MIT 6.S890 — Topics in Multiagent Learning

Lecture 7
Learning in games: Bandit feedback

Instructor: Prof. Gabriele Farina

The mathematical abstraction of a repeated decision maker we considered so far assumes that the entire utility function—evaluated in the strategies of the other players in the environment—is given to the decision maker as feedback. This is a strong assumption, and in many cases, the decision maker only receives partial feedback. In this lecture, we consider the case where the decision maker receives feedback only on the utility of the chosen action. This is known as the *bandit* feedback setting.

1 Setup and general considerations

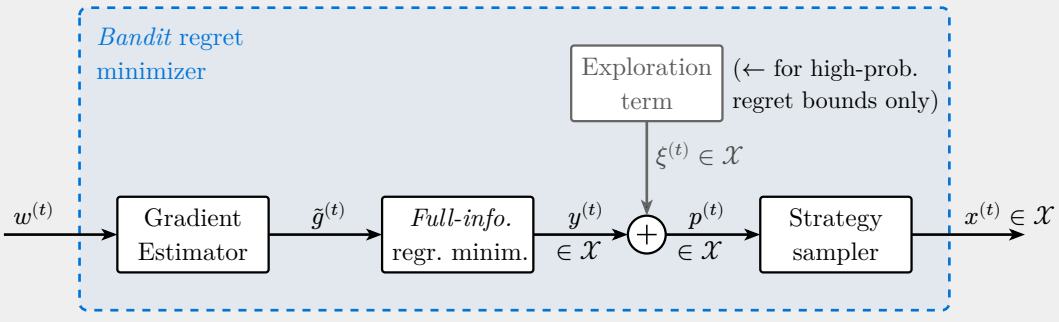
A bandit regret minimizer is identical to a full-information regret minimizer, except that the $\texttt{ObserveUtility}(u^{(t)})$ operation—where $u^{(t)} : x \mapsto \langle g^{(t)}, x \rangle$ is the utility function picked by the environment in the full-information case—is replaced with $\texttt{ObserveUtility}(w^{(t)})$, where $w^{(t)}$ is the scalar

$$w^{(t)} := \langle g^{(t)}, x^{(t)} \rangle \in \mathbb{R}.$$

As a general principle, algorithms for the *bandit* setting are constructed from regret minimizers for the full-information setting. Indeed, the key idea is to construct an *estimator* $\tilde{g}^{(t)}$ of the (unobserved) utility gradient $g^{(t)}$, and feed that into a full-information regret minimizer. The estimator is constructed from the observed utility $w^{(t)}$ and the chosen action $x^{(t)}$.

The utility function can still be picked adversarially by the environment. However, to get guarantees, it is necessary to reduce the power of the environment by letting the utility $u^{(t)}$ only depend on $x^{(1)}, ..., x^{(t-1)}$ but *not* on $x^{(t)}$. In other words, the environment can pick the utility adaptively, but has to decide the utility before the learner picks the strategy, and not after. This restriction still allows convergence to equilibria.

Example 1.1. Typically, the construction of bandit algorithms follows the following template.



The exploration term can be ignored if regret bounds *in expectation* are sought. Its role becomes important when *high-probability guarantees* are sought instead. We explain the difference next.

⋆These notes are class material that has not undergone formal peer review. The TAs and I are grateful for any reports of typos.

■ **Stochastic regret guarantees.** Because the estimation is stochastic, the regret of a bandit algorithm is a random variable. This adds a layer of complexity when approaching the analysis of bandit algorithms. As a rule of thumb, three "flavors" of guarantees tend to be considered in the literature. We will list them from weakest (and easiest to obtain) to strongest (and hardest to obtain):

- Guarantees on the *pseudoregret*, of the form

$$\mathrm{PseudoReg}^{(T)} := \max_{\hat{x}\in\mathcal{X}} \mathbb{E}\left[\sum_{t=1}^{T}\left\langle g^{(t)}, \hat{x}\right\rangle - \sum_{t=1}^{T}\left\langle g^{(t)}, x^{(t)}\right\rangle\right] = o(T).$$

- Guarantees on the *expected regret*, of the form

$$\mathbb{E}\left[\mathrm{Reg}^{(T)}\right] := \mathbb{E}\left[\max_{\hat{x}\in\mathcal{X}} \sum_{t=1}^{T}\left\langle g^{(t)}, \hat{x}\right\rangle - \sum_{t=1}^{T}\left\langle g^{(t)}, x^{(t)}\right\rangle\right] = o(T).$$

  Note the change of order between the expectation and the maximum compared with the pseudoregret introduced in the previous bullet point.

- *High-probability regret guarantees*, typically of the form

$$\mathbb{P}\left[\max_{\hat{x}\in\mathcal{X}} \sum_{t=1}^{T}\left\langle g^{(t)}, \hat{x}\right\rangle - \sum_{t=1}^{T}\left\langle g^{(t)}, x^{(t)}\right\rangle \leq o(T)\sqrt{\log\frac{1}{\delta}}\right] \geq 1-\delta$$

  for any $\delta > 0$ small enough.

Pseudoregret and expected regret guarantees are different, since $\max\mathbb{E} \leq \mathbb{E}\max$, but the converse is not true in general. Guarantees on the pseudoregret are *not* strong enough to conclude convergence to the set of equilibria, in general.

2 Adversarial bandit learning in normal-form games

Let's start from the case of normal-form games, in which our decision maker faces the choice of picking an action out of a finite set A. The setting in this case is also known as *adversarial multi-armed bandit problem*. We have $\mathcal{X} = \Delta(A)$.

■ **Strategy sampler.** In this settings, most algorithms use the natural strategy sampler: given a distribution $p^{(t)} \in \Delta(A)$, the decision maker samples an action $a^{(t)} \in A$ according to the probabilities in $p^{(t)}$. The vector $x^{(t)}$ is then set to the deterministic distribution $e_{a^{(t)}}$. Clearly, $\mathbb{E}_t\left[x^{(t)}\right] = p^{(t)}$.

■ **Gradient estimator.** For this setting, the standard gradient estimator is the *importance sampling* estimator. Given the utility scalar $w^{(t)} \in [0,1]$, the importance sampling estimator is defined as

$$\tilde{g}^{(t)} := \left(\frac{w^{(t)}}{p^{(t)}_{a^{(t)}}}\right) e_{a^{(t)}} \in \mathbb{R}^A.$$

Theorem 2.1. Let $w^{(t)} = \left\langle g^{(t)}, x^{(t)}\right\rangle$ where $g^{(t)}$ is some unknown utility gradient. Then, the importance sampling estimator $\tilde{g}^{(t)}$ is unbiased, that is, $\mathbb{E}_t\left[\tilde{g}^{(t)}\right] = g^{(t)}$.

Proof. The result follows by direct calculation. The randomness is due to the sampling of the action $a^{(t)}$. Each action $a \in A$ is sampled with probability $p^{(t)}_a$. Hence,

$$\mathbb{E}_t\left[\tilde{g}^{(t)}\right] = \sum_{a\in A} p^{(t)}_a \left(\frac{w^{(t)}}{p^{(t)}_a}\right) e_a = \sum_{a\in A} p^{(t)}_a \left(\frac{\left\langle g^{(t)}, e_{a^{(t)}}\right\rangle}{p^{(t)}_a}\right) e_a = \sum_{a\in A} g^{(t)}_a e_a = g^{(t)},$$

as we wanted to show. □

2.1 The Exp3 algorithm

The Exp3 (short for "exponential weights for exploration and exploitation") algorithm, introduced by Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. E. [Aue+02], applies the multiplicative weights update (MWU) algorithm on the importance sampling estimator. No exploration term is added, so that the deterministic strategy $x^{(t)}$ is sampled from the $y^{(t)}$ output by MWU directly (see also Example 1.1).

It is important to note that the analysis of MWU we did in Lectures 5 and 6 does not apply well to analyze the regret incurred by the full-information regret minimizer. The issue is that the estimated utilities potentially have a large range due to the division by the probabilities $p_a^{(t)}$. However, a better analysis of MWU in this case is possible.

Theorem 2.2. If the regret minimizer is set to MWU with learning rate $\eta = \sqrt{\log|A|/(T|A|)}$, the Exp3 algorithm guarantees *pseudoregret*

$$\text{PseudoReg}^{(T)} = O\Big(\sqrt{T|A|\log|A|}\Big).$$

2.2 Tsallis entropy

It can be shown that, information theoretically, no bandit learning algorithm for a finite set of actions $|A|$ can achieve better than $\Omega\big(\sqrt{T\ |A|}\big)$ expected regret in general. The regret guaranteed by the Exp3 algorithm is therefore optimal only up to a logarithmic factor. It remained open for a long time whether this logarithmic factor could be removed. A positive answer was given recently by Audibert, J.-Y., & Bubeck, S. [AB10], who proposed the idea of replacing the MWU algorithm with the FTRL algorithm instantiated with the (1/2)-Tsallis entropy regularizer

$$\psi(x) = 2 - 2\sum_{a\in A}\sqrt{x_a}.$$

Theorem 2.3. If the regret minimizer is set to FTRL algorithm with (1/2)-Tsallis entropy and learning rate $\eta = \sqrt{1/T}$, the resulting bandit algorithm guarantees pseudoregret

$$\text{PseudoReg}^{(T)} = O\Big(\sqrt{T|A|}\Big),$$

which is the optimal bound for bandit learning on finite probability distributions.

A simplified analysis can also be found in [ZS21].

2.3 The Exp3.P algorithm

The Exp3.P algorithm, introduced by Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. E. [Aue+02], is a variant of the Exp3 algorithm to achieve high-probability regret guarantees. Intuitively, the difficulty with getting high-probabilty bounds for the regret in Exp3 is due to the importance sampling: the gradient estimator has entries of magnitude inversely proportional to the prbabilities output by MWU. This makes the variance of the estimator large, and the concentration of the regret around its expectation difficult. To sidestep the issue, the Exp3.P algorithm uses the idea of superimposing a *uniform exploration term* to the output $y^{(t)}$ of MWU. More specifically, the input to the strategy sampler is set to

$$p^{(t)} := (1-\gamma)y^{(t)} + \gamma\frac{\mathbf{1}}{|A|} \in \Delta(A),$$

where $\gamma \in [0,1]$ is a parameter.

The exploration term increases the exploration of the algorithm, reducing the variance of the estimator. However, it is important to observe that this correction incurs a regret penalty due to the fact that MWU recommended $y^{(t)}$, and yet the decision maker sampled from $p^{(t)}$. The effect of such misalignment grows with the exploration parameter γ. Nonetheless, the following can be shown.

Theorem 2.4 ([AR09]). Consider the Exp3.P algorithm with exploration parameter $\gamma = \sqrt{|A|/T}$ and learning rate $\eta = \sqrt{\log|A|/(T|A|)}$. Then, for any $\delta \in (0,1)$,

$$\mathbb{P}\left[\mathrm{Reg}^{(T)} \leq O\left(\sqrt{T|A|\log\frac{|A|}{\delta}}\right)\right] \geq 1-\delta.$$

3 Adversarial bandit learning on more general convex domains

Today, we know that bandit optimization is possible well past probability simplexes. In fact, we can construct bandit algorithms for any convex and compact domain $\mathcal{X} \subseteq \mathbb{R}^d$. In particular, we mention the general general result by Abernethy, J. D., Hazan, E., & Rakhlin, A. [AHR08], who showed that the a bandit algorithm can be constructed starting from a full-information regret miminizer build using the FTRL algorithm with a self-concordant distance-generating function.

Bibliography

[Aue+02] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM journal on computing*, vol. 32, no. 1, pp. 48–77, 2002.

[AB10] J.-Y. Audibert and S. Bubeck, "Regret bounds and minimax policies under partial monitoring," *The Journal of Machine Learning Research*, vol. 11, pp. 2785–2836, 2010.

[ZS21] J. Zimmert and Y. Seldin, "Tsallis-INF: An optimal algorithm for stochastic and adversarial bandits," *Journal of Machine Learning Research*, vol. 22, no. 28, pp. 1–49, 2021.

[AR09] J. Abernethy and A. Rakhlin, "Beating the Adaptive Bandit with High Probability," Jan. 2009. [Online]. Available: `http://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-10.html`

[AHR08] J. D. Abernethy, E. Hazan, and A. Rakhlin, "Competing in the Dark: An Efficient Algorithm for Bandit Linear Optimization.," in *COLT*, 2008, pp. 263–274.

Changelog

- Sep 27: fixed a typo in Thereom 2.3.

6.S890 Topics in Multiagent Learning

Fall 2024