FALL 2026 BAIS 7900

Lecture 9: Contextual Bandits

Lecturer: *Mengxiao Zhang* Week 9

1 Contextual Bandits

In previous lectures, we studied multi-armed bandits and compared the learner with the best fixed arm. However, in many applications, the learner observes information about the current decision before choosing an action. For example, a news platform may observe a user's interests, browsing history, and the features of the available articles before recommending one article. The platform then observes whether the user clicks on the recommended article, but not whether the user would have clicked on the other articles. We can express this feedback as a loss, with loss zero for a click and loss one otherwise.

The preferred article can differ across users. Consequently, always recommending the same article is a weak benchmark, even if that article has the highest average click rate. We would like to learn a rule that uses the observed information to choose an article. This gives the *contextual bandit* problem: the feedback is still the loss of one selected action, but the learner can use a context when making its decision.

1.1 Problem formulation

Denote by $\mathcal{X}$ the context space, which may be infinite, and by $[K] \triangleq \{1, \dots, K\}$ the action set, where $K \geq 2$. Fix a horizon $T > 2$. At each round $t = 1, \dots, T$,

1. the environment chooses a context $x_t \in \mathcal{X}$ and a loss vector $\ell_t \in [0,1]^K$;
2. the learner observes x_t and chooses a distribution $p_t \in \Delta_K$, where $\Delta_K \triangleq \{p \in \mathbb{R}^K : p_i \geq 0 \text{ for every } i \in [K],\ \sum_{i=1}^K p_i = 1\}$;
3. the learner draws $I_t \sim p_t$ and incurs and observes only ℓ_{t,I_t}.

Different from MAB where we compare with the performance of the best fixed action, we A *policy* is a function $\pi : \mathcal{X} \to [K]$ that specifies an action for every context. The learner is given a nonempty policy class Π and aims to compete with its best policy. As in our earlier bandit lectures, we measure performance by pseudo-regret, now defined by

$$\overline{\mathrm{Reg}}_T \triangleq \sup_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=1}^{T} \big(\ell_{t,I_t} - \ell_{t,\pi(x_t)}\big)\right]. \tag{1}$$

The comparator uses one fixed policy throughout the interaction, instead of a fixed action as what we defined in MAB. Its action may change because the context changes. The expectation includes the learner's randomization and any randomness in the environment. For a fixed oblivious sequence of contexts and losses and a finite policy class, this equals the expected regret against the best policy in hindsight.

Example 1.1. Suppose that there are two user types, each arriving with probability 1/2, and two article categories, sports and business. A sports-oriented user has losses $(0, 1)$, while a business-oriented user has losses $(1, 0)$. Every constant policy incurs expected cumulative loss $T/2$. A policy that recommends sports to the first type and business to the second incurs zero loss. Thus competing with the best fixed article and competing with the best fixed policy are different objectives.

The non-contextual problem is a special case. If Π consists of the K constant policies $\pi_i(x) = i$, then equation (1) is exactly the pseudo-regret against the best fixed arm. Contextual bandits allow a richer comparison class while keeping the same bandit feedback. We first impose no relation between the contexts and the losses. Later, we assume that a function from a known class predicts the conditional mean losses.

1.2 Why do direct reductions not suffice?

One approach is to maintain a separate K-armed bandit algorithm for every context and run the corresponding algorithm whenever that context appears. However, a large or infinite context space may produce a different context on every round. Each separate algorithm then receives only one observation. The policy class is what allows us to learn across different contexts: observing one user's response can help distinguish rules that also make recommendations to other users.

Another approach is to treat each policy as an arm. After selecting a policy π_t, we play $I_t = \pi_t(x_t)$ and use ℓ_{t,I_t} as that policy's observed loss. For a finite class of size $N \triangleq |\Pi|$, applying EXP3 from Lecture 6 gives a regret bound of $\mathcal{O}(\sqrt{NT \log N})$. This dependence on N is costly when the class contains many policies. Fortunately, the observation tells us the loss of every policy recommending the selected action, not just the policy we sampled. We next use this shared information to replace the polynomial dependence on N by a logarithmic dependence.

2 Exponential-Weight Algorithm for Exploration and Exploitation using Expert Advice (EXP4)

In Lecture 2, we used Hedge to compete with a finite set of experts. In Lecture 6, we adapted Hedge to bandit feedback through importance-weighted loss estimates and obtained EXP3. We now apply the same idea with policies as experts. The new step is to construct estimates using the probability of observing an action, so that policies recommending that action share the same feedback. This gives the **Exp**onential-weight algorithm for **Exp**loration and **Exp**loitation using **Exp**ert advice (EXP4), introduced by Auer et al. (2002).

2.1 From policy weights to action probabilities

Assume that Π is finite. The idea is the same as Hedge/EXP3, which is to maintain a weight $w_{t,\pi}$ for each policy and denote by $q_{t,\pi} \triangleq w_{t,\pi} / \sum_{\rho \in \Pi} w_{t,\rho}$ its normalized weight. After observing x_t, sample an action according to

$$p_{t,i} \triangleq \sum_{\pi \in \Pi : \pi(x_t) = i} q_{t,\pi}, \tag{2}$$

for every $i \in [K]$. Equivalently, we could draw a policy from q_t and follow its recommendation. The induced probability $p_{t,i}$ combines the weights of all policies that recommend action i.

After observing ℓ_{t,I_t}, define the action loss estimates by

$$\widehat{\ell}_{t,i} \triangleq \frac{\ell_{t,I_t}}{p_{t,I_t}} \mathbf{1}\{I_t = i\}. \tag{3}$$

Define $\widehat{g}_{t,\pi} \triangleq \widehat{\ell}_{t,\pi(x_t)}$ as the loss estimate supplied to policy π. Every policy recommending I_t receives the same loss estimate, while the other policies receive zero. The denominator is the probability of the sampled action, which is positive. An action recommended by no policy can have probability zero, and its estimate is automatically zero.

Denote by $\mathbb{E}_t[\cdot]$ conditional expectation given the information before drawing I_t, including the current context and sampling distribution. Since the contexts and losses are fixed in this section, $\mathbb{E}_t[\widehat{\ell}_{t,i}] = \ell_{t,i}$ whenever $p_{t,i} > 0$. Every policy has positive Hedge weight and therefore recommends an action with positive probability. Consequently, $\mathbb{E}_t[\widehat{g}_{t,\pi}] = \ell_{t,\pi(x_t)}$ for every policy π.

Why is the action probability important? If we instead sampled a policy and divided its observed loss by $q_{t,\pi}$, the estimate would use only the event that this particular policy was sampled. The observation is available whenever *any* policy recommending the same action is sampled. Using $p_{t,\pi(x_t)} \geq q_{t,\pi}$ uses all of those events and reduces the second moment of the estimate.

Algorithm 1: EXP4

Input: policy class Π, learning rate $\eta > 0$
Initialize $w_{1,\pi} = 1$ for every $\pi \in \Pi$;
for $t = 1, \ldots, T$ **do**
  observe x_t and set $q_{t,\pi} = w_{t,\pi} / \sum_{\rho \in \Pi} w_{t,\rho}$ for every $\pi \in \Pi$;
  set $p_{t,i} = \sum_{\pi \in \Pi : \pi(x_t) = i} q_{t,\pi}$ for every $i \in [K]$;
  draw $I_t \sim p_t$ and observe ℓ_{t,I_t};
  set $\widehat{\ell}_{t,i} = (\ell_{t,I_t}/p_{t,I_t})\mathbf{1}\{I_t = i\}$ for every $i \in [K]$;
  set $w_{t+1,\pi} = w_{t,\pi} \exp(-\eta \widehat{\ell}_{t,\pi(x_t)})$ for every $\pi \in \Pi$;

2.2 Regret guarantee

The proof uses the same nonnegative-loss Hedge inequality that we established for EXP3 in Lecture 6. Although importance-weighted estimates can exceed one, that inequality still applies: its exponential bound $e^{-x} \leq 1 - x + x^2$ requires only $x \geq 0$. The essential calculation is that the weighted second moment depends on the number of actions, even though Hedge maintains a distribution over policies.

Theorem 2.1. *Fix $K \geq 2$, $T > 2$, a policy class of size $N \geq 2$, and $\eta > 0$. For every fixed oblivious sequence of contexts and loss vectors in $[0,1]^K$, Algorithm 1 guarantees that $\overline{\mathrm{Reg}}_T \leq \log N/\eta + \eta KT$. In particular, the choice $\eta = \sqrt{\log N/(KT)}$ guarantees that*

$$\overline{\mathrm{Reg}}_T \leq \mathcal{O}\left(\sqrt{KT \log N}\right). \tag{4}$$

Proof. Fix a comparator policy $\pi \in \Pi$. Applying the nonnegative-loss Hedge inequality from Lecture 6 to q_t and $\widehat{g}_t$ gives

$$\sum_{t=1}^{T} \left(\sum_{\rho \in \Pi} q_{t,\rho} \widehat{g}_{t,\rho} - \widehat{g}_{t,\pi} \right) \leq \frac{\log N}{\eta} + \eta \sum_{t=1}^{T} \sum_{\rho \in \Pi} q_{t,\rho} \widehat{g}_{t,\rho}^2. \tag{5}$$

This is a pathwise inequality, so it applies to the estimates produced by the algorithm. Moreover, grouping policies by their recommendations gives the identity $\sum_{\rho\in\Pi} q_{t,\rho}\widehat{g}_{t,\rho} = \sum_{i=1}^{K} p_{t,i}\widehat{\ell}_{t,i} = \ell_{t,I_t}$.

To bound the quadratic term, we group the policies in the same way and obtain

$$\mathbb{E}_t\left[\sum_{\rho\in\Pi} q_{t,\rho}\widehat{g}_{t,\rho}^2\right] = \sum_{\substack{i\in[K]:\\ p_{t,i}>0}} \left(\sum_{\rho\in\Pi:\rho(x_t)=i} q_{t,\rho}\right)\frac{\ell_{t,i}^2}{p_{t,i}} = \sum_{\substack{i\in[K]:\\ p_{t,i}>0}} \ell_{t,i}^2 \le K.$$

The first equality uses $\mathbb{E}_t[\widehat{\ell}_{t,i}^2] = \ell_{t,i}^2/p_{t,i}$ on the support of p_t, and the second uses equation (2). This is exactly the probability cancellation in the EXP3 analysis, after combining all policies that recommend the same action.

Taking expectations in equation (5) and using the unbiasedness of the comparator's estimate yields

$$\mathbb{E}\left[\sum_{t=1}^{T}\left(\ell_{t,I_t} - \ell_{t,\pi(x_t)}\right)\right] \le \frac{\log N}{\eta} + \eta KT.$$

Taking the maximum over the fixed comparator policies proves the first claim. Substituting the stated learning rate proves equation (4). □

Hedge contributes the $\log N$ term, while the quadratic term depends on K because policies recommending the same action share the same estimate. This is the information that the policy-as-arm reduction failed to use. For the K constant policies, EXP4 reduces to EXP3. However, its implementation still maintains N weights and evaluates all N policies on every round. A logarithmic dependence on N in regret therefore does not give a logarithmic dependence on N in computation. We next introduce a regression oracle to avoid explicitly enumerating the policies, under an additional assumption on the mean losses.

3 Reduction to an Online Regression Oracle

The computational difficulty of EXP4 comes from its explicit distribution over policies. To address this difficulty, we now assume that a function from a known class predicts the conditional mean losses and that we can learn this function through an online regression oracle. The resulting algorithm, SquareCB of Foster and Rakhlin (2020), uses the oracle's current predictions to choose an exploration distribution. Its computation depends on the oracle and the number of actions, without an additional enumeration of the policy class.

3.1 The realizable setting

Denote by $\mathcal{F} \subseteq \{f : \mathcal{X}\times[K] \to [0,1]\}$ a known regression class. We assume that there exists a fixed unknown $f^\star \in \mathcal{F}$ such that

$$\mathbb{E}[\ell_{t,i} \mid \mathcal{H}_{t-1}, x_t] = f^\star(x_t, i), \tag{6}$$

for every $t \in [T]$ and $i \in [K]$, where $\mathcal{H}_{t-1}$ denotes the information available from earlier rounds, including the learner's internal state. This is the *realizability assumption*: the same function describes the mean loss of each action at every context encountered. Contexts may depend on

the history. Conditional on this information and the current context, the learner's current action draw is independent of the current loss vector.

Every $f \in \mathcal{F}$ induces a policy $\pi_f(x) \triangleq \arg\min_{i\in[K]} f(x,i)$, with ties broken by a fixed rule. In this section, the comparison class is $\Pi \triangleq \{\pi_f : f \in \mathcal{F}\}$. Denote by $\pi^\star \triangleq \pi_{f^\star}$ the policy induced by the true mean-loss function. This policy minimizes the conditional mean loss at every context. Therefore, equation (1) becomes

$$\overline{\mathrm{Reg}}_T = \mathbb{E}\left[\sum_{t=1}^{T}\big(f^\star(x_t, I_t) - f^\star(x_t, \pi^\star(x_t))\big)\right]. \tag{7}$$

The function class specifies a common prediction rule across contexts. It can be infinite; we will access it through the oracle instead of maintaining a weight for every function.

3.2 The online regression oracle

In Lecture 3, we studied online regression as an instance of online convex optimization. We now use a regression algorithm as a subroutine. At round t, an online square-loss regression oracle receives an instance $z_t \in \mathcal{X} \times [K]$, predicts $\widehat{y}_t \in [0,1]$, and then observes an outcome $y_t \in [0,1]$. We assume that, for every possibly adaptive sequence of instances and outcomes, it guarantees

$$\sum_{t=1}^{T}(\widehat{y}_t - y_t)^2 - \inf_{f\in\mathcal{F}}\sum_{t=1}^{T}(f(z_t) - y_t)^2 \le B(T), \tag{8}$$

where $B(T) > 0$ is a known upper bound on its square-loss regret. Thus the oracle competes with the best fixed regression function on the instances it receives. We are interested in $B(T) = o(T)$.

For contextual bandits, we query the oracle's prediction at (x_t, i) for every $i \in [K]$, all from the same current state and before making any update. Denote these predictions by $\widehat{y}_{t,i}$. We then choose p_t, draw $I_t \sim p_t$, and update the oracle once using the instance $z_t = (x_t, I_t)$ and outcome $y_t = \ell_{t,I_t}$. The actual regression sequence is the sequence of selected context-action pairs, so equation (8) applies to the predictions $\widehat{y}_{t,I_t}$.

To see what the oracle guarantee gives us, compare its square loss with that of $f^\star$. Expanding the squares and using realizability, we obtain

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\left((\widehat{y}_{t,I_t} - \ell_{t,I_t})^2 - (f^\star(x_t, I_t) - \ell_{t,I_t})^2\right)\right] \\
&\quad= \mathbb{E}\left[\sum_{t=1}^{T}\left((\widehat{y}_{t,I_t} - f^\star(x_t, I_t))^2 + 2(\widehat{y}_{t,I_t} - f^\star(x_t, I_t))(f^\star(x_t, I_t) - \ell_{t,I_t})\right)\right] \\
&\quad= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{K} p_{t,i}(\widehat{y}_{t,i} - f^\star(x_t, i))^2\right].
\end{aligned}$$

The cross term has expectation zero after conditioning on the information before the action draw and then on I_t. Since $f^\star \in \mathcal{F}$, equation (8) consequently implies that

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{K} p_{t,i}(\widehat{y}_{t,i} - f^\star(x_t, i))^2\right] \le B(T). \tag{9}$$

This bounds prediction error weighted by the probabilities of the actions we actually observe. It need not control the error on actions that are rarely selected. Choosing the smallest predicted loss on every round can therefore fail: an overestimated loss can prevent an action from being selected and corrected. We need an exploration distribution that makes the weighted prediction error in equation (9) sufficient to control regret.

3.3 Deriving the exploration distribution

Fix a prediction vector $\widehat{y} \in [0,1]^K$ and an unknown mean-loss vector $\mu \in [0,1]^K$. We would like a distribution p such that its excess mean loss is at most a multiple of $\sum_{i=1}^K p_i(\widehat{y}_i - \mu_i)^2$, plus a small exploration term. Denote by e_j the jth coordinate vector. For every positive distribution p, comparator action $j \in [K]$, and parameter $\gamma > 0$, completing the square gives

$$\begin{aligned}\langle p - e_j, \mu\rangle - \frac{\gamma}{4}\sum_{i=1}^K p_i(\widehat{y}_i - \mu_i)^2 &\le \langle p - e_j, \widehat{y}\rangle + \frac{1}{\gamma}\sum_{i=1}^K \frac{(p_i - \mathbf{1}\{i=j\})^2}{p_i} \\ &= \langle p - e_j, \widehat{y}\rangle + \frac{1}{\gamma}\left(\frac{1}{p_j} - 1\right). \end{aligned} \tag{10}$$

For the inequality, apply $uv - (\gamma p_i/4)v^2 \le u^2/(\gamma p_i)$ with $u = p_i - \mathbf{1}\{i=j\}$ and $v = \mu_i - \widehat{y}_i$, then sum over the coordinates. The remaining expression depends only on the predictions and p, so we can use it to design the sampling rule.

Denote by $b \triangleq \arg\min_{i\in[K]} \widehat{y}_i$ an action with the smallest predicted loss and by $d_i \triangleq \widehat{y}_i - \widehat{y}_b$ the predicted loss gap of action i. Then $\langle p - e_j, \widehat{y}\rangle = \sum_{i=1}^K p_i d_i - d_j$. For $j \ne b$, choosing $1/p_j = K + \gamma d_j$ cancels the term $-d_j$ in equation (10). This motivates the distribution

$$p_i \triangleq \begin{cases} \dfrac{1}{K + \gamma d_i}, & i \ne b, \\ 1 - \displaystyle\sum_{\substack{j\in[K]:\\ j\ne b}} \frac{1}{K + \gamma d_j}, & i = b. \end{cases} \tag{11}$$

Since $d_i \ge 0$, each non-minimizing action receives probability at most $1/K$, leaving $p_b \ge 1/K$. Actions with a small predicted gap receive more exploration probability, while actions with a large predicted gap receive less. The parameter γ controls how strongly the distribution concentrates around the smallest predicted loss.

The following bound verifies that this choice gives the required relation between regret and weighted prediction error.

Lemma 3.1. *Fix $K \ge 2$, $\gamma > 0$, and $\widehat{y} \in [0,1]^K$. Let p be the distribution in equation (11). For every $\mu \in [0,1]^K$ and $j \in [K]$, we have*

$$\langle p - e_j, \mu\rangle \le \frac{\gamma}{4}\sum_{i=1}^K p_i(\widehat{y}_i - \mu_i)^2 + \frac{2(K-1)}{\gamma}. \tag{12}$$

Proof. By equation (10), it suffices to bound $\sum_{i=1}^K p_i d_i - d_j + (1/p_j - 1)/\gamma$. The definition of p gives

$$\sum_{i=1}^K p_i d_i = \sum_{\substack{i\in[K]:\\ i\ne b}} \frac{d_i}{K + \gamma d_i} \le \frac{K-1}{\gamma}.$$

If $j \neq b$, substituting $1/p_j = K + \gamma d_j$ gives $-d_j + (1/p_j - 1)/\gamma = (K-1)/\gamma$. If $j = b$, then $d_j = 0$ and $p_b \geq 1/K$ give the same upper bound. Substituting these bounds into equation (10) proves equation (12). □

3.4 SquareCB and its guarantee

We now apply the distribution above to the oracle's current predictions. The algorithm requires K prediction queries, one regression update, and $\mathcal{O}(K)$ additional arithmetic operations per round. In particular, it does not require weights for all policies induced by $\mathcal{F}$.

Algorithm 2: SquareCB

Input: online square-loss regression oracle, exploration parameter $\gamma > 0$
for $t = 1, \ldots, T$ **do**
observe x_t;
query the oracle at (x_t, i) to obtain $\widehat{y}_{t,i}$ for every $i \in [K]$, without updating its state;
set $b_t = \arg\min_{i \in [K]} \widehat{y}_{t,i}$, using the fixed tie-breaking rule;
set $p_{t,i} = 1/(K + \gamma(\widehat{y}_{t,i} - \widehat{y}_{t,b_t}))$ for every $i \neq b_t$;
set $p_{t,b_t} = 1 - \sum_{i \in [K]: i \neq b_t} p_{t,i}$;
draw $I_t \sim p_t$ and observe ℓ_{t,I_t};
update the oracle using the instance (x_t, I_t) and outcome ℓ_{t,I_t};

Theorem 3.2. *Fix $K \geq 2$, $T > 2$, and $\gamma > 0$. Assume that equation (6) holds and that the oracle satisfies equation (8). Algorithm 2 guarantees that*

$$\overline{\mathrm{Reg}}_T \leq \mathcal{O}\left(\gamma B(T) + \frac{KT}{\gamma}\right). \tag{13}$$

In particular, choosing $\gamma = \sqrt{8(K-1)T/B(T)}$ guarantees that $\overline{\mathrm{Reg}}_T \leq \mathcal{O}(\sqrt{KTB(T)})$.

Proof. Apply Lemma 3.1 with $\widehat{y}_i = \widehat{y}_{t,i}$, $\mu_i = f^\star(x_t, i)$, and $j = \pi^\star(x_t)$. Conditional on the information before the action draw, its left-hand side is the expected excess mean loss on round t. Summing over the rounds, taking expectations, and using equation (7) gives

$$\overline{\mathrm{Reg}}_T \leq \frac{\gamma}{4}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{K} p_{t,i}(\widehat{y}_{t,i} - f^\star(x_t, i))^2\right] + \frac{2(K-1)T}{\gamma} \leq \frac{\gamma}{4}B(T) + \frac{2(K-1)T}{\gamma},$$

where the second inequality uses equation (9). This proves equation (13). Substituting the stated value of γ gives $\overline{\mathrm{Reg}}_T \leq \sqrt{2(K-1)TB(T)}$, which proves the final claim. □

EXP4 estimates the losses of all policies and then applies Hedge. SquareCB instead predicts the action losses and explores according to the predicted gaps. Realizability makes the oracle's excess square loss equal to prediction error in expectation, and Lemma 3.1 converts that error into bandit regret. Thus the reduction transfers an online regression guarantee into a contextual-bandit guarantee. Its computational efficiency depends on the availability of an efficient oracle with the assumed guarantee.

3.5 Connections to Hedge and online convex optimization

The reduction lets us use regression algorithms already related to our earlier lectures. We give three examples to make the dependence on $B(T)$ explicit.

Example 3.3. Suppose that $\mathcal{F}$ is finite, with $|\mathcal{F}| \geq 2$. Initialize every unnormalized weight to one, use learning rate $1/2$, and predict the weighted average $\widehat{y}_t = \sum_{f \in \mathcal{F}} v_{t,f} f(z_t)$, where $v_{t,f}$ are the normalized weights. After receiving (z_t, y_t), multiply each function's weight by $\exp(-(f(z_t) - y_t)^2/2)$. In Lecture 2, we established a generic $\mathcal{O}(\sqrt{T \log |\mathcal{F}|})$ regret bound for the weighted average of the experts' losses. Here, the learner incurs the square loss of its averaged numerical prediction. This additional structure gives a stronger guarantee, as we now show.

Denote by W_t the total unnormalized weight before round t, defined by

$$W_t \triangleq \sum_{f \in \mathcal{F}} \exp\left(-\frac{1}{2}\sum_{s=1}^{t-1}(f(z_s) - y_s)^2\right).$$

Thus $W_1 = |\mathcal{F}|$ and $v_{t,f} = \exp(-\frac{1}{2}\sum_{s=1}^{t-1}(f(z_s) - y_s)^2)/W_t$. For fixed $y \in [0,1]$, the function $u \mapsto \exp(-(u-y)^2/2)$ has second derivative $((u-y)^2 - 1)\exp(-(u-y)^2/2) \leq 0$ on $[0,1]$, so it is concave. Applying Jensen's inequality to the averaged prediction gives

$$\exp\left(-\frac{(\widehat{y}_t - y_t)^2}{2}\right) \geq \sum_{f \in \mathcal{F}} v_{t,f} \exp\left(-\frac{(f(z_t) - y_t)^2}{2}\right) = \frac{W_{t+1}}{W_t}.$$

Taking logarithms and summing over the rounds, we obtain

$$\sum_{t=1}^{T}(\widehat{y}_t - y_t)^2 \leq -2\sum_{t=1}^{T}\log\frac{W_{t+1}}{W_t} = -2\log\frac{W_{T+1}}{W_1} \leq \min_{f \in \mathcal{F}}\sum_{t=1}^{T}(f(z_t) - y_t)^2 + 2\log|\mathcal{F}|.$$

The equality telescopes the logarithms, and the last inequality uses $W_{T+1} \geq \exp(-\frac{1}{2}\sum_{t=1}^{T}(f(z_t) - y_t)^2)$ for every $f \in \mathcal{F}$. Therefore, the oracle satisfies equation (8) with $B(T) = 2\log|\mathcal{F}|$.

SquareCB consequently guarantees $\overline{\mathrm{Reg}}_T \leq \mathcal{O}(\sqrt{KT \log|\mathcal{F}|})$. Enumerating a finite regression class is itself costly; the reduction is computationally useful when the oracle has an efficient implementation.

Example 3.4. Suppose that the regression model has a convex parameter set of Euclidean diameter at most $D > 0$, and that each square loss is convex in the parameter with Euclidean gradient norm at most $G > 0$. In Lecture 3, we established that OGD with step size $D/(G\sqrt{T})$ guarantees square-loss regret at most $DG\sqrt{T}$. Using this algorithm as the regression oracle therefore gives $B(T) = DG\sqrt{T}$ and

$$\overline{\mathrm{Reg}}_T \leq \mathcal{O}\left(\sqrt{KDG}\,T^{3/4}\right).$$

Thus our earlier OCO guarantee already gives a sublinear contextual-bandit guarantee when K, D, G are fixed. Any improved online square-loss regret bound can be substituted into Theorem 3.2 directly; the exploration rule and its analysis do not change.

Example 3.5. Consider the stochastic K-armed bandit problem from Lecture 5, with independent and identically distributed loss vectors, mean losses $\mu_i \triangleq \mathbb{E}[\ell_{t,i}]$ and no context. Take $\mathcal{X} = \{x_0\}$ and $\mathcal{F} \triangleq \{f_\theta : \theta \in [0,1]^K\}$, where $f_\theta(x_0, i) \triangleq \theta_i$. The function f_μ satisfies realizability, and the induced policy class consists of the K constant-arm policies. Denote by $N_{t,i} \triangleq \sum_{s=1}^{t} \mathbf{1}\{I_s = i\}$

the number of observations of arm i through round t, with $N_{0,i} = 0$. Use the empirical-average predictions

$$\widehat{y}_{t,i} \triangleq \begin{cases} \dfrac{\sum_{s=1}^{t-1} \ell_{s,I_s} \mathbf{1}\{I_s = i\}}{N_{t-1,i}}, & N_{t-1,i} > 0, \\ \dfrac{1}{2}, & N_{t-1,i} = 0. \end{cases}$$

Thus each prediction averages only the previously observed losses of that arm. SquareCB uses these predictions in its sampling rule and updates only the selected arm's empirical average after observing its loss.

To bound the oracle's square-loss regret, fix an arm observed $m \geq 1$ times and denote by $y_1, \ldots, y_m$ its observed losses. Denote by $\bar{y}_s \triangleq s^{-1} \sum_{r=1}^{s} y_r$ the empirical mean after $s \geq 1$ observations, and set $\bar{y}_0 = 1/2$. This is the FTL predictor on this arm: before observation $s \geq 2$, $\bar{y}_{s-1}$ minimizes $\sum_{r=1}^{s-1}(u - y_r)^2$ over $u \in [0, 1]$. Updating the empirical mean gives

$$\sum_{r=1}^{s}(y_r - \bar{y}_s)^2 - \sum_{r=1}^{s-1}(y_r - \bar{y}_{s-1})^2 = \frac{s-1}{s}(y_s - \bar{y}_{s-1})^2,$$

for $s \geq 2$. Since $\bar{y}_m$ minimizes the square loss among all constant predictions in $[0, 1]$, summing these increments yields

$$\sum_{s=1}^{m} \left((\bar{y}_{s-1} - y_s)^2 - (\bar{y}_m - y_s)^2\right) = (y_1 - 1/2)^2 + \sum_{s=2}^{m} \frac{(y_s - \bar{y}_{s-1})^2}{s} \leq \sum_{s=2}^{m} \frac{1}{s} \leq \frac{1}{4} + \log m.$$

Summing the bound over all observed arms gives

$$\sum_{\substack{i \in [K]: \\ N_{T,i} > 0}} \left(\frac{1}{4} + \log N_{T,i}\right) \leq 2K \log T.$$

Consequently, these empirical averages satisfy equation (8) with $B(T) = 2K \log T$, even when the arm selections depend on previous observations. With $\gamma = 2\sqrt{T/\log T}$, Theorem 3.2 therefore gives

$$\overline{\mathrm{Reg}}_T \leq \mathcal{O}\left(K\sqrt{T \log T}\right).$$

This gives a stochastic MAB algorithm using empirical averages and randomized exploration. Specialized stochastic-bandit algorithms from Lecture 5 achieve a better dependence on K.

References

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multi-armed bandit problem. *SIAM Journal on Computing*, 2002.

Dylan J. Foster and Alexander Rakhlin. Beyond UCB: Optimal and efficient contextual bandits with regression oracles. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.