Learning in Games: Robustness of Fast Convergence

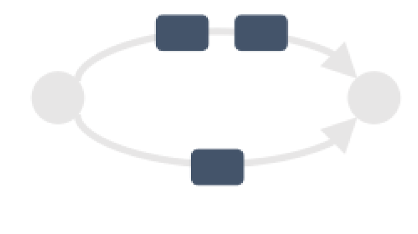
Dylan Foster, Zhiyuan Li, Thodoris Lykouris, Karthik Sridharan, and Éva Tardos

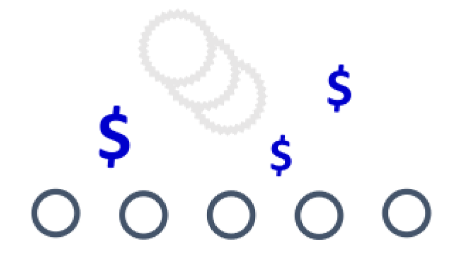
djfoster@cs.cornell.edu, lizhiyuan13@mails.tsinghua.edu.cn, teddylk@cs.cornell.edu, sridharan@cs.cornell.edu, eva@cs.cornell.edu

Cornell University

Overview

Repeated Games in Computer Systems


internet routing


advertising auctions

No-Regret learning converges to efficient outcomes:

$$\text{Cost} \leq \rho \cdot \text{Opt} + \text{Rate}.$$

- Standard algorithms: Rate $\approx O(1/\sqrt{T})$
- *Fast Convergence*: Rate $\approx O(1/T)$ [e.g. Syrgkanis et al.'15]

But seemingly only for restricted algorithms/settings.

Main result:

Fast Convergence of Learning is Robust!

via

Low Approximate Regret

$$\text{Cost} \leq (\rho + \epsilon) \cdot \text{Opt} + \text{ApxRate}.$$

- **Feedback**
  Prior work: Expectation feedback (information/computation issues).
  Our work: Realized (costs *given actions of the other players).*
- **Range of Algorithms**
  Prior work: Specialized algorithms.
  Our work: Broad class of standard algorithms.
- **Bandit Feedback**
  Fast convergence even when players only see loss of own action.
- **Dynamic Population Games**
  Fast convergence even when players come and go with time.

Prior work: [Syrgkanis-Agrawal-Luo-Schapire '15, Daskalakis-Deckelbaum-Kim '15, Rakhlin-Sridharan '13]

Repeated Game Model

- Normal form game with n players repeated for T time steps.
- Player i has action space S_i with $|S_i| = d$ and cost function $\textbf{cost}_i : S_1 \times \ldots \times S_n \to [0,1]$.
- At each time $t = 1, \ldots, T$, player i:
  1. Chooses action distribution $w_i^t \in \Delta(S_i)$ and samples action $s_i^t \sim w_i^t$.
  2. Observes cost vector $c_i^t = (\textbf{cost}_i(s, s_{-i}^t))_{s \in S_i}$ given other players' **sampled actions**. (**Realized Feedback**)

Define Social Cost: $\mathrm{SC}(s) = \sum_{i=1}^n \textbf{cost}_i(s)$. Measure performance with **Average Social Cost**: $\frac{1}{T}\sum_{t=1}^T \mathrm{SC}(s^t)$.

How does avg. cost approximate $\textsc{Opt} = \min_s \mathrm{SC}(s)$?

Efficiency of Low Approximate Regret

Definition: Low Approximate Regret

Player i satisfies the *Low Approximate Regret* property for parameter $\epsilon > 0$ and function $\textbf{ApxReg}(d,T)$ if for all action distributions $f \in \Delta(S_i)$,

$$(1-\epsilon)\sum_{t=1}^T \langle w_i^t, c_i^t\rangle \leq \sum_{t=1}^T \langle f, c_i^t\rangle + \frac{\textbf{ApxReg}(d,T)}{\epsilon}.$$

Multiplicative approximation to usual regret!

Definition: Smooth Games

A cost minimization game is (λ,μ)-smooth if for all strategy profiles s and s^*:

$$\sum_i \textbf{cost}_i(s_i^*, s_{-i}) \leq \lambda \cdot \textbf{cost}_i(s^*) + \mu \cdot \textbf{cost}_i(s).$$

- No-regret converges to approximately optimal social welfare $\frac{\lambda}{1-\mu}\textsc{Opt}$.
- Ex: atomic congestion games w/ affine cost functions are $(\frac{5}{3}, \frac{1}{3})$-smooth [Christodoulou-Koutsoupias'05] $\implies$ social cost within $2.5 \times \textsc{Opt}$.

Key Result: Efficiency of Low Approximate Regret

In any (λ,μ)-smooth game, if all players use Low Approximate Regret algorithms with parameters ϵ and $A(d,T)$, then

$$\frac{1}{T}\sum_t \mathbb{E}[\mathrm{SC}(s^t)] \leq \frac{\lambda}{1-\mu-\epsilon}\textsc{Opt} + \underbrace{\frac{n}{T}}_{O(1/T)\text{ rate}} \cdot \frac{1}{1-\mu-\epsilon} \cdot \underbrace{\frac{\textbf{ApxReg}(d,T)}{\epsilon}}_{\text{Typically } \textbf{ApxReg}=O(\log d)}.$$

Also holds in high probability!

Proof: Let $\mathrm{SC}(s^\star) = \mathrm{OPT}$.

$$\frac{(1-\epsilon)}{T}\mathbb{E}[\mathrm{SC}(s^t)] = \mathbb{E}\left[\frac{(1-\epsilon)}{T}\sum_{i=1}^n\sum_{t=1}^T\langle w_i^t, c_i^t\rangle\right] \overset{\text{(L.A.R.)}}{\leq} \mathbb{E}\left[\frac{1}{T}\sum_{i=1}^n\sum_{t=1}^T\langle s_i^\star, c_i^t\rangle\right] + \frac{n}{T}\cdot\frac{\textbf{ApxReg}(d,T)}{\epsilon}$$

$$\overset{\text{(Smoothness)}}{\leq} \lambda\textsc{Opt} + \frac{\mu}{T}\mathbb{E}[\mathrm{SC}(s^t)] + \frac{n}{T}\cdot\frac{\textbf{ApxReg}(d,T)}{\epsilon}.$$ Now just rearrange!

Insight: Same trick works in robust range of settings, e.g. bandit, dynamic.

Low Approximate Regret Algorithms

Many algorithms have Low Approximate Regret!

Hedge/Multiplicative Weights

With learning rate η, standard mirror descent analysis gives:

$$\sum_{t=1}^T\langle w^t - f, c^t\rangle \leq \eta\sum_{t=1}^T\langle f, c^t\rangle + \frac{\log d}{\eta} \quad \forall f \in \Delta(S_i).$$

So Hedge has Low Approximate Regret with $\textbf{ApxReg}(d,T) = \log d$, $\epsilon = \eta$.

Small-Loss Bounds

$$\textbf{Reg}_T \leq \sqrt{(\text{Learner's cost})\cdot\textbf{ApxReg}} \quad \text{or} \quad \textbf{Reg}_T \leq \sqrt{(\text{Cost of best action})\cdot\textbf{ApxReg}}.$$

implies low approximate regret for all $\epsilon > 0$ simultaneously via $\sqrt{ab} \propto \inf_{\epsilon>0}[\epsilon\cdot a + b/\epsilon]$.

Specific examples: (achieve any ϵ by setting learning rate.)

- Small-loss/self-confident algorithms: [Freund,Schapire'97], [Auer,Cesa-Bianchi,Gentile'02], [Yaroshinsky,El-Yaniv,Seiden'04]
- Algorithm from [Cesa-Bianchi-Mansour-Stoltz'07]
- Variation MW [Hazan-Kale'10]
- AEG-Path [Steinhardt-Liang'14]
- AdaNormalHedge [Luo-Schapire'15]
- Squint [Koolen-van Erven'15]

Revisiting Optimistic Hedge

[Syrgkanis-Agarwal-Luo-Schapire'15]: **Optimistic Hedge** has $O(1/T)$ convergence to $\frac{\lambda}{1-\mu}\textsc{Opt}$ with expectation feedback.
Lemma: Optimistic Hedge has $O(1/T)$ convergence to $\frac{\lambda}{1-\mu-\epsilon}\textsc{Opt}$ with **realized feedback**.

$\implies$ **Strengthens Low Approximate Regret as a behavioral assumption!**

Bandit Feedback

Bandit feedback: Player i sees only $\textbf{cost}_i(s_i^t, s_{-i}^t)$, not whole vector $(\textbf{cost}_i(s, s_{-i}^t))_{s\in S_i}$.

If Low Approximate Regret is satisfied (in expectation) under bandit feedback, same efficiency guarantee as in full information setting.

Bandit Algorithms with Low Approximate Regret

New Algorithm: Initialize w^1 to the uniform distribution. On each round t, perform update:

$$\mathbf{w}^t_{\mathbf{s}^{t-1}} = \frac{\mathbf{w}^{t-1}_{\mathbf{s}^{t-1}}}{1+\eta\mathbf{c}^t_{\mathbf{s}^{t-1}}+\gamma\mathbf{w}^{t-1}_{\mathbf{s}^{t-1}}} \quad \text{and} \quad \forall \mathbf{j} \neq \mathbf{s}^{t-1} \quad \mathbf{w}^t_{\mathbf{j}} = \frac{\mathbf{w}^{t-1}_{\mathbf{j}}}{1+\gamma\mathbf{w}^{t-1}_{\mathbf{j}}},$$

where $\gamma \leq 0$ is chosen so that w^t is a valid probability distribution.

- Achieves $A(d,T) = d\log(T/d)$ (best possible d dependence) for both *costs* and *utilities*.
- Derivation: Mirror descent with log-barrier regularizer + importance sampling.

- Exp3Light [Stoltz'05]: $\textbf{ApxReg}(d,T) = d^2\log T$.
- SCriBLe [Abernethy,Hazan,Rakhlin'08,Rakhlin,Sridharan'13] has $\textbf{ApxReg}(d,T) = d^3\log T$.
- GREEN [Allenberg-Auer-Gyorfi-Ottucsak'06] has $\textbf{ApxReg}(d,T) = d\log T$ (only in cost setting).

General challenge in bandit setting: Can't assume oblivious adversary. On the bright side, we are fine with pseudo-regret.

Dynamic Population Games

Beyond static population

Definition: Dynamic Population Games (Lykouris-Syrgkanis-Tardos 2016)

Interpolation between stochastic and worst-case changes in population.

- At every round t, every player i is replaced with ***turnover probability*** p.
- The types of the replacing players are **adversarially** selected.

Behavioral assumption

Players encounter Low Approximate approximate Regret compared to a shifting benchmark.

Definition: Low Approximate Regret for shifting experts

Player i satisfies the *Low Approximate Regret for shifting experts* property for parameter $\epsilon > 0$ and function $A(d,T)$ if for all action distribution sequences $(f_i^1 \ldots f_i^T) \in \Delta(S_i)^T$ such that $\sum_{t=2}^T\{f_i^t \neq f_i^{t-1}\} \leq K_i$,

$$(1-\epsilon)\sum_{t=1}^T\langle w_i^t, c_i^t\rangle \leq \sum_{t=1}^T\langle f_i^t, c_i^t\rangle + (1+K_i)\frac{\textbf{ApxReg}(d,T)}{\epsilon}.$$

Efficiency of Low Approximate Regret with shifting experts in Dynamic Population Games

In any (λ,μ)-smooth game with a sequence $(s^{*1},\ldots s^{*T})$ of solutions that is *stable* ($\sum_{t=2}^T\|s^{*t} - s^{*,t-1}\| \leq K_i$) and *near-optimal* ($\beta$-approx.), if all players use Low Approximate Regret algorithms for shifting experts then

$$\frac{1}{T}\sum_t\mathbb{E}[\mathrm{SC}(s^t)] \leq \frac{\lambda\cdot\beta}{1-\mu-\epsilon}\cdot\frac{1}{T}\sum_{t=1}^T\textsc{Opt}(t) + \frac{\sum_i(1+\mathbb{E}[K_i])}{T}\cdot\frac{1}{1-\mu-\epsilon}\cdot\frac{\textbf{ApxReg}(d,T)}{\epsilon}.$$

Algorithms with Low Approximate Regret for shifting experts

Noisy Hedge: Hedge update that mixes $\theta = 1/T$ uniform noise at every round at every round.

- Achieves $\textbf{ApxReg}(d,T) = \log(dT)$.

AdaNormalHedge (Luo-Schapire '15) achieves $\textbf{ApxReg}(d,T) = \log(dT)$.

Proof for Noisy Hedge: Let f^t be the comparator at round t. At this round t, it holds:

$$\langle w^t, c^t\rangle - \langle f^t, c^t\rangle \leq \eta\langle f^t, c^t\rangle + \frac{KL(f^t|w^t) - KL(f^t|w^{t+1})}{\eta}$$

- Telescope over the parts where the comparator does not change.
- Incur a $\log(d)$ penalty from the KL divergence at the first round.
- Incur a $\log(d/\theta)$ from the KL divergence at every round that the comparator changes.

Comparison to Previous work

- Broader range of algorithms
  Lykouris-Syrgkanis-Tardos '16: Small adaptive regret (e.g. AdaNormalHedge)$\Rightarrow$ Typically requires high memory.
  Our work: Broader class that includes simple algorithms like Noisy Hedge.
- Improved rate of turnover