27 Exp3 for Adversarial Linear Bandits

The model for adversarial linear bandits is as follows. The learner is given an action set $\mathcal{A} \subset \mathbb{R}^d$ and the number of rounds n. An instance of the adversarial problem is a sequence of loss vectors $y_1, \dots, y_n$ where $y_t \in \mathbb{R}^d$ for each t. As usual in the adversarial setting, it is convenient to switch to losses. In each round t the learner selects an action $A_t \in \mathcal{A}$ and observes a loss $Y_t = \langle A_t, y_t \rangle$. The learner does not observe the loss vector y_t (if the loss vector is observed, then we call it the **full information setting**, but this is a topic for another book). Our standing assumption will be that the scalar loss for any of the action is in $[-1, 1]$, which corresponds to assuming that y_t is chosen from the polar of $\mathcal{A}$. For the rest of this chapter we assume that $y_t \in \mathcal{A}^\circ$ for all t. We furthermore assume that $\mathcal{A}$ spans $\mathbb{R}^d$. The latter of these assumptions is for convenience only and may be relaxed with just a little care (Exercise 27.6). The regret of the learner after n rounds is

$$R_n = \mathbb{E}\left[\sum_{t=1}^n Y_t\right] - \min_{a \in \mathcal{A}} \sum_{t=1}^n \langle a, y_t \rangle .$$

Clearly, the finite-armed adversarial bandits discussed in Chapter 11 is a special case of adversarial linear bandits corresponding to the choice $\mathcal{A} = \{e_1, \dots, e_d\}$ where $e_1, \dots, e_d$ are the unit vectors of the d-dimensional standard Euclidean basis.

27.1 Exponential weights for linear bandits

We adapt the exponential weights algorithm of Chapter 11. Like in that setting we need a way to estimate the individual losses for each action, but now we make use of the linear structure to share information between the arms and decrease the variance of our estimators. For now we assume that $\mathcal{A}$ is finite, which we relax in Section 27.3. Let $t \in [n]$ be the index of the current round. Assuming the loss estimate for action $a \in \mathcal{A}$ in round $s \in [n]$ is $\hat{Y}_s(a)$, then the distribution proposed by exponential weights is $\tilde{P}_t : \mathcal{A} \to [0, 1]$ given by

$$\tilde{P}_t(a) \propto \exp\left(-\eta \sum_{s=1}^{t-1} \hat{Y}_s(a)\right) ,$$

where $\eta > 0$ is the learning rate. To control the variance of the loss estimates, it will be useful to mix this distribution with an exploration distribution $\pi : \mathcal{A} \to [0,1]$ with $\sum_{a \in \mathcal{A}} \pi(a) = 1$. The mixture distribution is

$$P_t(a) = (1-\gamma)\tilde{P}_t(a) + \gamma\pi(a)\,,$$

where γ is a constant mixing factor to be chosen later. The algorithm then simply samples its action A_t from P_t:

$$A_t \sim P_t\,.$$

Recall that $Y_t = \langle A_t, y_t\rangle$ is the observed loss after taking action A_t. We need a way to estimate $y_t(a) = \langle a, y_t\rangle$. The idea is to use least squares to estimate y_t with $\hat{Y}_t = R_t A_t Y_t$ where $R_t \in \mathbb{R}^{d\times d}$ is selected so that $\hat{Y}_t$ is an unbiased estimate of y_t given the history. Then the loss for a given action is estimated by $\hat{Y}_t(a) = \langle a, \hat{Y}_t\rangle$. To find the choice of R_t that makes $\hat{Y}_t$ unbiased let $\mathbb{E}_t[\cdot] = \mathbb{E}\left[\cdot | A_1, \dots, A_{t-1}\right]$ and calculate

$$\mathbb{E}_t[\hat{Y}_t] = R_t \mathbb{E}_t[A_t A_t^\top] y_t = R_t \underbrace{\left(\sum_a P_t(a) a a^\top\right)}_{Q_t} y_t\,.$$

Using $R_t = Q_t^{-1}$ leads to $\mathbb{E}_t[\hat{Y}_t] = y_t$ as desired. Of course Q_t should be non-singular, which will follow by choosing π so that

$$Q(\pi) = \sum_{a \in \mathcal{A}} \pi(a) a a^\top$$

is non-singular. The complete algorithm is summarized in Algorithm 14.

```
Input  Action set $\mathcal{A} \subset \mathbb{R}^d$, learning rate $\eta$, exploration distribution $\pi$,
       exploration parameter $\gamma$
for $t = 1, 2, \dots, n$ do
    Compute sampling distribution:
        $P_t(a) = \gamma\pi(a) + (1-\gamma)\frac{\exp\left(-\eta \sum_{s=1}^{t-1} \hat{Y}_s(a)\right)}{\sum_{a' \in \mathcal{A}} \exp\left(-\eta \sum_{s=1}^{t-1} \hat{Y}_s(a')\right)}\,.$
    Sample action:
        $A_t \sim P_t\,.$
    Observe loss $Y_t = \langle A_t, y_t\rangle$ and compute loss estimates:
        $\hat{Y}_t = Q_t^{-1} A_t Y_t$     and     $\hat{Y}_t(a) = \langle a, \hat{Y}_t\rangle\,.$
end for
```

Algorithm 14: Exp3 for Linear Bandits

27.2 Regret analysis

THEOREM 27.1 *Assume that* $\operatorname{span}(\mathcal{A}) = \mathbb{R}^d$. *There exists an exploration distribution* π *and parameters* η *and* γ *such that for all* $(y_t)_t$ *with* $y_t \in \mathcal{A}^\circ$ *the regret of Algorithm 14 is at most* $R_n \leq 2\sqrt{3dn\log(K)}$.

Proof Assume that the learning rate η is chosen so that for each round t the loss estimates satisfy

$$\eta\hat{Y}_t(a) \geq -1, \qquad \forall a \in \mathcal{A}\,. \tag{27.1}$$

Then by modifying the proof of Theorem 11.1 (see Exercise 27.1) the regret is bounded by

$$R_n \leq \frac{\log K}{\eta} + 2\gamma n + \eta \sum_t \mathbb{E}\left[\sum_a P_t(a)\hat{Y}_t^2(a)\right]\,. \tag{27.2}$$

Note that we cannot use the proof that leads to the tighter constant (η getting replaced by $\eta/2$ in the second term above) because there is no guarantee that the loss estimates will be upper bounded by one. To get a regret bound it remains to set γ and η so that (27.1) is satisfied and to bound $\mathbb{E}\left[\sum_a P_t(a)\hat{Y}_t^2(a)\right]$. We start with the latter. Let $M_t = \sum_a P_t(a)\hat{Y}_t^2(a)$. By the definition of the loss estimate,

$$\hat{Y}_t^2(a) = (a^\top Q_t^{-1}A_tY_t)^2 = Y_t^2 A_t^\top Q_t^{-1}aa^\top Q_t^{-1}A_t\,,$$

which means that $M_t = \sum_a P_t(a)\hat{Y}_t^2(a) = Y_t^2A_t^\top Q_t^{-1}A_t \leq A_t^\top Q_t^{-1}A_t$ and

$$\mathbb{E}_t[M_t] \leq \operatorname{trace}\left(\sum_a P_t(a)aa^\top Q_t^{-1}\right) = d\,.$$

It remains to choose γ and η. Strengthen (27.1) to $|\eta\hat{Y}_t(a)| \leq 1$ and note that since $|Y_t| \leq 1$,

$$|\eta\hat{Y}_t(a)| = |\eta a^\top Q_t^{-1}A_tY_t| \leq \eta|a^\top Q_t^{-1}A_t|\,.$$

Let $Q(\pi) = \sum_{\nu\in\mathcal{A}} \pi(\nu)\nu\nu^\top$. Clearly $Q_t \succeq \gamma Q(\pi)$ and hence $Q_t^{-1} \preceq Q(\pi)^{-1}/\gamma$ by Exercise 27.3. Using this and the Cauchy-Schwartz inequality shows that

$$|a^\top Q_t^{-1}A_t| \leq \|a\|_{Q_t^{-1}}\|A_t\|_{Q_t^{-1}} \leq \max_{\nu\in\mathcal{A}} \nu^\top Q_t^{-1}\nu \leq \frac{1}{\gamma}\max_{\nu\in\mathcal{A}} \nu^\top Q^{-1}(\pi)\nu\,,$$

which implies that

$$|\eta\hat{Y}_t(a)| \leq \frac{\eta}{\gamma}\max_{\nu\in\mathcal{A}} \nu^\top Q^{-1}(\pi)\nu = \frac{\eta}{\gamma}\max_{\nu\in\mathcal{A}} \|\nu\|^2_{Q^{-1}(\pi)}\,. \tag{27.3}$$

From Theorem 21.1 (Kiefer–Wolfowitz) we know that there exists a sampling distribution π such that $\max_{\nu\in\mathcal{A}}\|\nu\|^2_{Q^{-1}(\pi)} = d$. By choosing $\gamma = \eta d$ and plugging into (27.2) we get

$$R_n \leq \frac{\log K}{\eta} + 3\eta dn = 2\sqrt{3dn\log(K)}\,,$$

where the last equality is derived by choosing $\eta = \sqrt{\frac{\log(K)}{3dn}}$. □

27.3 Continuous exponential weights

As the number of arms becomes extremely large or infinite, the dependence on $\log(K)$ may be undesirable. Suppose that $\mathcal{A} \subset [-1,1]^d$ is a subset of the hypercube and K is extremely large. Letting $\varepsilon = 1/n$ define $\mathcal{A}' \subset \mathcal{A}$ to be the smallest set such that for all $x \in \mathcal{A}$ there exists a $y \in \mathcal{A}'$ with $|\langle x - y, u\rangle| \le \varepsilon$ for all $u \in \mathcal{A}^\circ$. That is, $\mathcal{A}'$ is an ε-accurate approximation (loss-wise) to $\mathcal{A}$. A standard calculation (cf. Exercise 27.5) shows that no matter how large is K, the set $\mathcal{A}'$ is guaranteed to satisfy $\log|\mathcal{A}'| = O(d\log n)$. Then it is easy to check that playing Exp3 on $\mathcal{A}'$ suffers regret at most $R_n = O(d\sqrt{n\log(n)})$. Notice that this even works when $|\mathcal{A}| = \infty$. The problem with this approach is that $\mathcal{A}'$ is still exponentially large in d, which quickly renders this algorithm impossible to use as d and n get larger. When $\mathcal{A}$ is itself a convex set, then a more computationally tractable approach is to switch to the **continuous exponential weights** algorithm, which is the topic of this section. Hence, from now on we assume that $\mathcal{A}$ is a convex set with positive Lebesgue measure. Again, the condition that $\mathcal{A}$ has a positive Lebesgue measure can be relaxed with some care.

Let π be a probability measure supported on $\mathcal{A}$. The continuous exponential weights policy samples A_t from $P_t = (1-\gamma)\tilde{P}_t + \gamma\pi$. That is, P_t is the measure defined via $P_t(A) = (1-\gamma)\tilde{P}_t(A) + \gamma\pi(A)$ for $A \in \mathfrak{B}(\mathbb{R}^d)$, while $\tilde{P}_t$ is a measure supported on $\mathcal{A}$ defined by

$$\tilde{P}_t(A) = \frac{\int_A \exp\left(-\eta\sum_{s=1}^{t-1}\hat{Y}_s(a)\right)da}{\int_{\mathcal{A}} \exp\left(-\eta\sum_{s=1}^{t-1}\hat{Y}_s(a)\right)da}\,. \tag{27.4}$$

We will shortly see that the analysis in the previous section can be copied almost verbatim to prove a regret bound for this strategy. But what has been bought here? Rather than sampling from a discrete distribution on a large number of arms we now have to sample from a probability distribution on a convex set. Sampling from arbitrary probability distributions is itself a challenging problem, but under certain conditions there are polynomial time algorithms for this problem. The factors that play the biggest role in the feasibility of sampling from a distribution are *(a)* what is the form of the distribution and *(b)* how is the convex set represented. As it happens the measure defined in the last display is **log-concave**, which means that the logarithm of the density, with respect to the Lebesgue-measure on $\mathcal{A}$, is a concave function (in this case it is even a linear function). Suppose that $p_t(a) \propto \mathbb{I}_{\mathcal{A}}(a)\exp(-f(a))$ is a density with respect to the Lebesgue measure on $\mathcal{A}$, then there exists a polynomial-time algorithm for sampling from p provided one can compute the following efficiently:

1 (first-order information): $\nabla f(a)$ where $a \in \mathcal{A}$;

2 (projections): $\operatorname{argmin}_{x \in \mathcal{A}} \|x - y\|_2$ where $y \in \mathbb{R}^d$.

Clearly, the probability distribution defined by Eq. (27.4) satisfies the first condition. Efficiently computing a projection onto a convex set is a more delicate issue. A general criteria that makes this efficient is access to a **separation oracle**, which is a computational procedure to evaluate some function ϕ on $\mathbb{R}^n$ with $\phi(x) = \text{True}$ for $x \in \mathcal{A}$ and otherwise $\phi(x) = u$ with $\langle y, u\rangle > \langle x, u\rangle$ for all $y \in \mathcal{A}$. That is, the separation oracle accepts points in $\mathbb{R}^d$ as input and responds as output whether or not that point is inside the set and if it is not provides a separating hyperplane (see Fig. 27.1).

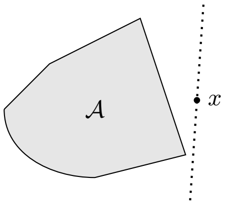

Figure 27.1 Separation oracle returns the normal of a hyperplane that separates x from $\mathcal{A}$ whenever $x \notin \mathcal{A}$. When $x \in \mathcal{A}$, the separation oracle returns True.

The analysis of the exponential weights algorithm goes through almost unchanged. By repeating the analysis in the previous section, but replacing sums with integrals one obtains the following bound on the regret:

THEOREM 27.2 *The regret of continuous exponential weights algorithm is bounded by*

$$R_n \le \frac{1}{\eta} \log \left(\frac{\operatorname{vol}(\mathcal{A})}{\int_{\mathcal{A}} \exp\left(-\eta \sum_{t=1}^n (\hat{Y}_t(a) - \hat{Y}_t(a^*))\right) da} \right) + \gamma n + \eta d n \,, \tag{27.5}$$

where $\operatorname{vol}(\mathcal{A}) = \int_{\mathcal{A}} da$ *is the volume of the action set* $\mathcal{A}$.

The term inside the logarithm is bounded using the following proposition, the proof of which we leave as an exercise to the reader.

PROPOSITION 27.1 *Let* $\mathcal{K} \subset \mathbb{R}^d$ *be a compact convex set with* $\operatorname{vol}(\mathcal{K}) > 0$ *and* $u \in \mathbb{R}^d$ *and* $x^* = \operatorname{argmin}_{x \in \mathcal{K}} \langle x, u\rangle$. *Then*

$$\log \left(\frac{\operatorname{vol}(\mathcal{K})}{\int_{\mathcal{K}} \exp\left(-\langle x^* - x, u\rangle\right) dx} \right) \le 1 + \max\left(0, d \log\left(\sup_{x,y \in \mathcal{K}} \langle x - y, u\rangle \right)\right) .$$

Substituting this result into Eq. (27.5) and choosing $\eta = \sqrt{\log(n)/n}$ leads to

$$R_n = O(d\sqrt{n \log(n)}) \,,$$

which matches the bound we got from the discretization approach.

27.4 Notes

1 A naive implementation of Algorithm 14 has computation complexity $O(Kd + d^3)$ per round. There is also the one-off cost of computing the exploration distribution, the complexity of which was discussed in Chapter 21. The real problem is that K can be extremely large. This is especially true when the action set is combinatorial. For example, when $\mathcal{A} = \{a \in \mathbb{R}^d : a_i = \pm 1\}$ is the corners of the hypercube $|\mathcal{A}| = 2^d$, which is much too large unless the dimension is small. Such problems call for a different approach that we present in the next chapter and in Chapter 30.

2 It is not important to find exactly the optimal exploration distribution. All that is needed is a bound on Eq. (27.3), which for the exploration distribution based on the Kiefer-Wolfowitz theorem is just d.

3 The $O(\sqrt{n})$ dependence of the regret on the horizon is not improvable, but the linear dependence on the dimension is suboptimal for certain action sets and optimal for others. An example where improvement is possible occurs when $\mathcal{A}$ is the unit ball, which is analyzed in the next chapter. Lower bounds are discussed in Chapter 29.

4 Like for stochastic bandits, Algorithm 14 and Theorem 27.1 can be generalized to the case when the action set is different in each round. The only adjustment to the algorithm is that now the exploration distribution must be recomputed in every round. The analysis goes through without change.

27.5 Bibliographic remarks

The results in Sections 27.1 and 27.2 follow the article by Bubeck et al. [2012] with minor modifications to make the argument more pedagogical. The main difference is that they used John's ellipsoid over the action set for exploration, which is only the 'right thing' when the John's ellipsoid is also a central ellipsoid. Here we use Kiefer–Wolfowitz, which is equivalent to finding the minimum volume central ellipsoid containing the action set as described in Chapter 21, where we also discuss the computation properties of finding the core set necessary to define the exploration distribution. A polynomial time sampling algorithm for convex sets with gradient information and projections is by Bubeck et al. [2015b]. We warn the reader that these algorithms are perhaps not the most practical, especially if theoretically justified parameters are used. The study of sampling from convex bodies is quite fascinating. There is a overview by Lovász and Vempala [2007], though it is a little old. Another path towards an efficient $O(d\sqrt{n}\log(\cdot))$ policy for convex action sets is to use the tools from online optimization. We explain these ideas in more detail in the next chapter, but the reader is referred to the paper by Bubeck and Eldan [2015]. The continuous exponential weights algorithm is perhaps attributable to Cover [1991] in the special setting of online learning called universal portfolio optimization. The first application to linear bandits is by Hazan et al. [2016]. Their algorithm and analysis is more complicated because they seek to improve the computation properties by replacing the exploration distribution based on Kiefer–Wolfowitz with an adaptive randomized exploration basis that can be computed in polynomial time under weaker assumptions. Continuous exponential weights for linear bandits using the core set of John's ellipsoid for exploration (rather than Kiefer–Wolfowitz) was recently analyzed by van der Hoeven et al. [2018].

27.6 Exercises

27.1 Prove Eq. (27.2).

27.2 Suppose that instead of assuming $y_t \in \mathcal{A}^\circ$ we assume that $y_t \in \{y \in \mathbb{R}^d : \sup_{a \in \mathcal{A}} |\langle a, y \rangle| \leq b\}$ for some known $b > 0$. Modify the algorithm to accommodate this change and explain how the regret guarantee changes.

27.3 Let $A, B \in \mathbb{R}^{d \times d}$ and suppose that $A \succeq B$ and B is invertible. Show that $A^{-1} \preceq B^{-1}$.

27.4 Now suppose that $a < b$ are known and $y_t \in \{y \in \mathbb{R}^d : \langle a, y \rangle \in [a, b] \text{ for all } a \in \mathcal{A}\}$. How can you adapt the algorithm now and what is its regret?

27.5 Let $\mathcal{A} \subset \mathbb{R}^d$ be bounded, $\|x\| = \sup_{u \in \mathcal{A}^\circ} |\langle x, u \rangle|$, where we also allow $\|x\| = \infty$. For $\mathcal{A}' \subset \mathcal{A}$ let $d(\mathcal{A}', \mathcal{A}) = \sup_{x \in \mathcal{A}} \inf_{y \in \mathcal{A}'} \|x - y\|$. Finally, for $\varepsilon > 0$, we let $N(\varepsilon, \mathcal{A})$ be the ε-covering number of $\mathcal{A}$, which is defined as in Definition 20.2 except that the Euclidean norm $\|\cdot\|_2$ used there is replaced by $\|\cdot\|$ defined above. Show that the following hold:

(a) $\|\cdot\|$ satisfies the triangle inequality, $\|0\| = 0$ and $\|cx\| = |c|\|x\|$ for any $x \in \mathbb{R}^d$ and $c \in \mathbb{R}$;

(b) Let $\mathcal{A}' \subset \mathcal{A}$ be finite. Show that $d(\mathcal{A}', \mathcal{A}) \leq \varepsilon$ if and only if $\mathcal{A}'$ is an ε-cover of $\mathcal{A}$;

(c) Let $B = \{z : \|z\| \leq 1\}$ and let $\mathcal{A}^\star = \mathcal{A} \cup -\mathcal{A}$ denote the reflection of $\mathcal{A}$ in the origin. Then, $\text{co}(\mathcal{A}^\star) \subset B \subset \text{span}(\mathcal{A})$ and $\{z \in \mathbb{R}^d : \|z\| < \infty\} = \text{span}(\mathcal{A})$;

(d) Let $p = \dim(\text{span}(\mathcal{A}))$. There exists a constant $c > 0$ that depends on $\mathcal{A}$ such that for any $\varepsilon \leq 1/2$ the inequality $\log N(\varepsilon, \mathcal{A}) \leq cp \log(1/\varepsilon)$ holds;

(e) For any $\varepsilon \leq 1/2$ there exists $\mathcal{A}'$ such that $d(\mathcal{A}', \mathcal{A}) \leq \varepsilon$ and $\log |\mathcal{A}'| \leq cp \log(1/\varepsilon) \leq cd \log(1/\varepsilon)$.

(**Hint:** You should be aware of Exercise 20.1.)

One can also show that there exists some constant $c > 0$ such that for all

$x \in \operatorname{span}(\mathcal{A})$, $\|x\| \leq c\|x\|_2$. This implies that $\|\cdot\|$, when restricted to $\operatorname{span}(\mathcal{A})$, is a norm.

27.6 In the definition of the algorithm and the proof of Theorem 27.1 we assumed that $\mathcal{A}$ spans $\mathbb{R}^d$. Show that this assumption may be relaxed by carefully adapting the algorithm and analysis.

27.7 We saw in Chapter 11 that the exponential weights algorithm achieved near-optimal regret without mixing additional exploration. Show that exploration is crucial here. More precisely, construct a finite action set $\mathcal{A}$ and reward sequence $y_t \in \mathcal{A}^\circ$ such that the regret of Algorithm 14 with $\gamma = 0$ leads to a very bad algorithm relative to the optimal choice.

27.8 Prove Theorem 27.2.

27.9 Prove Proposition 27.1.