MIT 6.7980 — Topics in Multiagent Learning Thu, Oct 1, 2026

Lecture 6

Bandit feedback

Instructor: Prof. Constantinos Daskalakis (costis@mit.edu)

The mathematical abstraction of a sequential decision-maker we considered in previous lectures assumes that the learner receives enough information from the environment that she can compute her utility not only on the strategy she selected to play at each round of the interaction but also any counter-factual strategy that she could have played at that round. That is, we assume *full-information* 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 strategy they played. This is known as *bandit feedback*.

L6.1 Setup and general considerations

Like in our prior lectures, we will study *linear* settings, where our sequential decision-maker chooses a strategy $x^{(t)} \in X$ in some strategy set satisfying $X \subseteq \mathbb{R}^n$. In the full-information setting that we have already studied, the feedback that our decision maker receives, at every round t, is a utility function $u^{(t)} : x \mapsto \langle g^{(t)}, x \rangle$, from which they can compute their realized utility, $w^{(t)} := \langle g^{(t)}, x^{(t)} \rangle$, for the strategy they played as well as the counterfactual utility they would have received from any strategy they could have played. In the *bandit* setting, the decision-maker only receives as feedback their realized utility $w^{(t)}$ as opposed to their complete utility function $u^{(t)}$.

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 strategy $x^{(t)}$.

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 must decide the utility *before* the learner picks the strategy, and not after. This restriction still allows convergence to equilibria if bandit algorithms are used by players to iteratively update their strategies in games.

Typically, the construction of bandit algorithms follows the template shown in Figure 1.

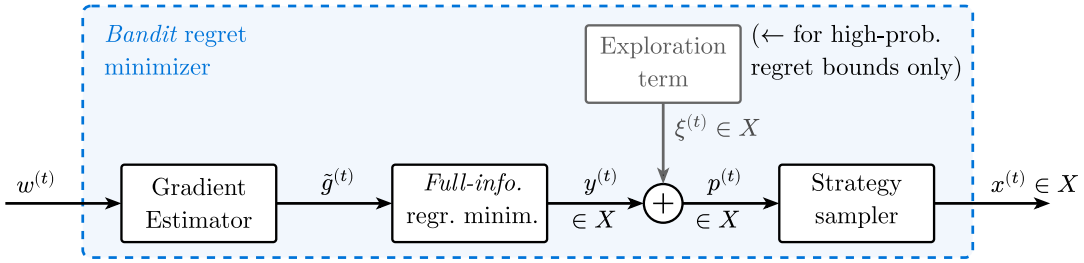

Figure 1: General template for bandit learning algorithms. X is the space of strategies of the learner. The output of the "strategy sampler" is a strategy from a restricted set of strategies. The name "strategy sampler" is generally a misnomer, but it is fitting in the widely-studied setting where $X = \Delta(A)$ for some finite set of actions A. In this case, a common instantiation of the strategy sampler is to take as input a distribution $p^{(t)} \in \Delta(A)$ and sample an action $a^{(t)} \sim p^{(t)}$. In this case, $x^{(t)}$ would be a single-atom distribution with an atom at $a^{(t)}$. But, in general, we allow for more general X's and more general samplers.

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.

■ **Stochastic regret guarantees.** Because online learning algorithms benefit from randomization, as is crucially the case in bandit settings, 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 are typically considered in the literature. We list them from the weakest (and easiest to obtain) to the strongest (and hardest to obtain):

- Guarantees on the *pseudoregret*, namely guarantees of the following form:

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

- Guarantees on the *expected regret*, namely guarantees of the following form:

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

  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*, namely guarantees of the following form:

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

  for any $\delta > 0$ small enough.

To make sense of the measures with respect to which the expectations and probabilities are computed in the above definitions, consider a randomized algorithm for the learner that

takes as input the history, $\left(x^{(\tau)}, w^{(\tau)}\right)_{\tau<t}$, observable to the learner so far and produces a strategy $x^{(t)}$ and, similarly, a randomized algorithm for the adversary that takes as input the history, $\left(x^{(\tau)}, u^{(\tau)}\right)_{\tau<t-1}$, observable to the adversary so far and chooses a loss function $u^{(t)}$. Pitting the two algorithms against each other defines a probability measure with respect to which the above expectations and probabilities are defined.

Finally, notice that Pseudoregret and expected regret guarantees are different, since $\max \mathbb{E} \leq \mathbb{E} \max$, but the converse is not true in general. This means that bounding expected regret automatically bounds the pseudoregret but the opposite is not necessarily the case. In fact, bounds on the pseudoregret are *not* strong enough to conclude convergence to the set of equilibria, in general.

L6.2 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 $X = \Delta(A)$.

■ **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 L6.1. 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,

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

□

L6.2.1 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 Figure 1).

It is important to note that the analysis of MWU we did in Lecture 5 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 L6.2. If the regret minimizer is set to MWU with learning rate $\eta = \sqrt{\log|A|/(T|A|)}$, the Exp3 algorithm guarantees pseudoregret

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

L6.2.2 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 negative (1/2)-Tsallis entropy regularizer

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

Theorem L6.3. 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].

L6.2.3 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 probabilities 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 L6.4 ([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 $\delta \in (0,1)$,

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

L6.3 Adversarial bandit learning on 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 $X \subseteq \mathbb{R}^d$. 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 built 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," technical report UCB/EECS-2009–10, Jan. 2009. [Online]. Available: https://digicoll.lib.berkeley.edu/record/138169

[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.