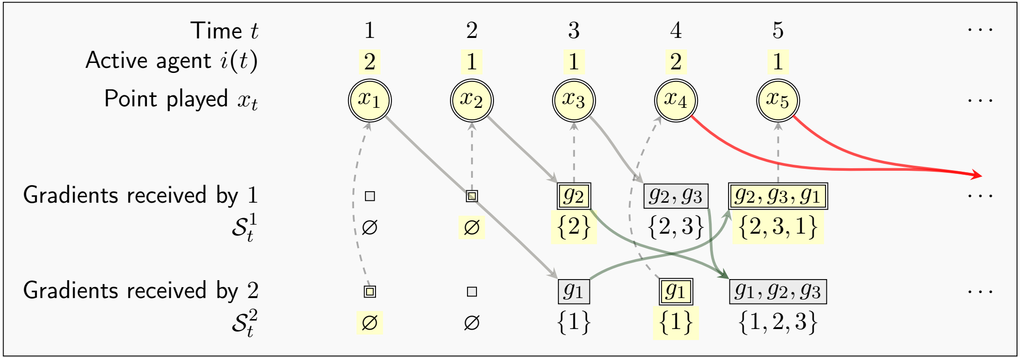
Time t	1	2	3	4	5
Active agent $i(t)$	2	1	1	2	
Point played x_t	x_1	x_2	x_3	x_4	
Gradients received by 1	□	▣	g_2	g_2, g_3	g_2, g_3, g_1
$\mathcal{S}_t^1$	$\varnothing$	$\varnothing$	$\{2\}$	$\{2, 3\}$	$\{2, 3, 1\}$
Gradients received by 2	▣	□	g_1	g_1	g_1, g_2, g_3
$\mathcal{S}_t^2$	$\varnothing$	$\varnothing$	$\{1\}$	$\{1\}$	$\{1, 2, 3\}$

An Example With Two Agents



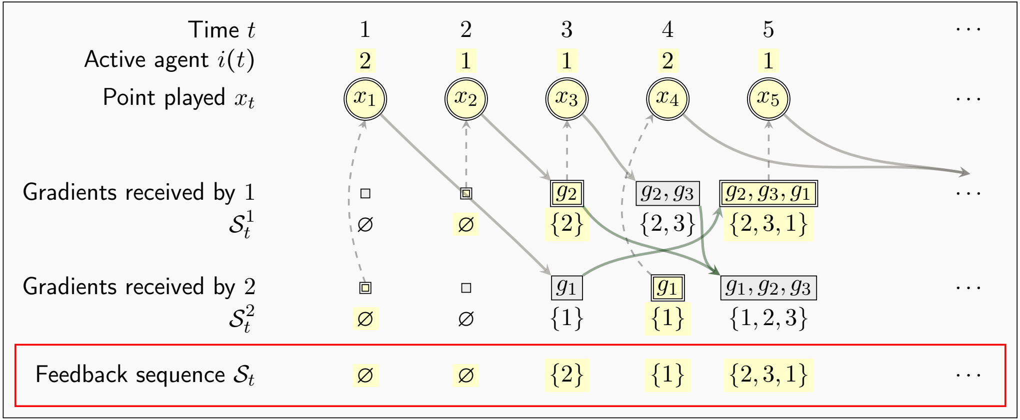
Time t	1	2	3	4	5
Active agent $i(t)$	2	1	1	2	1
Point played x_t	x_1	x_2	x_3	x_4	x_5
Gradients received by 1	□	▣	g_2	g_2, g_3	g_2, g_3, g_1
$\mathcal{S}_t^1$	$\varnothing$	$\varnothing$	$\{2\}$	$\{2, 3\}$	$\{2, 3, 1\}$
Gradients received by 2	▣	□	g_1	g_1	g_1, g_2, g_3
$\mathcal{S}_t^2$	$\varnothing$	$\varnothing$	$\{1\}$	$\{1\}$	$\{1, 2, 3\}$

An Example With Two Agents



Time t	1	2	3	4	5	$\cdots$
Active agent $i(t)$	2	1	1	2	1	
Point played x_t	x_1	x_2	x_3	x_4	x_5	$\cdots$
Gradients received by 1	□	▣	g_2	g_2, g_3	g_2, g_3, g_1	$\cdots$
$\mathcal{S}_t^1$	$\varnothing$	$\varnothing$	$\{2\}$	$\{2, 3\}$	$\{2, 3, 1\}$	
Gradients received by 2	▣	□	g_1	g_1	g_1, g_2, g_3	$\cdots$
$\mathcal{S}_t^2$	$\varnothing$	$\varnothing$	$\{1\}$	$\{1\}$	$\{1, 2, 3\}$	

Feedback Sequence



Time t	1	2	3	4	5	$\cdots$
Active agent $i(t)$	2	1	1	2	1	
Point played x_t	x_1	x_2	x_3	x_4	x_5	$\cdots$
Gradients received by 1	□	□	g_2	g_2, g_3	g_2, g_3, g_1	$\cdots$
$\mathcal{S}_t^1$	$\varnothing$	$\varnothing$	$\{2\}$	$\{2, 3\}$	$\{2, 3, 1\}$	
Gradients received by 2	□	□	g_1	g_1	g_1, g_2, g_3	$\cdots$
$\mathcal{S}_t^2$	$\varnothing$	$\varnothing$	$\{1\}$	$\{1\}$	$\{1, 2, 3\}$	
Feedback sequence $\mathcal{S}_t$	$\varnothing$	$\varnothing$	$\{2\}$	$\{1\}$	$\{2, 3, 1\}$	$\cdots$

Delayed Dual Averaging

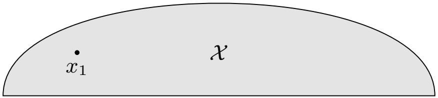
Dual averaging [Nesterov 09]

$$x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s=1}^{t-1} g_s\right)$$

Delayed dual averaging [H. et al. 22]

$$x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s\in\mathcal{S}_t} g_s\right)$$

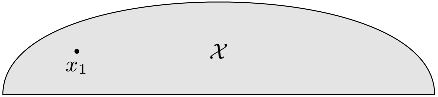
- $\{g_s : s \in \mathcal{S}_t\}$ are the gradients the active agent $i(t)$ can use to compute x_t
- $\Pi_{\mathcal{X}}$ is Euclidean projection onto the set $\mathcal{X}$



Delayed Dual Averaging

Dual averaging [Nesterov 09]	Delayed dual averaging [H. et al. 22]
$x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s=1}^{t-1} g_s\right)$	$x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s \in \mathcal{S}_t} g_s\right)$

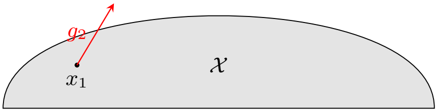
- $\{g_s : s \in \mathcal{S}_t\}$ are the gradients the active agent $i(t)$ can use to compute x_t
- $\Pi_{\mathcal{X}}$ is Euclidean projection onto the set $\mathcal{X}$
- Example: $\mathcal{S}_5 = \{2, 3, 1\}$



Delayed Dual Averaging

Dual averaging [Nesterov 09]	Delayed dual averaging [H. et al. 22]
$$x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s=1}^{t-1} g_s\right)$$	$$x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s \in \mathcal{S}_t} g_s\right)$$

- $\{g_s : s \in \mathcal{S}_t\}$ are the gradients the active agent $i(t)$ can use to compute x_t
- $\Pi_{\mathcal{X}}$ is Euclidean projection onto the set $\mathcal{X}$
- Example: $\mathcal{S}_5 = \{2, 3, 1\}$



Delayed Dual Averaging

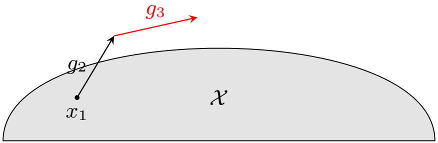
Dual averaging [Nesterov 09]

$$x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s=1}^{t-1} g_s\right)$$

Delayed dual averaging [H. et al. 22]

$$x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s\in\mathcal{S}_t} g_s\right)$$

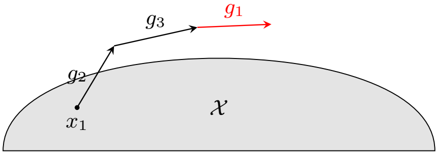
- $\{g_s : s \in \mathcal{S}_t\}$ are the gradients the active agent $i(t)$ can use to compute x_t
- $\Pi_{\mathcal{X}}$ is Euclidean projection onto the set $\mathcal{X}$
- Example: $\mathcal{S}_5 = \{2, 3, 1\}$



Delayed Dual Averaging

Dual averaging [Nesterov 09]	Delayed dual averaging [H. et al. 22]
$x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s=1}^{t-1} g_s\right)$	$x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s \in \mathcal{S}_t} g_s\right)$

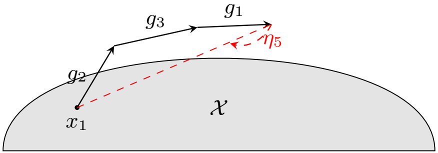
- $\{g_s : s \in \mathcal{S}_t\}$ are the gradients the active agent $i(t)$ can use to compute x_t
- $\Pi_{\mathcal{X}}$ is Euclidean projection onto the set $\mathcal{X}$
- Example: $\mathcal{S}_5 = \{2, 3, 1\}$



Delayed Dual Averaging

Dual averaging [Nesterov 09]	Delayed dual averaging [H. et al. 22]
$x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s=1}^{t-1} g_s\right)$	$x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s \in \mathcal{S}_t} g_s\right)$

- $\{g_s : s \in \mathcal{S}_t\}$ are the gradients the active agent $i(t)$ can use to compute x_t
- $\Pi_{\mathcal{X}}$ is Euclidean projection onto the set $\mathcal{X}$
- Example: $\mathcal{S}_5 = \{2, 3, 1\}$



Delayed Dual Averaging

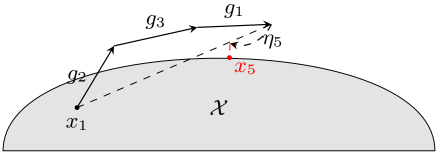
Dual averaging [Nesterov 09]

$$x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s=1}^{t-1} g_s\right)$$

Delayed dual averaging [H. et al. 22]

$$x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s \in \mathcal{S}_t} g_s\right)$$

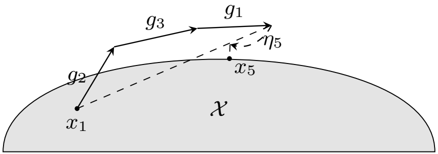
- $\{g_s : s \in \mathcal{S}_t\}$ are the gradients the active agent $i(t)$ can use to compute x_t
- $\Pi_{\mathcal{X}}$ is Euclidean projection onto the set $\mathcal{X}$
- Example: $\mathcal{S}_5 = \{2, 3, 1\}$



Delayed Dual Averaging

Dual averaging [Nesterov 09]	Delayed dual averaging [H. et al. 22]
$x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s=1}^{t-1} g_s\right)$	$x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s \in \mathcal{S}_t} g_s\right)$

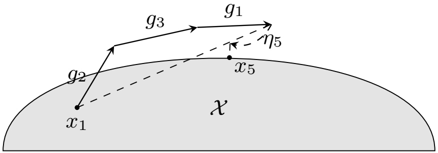
- $\{g_s : s \in \mathcal{S}_t\}$ are the gradients the active agent $i(t)$ can use to compute x_t
- $\Pi_{\mathcal{X}}$ is Euclidean projection onto the set $\mathcal{X}$
- Example: $\mathcal{S}_5 = \{2, 3, 1\}$
- All the gradients have the same weight



Delayed Dual Averaging

Dual averaging [Nesterov 09]	Delayed dual averaging [H. et al. 22]
$x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s=1}^{t-1} g_s\right)$	$x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s \in \mathcal{S}_t} g_s\right)$

- $\{g_s : s \in \mathcal{S}_t\}$ are the gradients the active agent $i(t)$ can use to compute x_t
- $\Pi_{\mathcal{X}}$ is Euclidean projection onto the set $\mathcal{X}$
- Example: $\mathcal{S}_5 = \{2, 3, 1\}$
- All the gradients have the same weight
- Issue: learning rate η_t needs to be non-increasing



Dependency Graph

Key observation: only $\mathcal{S}_t$ counts for the algorithm

- Dependency graph $\mathcal{G}$: Each vertex is a timestamp, and we put a directed edge from s to t if and only if $s \in \mathcal{S}_t$
- Example: $\mathcal{S}_1 = \mathcal{S}_2 = \varnothing$; $\mathcal{S}_3 = \{2\}$; $\mathcal{S}_4 = \{1\}$; $\mathcal{S}_5 = \{2, 3, 1\}$

Dependency Graph

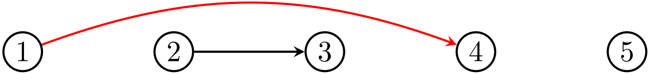
Key observation: only $\mathcal{S}_t$ counts for the algorithm

- Dependency graph $\mathcal{G}$: Each vertex is a timestamp, and we put a directed edge from s to t if and only if $s \in \mathcal{S}_t$
- Example: $\mathcal{S}_1 = \mathcal{S}_2 = \varnothing$; $\mathcal{S}_3 = \{2\}$; $\mathcal{S}_4 = \{1\}$; $\mathcal{S}_5 = \{2, 3, 1\}$

Dependency Graph

Key observation: only $\mathcal{S}_t$ counts for the algorithm

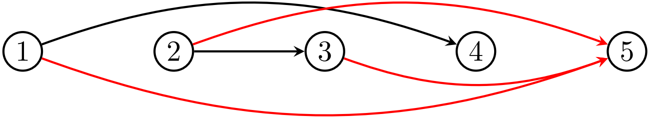
- Dependency graph $\mathcal{G}$: Each vertex is a timestamp, and we put a directed edge from s to t if and only if $s \in \mathcal{S}_t$
- Example: $\mathcal{S}_1 = \mathcal{S}_2 = \varnothing$; $\mathcal{S}_3 = \{2\}$; $\mathcal{S}_4 = \{1\}$; $\mathcal{S}_5 = \{2, 3, 1\}$



Dependency Graph

Key observation: only $\mathcal{S}_t$ counts for the algorithm

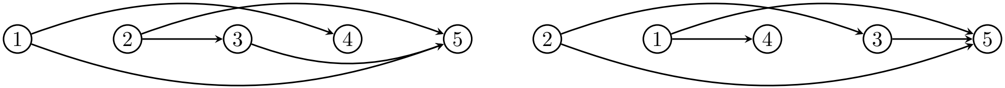
- Dependency graph $\mathcal{G}$: Each vertex is a timestamp, and we put a directed edge from s to t if and only if $s \in \mathcal{S}_t$
- Example: $\mathcal{S}_1 = \mathcal{S}_2 = \varnothing$; $\mathcal{S}_3 = \{2\}$; $\mathcal{S}_4 = \{1\}$; $\mathcal{S}_5 = \{2, 3, 1\}$



Faithful Permutation

Key observation: only $\mathcal{S}_t$ counts for the algorithm

- Faithful permutation: A permutation π of $\{1, 2, \ldots, T\}$ is *faithful* if and only if $\pi(1), \ldots, \pi(T)$ is a topological ordering of $\mathcal{G}$
- Example: $\{1, 2, 3, 4, 5\}$ and $\{2, 1, 4, 3, 5\}$ are faithful for $\mathcal{S}_1 = \mathcal{S}_2 = \varnothing$; $\mathcal{S}_3 = \{2\}$; $\mathcal{S}_4 = \{1\}$; $\mathcal{S}_5 = \{2, 3, 1\}$



Template Regret Bound

Theorem [H. et al. 22]

Let π be a faithful permutation of $\{1,\dots,T\}$, and assume that delayed dual averaging is run with η_t satisfying that $\boxed{\eta_{\pi(t+1)} \leq \eta_{\pi(t)}}$ for all t. Then,

$$\mathrm{Reg}_T(p) \leq \frac{\|x_1 - p\|^2}{2\eta_{\pi(T)}} + \frac{1}{2}\sum_{t=1}^{T} \eta_{\pi(t)} \left(\|g_{\pi(t)}\|^2 + 2\|g_{\pi(t)}\| \sum_{s\in\mathcal{U}_t^\pi} \|g_s\| \right).$$

From undelayed dual averaging

Induced by delays

Here $\mathcal{U}_t^\pi = \{\pi(1),\dots,\pi(t)\} \setminus \mathcal{S}_{\pi(t)}$

Template Regret Bound

Theorem [H. et al. 22]

Let π be a faithful permutation of $\{1, \ldots, T\}$, and assume that delayed dual averaging is run with η_t satisfying that $\boxed{\eta_{\pi(t+1)} \leq \eta_{\pi(t)}}$ for all t. Then,

$$\operatorname{Reg}_T(p) \leq \frac{\|x_1 - p\|^2}{2\eta_{\pi(T)}} + \frac{1}{2} \sum_{t=1}^{T} \eta_{\pi(t)} \left(\left\|g_{\pi(t)}\right\|^2 + 2\left\|g_{\pi(t)}\right\| \sum_{s \in \mathcal{U}_t^\pi} \|g_s\| \right).$$

From undelayed dual averaging

Induced by delays

Here $\mathcal{U}_t^\pi = \{\pi(1), \ldots, \pi(t)\} \setminus \mathcal{S}_{\pi(t)}$

Q1: What is the optimal regret bound?

Template Regret Bound

Theorem [H. et al. 22]

Let π be a faithful permutation of $\{1, \ldots, T\}$, and assume that delayed dual averaging is run with η_t satisfying that $\boxed{\eta_{\pi(t+1)} \leq \eta_{\pi(t)}}$ for all t. Then,

$$\mathrm{Reg}_T(p) \leq \frac{\|x_1 - p\|^2}{2\eta_{\pi(T)}} + \frac{1}{2}\sum_{t=1}^{T} \eta_{\pi(t)} \left(\|g_{\pi(t)}\|^2 + 2\|g_{\pi(t)}\| \sum_{s \in \mathcal{U}_t^{\pi}} \|g_s\| \right).$$

From undelayed dual averaging

Induced by delays

Here $\mathcal{U}_t^{\pi} = \{\pi(1), \ldots, \pi(t)\} \setminus \mathcal{S}_{\pi(t)}$

Q1: What is the optimal regret bound? **Q2:** How to interpret the additional terms?

Lag and Ideal Regret Bound

The lag with respect to π up to time t is

$$\Lambda_t^\pi = \sum_{s=1}^{t} \left(\|g_{\pi(s)}\|^2 + 2\|g_{\pi(s)}\| \sum_{l \in \mathcal{U}_s^\pi} \|g_l\| \right).$$

Corollary

Let π be a faithful permutation of $\{1, \dots, T\}$, and assume that delayed dual averaging is run with $\eta_{\pi(t)} = 1/\sqrt{\Lambda_T^\pi}$ or $\eta_{\pi(t)} = 1/\sqrt{\Lambda_t^\pi}$, then the regret is

$$\mathrm{Reg}_T(p) = \mathcal{O}(\sqrt{\Lambda_T^\pi})$$

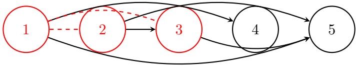
Interpretation of Lag

The lag with respect to π up to time t is

$$\Lambda_t^\pi = \sum_{s=1}^{t} \left(\|g_{\pi(s)}\|^2 + 2\|g_{\pi(s)}\| \sum_{l \in \mathcal{U}_s^\pi} \|g_l\| \right).$$

Proposition

The term $\sum_{s=1}^{t} 2\|g_{\pi(s)}\| \sum_{l \in \mathcal{U}_s^\pi} \|g_l\|$ contains the pairs of non-adjacent vertices in $\mathcal{G}_{\upharpoonright\{\pi(1),\dots,\pi(t)\}}$.



Consequences:
- $\Lambda_T^\pi = \Lambda_T^{\mathrm{id}}$
- Lag is both data- and delay-dependent

Regret Bound in the Case of Bounded Delay

The lag with respect to π up to time t is

$$\Lambda_t^\pi = \sum_{s=1}^{t} \left(\left\| g_{\pi(s)} \right\|^2 + 2 \underbrace{\left\| g_{\pi(s)} \right\| \sum_{l \in \mathcal{U}_s^\pi} \left\| g_l \right\|}_{\text{Pairs of non-adjacent vertices in } \mathcal{G}_{\upharpoonright \{\pi(1),\ldots,\pi(t)\}}} \right).$$

- If $\|g_t\| \le G$ and delay is bounded by τ, then $\Lambda_t^{\mathrm{id}} \le (2\tau + 1)tG^2$
- Setting $\eta_t = 1/\sqrt{\tau t}$ gives $\mathcal{O}(\sqrt{\tau T})$ regret

  Similar result in [Weinberger and Ordentlich 02, Langford et al. 09] for constant delay τ

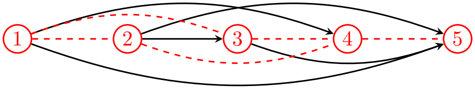
Non-Implementability of the Algorithms

The lag with respect to π up to time t is

$$\Lambda_t^\pi = \sum_{s=1}^{t} \left(\left\| g_{\pi(s)} \right\|^2 + 2 \underbrace{\left\| g_{\pi(s)} \right\| \sum_{l \in \mathcal{U}_s^\pi} \left\| g_l \right\|}_{\text{Pairs of non-adjacent vertices in } \mathcal{G}_{\restriction \{\pi(1),\dots,\pi(t)\}}} \right).$$

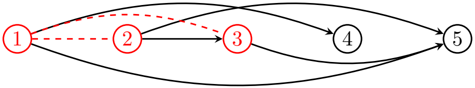
- $\eta_{\pi(t)} = 1/\sqrt{\Lambda_t^\pi}$: Λ_t^π cannot be computed at time $\pi(t)$
- $\eta_t = 1/\sqrt{\tau t}$: Even τ and t might be unknown

Adaptive Learning Rate



Approximate

$$\Lambda_t^\pi = \sum_{s=1}^{t} \|g_{\pi(s)}\|^2 + \sum_{\substack{s,l \le t,\\ \pi(s) \not- \pi(l)}} 2\|g_{\pi(s)}\|\|g_{\pi(l)}\|$$

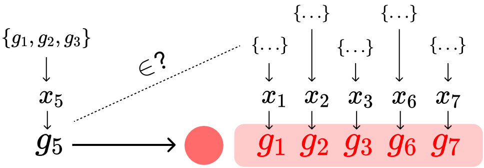


By

$$\sum_{s \in \mathcal{S}_{\pi(t)}} \|g_s\|^2 + \sum_{\substack{s,l \in \mathcal{S}_{\pi(t)}\\ s \not- l}} 2\|g_s\|\|g_l\|$$

- $\mathcal{S}_5 = \{2, 3, 1\}$
- $\not-$ indicates "non-adjacent in the dependency graph"

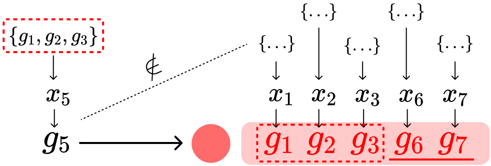
Adaptive Learning Rate: Issues



Two issues:

1. Naive implementation of the algorithm requires to identify each gradient, unbounded memory, and high time complexity.
2. Is the induced learning rate non-increasing along some faithful permutation?

Adaptive Learning Rate: Assumption



Assumption: When an agent receives g_t, it must have received $\{g_s : s \in \mathcal{S}_t\}$

Satisfied if all the gradients are transmitted in order

Adaptive Learning Rate: Pseudo-Code

Algorithm AdaDelay-Dist – from the point of view of agent i

```
**Initialize:** $\mathcal{G}_i \leftarrow \varnothing$, $\Gamma^i \leftarrow \beta > 0$, $R > 0$
**while** not stopped **do**
    **asynchronously** receive $g_t$ (along with $\sum_{s \in \mathcal{S}_t} \|g_s\|$ if sent by other agents)
        $\Gamma^i \leftarrow \Gamma^i + \|g_t\|^2 + 2\|g_t\|(\sum_{s \in \mathcal{G}^i} \|g_s\| - \sum_{s \in \mathcal{S}_t} \|g_s\|)$
        $\mathcal{G}^i \leftarrow \mathcal{G}^i \cup \{g_t\}$
    **if** the agent becomes active, i.e., $i(t) = i$ **then**
        $\mathcal{S}_t \leftarrow \mathcal{G}_i$
        $\eta_t \leftarrow R/\sqrt{\Gamma^i}$
        Play $x_t = \Pi_{\mathcal{X}}\left(x_1 - \eta_t \sum_{s \in \mathcal{S}_t} g_s\right)$
    **end if**
**end while**
```

Regret Bound for AdaDelay-Dist

Theorem [H. et al. 22]

Assume that

1. For all t, $\|g_t\| \leq G$
2. Delays are bounded by τ (possibly unknown)
3. When an agent receives g_t, they have already received $\{g_s : s \in \mathcal{S}_t\}$

Then, if $\|x_1 - p\|^2 \leq 2R^2$, the algorithm AdaDelay-Dist enjoys the regret bound

$$\mathrm{Reg}_T(p) \leq \underbrace{2R\sqrt{\Lambda_T}}_{\text{Lag: data- and delay-dependent}} + \underbrace{2R\sqrt{\beta} + \frac{R}{\sqrt{\beta}}G^2(2\tau+1)^2}_{\text{price of adaptivity}}$$

Regret Bound for AdaDelay-Dist

Theorem [H. et al. 22]

Assume that

1. For all t, $\|g_t\| \leq G$
2. Delays are bounded by τ (possibly unknown)
3. When an agent receives g_t, they have already received $\{g_s : s \in \mathcal{S}_t\}$

Then, if $\|x_1 - p\|^2 \leq 2R^2$, the algorithm AdaDelay-Dist enjoys the regret bound

$$\mathrm{Reg}_T(p) \leq \underbrace{2R\sqrt{\Lambda_T}}_{\text{Lag: data- and delay-dependent}} + \underbrace{2R\sqrt{\beta} + \frac{R}{\sqrt{\beta}} G^2 (2\tau + 1)^2}_{\text{price of adaptivity}}$$

Regret Bound for AdaDelay-Dist

Theorem [H. et al. 22]

Assume that

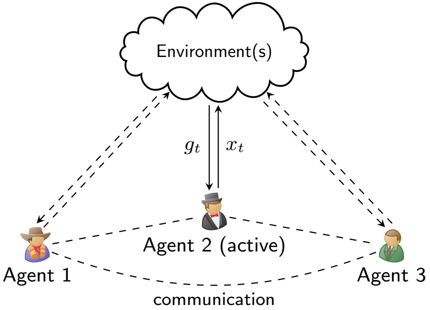
1. For all t, $\|g_t\| \le G$
2. Delays are bounded by τ (possibly unknown)
3. When an agent receives g_t, they have already received $\{g_s : s \in \mathcal{S}_t\}$

Then, if $\|x_1 - p\|^2 \le 2R^2$, the algorithm AdaDelay-Dist enjoys the regret bound

$$\mathrm{Reg}_T(p) \le \underbrace{2R\sqrt{\Lambda_T}}_{\text{Lag: data- and delay-dependent}} + \underbrace{2R\sqrt{\beta} + \frac{R}{\sqrt{\beta}} G^2 (2\tau + 1)^2}_{\text{price of adaptivity}}$$

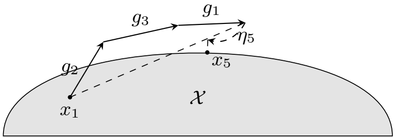
What We Have Seen in This Part

- **A framework for decentralized online learning**
- Simple algorithm template with data- and delay-adaptive learning rate
- Examined Challenges
  - Asynchronicity and delays
  - Non-stationarity
  - Lack of coordination
  - Adaptive learning



What We Have Seen in This Part

- A framework for decentralized online learning
- **Simple algorithm template with data- and delay-adaptive learning rate**
- Examined Challenges
  - Asynchronicity and delays
  - Non-stationarity
  - Lack of coordination
  - Adaptive learning



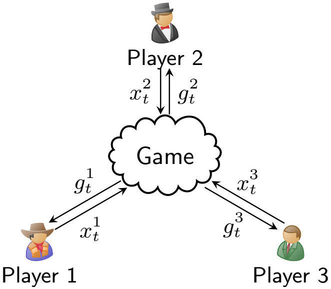
Part II: Adaptive Learning in Continuous Games

Learning in Continuous Games With Gradient Feedback

At each round $t = 1, 2, \ldots$, *each* player $i \in \mathcal{N} \coloneqq \{1, \ldots, N\}$

- Plays an action $x_t^i \in \mathcal{X}^i$
- Suffers loss $\ell^i(\mathbf{x}_t)$ and receives as feedback $g_t^i \approx \nabla_i \, \ell^i(\mathbf{x}_t)$

- Each player i has a convex closed action set $\mathcal{X}^i$ and a loss function $\ell^i \colon \mathcal{X}^1 \times \ldots \times \mathcal{X}^N \to \mathbb{R}$
- Joint action of all players $\mathbf{x} = (x^i)_{i \in \mathcal{N}} = (x^i, \mathbf{x}^{-i})$

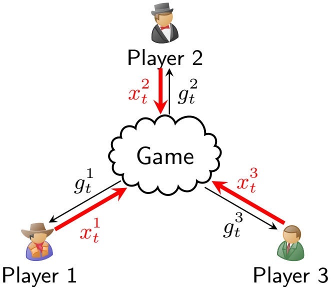


Learning in Continuous Games With Gradient Feedback

At each round $t = 1, 2, \ldots$, *each* player $i \in \mathcal{N} \coloneqq \{1, \ldots, N\}$

- Plays an action $x_t^i \in \mathcal{X}^i$
- Suffers loss $\ell^i(\mathbf{x}_t)$ and receives as feedback $g_t^i \approx \nabla_i \ell^i(\mathbf{x}_t)$

- Each player i has a convex closed action set $\mathcal{X}^i$ and a loss function $\ell^i : \mathcal{X}^1 \times \ldots \times \mathcal{X}^N \to \mathbb{R}$
- Joint action of all players $\mathbf{x} = (x^i)_{i \in \mathcal{N}} = (x^i, \mathbf{x}^{-i})$

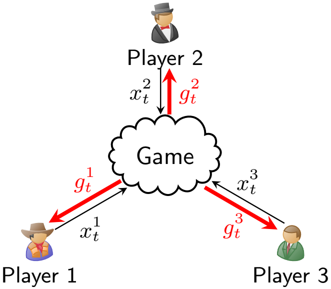


Learning in Continuous Games With Gradient Feedback

At each round $t = 1, 2, \ldots$, *each* player $i \in \mathcal{N} \coloneqq \{1, \ldots, N\}$

- Plays an action $x_t^i \in \mathcal{X}^i$
- Suffers loss $\ell^i(\mathbf{x}_t)$ and receives as feedback $g_t^i \approx \nabla_i \ell^i(\mathbf{x}_t)$

- Each player i has a convex closed action set $\mathcal{X}^i$ and a loss function $\ell^i : \mathcal{X}^1 \times \ldots \times \mathcal{X}^N \to \mathbb{R}$
- Joint action of all players $\mathbf{x} = (x^i)_{i \in \mathcal{N}} = (x^i, \mathbf{x}^{-i})$

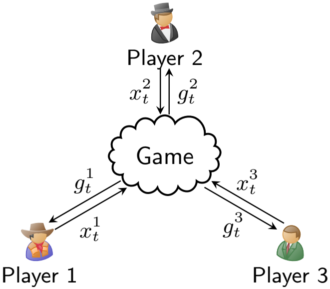


Learning in Continuous Games With Gradient Feedback

At each round $t = 1, 2, \ldots$, *each* player $i \in \mathcal{N} := \{1, \ldots, N\}$

- Plays an action $x_t^i \in \mathcal{X}^i$
- Suffers loss $\ell^i(\mathbf{x}_t)$ and receives as feedback $g_t^i \approx \nabla_i \ell^i(\mathbf{x}_t)$

- Each player i has a convex closed action set $\mathcal{X}^i$ and a loss function $\ell^i \colon \mathcal{X}^1 \times \ldots \times \mathcal{X}^N \to \mathbb{R}$
- Joint action of all players $\mathbf{x} = (x^i)_{i \in \mathcal{N}} = (x^i, \mathbf{x}^{-i})$
- $\ell^i(\cdot, \mathbf{x}^{-i})$ is convex and $\nabla_i \ell^i(\mathbf{x}_t)$ is Lipschitz continuous



Evaluating Learning-in-Games Algorithms

Two interaction scenarios

- Adversarial: the actions of the other players are arbitrary
- Self-play: all the players use the same algorithm

Two evaluation criteria

- Regret of player i with respect to $p^i \in \mathcal{X}^i$ is

$$\mathrm{Reg}_T^i(p^i) = \sum_{t=1}^{T} \Big(\underbrace{\ell^i(x_t^i, \mathbf{x}_t^{-i}) - \ell^i(p^i, \mathbf{x}_t^{-i})}_{\text{cost of not playing } p^i \text{ in round } t} \Big) \qquad [\text{i.e. } \ell_t = \ell^i(\cdot, \mathbf{x}_t^{-i})]$$

- Whether the sequence of play $\mathbf{x}_t$ converges to a Nash equilibrium $\mathbf{x}_\star$

Evaluating Learning-in-Games Algorithms

Two interaction scenarios

- **Adversarial: the actions of the other players are arbitrary**
- Self-play: all the players use the same algorithm

Two evaluation criteria

- **Regret of player i with respect to $p^i \in \mathcal{X}^i$ is**

$$\mathrm{Reg}_T^i(p^i) = \sum_{t=1}^{T} \Big(\underbrace{\ell^i(x_t^i, \mathbf{x}_t^{-i}) - \ell^i(p^i, \mathbf{x}_t^{-i})}_{\textbf{cost of not playing } p^i \textbf{ in round } t} \Big) \qquad [\textbf{i.e. } \ell_t = \ell^i(\cdot, \mathbf{x}_t^{-i})]$$

- Whether the sequence of play $\mathbf{x}_t$ converges to a Nash equilibrium $\mathbf{x}_\star$

Evaluating Learning-in-Games Algorithms

Two interaction scenarios

- Adversarial: the actions of the other players are arbitrary
- Self-play: **all the players use the same algorithm**

Two evaluation criteria

- Regret of player i with respect to $p^i \in \mathcal{X}^i$ is

$$\mathrm{Reg}_T^i(p^i) = \sum_{t=1}^{T} \Big(\underbrace{\ell^i(x_t^i, \mathbf{x}_t^{-i}) - \ell^i(p^i, \mathbf{x}_t^{-i})}_{\text{cost of not playing } p^i \text{ in round } t} \Big) \qquad [\text{i.e. } \ell_t = \ell^i(\cdot, \mathbf{x}_t^{-i})]$$

- **Whether the sequence of play $\mathbf{x}_t$ converges to a Nash equilibrium $\mathbf{x}_\star$**

Variational Stability for Convergence to Nash Equilibrium

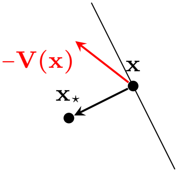
A continuous convex game is variationally stable (VS) if the set $\mathcal{X}_\star$ of Nash equilibria of the game is nonempty and

$$\langle \mathbf{V}(\mathbf{x}), \mathbf{x} - \mathbf{x}_\star \rangle = \sum_{i=1}^{N} \langle \nabla_i \ell^i(\mathbf{x}), x^i - x^i_\star \rangle \geq 0 \quad \text{for all } \mathbf{x} \in \mathcal{X},\ \mathbf{x}_\star \in \mathcal{X}_\star$$

- **Finding Nash Equilibrium is hard [Daskalakis et al. 08]**
- Game vector field / Psudeo-gradient

$$\mathbf{V} = (\nabla_1 \ell^1, \ldots, \nabla_N \ell^N)$$

- $\mathbf{V}$ monotone ⇒ VS satisfied



Variational Stability for Convergence to Nash Equilibrium

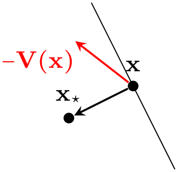
A continuous convex game is variationally stable (VS) if the set $\mathcal{X}_\star$ of Nash equilibria of the game is nonempty and

$$\langle \mathbf{V}(\mathbf{x}), \mathbf{x} - \mathbf{x}_\star \rangle = \sum_{i=1}^{N} \langle \nabla_i \ell^i(\mathbf{x}), x^i - x^i_\star \rangle \geq 0 \quad \text{for all } \mathbf{x} \in \mathcal{X},\ \mathbf{x}_\star \in \mathcal{X}_\star$$

- Finding Nash Equilibrium is hard [Daskalakis et al. 08]
- **Game vector field / Psudeo-gradient**

$$\mathbf{V} = (\nabla_1 \ell^1, \dots, \nabla_N \ell^N)$$

- $\mathbf{V}$ monotone $\Rightarrow$ VS satisfied



Variational Stability for Convergence to Nash Equilibrium

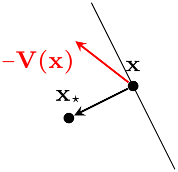
A continuous convex game is variationally stable (VS) if the set $\mathcal{X}_\star$ of Nash equilibria of the game is nonempty and

$$\langle \mathbf{V}(\mathbf{x}), \mathbf{x} - \mathbf{x}_\star \rangle = \sum_{i=1}^{N} \langle \nabla_i \ell^i(\mathbf{x}), x^i - x^i_\star \rangle \geq 0 \quad \text{for all } \mathbf{x} \in \mathcal{X},\ \mathbf{x}_\star \in \mathcal{X}_\star$$

- Finding Nash Equilibrium is hard [Daskalakis et al. 08]
- Game vector field / Psudeo-gradient

$$\mathbf{V} = (\nabla_1 \ell^1, \ldots, \nabla_N \ell^N)$$

- $\mathbf{V}$ monotone $\Rightarrow$ VS satisfied

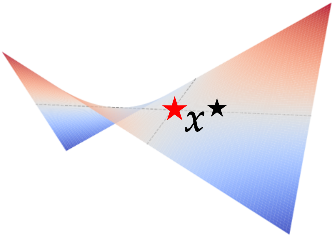


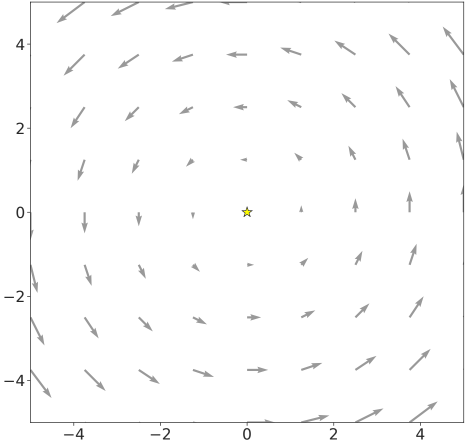
Failure of the Vanilla Gradient Method in Bilinear Games

- Two-player planar bilinear zero-sum game

$$\ell^1(\mathbf{x}) = -\ell^2(\mathbf{x}) = \theta\phi \quad [\mathbf{x} = (\theta, \phi) \in \mathbb{R}^2]$$

Unique Nash equilibrium: $(0,0)$





Failure of the Vanilla Gradient Method in Bilinear Games

- Two-player planar bilinear zero-sum game

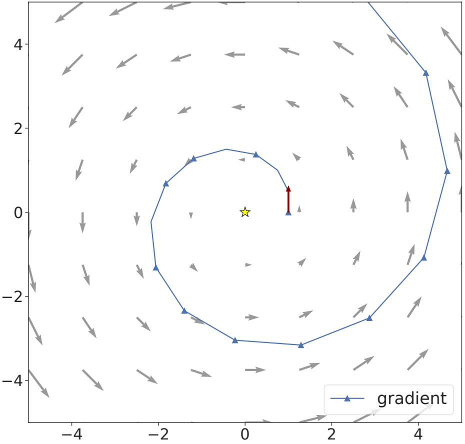
$$\ell^1(\mathbf{x}) = -\ell^2(\mathbf{x}) = \theta\phi \quad [\mathbf{x} = (\theta, \phi) \in \mathbb{R}^2]$$

- Game vector field

$$\mathbf{V}(\mathbf{x}) = (\nabla_\theta \ell^1(\mathbf{x}), \nabla_\phi \ell^2(\mathbf{x})) = (\phi, -\theta)$$

- Gradient descent

$$\mathbf{X}_{t+1} = \mathbf{X}_t - \eta_t \mathbf{V}(\mathbf{X}_t)$$



Optimistic Gradient to the Rescue

- Two-player planar bilinear zero-sum game

$$\ell^1(\mathbf{x}) = -\ell^2(\mathbf{x}) = \theta\phi \quad [\mathbf{x} = (\theta, \phi) \in \mathbb{R}^2]$$

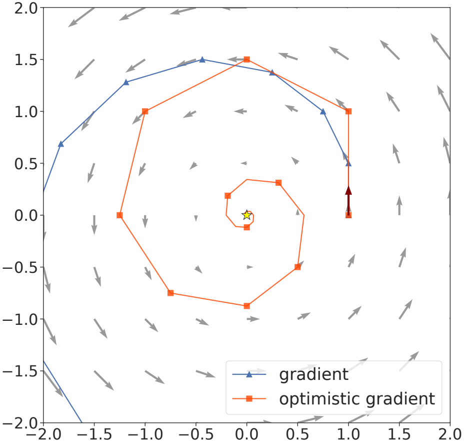
- Game vector field

$$\mathbf{V}(\mathbf{x}) = (\nabla_\theta \ell^1(\mathbf{x}), \nabla_\phi \ell^2(\mathbf{x})) = (\phi, -\theta)$$

- Optimistic gradient descent [Popov 80]

$$\mathbf{X}_{t+\frac{1}{2}} = \mathbf{X}_t - \eta_t \boxed{\mathbf{V}(\mathbf{X}_{t-\frac{1}{2}})}$$

$$\mathbf{X}_{t+1} = \mathbf{X}_t - \eta_{t+1} \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}})$$



Optimistic Gradient in Purely Online Setup

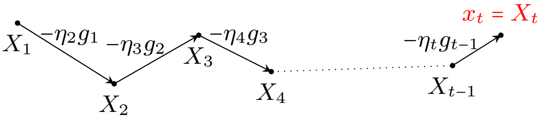
$$\boxed{X_{t+1} = \Pi_{\mathcal{X}}(X_t - \eta_{t+1} g_t)}$$

Online gradient descent: $x_t = X_t$

$$\mathrm{Reg}_T(p) = \mathcal{O}\left(\sqrt{\sum_{t=1}^{T} \|g_t\|^2}\right) = \mathcal{O}(\sqrt{T})$$

[Zinkevich 03]

Optimal in the *worst* case



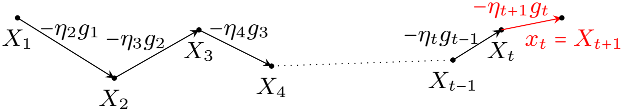
Optimistic Gradient in Purely Online Setup

$$X_{t+1} = \Pi_{\mathcal{X}}(X_t - \eta_{t+1} g_t)$$

A conceptual algorithm: $x_t = X_{t+1} = \Pi_{\mathcal{X}}(X_t - \eta_{t+1} g_t)$

$$\mathrm{Reg}_T(p) = \mathcal{O}(1)$$

This strategy is not implementable as it requires to know g_t before playing x_t



Optimistic Gradient in Purely Online Setup

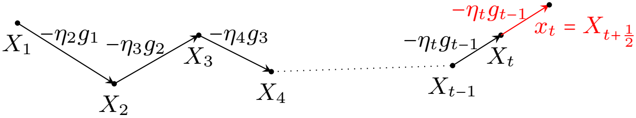
$$X_{t+1} = \Pi_{\mathcal{X}}(X_t - \eta_{t+1} g_t)$$

Optimistic gradient descent: $x_t = \Pi_{\mathcal{X}}(X_t - \eta_t g_{t-1})$

$$\mathrm{Reg}_T(p) = \mathcal{O}\left(\sqrt{\sum_{t=1}^{T} \|g_t - g_{t-1}\|^2}\right)$$

[Chiang et al. 12]

We are optimistic because we expect g_{t-1} to be close to g_t



Contributions for Part II

- Adaptive algorithm
- Robustness against noise
- Sublinear regret against adversarial opponents
- Constant regret in self-play
- Convergence to Nash Equilibrium in self-play
- Convergence rates under error bound condition
- Local convergence results

In this defense

---

1. H., Iutzeler, Malick, and Mertikopoulos. *Explore Aggressively, Update Conservatively: Stochastic Extragradient Methods with Variable Stepsize Scaling.* NeurIPS, 2020.

2. H., Antonakopoulos, Mertikopoulos. *Adaptive Learning in Continuous Games: Optimal Regret Bounds and Convergence to Nash Equilibrium.* COLT, 2021.

3. H., Antonakopoulos, Cevher, Mertikopoulos. *No-Regret Learning in Games with Noisy Feedback: Faster Rates and Adaptivity via Learning Rate Separation.* NeurIPS, 2022.

Contributions for Part II: Case of Perfect Feedback

- Adaptive algorithm
- Robustness against noise
- Sublinear regret against adversarial opponents
- Constant regret in self-play
- Convergence to Nash Equilibrium in self-play
- Convergence rates under error bound condition
- Local convergence results

In this defense

---

1. H., Iutzeler, Malick, and Mertikopoulos. *Explore Aggressively, Update Conservatively: Stochastic Extragradient Methods with Variable Stepsize Scaling.* NeurIPS, 2020.

2. H., Antonakopoulos, Mertikopoulos. *Adaptive Learning in Continuous Games: Optimal Regret Bounds and Convergence to Nash Equilibrium.* COLT, 2021.

3. H., Antonakopoulos, Cevher, Mertikopoulos. *No-Regret Learning in Games with Noisy Feedback: Faster Rates and Adaptivity via Learning Rate Separation.* NeurIPS, 2022.

Toward Adaptive Learning Rate

All the favorable guarantees break if learning rates are not properly tuned

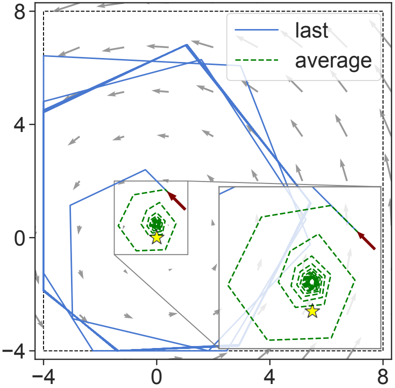
- Two-player planar bilinear zero-sum game

$$\ell^1(\mathbf{x}) = -\ell^2(\mathbf{x}) = \theta\phi \quad \text{where} \quad \mathcal{X}^1 = \mathcal{X}^2 = [-4, 8]$$

- The two players play optimistic gradient with constant $\eta = 0.7$ and $T = 100$

Problem

Convergence only holds for small enough η



Toward Adaptive Learning Rate

All the favorable guarantees break if learning rates are not properly tuned

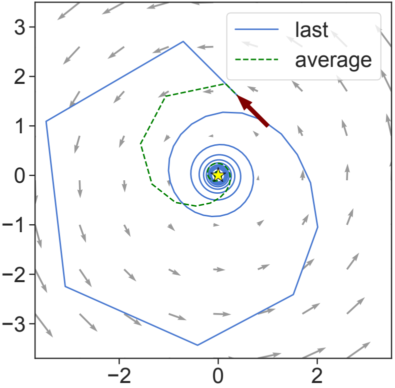
- Two-player planar bilinear zero-sum game

$$\ell^1(\mathbf{x}) = -\ell^2(\mathbf{x}) = \theta\phi \quad \text{where} \quad \mathcal{X}^1 = \mathcal{X}^2 = [-4, 8]$$

- The two players play optimistic gradient with decreasing $\eta_t = 1/\sqrt{t}$ and $T = 100$

Solution?

$\eta_t \propto 1/\sqrt{t}$ → slow convergence



Toward Adaptive Learning Rate

All the favorable guarantees break if learning rates are not properly tuned

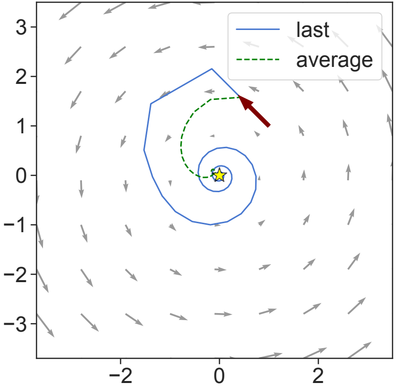
- Two-player planar bilinear zero-sum game

$$\ell^1(\mathbf{x}) = -\ell^2(\mathbf{x}) = \theta\phi \quad \text{where} \quad \mathcal{X}^1 = \mathcal{X}^2 = [-4, 8]$$

- The two players play optimistic gradient with adaptive η_t and $T = 100$

Solution

Adaptive learning rate



Toward Adaptive Learning Rate

$$\boxed{\sum_{t=1}^{T}\langle g_t^i, X_{t+\frac{1}{2}}^i - p^i\rangle \le \frac{h^i(p^i) - \min h^i}{\eta_{T+1}^i} + \sum_{t=1}^{T} \eta_t^i \|g_t^i - g_{t-1}^i\|^2 - \sum_{t=2}^{T} \frac{1}{8\eta_{t-1}^i}\|X_{t+\frac{1}{2}}^i - X_{t-\frac{1}{2}}^i\|^2}$$

Take the adaptive learning rate

$$\eta_t^i = \frac{1}{\sqrt{\tau^i + \sum_{s=1}^{t-1}\|g_t^i - g_{t-1}^i\|^2}} \qquad \text{(Adapt)}$$

- $\tau^i > 0$ can be chosen freely by the player
- η_t^i is thus computed solely based on local information available to each player

Theoretical Guarantees

Theorem [H. et al. 21]

Assume that player i runs OptDA with learning rate (Adapt), we have the following guarantees under different situations:

1. **[Adversarial]** Player i's regret is bounded as $\mathcal{O}\left(\sqrt{\sum_{t=1}^{T}\|g_t^i - g_{t-1}^i\|^2}\right)$
2. [Self-play] All the players have constant regret and the trajectory of play converges to Nash equilibrium if the game is variationally stable.

Theoretical Guarantees

Theorem [H. et al. 21]

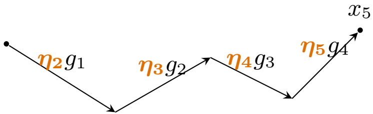
Assume that player i runs OptDA with learning rate (Adapt), we have the following guarantees under different situations:

1. [Adversarial] Player i's regret is bounded as $\mathcal{O}\left(\sqrt{\sum_{t=1}^{T}\|g_t^i - g_{t-1}^i\|^2}\right)$
2. **[Self-play]** All the players have constant regret and the trajectory of play converges to Nash equilibrium if the game is variationally stable.

Optimistic Gradient Descent and Optimistic Dual Averaging

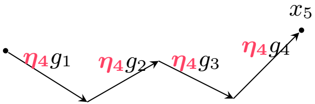
Optimistic gradient descent [Popov 80]

$$X^i_{t+\frac{1}{2}} = \Pi_{\mathcal{X}}(X^i_t - \eta^i_t g^i_{t-1})$$
$$X^i_{t+1} = \Pi_{\mathcal{X}}(X^i_t - \eta^i_{t+1} g^i_t)$$



Optimistic dual averaging [Song et al. 20]

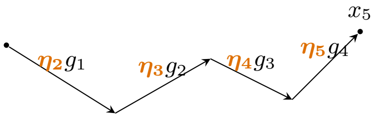
$$X^i_{t+\frac{1}{2}} = \Pi_{\mathcal{X}}(X^i_t - \eta^i_t g^i_{t-1})$$
$$X^i_{t+1} = \Pi_{\mathcal{X}}(X^i_1 - \eta^i_{t+1} \sum_{s=1}^{t} g^i_s)$$



Optimistic Gradient Descent and Optimistic Dual Averaging

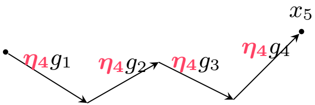
Optimistic gradient descent [Popov 80]

$$\begin{aligned}\rightarrow X^i_{t+\frac{1}{2}} &= \Pi_{\mathcal{X}}(X^i_t - \eta^i_t g^i_{t-1})\\ X^i_{t+1} &= \Pi_{\mathcal{X}}(X^i_t - \eta^i_{t+1} g^i_t)\end{aligned}$$



Optimistic dual averaging [Song et al. 20]

$$\begin{aligned}\rightarrow X^i_{t+\frac{1}{2}} &= \Pi_{\mathcal{X}}(X^i_t - \eta^i_t g^i_{t-1})\\ X^i_{t+1} &= \Pi_{\mathcal{X}}(X^i_1 - \eta^i_{t+1} \sum_{s=1}^{t} g^i_s)\end{aligned}$$

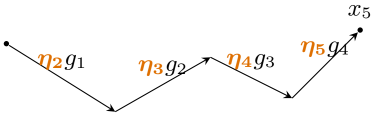


Optimistic Gradient Descent and Optimistic Dual Averaging

Optimistic gradient descent [Popov 80]

$$X^i_{t+\frac{1}{2}} = \Pi_{\mathcal{X}}(X^i_t - \eta^i_t g^i_{t-1})$$

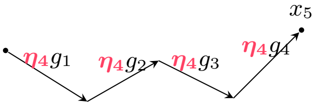
$$\to X^i_{t+1} = \Pi_{\mathcal{X}}(X^i_t - \eta^i_{t+1} g^i_t)$$



Optimistic dual averaging [Song et al. 20]

$$X^i_{t+\frac{1}{2}} = \Pi_{\mathcal{X}}(X^i_t - \eta^i_t g^i_{t-1})$$

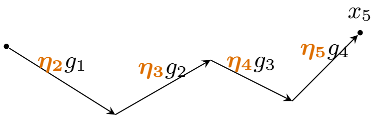
$$X^i_{t+1} = \Pi_{\mathcal{X}}(X^i_1 - \eta^i_{t+1} \sum_{s=1}^{t} g^i_s)$$



Optimistic Gradient Descent and Optimistic Dual Averaging

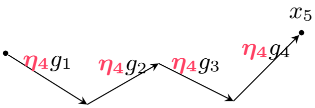
Optimistic gradient descent [Popov 80]

$$X^i_{t+\frac{1}{2}} = \Pi_{\mathcal{X}}(X^i_t - \eta^i_t g^i_{t-1})$$
$$X^i_{t+1} = \Pi_{\mathcal{X}}(X^i_t - \eta^i_{t+1} g^i_t)$$



Optimistic dual averaging [Song et al. 20]

$$X^i_{t+\frac{1}{2}} = \Pi_{\mathcal{X}}(X^i_t - \eta^i_t g^i_{t-1})$$
$$\rightarrow X^i_{t+1} = \Pi_{\mathcal{X}}(X^i_1 - \eta^i_{t+1} \sum_{s=1}^{t} g^i_s)$$



Contributions for Part II: What We Have Seen

- Adaptive algorithm
- Robustness against noise
- Sublinear regret against adversarial opponents
- Constant regret in self-play
- Convergence to Nash Equilibrium in self-play
- Convergence rates under error bound condition
- Local convergence results

In this defense

---

1. H., Iutzeler, Malick, and Mertikopoulos. *Explore Aggressively, Update Conservatively: Stochastic Extragradient Methods with Variable Stepsize Scaling.* NeurIPS, 2020.

2. H., Antonakopoulos, Mertikopoulos. *Adaptive Learning in Continuous Games: Optimal Regret Bounds and Convergence to Nash Equilibrium.* COLT, 2021.

3. H., Antonakopoulos, Cevher, Mertikopoulos. *No-Regret Learning in Games with Noisy Feedback: Faster Rates and Adaptivity via Learning Rate Separation.* NeurIPS, 2022.

Contributions for Part II: What Comes Next

- Adaptive algorithm
- Robustness against noise
- Sublinear regret against adversarial opponents
- Constant regret in self-play under multiplicative noise
- Convergence to Nash Equilibrium in self-play
- Convergence rates under error bound condition
- Local convergence results

$\}$ In this defense

---

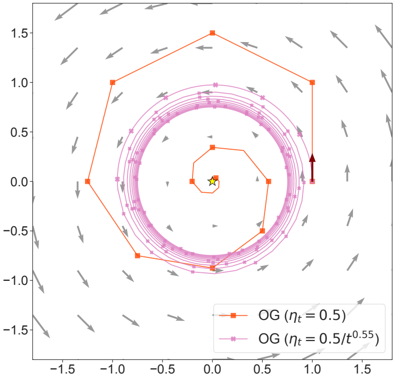
1. H., Iutzeler, Malick, and Mertikopoulos. *Explore Aggressively, Update Conservatively: Stochastic Extragradient Methods with Variable Stepsize Scaling.* NeurIPS, 2020.

2. H., Antonakopoulos, Mertikopoulos. *Adaptive Learning in Continuous Games: Optimal Regret Bounds and Convergence to Nash Equilibrium.* COLT, 2021.

3. H., Antonakopoulos, Cevher, Mertikopoulos. *No-Regret Learning in Games with Noisy Feedback: Faster Rates and Adaptivity via Learning Rate Separation.* NeurIPS, 2022.

Stochasticity Breaks Optimistic Gradient

All the favorable guarantees break if feedback is stochastic



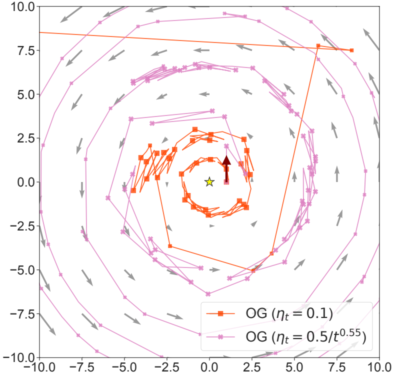
Stochastic Feedback

$\mathbb{E}[\hat{V}_{t+\frac{1}{2}}] = (\phi_{t+\frac{1}{2}}, -\theta_{t+\frac{1}{2}})$

$\longrightarrow$

$\hat{\mathbf{V}}_{t+\frac{1}{2}}$ is $(-\phi_{t+\frac{1}{2}}, \theta_{t+\frac{1}{2}})$

or $(3\phi_{t+\frac{1}{2}}, -3\theta_{t+\frac{1}{2}})$



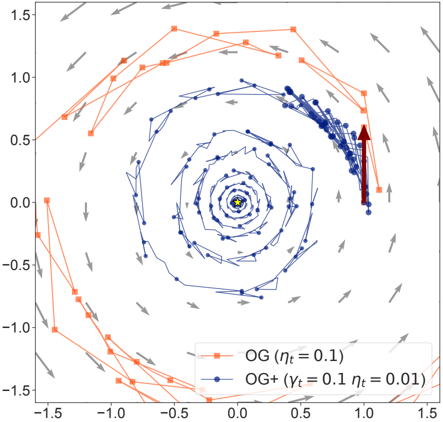
Toward Robustness Against Noise: Learning Rate Separation

Problem: Noise present in the two steps

- OG+ $[\mathbf{x}_t = \mathbf{X}_{t+\frac{1}{2}}]$

$$\mathbf{X}_{t+\frac{1}{2}} = \mathbf{X}_t - \gamma_t \hat{\mathbf{V}}_{t-\frac{1}{2}}$$
$$\mathbf{X}_{t+1} = \mathbf{X}_t - \eta_t \hat{\mathbf{V}}_{t+\frac{1}{2}}$$

  With $\gamma_t \geq \eta_t$
- Similar to mini-batching of the update step



Toward Robustness Against Noise: Learning Rate Separation

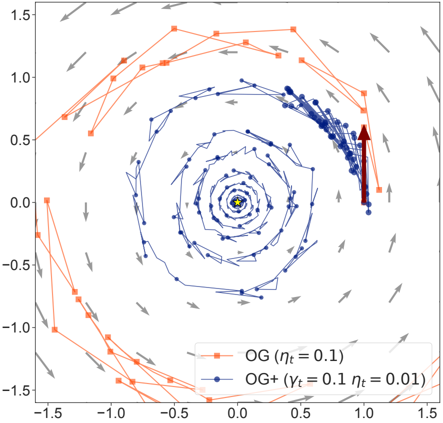
Problem: Noise present in the two steps

- OG+ $[\mathbf{x}_t = \mathbf{X}_{t+\frac{1}{2}}]$

$$\mathbf{X}_{t+\frac{1}{2}} = \mathbf{X}_t - \gamma_t \hat{\mathbf{V}}_{t-\frac{1}{2}}$$

$$\mathbf{X}_{t+1} = \mathbf{X}_t - \eta_t \hat{\mathbf{V}}_{t+\frac{1}{2}}$$

  With $\gamma_t \geq \eta_t$
- Similar to mini-batching of the update step



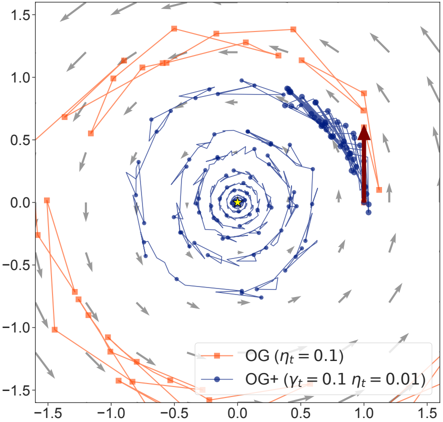
Toward Robustness Against Noise: Learning Rate Separation

- OG+ is guaranteed to converge to Nash equilibrium under VS if

$$\sum_{t=1}^{+\infty} \gamma_t \eta_{t+1} = +\infty,$$

$$\sum_{t=1}^{+\infty} \gamma_t^2 \eta_{t+1} < +\infty, \quad \sum_{t=1}^{+\infty} \eta_t^2 < +\infty$$

- We can take constant learning rates if the noise is multiplicative

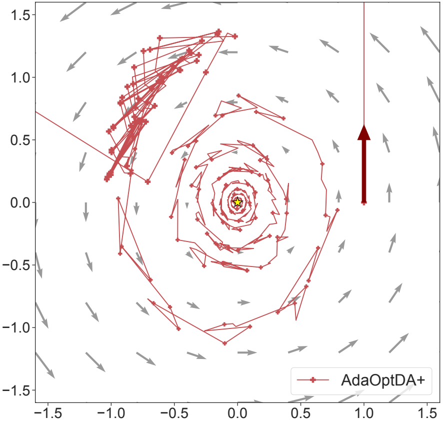


Toward Robustness Against Noise: Adaptive Learning Rates

- AdaOptDA+ uses learning rates

$$\gamma_t^i = \frac{1}{\left(1 + \sum_{s=1}^{t-2} \|g_s^i\|^2\right)^{\boxed{\frac{1}{4}}}}$$

$$\eta_t^i = \frac{1}{\sqrt{1 + \sum_{s=1}^{t-2} \left(\|g_s^i\|^2 + \|X_s^i - X_{s+1}^i\|^2\right)}}$$



Stochastic Oracle

- We focus on the unconstrained setup $\mathcal{X}^i = \mathbb{R}^{d^i}$
- Stochastic feedback $g_t^i = \nabla_i \ell^i(\mathbf{x}_t) + \xi_t^i$ with noise satisfying
  1. *Zero-mean:* $\mathbb{E}_t[\xi_t^i] = 0$
  2. *Variance control:* $\mathbb{E}_t[\|\xi_t^i\|^2] \leq \sigma_{\mathsf{add}}^2 + \sigma_{\mathsf{mult}}^2 \|\nabla_i \ell^i(\mathbf{x}_t)\|^2$
- We say that the noise is multiplicative if $\sigma_{\mathsf{add}}^2 = 0$

  Examples:
  - Randomized coordinate descent
  - Finite sum of operators whose solution sets intersect

Theoretical Guarantees for Learning with Noisy Feedback: OG

		Adversarial	Self-Play + Variational Stability		
		Bounded feedback	-	-	Strongly M
	Noise	Reg_t	Reg_t	Cvg?	$\mathrm{dist}(\mathbf{X}_t, \mathcal{X}_\star)$
OG	-	$\sqrt{t}$ [Chiang et al. 12]	$\sqrt{t}$ [Gidel et al. 19]	✗ [H. et al. 20]	$1/\sqrt{t}$ [H. et al. 19]

Theoretical Guarantees for Learning with Noisy Feedback [H. et al. 22]

		Adversarial	Self-Play + Variational Stability			
		Bounded feedback	-	-	Strongly M	Error bound
	Noise	Reg_t	Reg_t	Cvg?	$\mathrm{dist}(\mathbf{X}_t, \mathcal{X}_\star)$	$\mathrm{dist}(\mathbf{X}_t, \mathcal{X}_\star)$
OG+	-	$\sqrt{t}$	$\sqrt{t}$	✓	$1/\sqrt{t}$	$1/t^{1/6}$
	Mul.		1	✓	$e^{-\rho t}$	$e^{-\rho t}$
OptDA+	-	$\sqrt{t}$	$\sqrt{t}$	-	$1/\sqrt{t}$	$1/t^{1/6}$
	Mul.		1	✓	-	-
AdaOptDA+	-	$t^{3/4}$	$\sqrt{t}$	-	-	-
	Mul.		1	✓	-	-

Theoretical Guarantees With Unknown Time Horizon [H. et al. 22]

		Adversarial	Self-Play + Variational Stability			
		Bounded feedback	-	-	Strongly M	Error bound
	Noise	Reg_t	Reg_t	Cvg?	$\mathrm{dist}(\mathbf{X}_t, \mathcal{X}_\star)$	$\mathrm{dist}(\mathbf{X}_t, \mathcal{X}_\star)$
OG+	-	✗	$\sqrt{t}$	✓	$1/\sqrt{t}$	$1/t^{1/6}$
	Mul.		1	✓	$e^{-\rho t}$	$e^{-\rho t}$
OptDA+	-	$\sqrt{t}$	$\sqrt{t}$	-	-	-
	Mul.		1	✓	-	-
AdaOptDA+	-	$t^{3/4}$	$\sqrt{t}$	-	-	-
	Mul.		1	✓	-	-

Theoretical Guarantees With Unknown Time Horizon [H. et al. 22]

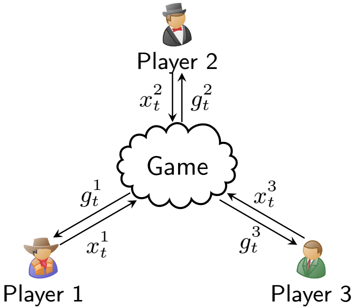
		Adversarial	Self-Play + Variational Stability			
		Bounded feedback	-	-	Strongly M	Error bound
	Noise	Reg_t	Reg_t	Cvg?	$\mathrm{dist}(\mathbf{X}_t, \mathcal{X}_\star)$	$\mathrm{dist}(\mathbf{X}_t, \mathcal{X}_\star)$
OG+	-	✗	$\sqrt{t}$	✓	$1/\sqrt{t}$	$1/t^{1/6}$
	Mul.		1	✓	$e^{-\rho t}$	$e^{-\rho t}$
OptDA+	-	$\sqrt{t}$	$\sqrt{t}$	-	-	-
	Mul.		1	✓	-	-
AdaOptDA+	-	$t^{3/4}$	$\sqrt{t}$	-	-	-
	Mul.		1	✓	-	-

Theoretical Guarantees With Unknown Time Horizon [H. et al. 22]

		Adversarial	Self-Play + Variational Stability			
		Bounded feedback	-	-	Strongly M	Error bound
	Noise	Reg_t	Reg_t	Cvg?	$\mathrm{dist}(\mathbf{X}_t, \mathcal{X}_\star)$	$\mathrm{dist}(\mathbf{X}_t, \mathcal{X}_\star)$
OG+	-	✗	$\sqrt{t}$	✓	$1/\sqrt{t}$	$1/t^{1/6}$
	Mul.		1	✓	$e^{-\rho t}$	$e^{-\rho t}$
OptDA+	-	$\sqrt{t}$	$\sqrt{t}$	-	-	-
	Mul.		1	✓	-	-
AdaOptDA+	-	$t^{3/4}$	$\sqrt{t}$	-	-	-
	Mul.		1	✓	-	-

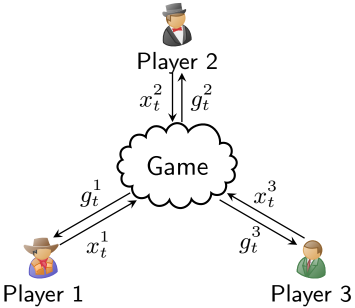
What We Have Seen in This Part

- **Learning-in-game algorithms run individually without knowledge about the game**
- Nearly optimal guarantees in different situations, potentially under noisy feedback
- Examined Challenges
  - Conflicting interests
  - Non-stationarity
  - Lack of coordination
  - Adaptive learning
  - Uncertainty



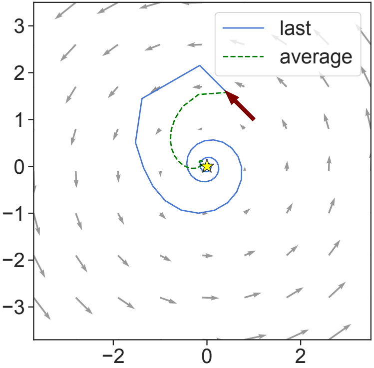
What We Have Seen in This Part

- **Learning-in-game algorithms run individually without knowledge about the game**
- Nearly optimal guarantees in different situations, potentially under noisy feedback
- Examined Challenges
  - Conflicting interests
  - Non-stationarity
  - Lack of coordination
  - Adaptive learning
  - Uncertainty



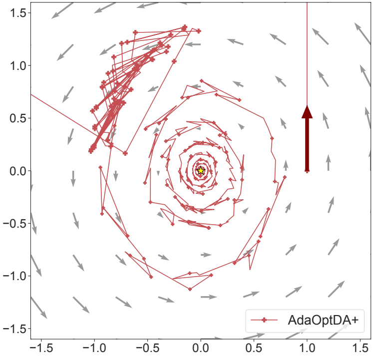
What We Have Seen in This Part

- Learning-in-game algorithms run individually without knowledge about the game
- **Nearly optimal guarantees in different situations, potentially under noisy feedback**
- Examined Challenges
  - Conflicting interests
  - Non-stationarity
  - Lack of coordination
  - Adaptive learning
  - Uncertainty



What We Have Seen in This Part

- Learning-in-game algorithms run individually without knowledge about the game
- **Nearly optimal guarantees in different situations, potentially under noisy feedback**
- Examined Challenges
  - Conflicting interests
  - Non-stationarity
  - Lack of coordination
  - Adaptive learning
  - Uncertainty



Perspectives

Perspectives

- **Better understanding of the algorithms**
  - Guarantees for broader family of algorithms as in no-regret [Sorin 23]
  - Other guarantees: Policy regret [Arora et al. 12], dynamic regret [Zinkevich 03]
- **Different setups**
  - Bandit [Cesa-Bianchi et al. 19, Tatarenko and Kamgarpour 18]
  - Non-convex games [Daskalakis 22]
  - Stochastic games [Shapley 53]

Perspectives

- **Better understanding of the algorithms**
  - Guarantees for broader family of algorithms as in no-regret [Sorin 23]
  - Other guarantees: Policy regret [Arora et al. 12], dynamic regret [Zinkevich 03]
- **Different setups**
  - Bandit [Cesa-Bianchi et al. 19, Tatarenko and Kamgarpour 18]
  - Non-convex games [Daskalakis 22]
  - Stochastic games [Shapley 53]

Perspectives

- **Better understanding of the algorithms**
  - Guarantees for broader family of algorithms as in no-regret [Sorin 23]
  - Other guarantees: Policy regret [Arora et al. 12], dynamic regret [Zinkevich 03]
- **Different setups**
  - Bandit [Cesa-Bianchi et al. 19, Tatarenko and Kamgarpour 18]
  - Non-convex games [Daskalakis 22]
  - Stochastic games [Shapley 53]

Perspectives

- **Better understanding of the algorithms**
  - Guarantees for broader family of algorithms as in no-regret [Sorin 23]
  - Other guarantees: Policy regret [Arora et al. 12], dynamic regret [Zinkevich 03]
- **Different setups**
  - Bandit [Cesa-Bianchi et al. 19, Tatarenko and Kamgarpour 18]
  - Non-convex games [Daskalakis 22]
  - Stochastic games [Shapley 53]

Perspectives

- **Better understanding of the algorithms**
  - Guarantees for broader family of algorithms as in no-regret [Sorin 23]
  - Other guarantees: Policy regret [Arora et al. 12], dynamic regret [Zinkevich 03]
- **Different setups**
  - Bandit [Cesa-Bianchi et al. 19, Tatarenko and Kamgarpour 18]
  - Non-convex games [Daskalakis 22]
  - Stochastic games [Shapley 53]

Perspectives

- **Better understanding of the algorithms**
  - Guarantees for broader family of algorithms as in no-regret [Sorin 23]
  - Other guarantees: Policy regret [Arora et al. 12], dynamic regret [Zinkevich 03]
- **Different setups**
  - Bandit [Cesa-Bianchi et al. 19, Tatarenko and Kamgarpour 18]
  - Non-convex games [Daskalakis 22]
  - Stochastic games [Shapley 53]

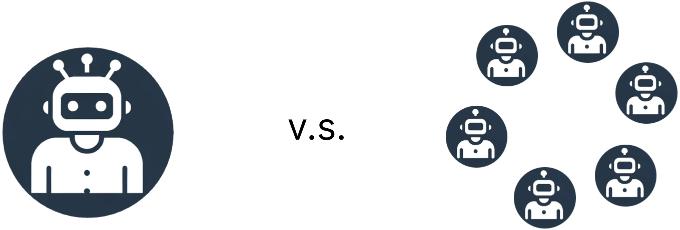
Perspectives

- **Better understanding of the algorithms**
    - Guarantees for broader family of algorithms as in no-regret [Sorin 23]
    - Other guarantees: Policy regret [Arora et al. 12], dynamic regret [Zinkevich 03]
- **Different setups**
    - Bandit [Cesa-Bianchi et al. 19, Tatarenko and Kamgarpour 18]
    - Non-convex games [Daskalakis 22]
    - Stochastic games [Shapley 53]

Perspectives: Other Interesting Related Directions

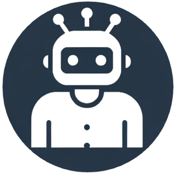
- **Evaluation and alignment of generative models with preference-based feedback**
    - Dueling bandit and learning in two-player zero-sum finite games [Ailon et al. 14]
    - RLHF and learning in stochastic games [Wang et al. 23]

- **One big model or many small models**

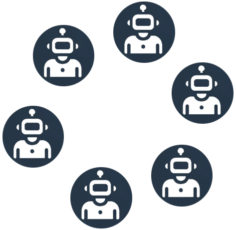


Perspectives: Other Interesting Related Directions

- **Evaluation and alignment of generative models with preference-based feedback**
  - Dueling bandit and learning in two-player zero-sum finite games [Ailon et al. 14]
  - RLHF and learning in stochastic games [Wang et al. 23]
- **One big model or many small models**

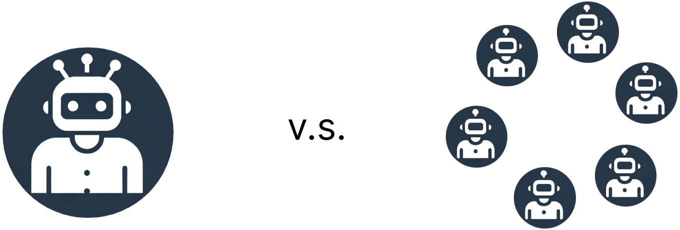


v.s.



Perspectives: Other Interesting Related Directions

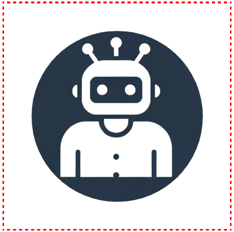
- **Evaluation and alignment of generative models with preference-based feedback**
  - Dueling bandit and learning in two-player zero-sum finite games [Ailon et al. 14]
  - RLHF and learning in stochastic games [Wang et al. 23]
- **One big model or many small models**



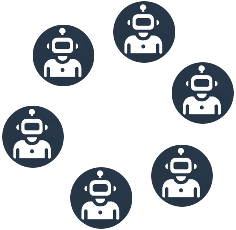
v.s.

Perspectives: Other Interesting Related Directions

- **Evaluation and alignment of generative models with preference-based feedback**
    - Dueling bandit and learning in two-player zero-sum finite games [Ailon et al. 14]
    - RLHF and learning in stochastic games [Wang et al. 23]
- **One big model or many small models**

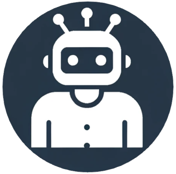


v.s.

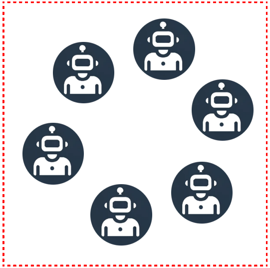


Perspectives: Other Interesting Related Directions

- **Evaluation and alignment of generative models with preference-based feedback**
  - Dueling bandit and learning in two-player zero-sum finite games [Ailon et al. 14]
  - RLHF and learning in stochastic games [Wang et al. 23]
- **One big model or many small models**



v.s.



My Publications

[1] Shin-Ying Yeh, **Y. H.**, Zhidong Gao, Bernard B W Yang, Giyeong Oh, and Yanmin Gong. *Navigating Text-To-Image Customization: From LyCORIS Fine-Tuning to Model Evaluation.* Submitted to ICLR, 2023.

[2] **Y. H.**, Shiva Kasiviswanathan, Branislav Kveton, and Patrick Bloebaum. *Thompson Sampling with Diffusion Generative Prior.* In **ICML**, 2023.

[3] **Y. H.**, Yassine Laguel, Franck Iutzeler, and Jérôme Malick. *Push–Pull with Device Sampling.* **TACON**, 2023.

[4] **Y. H.**, Kimon Antonakopoulos, Volkan Cevher, and Panayotis Mertikopoulos. *No-Regret Learning in Games with Noisy Feedback: Faster Rates and Adaptivity via Learning Rate Separation.* In **NeurIPS**, 2022.

[5] **Y. H.**, Shiva Kasiviswanathan, and Branislav Kveton. *Uplifting Bandits.* In **NeurIPS**, 2022.

[6] **Y. H.**, Franck Iutzeler, Jérôme Malick, and Panayotis Mertikopoulos. *Multi-agent Online Optimization with Delays: Asynchronicity, Adaptivity, and Optimism.* **JMLR**, 2022.

[7] **Y. H.**, Franck Iutzeler, Jérôme Malick, and Panayotis Mertikopoulos. *Optimization in Open Networks via Dual Averaging.* In **CDC**, 2021.

[8] **Y. H.**, Kimon Antonakopoulos, and Panayotis Mertikopoulos. *Adaptive Learning in Continuous Games: Optimal Regret Bounds and Convergence to Nash Equilibrium.* In **COLT**, 2021.

[9] **Y. H.**, Franck Iutzeler, Jérôme Malick, and Panayotis Mertikopoulos. *Explore Aggressively, Update Conservatively: Stochastic Extragradient Methods with Variable Stepsize Scaling.* In **NeurIPS**, 2020.

[10] **Y. H.**, Franck Iutzeler, Jérôme Malick, and Panayotis Mertikopoulos. *On the Convergence of Single-Call Stochastic Extra-Gradient Methods.* In **NeurIPS**, 2019.

[11] **Y. H.**, Gang Niu, and Masashi Sugiyama. *Classification from Positive, Unlabeled and Biased Negative Data.* In **ICML**, 2019.

Proof Sketch for Regret Bound of AdaDelay-Dist

- Show that the learning rate is non-increasing along a faithful permutation π
- Show that $\eta_{\pi(t)+2\tau+1} \leq R/\sqrt{\Lambda_t^\pi}$
- Apply template regret bound to conclude

Stochastic Feedback in Asynchronous Decentralized Online Learning

- The template regret bound still holds when noise variance is bounded
- We recover the same results for learning rates that do not depend on the realization
- For AdaDelay-Dist we require the feedback to be bounded almost surely, and we get regret with a $\mathbb{E}[\sqrt{\Lambda_T}]$ term

Bandit Feedback in Asynchronous Decentralized Online Learning

The vector z_t randomly drawn from the sphere

- Two-point estimate:

$$g_t = \frac{d}{2\delta}(\ell_t(y_t + \delta z_t) - \ell_t(y_t + \delta z_t))z_t$$

  We have $\|g_t\| \leq Gd$, so everything still holds and δ should be as small as possible

- Single-point estimate:

$$g_t = \frac{d}{\delta}(\ell_t(y_t + \delta z_t))z_t$$

  We have $\|g_t\| \leq \frac{Fd}{\delta}$, bias is in δ, regret is $\mathcal{O}(D^{1/4}T^{1/2})$ if everything properly tuned

In both cases, $\mathbb{E}[g_t] = \nabla \tilde{\ell}(y_t)$ for $\tilde{\ell}(x) = \mathbb{E}_{z\in\mathbb{B}}[\ell(x + \delta z)]$

Bandit Feedback in Asynchronous Decentralized Online Learning

- We need to know the sampled vector associated to each feedback loss
- How can we adaptively tune δ?

Mirror Descent versus Dual Averaging

Mirror descent type methods with dynamic learning rates **may incur regret**

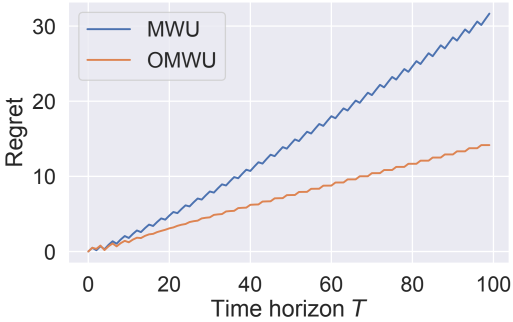
Assume that player 1 has a linear loss and simplex-constrained action set.

- $\mathcal{X}^1 = \Delta^1 = \{(w_1, w_2) \in \mathbb{R}^2_+, w_1 + w_2 = 1\}$
- Feedback sequence:

$$[\underbrace{-e_1, \ldots, -e_1}_{\lceil T/3 \rceil}, \underbrace{-e_2, \ldots, -e_2}_{\lfloor 2T/3 \rfloor}]$$

- Adaptive (Optimistic) Multiplicative Weight Update

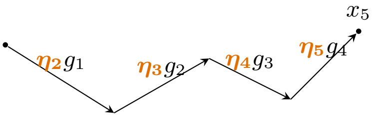
(Example from [Orabona and Pal 16])



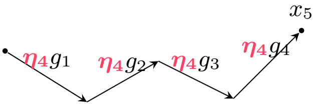
Mirror Descent versus Dual Averaging

Mirror descent type methods with dynamic learning rates **may incur regret**

- Cause: new information enters MD with a decreasing weight
- Solution: enter each feedback with equal weight
  E.g. Dual averaging or stabilization technique



MD-type update



DA-type update

Mirror Descent versus Dual Averaging

Mirror descent type methods with dynamic learning rates **may incur regret**

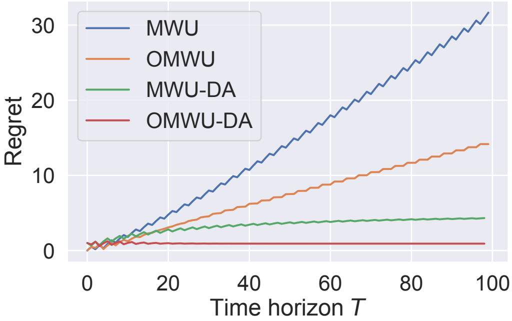
Assume that player 1 has a linear loss and simplex-constrained action set.

- $\mathcal{X}^1 = \Delta^1 = \{(w_1, w_2) \in \mathbb{R}^2_+, w_1 + w_2 = 1\}$
- Feedback sequence:

$$[\underbrace{-e_1, \dots, -e_1}_{\lceil T/3 \rceil}, \underbrace{-e_2, \dots, -e_2}_{\lfloor 2T/3 \rfloor}]$$

- Adaptive (Optimistic) Multiplicative Weight Update with Dual Averaging

(Example from [Orabona and Pal 16])



Optimistic Algorithm: A General Procedure

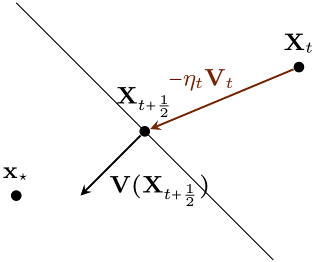
Two types of states: memory y_t and action x_t

- Optimistic step: Compute x_t from y_t using a guess $\tilde{g}_t$, play x_t
- Update step: Update the memory from y_t to y_{t+1} using feedback g_t

Examples: • Mirror-prox [Nemirovski 04] • optimistic FTRL [Joulani et al. 17]

Optimistic Gradient as Relaxed Projection onto Separating Hyperplane

$$\boxed{\mathbf{X}_{t+\frac{1}{2}} = \mathbf{X}_t - \eta_t \mathbf{V}_t, \quad \mathbf{X}_{t+1} = \mathbf{X}_t - \eta_{t+1} \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}})}$$

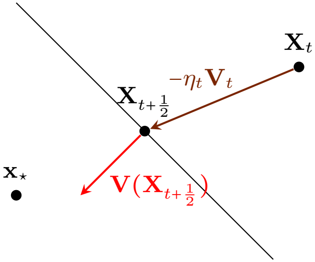


Optimistic Gradient as Relaxed Projection onto Separating Hyperplane

$$\boxed{\mathbf{X}_{t+\frac{1}{2}} = \mathbf{X}_t - \eta_t \mathbf{V}_t, \quad \mathbf{X}_{t+1} = \mathbf{X}_t - \eta_{t+1} \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}})}$$

- Consider the hyperplan

$$\mathcal{H} \coloneqq \{\mathbf{x} : \langle \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}), \mathbf{X}_{t+\frac{1}{2}} - \mathbf{x} \rangle = 0\}$$



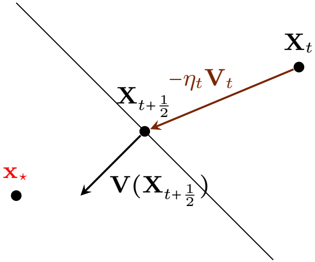
Optimistic Gradient as Relaxed Projection onto Separating Hyperplane

$$\boxed{\mathbf{X}_{t+\frac{1}{2}} = \mathbf{X}_t - \eta_t \mathbf{V}_t, \quad \mathbf{X}_{t+1} = \mathbf{X}_t - \eta_{t+1} \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}})}$$

- Consider the hyperplan

$$\mathcal{H} \coloneqq \{\mathbf{x} : \langle \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}), \mathbf{X}_{t+\frac{1}{2}} - \mathbf{x} \rangle = 0\}$$

- Assumption: $\langle \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}), \mathbf{X}_{t+\frac{1}{2}} - \mathbf{x}_\star \rangle \geq 0$



Optimistic Gradient as Relaxed Projection onto Separating Hyperplane

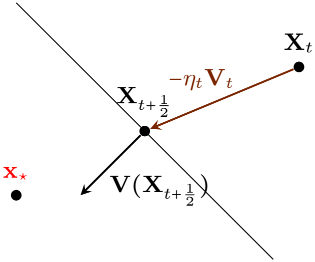
$$\boxed{\mathbf{X}_{t+\frac{1}{2}} = \mathbf{X}_t - \eta_t \mathbf{V}_t, \quad \mathbf{X}_{t+1} = \mathbf{X}_t - \eta_{t+1}\mathbf{V}(\mathbf{X}_{t+\frac{1}{2}})}$$

- Consider the hyperplan

$$\mathcal{H} \coloneqq \{\mathbf{x} : \langle \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}), \mathbf{X}_{t+\frac{1}{2}} - \mathbf{x}\rangle = 0\}$$

- Assumption: $\langle \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}), \mathbf{X}_{t+\frac{1}{2}} - \mathbf{x}_\star\rangle \geq 0$

  Monotone: $\langle \mathbf{V}(\mathbf{x}') - \mathbf{V}(\mathbf{x}), \mathbf{x}' - \mathbf{x}\rangle \geq 0$



Optimistic Gradient as Relaxed Projection onto Separating Hyperplane

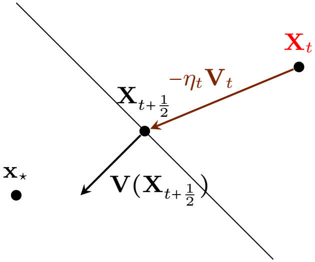
$$\boxed{\mathbf{X}_{t+\frac{1}{2}} = \mathbf{X}_t - \eta_t \mathbf{V}_t, \quad \mathbf{X}_{t+1} = \mathbf{X}_t - \eta_{t+1} \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}})}$$

- Consider the hyperplan

$$\mathcal{H} \coloneqq \{\mathbf{x} : \langle \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}), \mathbf{X}_{t+\frac{1}{2}} - \mathbf{x} \rangle = 0\}$$

- Assumption: $\langle \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}), \mathbf{X}_{t+\frac{1}{2}} - \mathbf{x}_\star \rangle \geq 0$
- If $\mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}) \approx \mathbf{V}_t$ then

$$\langle \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}), \mathbf{X}_{t+\frac{1}{2}} - \mathbf{X}_t \rangle = -\eta_t \langle \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}), \mathbf{V}_t \rangle \leq 0$$



Optimistic Gradient as Relaxed Projection onto Separating Hyperplane

$$\boxed{\mathbf{X}_{t+\frac{1}{2}} = \mathbf{X}_t - \eta_t \mathbf{V}_t, \quad \mathbf{X}_{t+1} = \mathbf{X}_t - \eta_{t+1} \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}})}$$

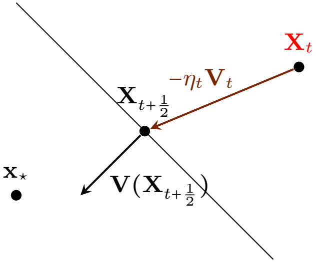
- Consider the hyperplan

$$\mathcal{H} \coloneqq \{\mathbf{x} : \langle \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}), \mathbf{X}_{t+\frac{1}{2}} - \mathbf{x} \rangle = 0\}$$

- Assumption: $\langle \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}), \mathbf{X}_{t+\frac{1}{2}} - \mathbf{x}_\star \rangle \geq 0$
- If $\mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}) \approx \mathbf{V}_t$ then

$$\langle \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}), \mathbf{X}_{t+\frac{1}{2}} - \mathbf{X}_t \rangle = -\eta_t \langle \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}), \mathbf{V}_t \rangle \leq 0$$

This is why we require Lipschitz continuity



Optimistic Gradient as Relaxed Projection onto Separating Hyperplane

$$\boxed{\mathbf{X}_{t+\frac{1}{2}} = \mathbf{X}_t - \eta_t \mathbf{V}_t, \quad \mathbf{X}_{t+1} = \mathbf{X}_t - \eta_{t+1}\mathbf{V}(\mathbf{X}_{t+\frac{1}{2}})}$$

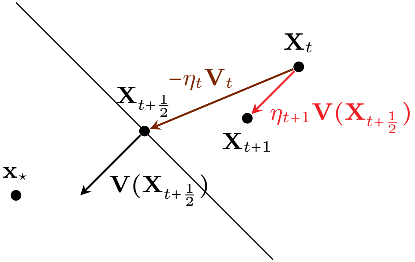
- Consider the hyperplan

$$\mathcal{H} \coloneqq \{\mathbf{x} : \langle \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}), \mathbf{X}_{t+\frac{1}{2}} - \mathbf{x}\rangle = 0\}$$

- Assumption: $\langle \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}), \mathbf{X}_{t+\frac{1}{2}} - \mathbf{x}_\star\rangle \geq 0$
- If $\mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}) \approx \mathbf{V}_t$ then

$$\langle \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}), \mathbf{X}_{t+\frac{1}{2}} - \mathbf{X}_t\rangle = -\eta_t \langle \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}}), \mathbf{V}_t\rangle \leq 0$$

- The update step moves the iterate closer to the solutions



An Intuition Behind Scale Separation of Learning Rates

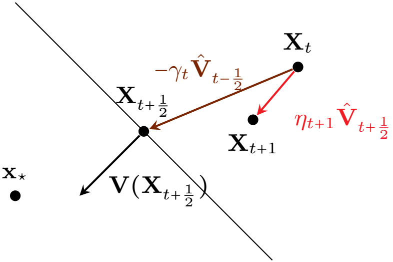
$$X^i_{t+\frac{1}{2}} = X^i_t - \boxed{\gamma^i_t} \, g^i_{t-1}, \qquad X^i_{t+1} = X^i_t - \boxed{\eta^i_{t+1}} \, g^i_t \tag{OG+}$$

$$X^i_{t+\frac{1}{2}} = X^i_t - \boxed{\gamma^i_t} \, g^i_{t-1}, \qquad X^i_{t+1} = X^i_1 - \boxed{\eta^i_{t+1}} \sum_{s=1}^{t} g^i_s \tag{OptDA+}$$

- Variational stability

$$\langle \mathbf{V}(\mathbf{x}), \mathbf{x} - \mathbf{x}_\star \rangle \geq 0$$

- Stochastic update: relaxation of an approximate projection step with relaxation factor of the order of η_{t+1}/γ_t → the ratio η_{t+1}/γ_t should go to 0



Sketch of Proof: Energy Inequality of OptDA+

$$
\begin{aligned}
&\mathbb{E}_{t-1}\left[\frac{\|X^i_{t+1}-p^i\|^2}{\eta^i_{t+1}}\right] \leq \mathbb{E}_{t-1}\Bigg[\frac{\|X^i_t-p^i\|^2}{\eta^i_t}+\left(\frac{1}{\eta^i_{t+1}}-\frac{1}{\eta^i_t}\right)\|X^i_1-p^i\|^2 \\
&\text{(linearized regret)} \qquad -2\langle V^i(\mathbf{X}_{t+\frac{1}{2}}), X^i_{t+\frac{1}{2}}-p^i\rangle \\
&\text{(negative drift)} \qquad -\boxed{\gamma^i_t}\,(\|\nabla_i\,\ell^i(\mathbf{X}_{t+\frac{1}{2}})\|^2+\|\nabla_i\,\ell^i(\mathbf{X}_{t-\frac{1}{2}})\|^2) \\
&\text{(variation)} \qquad -\frac{\|X^i_t-X^i_{t+1}\|^2}{2\eta^i_t}+\gamma^i_t\|\nabla_i\,\ell^i(\mathbf{X}_{t+\frac{1}{2}})-\nabla_i\,\ell^i(\mathbf{X}_{t-\frac{1}{2}})\|^2 \\
&\text{(noise)} \qquad +\boxed{(\gamma^i_t)^2}\,L\|\xi^i_{t-1}\|^2+L\|\boldsymbol{\xi}_{t-\frac{1}{2}}\|^2\underset{\boxed{(\boldsymbol{\eta}_t+\boldsymbol{\gamma}_t)^2}}{}+2\,\boxed{\eta^i_t}\,\|g^i_t\|^2\Bigg]
\end{aligned}
$$

Sketch of Proof: Adaptive Learning Rate

$$\Lambda_t^i = \sum_{s=1}^{t} \|g_s^i\|^2, \qquad \Gamma_t^i = \sum_{s=1}^{t} \|X_s^i - X_{s+1}^i\|^2$$

- For some constants c_1, c_2,

$$\sum_{t=1}^{T} \mathbb{E}[\|\mathbf{V}(\mathbf{X}_{t+\frac{1}{2}})\|_{\gamma_t}^2] + \frac{1}{8}\sum_{t=1}^{T} \mathbb{E}[\|\mathbf{X}_t - \mathbf{X}_{t+1}\|^2] \le c_1 \sum_{i=1}^{N} \mathbb{E}\left[\sqrt{\Lambda_T^i}\right] + c_2,$$

Sketch of Proof: Adaptive Learning Rate

$$\boxed{\Lambda_t^i = \sum_{s=1}^{t} \|g_s^i\|^2, \qquad \Gamma_t^i = \sum_{s=1}^{t} \|X_s^i - X_{s+1}^i\|^2}$$

- For some constants c_1, c_2,

$$\underbrace{\sum_{t=1}^{T} \mathbb{E}[\|\mathbf{V}(\mathbf{X}_{t+\frac{1}{2}})\|_{\gamma_t}^2]}_{\text{Bound from below with } \Lambda_T^i} + \frac{1}{8}\sum_{t=1}^{T} \mathbb{E}[\|\mathbf{X}_t - \mathbf{X}_{t+1}\|^2] \leq c_1 \sum_{i=1}^{N} \mathbb{E}\left[\sqrt{\Lambda_T^i}\right] + c_2,$$

- Multiplicative noise: for some constant C, $\sum_{i=1}^{N} \mathbb{E}\left[\sqrt{1 + \Lambda_T^i}\right] \leq C$ and $\sum_{i=1}^{N} \mathbb{E}[\Gamma_T^i] \leq C$

Sketch of Proof: Adaptive Learning Rate

$$\Lambda_t^i = \sum_{s=1}^{t} \|g_s^i\|^2, \qquad \Gamma_t^i = \sum_{s=1}^{t} \|X_s^i - X_{s+1}^i\|^2$$

- For some constants c_1, c_2,

$$\underbrace{\sum_{t=1}^{T} \mathbb{E}[\|\mathbf{V}(\mathbf{X}_{t+\frac{1}{2}})\|_{\gamma_t}^2]}_{\text{Bound from below with } \Lambda_T^i} + \frac{1}{8} \sum_{t=1}^{T} \mathbb{E}[\|\mathbf{X}_t - \mathbf{X}_{t+1}\|^2] \le c_1 \sum_{i=1}^{N} \mathbb{E}\left[\sqrt{\Lambda_T^i}\right] + c_2,$$

- Multiplicative noise: for some constant C, $\sum_{i=1}^{N} \mathbb{E}\left[\sqrt{1 + \Lambda_T^i}\right] \le C$ and $\sum_{i=1}^{N} \mathbb{E}[\Gamma_T^i] \le C$
- Convergence: Apply Robbins–Siegmund's theorem and define $\tilde{X}_t^i = X_t^i + \eta_t^i \xi_{t-1}^i$

Other Ways to Handle Noise

- Mini-batching: its naive implementation does not work for online learning

  $$1, -1, -1, 1, 1, 1, -1, \ldots$$

  The player plays

  $$0, -\eta, -\eta, 0, 0, 0, -\eta, \ldots$$

  Regret with respect to 0 is $(a_2 + a_4 + \ldots)\eta \geq \eta T/2$
- Anchoring: we lose the faster convergence rate under error bound condition

Toward Robustness Against Noise

All the favorable guarantees break if feedback is noisy

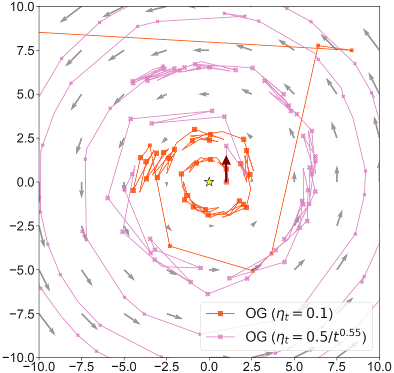
- Stochastic estimate $\mathbb{E}[\hat{\mathbf{V}}_{t+\frac{1}{2}}] = \mathbf{V}(\mathbf{X}_{t+\frac{1}{2}})$

$$\hat{\mathbf{V}}_{t+\frac{1}{2}} = \begin{cases} (3\phi_{t+\frac{1}{2}}, -3\theta_{t+\frac{1}{2}}) & \text{with prob. } 1/2 \\ (-\phi_{t+\frac{1}{2}}, \theta_{t+\frac{1}{2}}) & \text{with prob. } 1/2 \end{cases}$$

- The two players play optimistic gradient with decreasing $\eta_t = 0.1/\sqrt{t}$

Problem

We observe non-convergence and linear regret



Toward Robustness Against Noise

All the favorable guarantees break if feedback is noisy

- OG+ $[\mathbf{x}_t = \mathbf{X}_{t+\frac{1}{2}}]$

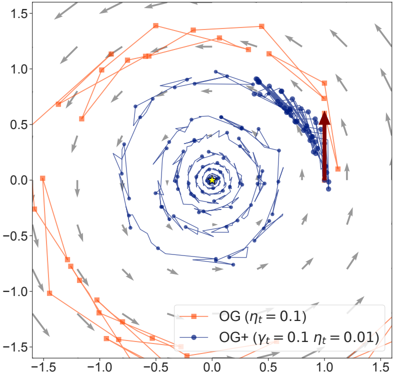
$$\mathbf{X}_{t+\frac{1}{2}} = \mathbf{X}_t - \boxed{\gamma_t}\,\hat{\mathbf{V}}_{t-\frac{1}{2}}$$

$$\mathbf{X}_{t+1} = \mathbf{X}_t - \eta_t \hat{\mathbf{V}}_{t+\frac{1}{2}}$$

With $\gamma_t \geq \eta_t$

Solution

Scale separation of learning rates



Toward Robustness Against Noise

All the favorable guarantees break if feedback is noisy

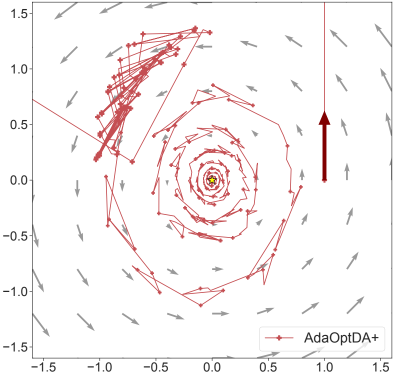
- AdaOptDA+ uses learning rates

$$\gamma_t^i = \frac{1}{\left(1 + \sum_{s=1}^{t-2} \|g_s^i\|^2\right)^{\boxed{\frac{1}{4}}}}$$

$$\eta_t^i = \frac{1}{\sqrt{1 + \sum_{s=1}^{t-2} \left(\|g_s^i\|^2 + \|X_s^i - X_{s+1}^i\|^2\right)}}$$

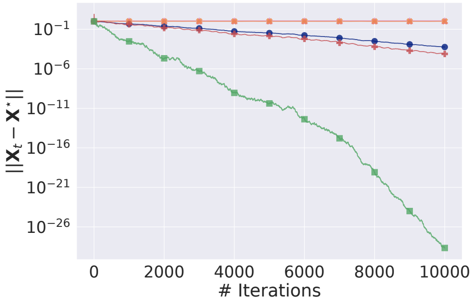
Solution

Scale separation of learning rates + Adaptivity

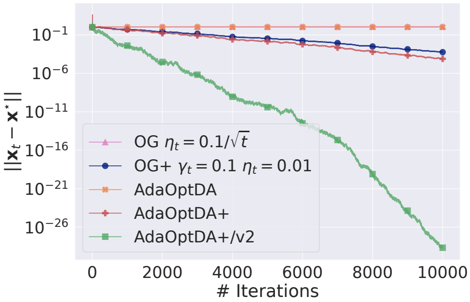


Convergence to Solution Under Multiplicative Noise

- $\hat{\mathbf{V}}_{t+\frac{1}{2}}$ is $(3\phi_{t+\frac{1}{2}}, -3\theta_{t+\frac{1}{2}})$ or $(-\phi_{t+\frac{1}{2}}, \theta_{t+\frac{1}{2}})$ with probability one half for each



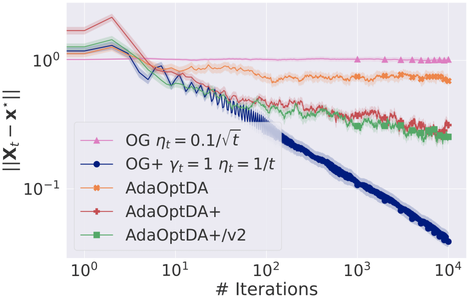
Base state $\mathbf{X}_t$



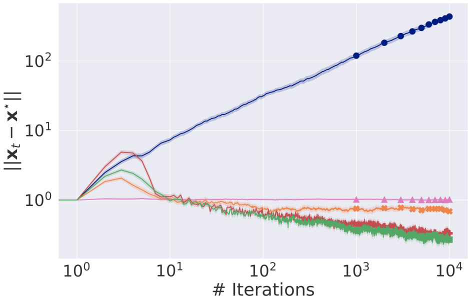
Played action $\mathbf{x}_t = \mathbf{X}_{t+\frac{1}{2}}$

Convergence to Solution Under Additive Noise

- $\hat{\mathbf{V}}_{t+\frac{1}{2}} = (\phi_{t+\frac{1}{2}} + \xi_t^1, -\theta_{t+\frac{1}{2}} + \xi_t^2)$ where $\xi_t^1, \xi_t^2 \sim \mathcal{N}(0,1)$



Base state $\mathbf{X}_t$



Played action $\mathbf{x}_t = \mathbf{X}_{t+\frac{1}{2}}$