Online Learning

Pierre Gaillard

December 5, 2023

INRIA

References

These monographs are available online.

- Cesa-Bianchi and Lugosi, Prediction, learning, and games, 2006
- Shalev-Shwartz et al., “Online learning and online convex optimization”, 2012
- Hazan et al., “Introduction to online convex optimization”, 2016
- Lattimore and Szepesvári, “Bandit algorithms”, 2019

Introduction: What is online learning?

Online Linear Optimization

Online Convex Optimization

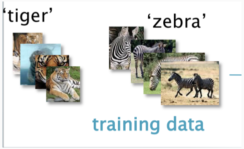
Adversarial bandits

Stochastic bandits

Classical Machine Learning

In classical supervised machine learning, the learner

1. observes training data with labels,
2. builds a program to minimize the training error
3. controls the error of new data if they are similar to the training data



→ Learning method → Prediction on test data

Sequential Learning

In some applications, the environment may evolve over time and the data may be available sequentially.

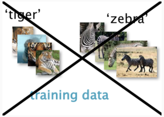
Spam detection: can be seen as a game between spammer and spam filters. Each trying to fool the other one. The data is possibly adversarial.

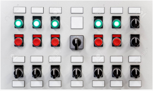
Necessity to take a robust approach by learning as ones goes along from experiences as more aspects of the problem are observed.

This is the goal of sequential learning (or sequential learning).

Sequential learning

In sequential learning, we do not have any training data.
Data are acquired and treated on the fly.
Feedbacks are received and algorithms updated step by step.



⇒  ⇒ zebra ⇒ Change parameters ⇒ ⇒ . . .

This field has received a lot of attention recently because of the possible applications coming from internet:

- ads to display,
- repeated auctions,
- spam detection,
- experts/algorithm aggregation

Setting of an online learning problem/online convex optimization

At each time step $t = 1, \ldots, T$

- the player observes a context $c_t \in \mathcal{X}$ (optional step)
- the player chooses an action $x_t \in \mathcal{K}$ (compact decision/parameter set);
- the environment chooses a loss function $f_t : \mathcal{K} \to [0, 1]$;
- the player suffers loss $f_t(x_t)$ and observes
  - the losses of every actions: $f_t(x)$ for all $x \in \mathcal{K}$ → full-information feedback
  - the loss of the chosen action only: $f_t(x_t)$ → bandit feedback.

Goal. Minimize the cumulative loss:

$$\widehat{L}_T \stackrel{\text{def}}{=} \sum_{t=1}^{T} f_t(x_t)\,.$$

Example: Multi-armed bandit (Thomson, 1933)

A simple stochastic model:

- K arms (actions: here price signals)
- Each arm k is associated an unknown probability distribution with mean μ_k

Setting: sequentially pick an arm k_t and get reward $X_{k_t,t}$ with mean μ_{k_t}

Goal: maximize the expected cumulative reward

$$\mathbb{E}\left[\sum_{t=1}^{T} X_{k_t,t}\right]$$

Exploration vs Exploitation trade-off.

Bandit applications

Maximize one's gains in casino? Hopeless ...

Historical motivation (Thomson, 1933): clinical trials, for each patient t in a clinical study

- choose a treatment k_t
- observe response to the treatment $X_{k_t,t}$

Goal: maximize the number of patient healed (or find the best treatment)

Successful because of many applications coming from Internet: recommender systems, online advertisements,...

Setting of an online learning problem – Multi-armed bandits

At each time step $t = 1, \ldots, T$

- ~~the player observes a context $x_t \in \mathcal{X}$ (optional step)~~
- the player chooses an action $x_t = k_t \in \mathcal{K} := \{1, \ldots, K\}$ (compact decision/parameter set);
- the environment chooses a loss function $f_t : \mathcal{K} \to [0, 1]$ (by sampling the arms);
- the player suffers loss $f_t(x_t) = 1 - X_{k_t,t}$ and observes
  - the losses of every actions: $f_t(x)$ for all $x \in \mathcal{K}$ → full-information feedback
  - the loss of the chosen action only: $f_t(x_t) = X_{k_t,t}$ → bandit feedback.

The goal of the player is to minimize his cumulative loss:

$$\widehat{L}_T \overset{\text{def}}{=} \sum_{t=1}^{T} f_t(x_t) \,.$$

Example 2: Prediction with expert advice

There is some sequence of observations $y_1, \dots, y_T \in [0,1]$ to be predicted step by step with the help of expert forecasts.

At each time step $t \geqslant 1$

- the environment reveals experts forecasts $c_t(k)$ for $k = 1, \dots, K$
- the player chooses a weight vector $p_t \in \Delta_K \overset{\text{def}}{=} \{p \in [0,1]^K : \sum_{k=1}^K p_k = 1\}$ (here x_t is denoted p_t and $\mathcal{K} = \Delta_K$)
- the player forecasts $\widehat{y}_t = \sum_{k=1}^K p_t(k) c_t(k)$
- the environment reveals $y_t \in [0,1]$ and the player suffers loss $f_t(p_t) = f(\widehat{y}_t, y_t)$ where $f : [0,1]^2 \to [0,1]$ is a loss function.

Considering $\mathcal{K} := \Delta_K$ and $x_t := p_t$, we recover the general setting. The inputs correspond to the expert advice $c_t(k)$ that are often revealed before the learner makes his decision p_t.

Example 2: Prediction with expert advice

There is some sequence of observations $y_1, \ldots, y_T \in [0,1]$ to be predicted step by step with the help of expert forecasts.

At each time step $t \geqslant 1$

- the environment reveals experts forecasts $c_t(k)$ for $k = 1, \ldots, K$
- the player chooses a weight vector $p_t \in \Delta_K \overset{\text{def}}{=} \{p \in [0,1]^K : \sum_{k=1}^K p_k = 1\}$ (here x_t is denoted p_t and $\mathcal{K} = \Delta_K$)
- the player forecasts $\widehat{y}_t = \sum_{k=1}^K p_t(k) c_t(k)$
- the environment reveals $y_t \in [0,1]$ and the player suffers loss $f_t(p_t) = f(\widehat{y}_t, y_t)$ where $f : [0,1]^2 \to [0,1]$ is a loss function.

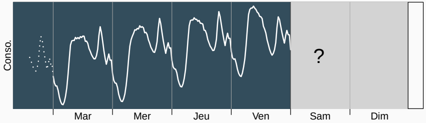
Player's performance is then measured via a loss function $f_t(p_t) = f(\widehat{y}_t, y_t)$ which measures the distance between the prediction $\widehat{y}_t$ and the output y_t:

- squared loss $f(\widehat{y}_t, y_t) = (\widehat{y}_t - y_t)^2$
- absolute loss $f(\widehat{y}_t, y_t) = |\widehat{y}_t - y_t|$
- absolute percentage of error $f(\widehat{y}_t, y_t) = |\widehat{y}_t - y_t| / |y_t|$
- pinball loss.

All these loss functions are convex, which will play an important role in the analysis.

Example: Prediction with expert advice for electricity forecasting

Short term prediction (one day ahead) of the French electricity consumption

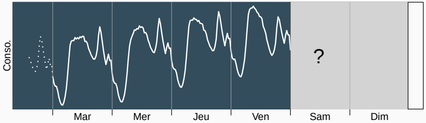


Important because electricity is hard to store.

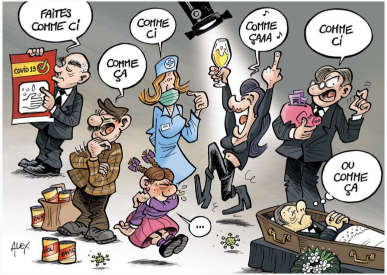
Production ⚖ Demand

Example: Prediction with expert advice for electricity forecasting

Short term prediction (one day ahead) of the French electricity consumption



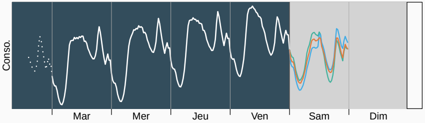
Many experts (statisticians or data scientists) design prediction models:



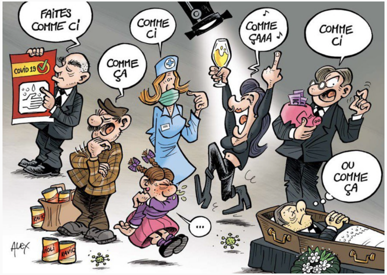
Simultaneously, the French electricity market is evolving (electric cars,. . .)

Example: Prediction with expert advice for electricity forecasting

Short term prediction (one day ahead) of the French electricity consumption



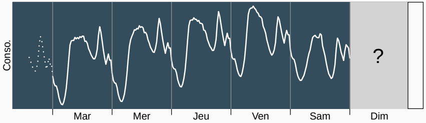
Many experts (statisticians or data scientists) design prediction models:



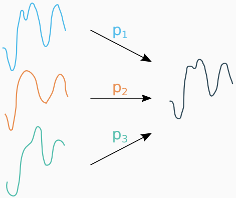
Simultaneously, the French electricity market is evolving (electric cars,. . .)

Example: Prediction with expert advice for electricity forecasting

Short term prediction (one day ahead) of the French electricity consumption



Combine the predictions using adaptive methods:



Each day,

1. Assign a weight to each expert based on past performance

$$x_t = \text{weight vector}$$

2. Predict the weighted average $\widehat{y}_t = \langle x_t, c_t \rangle$ and suffer loss

$$f_t(x_t) = \left(y_t - \widehat{y}_t\right)^2$$

How to measure the performance? The regret

If the environment chooses large losses $f_t(x)$ for all decisions $x \in \mathcal{K}$, it is impossible for the player to ensure small cumulative loss.

$\rightarrow$ Relative criterion: the regret of the player is the difference between the cumulative loss he incurred and that of the best fixed decision in hindsight.

Definition (Regret)

The regret of the player with respect to a fixed parameter $x^* \in \mathcal{K}$ after T time steps is

$$R_T(x^*) \overset{\text{def}}{=} \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^*) \,.$$

The regret (or uniform regret) is defined as $R_T \overset{\text{def}}{=} \sup_{x^* \in \mathcal{K}} R_T(x^*)$.

Regret decomposition

We have some approximation-estimation decomposition:

$$\sum_{t=1}^{T} f_t(x_t) = \underbrace{\inf_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x)}_{\text{Approximation error = how good the possible actions are.}} + \underbrace{R_T}_{\text{Sequential estimation error of the best action}}$$

We will focus on the regret in these lectures.

The goal of the player is to ensure a sublinear regret $R_T = o(T)$ as $T \to \infty$ and this for any possible sequence of losses $f_1, \dots, f_T$.
→ the average performance of the player will approach on the long term the one of the best decision.

Adversarial / Stochastic setting

The losses f_t are unknown to the player beforehand and may be:

- Adversarial setting (lessons 1, 2, and 3): No stochastic assumption on the process generating the losses f_t. The latter are deterministic and may be chosen by some adversary. Typically, the problem can be seen as a game between the player who aims at optimizing with respect to $x_1, \dots, x_T$ against an environment who aims at mazimizing with respect to $loss_t, \dots, loss_T$ and x^*. Players's goal is to control the quantity:

$$\inf_{x_1} \sup_{f_1} \inf_{x_2} \sup_{f_2} \dots \inf_{x_T} \sup_{f_T} \sup_{x^* \in \mathcal{K}} R_T(x^*) \,.$$

- Stochastic setting (lessons 4, 5, and 6): the losses are generated by some stochastic process (e.g., i.i.d.). The regret bounds hold then in expectation or with high probability.

Why a different loss at every round t?

This may be caused by many phenomena, e.g. by

- some observation to be predicted if $f_t(x) = f(x, y_t)$. For instance, if the goal is to predict the evolution of the temperature $y_1, \dots, y_T$, the latter changes over time and a prediction x is evaluated with $f_t(x) = (x - y_t)^2$.
- noise: the environment is stochastic and the variation over time t models some noise effect.
- a changing environment. For instance, if the player is playing a game against some adversary that evolves and adapts to its strategy. A typical example is the case of spam detections. If the player tries to detect spams, while some spammers (the environment) try at the same time to fool the player with new spam strategies.

Exercise: what about best x_t^* at every round?

Regret

$$R_T = \sum_{t=1}^{T} f_t(x_t) - \inf_{x^* \in \mathcal{K}} \sum_{t=1}^{T} f_t(x^*)$$

Instead considering the regret with respect to a fixed $x^* \in \mathcal{K}$, one would be tempted to minimize the quantity

$$R_T^* \stackrel{\text{def}}{=} \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} \inf_{x \in \mathcal{K}} f_t(x)$$

where the infimum is inside the sum.

Exercise: Show that the environment can ensure R_T^* to be linear in T by choosing properly the loss functions f_t.

Introduction: What is online learning?

Online Linear Optimization

Online Convex Optimization

Adversarial bandits

Stochastic bandits

Online Linear Optimization

We will start with the simple case where the decision set $\mathcal{K}$ is the K-dimensional simplex

$$\Delta_K \overset{\text{def}}{=} \left\{ p \in [0,1]^K : \sum_{k=1}^{K} p_k = 1 \right\}. \qquad \text{(decision set)}$$

Since the decisions x_t are probability distributions in $\mathcal{K} = \Delta_K$, in this part we will denote them by p_t instead of x_t. We assume that the loss functions f_t are linear

$$\forall p \in \mathcal{K}, \qquad f_t(p) = \sum_{k=1}^{K} p(k) g_t(k) \in [-1,1] \qquad \text{(linear loss)}$$

where $g_t = (g_t(1), \dots, g_t(K)) \in [-1,1]^K$ is a loss vector chosen by the environment at round t.

How to choose the weights

At round t the player needs to choose a weight vector $p_t \in \Delta_K$.

How to choose the weights? The player should

- give more weight to actions that performed well in the past.
- not give all the weight to the current best action, otherwise it would not work (see Exercise next).

The exponentially weighted average forecaster (EWA) also called Hedge performs this trade-off by choosing a weight that decreases exponentially fast with the past errors.

Introduction: What is online learning?

Online Linear Optimization

- The exponentially weighted average forecaster (EWA)
- Application to prediction with expert advice

Online Convex Optimization

Adversarial bandits

Stochastic bandits

The exponentially weighted average forecaster (EWA)

The exponentially weighted average forecaster

Parameter: $\eta > 0$

Initialize: $p_1 = \left(\frac{1}{K}, \dots, \frac{1}{K}\right)$

For $t = 1, \dots, T$

- select p_t; incur loss $f_t(p_t) = p_t^\top g_t$ and observe $g_t \in [-1, 1]^K$;
- update for all $k \in \{1, \dots, K\}$

$$p_{t+1}(k) = \frac{e^{-\eta \sum_{s=1}^t g_s(k)}}{\sum_{j=1}^K e^{-\eta \sum_{s=1}^t g_s(j)}} .$$

Exercise

Consider the strategy, called "Follow The Leader" (FTL) that puts all the mass on the best action so far:

$$p_t \in \operatorname*{arg\,min}_{p \in \mathcal{K}} \sum_{s=1}^{t-1} f_s(p)\,. \tag{FTL}$$

Exercise:

1. Show that $p_t(k) > 0$ implies that $k \in \arg\min_j \sum_{s=1}^{t-1} g_s(j)$
2. Show that the regret of FTL might be linear: i.e., there exists a sequence $g_1, \dots, g_T \in [-1, 1]^K$ such that $R_T \geqslant \Omega(T)$.

Solution

Consider the strategy, called "Follow The Leader" (FTL) that puts all the mass on the best action so far:

$$p_t \in \arg\min_{p \in \mathcal{K}} \sum_{s=1}^{t-1} f_s(p) \,. \tag{FTL}$$

Exercise:

1. Show that $p_t(k) > 0$ implies that $k \in \arg\min_j \sum_{s=1}^{t-1} g_s(j)$

Solution

Assume that there exists $k \in [K]$ such that $p_t(k) > 0$ and $k \notin \arg\min_j \sum_{s=1}^{t-1} g_s(j)$. Then, there exists $k' \neq k$ such that $\sum_{s=1}^{t-1} g_s(k') < \sum_{s=1}^{t-1} g_s(k)$. Therefore,

$$\begin{aligned}\sum_{s=1}^{t-1} f_s(p_t) &= \sum_{s=1}^{t-1} \sum_{j=1}^{K} p_s(j) g_s(j) = \sum_{s=1}^{t-1} \sum_{j \neq k} p_s(j) g_s(j) + p_s(k) \sum_{s=1}^{t-1} g_s(k) \\ &> \sum_{s=1}^{t-1} \sum_{j \neq k} p_s(j) g_s(j) + p_s(k') \sum_{s=1}^{t-1} g_s(k) = \sum_{s=1}^{t-1} f_s(q_t) \,,\end{aligned}$$

where $q_t(j) = p_t(j)$ if $j \notin \{k, k'\}$ and $q_t(k) = 0$ and $q_t(k') = p_t(k') + q_t(k')$. This yields a contradiction.

Solution

Consider the strategy, called "Follow The Leader" (FTL) that puts all the mass on the best action so far:

$$p_t \in \operatorname*{arg\,min}_{p \in \mathcal{K}} \sum_{s=1}^{t-1} f_s(p)\,. \tag{FTL}$$

Exercise:

1. Show that $p_t(k) > 0$ implies that $k \in \arg\min_j \sum_{s=1}^{t-1} g_s(j)$
2. Show that the regret of FTL might be linear: i.e., there exists a sequence $g_1, \dots, g_T \in [0,1]^K$ such that $R_T \geqslant \Omega(T)$.

Solution

It suffices to choose $g_t(k) = 1$ if $p_t(k) > 0$ and $g_t(k) = 0$ otherwise. The cumulative loss of FTL is T while there exists an action with cumulative loss smaller then T/K.

Regret guarantee for EWA

Theorem 1 (Regret bound for EWA)

Let $T \geqslant 1$. For all sequences of loss vectors $g_1, \dots, g_T \in [-1,1]^K$, EWA achieves the bound

$$R_T \overset{\text{def}}{=} \sum_{t=1}^{T} f_t(p_t) - \min_{p \in \Delta_K} \sum_{t=1}^{T} f_t(p) \leqslant \eta \sum_{t=1}^{T} \sum_{k=1}^{K} p_t(k) g_t(k)^2 + \frac{\log K}{\eta}\,, \tag{1}$$

where we recall $f_t : p \in \Delta_K \mapsto p^\top g_t$.

Therefore, for the choice $\eta = \sqrt{\frac{\log K}{T}}$, EWA satisfies the regret bound $R_T \leqslant 2\sqrt{T \log K}$.

This regret bound is optimal (see [1]).

Exercise: Generalize the above theorem when the losses $g_1, \dots, g_T \in [-B, B]^K$ for some $B > 0$.

[1] Cesa-Bianchi and Lugosi, Prediction, learning, and games, 2006.

Proof (Step 1 - Reformulation of the regret for linear losses)

First, we remark that by definition of $f_t : p \mapsto p \cdot g_t$ we have

$$\begin{aligned} R_T &\overset{\mathrm{def}}{=} \sum_{t=1}^{T} f_t(p_t) - \min_{p \in \Delta_K} \sum_{t=1}^{T} f_t(p) \\ &= \sum_{t=1}^{T} p_t \cdot g_t - \min_{p \in \Delta_K} \sum_{t=1}^{T} p \cdot g_t \\ &= \sum_{t=1}^{T} p_t \cdot g_t - \min_{p \in \Delta_K} \sum_{k=1}^{K} \sum_{t=1}^{T} p(k) g_t(k) \,. \end{aligned}$$

Now, we can see that the minimum over $p \in \Delta_K$ is reached on a corner of the simplex. Therefore

$$R_T = \sum_{t=1}^{T} p_t \cdot g_t - \min_{1 \leqslant k \leqslant K} \sum_{t=1}^{T} g_t(k) \,.$$

Proof (Step 2 – Upper-bound of W_T)

We denote $W_t(j) = e^{-\eta \sum_{s=1}^{t} g_t(j)}$ and $W_t = \sum_{j=1}^{K} W_t(j)$. The proof will consist in upper-bounding and lower-bounding W_T. We have

$$\begin{aligned}
W_t &= \sum_{j=1}^{K} W_{t-1}(j) e^{-\eta g_t(j)} & &\leftarrow \quad W_t^{(j)} = W_{t-1}(j) e^{-\eta g_t(j)} \\
&= W_{t-1} \sum_{j=1}^{K} \frac{W_{t-1}(j)}{W_{t-1}} e^{-\eta g_t(j)} & & \\
&= W_{t-1} \sum_{j=1}^{K} p_t(j) e^{-\eta g_t(j)} & &\leftarrow \quad p_t(j) = \frac{e^{-\eta \sum_{s=1}^{t-1} g_s(j)}}{\sum_{k=1}^{K} e^{-\eta \sum_{s=1}^{t-1} g_s(k)}} = \frac{W_{t-1}(j)}{W_{t-1}} \\
&\leqslant W_{t-1} \sum_{j=1}^{K} p_t(j) \big(1 - \eta g_t(j) + \eta^2 g_t(j)^2\big) & &\leftarrow \quad e^x \leqslant 1 + x + x^2 \text{ for } x \leqslant 1 \\
&= W_{t-1}(1 - \eta p_t \cdot g_t + \eta^2 p_t \cdot g_t^2)\,, & &
\end{aligned}$$

where we assumed in the inequality $-\eta g_t(j) \leqslant 1$ and where we denote $g_t = (g_t(1), \ldots, g_t(K))$, $g_t^2 = \big(g_t(1)^2, \ldots, g_t(K)^2\big)$ and $p_t = (p_t(1), \ldots, p_t(K))$.

Proof (Step 2 - Upper-bound of W_T)

Now, using $1 + x \leqslant e^x$, we get:

$$W_t \leqslant W_{t-1}(1 - \eta p_t \cdot g_t + \eta^2 p_t \cdot g_t^2) \leqslant W_{t-1} \exp\left(-\eta p_t \cdot g_t + \eta^2 p_t \cdot g_t^2\right).$$

By induction on $t = 1, \ldots, T$, this yields using $W_0 = K$

$$W_T \leqslant K \exp\left(-\eta \sum_{t=1}^{T} p_t \cdot g_t + \eta^2 \sum_{t=1}^{T} p_t \cdot g_t^2\right). \tag{2}$$

Proof (Step 3 – Lower-bound of W_T)

On the other hand, upper-bounding the maximum with the sum,

$$\exp\left(-\eta \min_{j\in[K]} \sum_{t=1}^{T} g_t(j)\right) \leqslant \sum_{j=1}^{K} \exp\left(-\eta \sum_{t=1}^{T} g_t(j)\right) \leqslant W_T\,.$$

Combining the above inequality with Inequality (2) and taking the log, we get

$$-\eta \min_{j\in[K]} \sum_{t=1}^{T} g_t(j) \leqslant -\eta \sum_{t=1}^{T} p_t \cdot g_t + \eta^2 \sum_{t=1}^{T} p_t \cdot g_t^2 + \log K\,. \tag{3}$$

Dividing by η and reorganizing the terms proves the first inequality:

$$R_T = \sum_{t=1}^{T} p_t \cdot g_t - \min_{1\leqslant j\leqslant K} \sum_{t=1}^{T} g_t(j) \leqslant \eta \sum_{t=1}^{T} p_t \cdot g_t^2 + \frac{\log K}{\eta}$$

Optimizing η and upper-bounding $p_t \cdot g_t^2 \leqslant 1$ concludes the second inequality. □

Regret guarantee for EWA

Theorem 1 (Regret bound for EWA)

Let $T \geqslant 1$. For all sequences of loss vectors $g_1, \dots, g_T \in [-1,1]^K$, EWA achieves the bound

$$R_T \overset{\text{def}}{=} \sum_{t=1}^T f_t(p_t) - \min_{p \in \Delta_K} \sum_{t=1}^T f_t(p) \leqslant \eta \sum_{t=1}^T \sum_{k=1}^K p_t(k) g_t(k)^2 + \frac{\log K}{\eta}\,, \tag{1}$$

where we recall $f_t : p \in \Delta_K \mapsto p^\top g_t$.

Therefore, for the choice $\eta = \sqrt{\frac{\log K}{T}}$, EWA satisfies the regret bound $R_T \leqslant 2\sqrt{T \log K}$.

This regret bound is optimal (see [1]).

Exercise: Generalize the above theorem when the losses $g_1, \dots, g_T \in [-B, B]^K$ for some $B > 0$.

[1] Cesa-Bianchi and Lugosi, Prediction, learning, and games, 2006.

Anytime algorithm

The previous algorithms EWA depends on a parameter $\eta > 0$ that needs to be optimized according to K and T. For instance, for EWA using the value

$$\eta = \sqrt{\frac{\log K}{KT}}\,.$$

The bound of Theorem 1 is only valid for horizon T.

However, the learner might not know the time horizon in advance and one might want an algorithm with guarantees valid simultaneously for all $T \geqslant 1$.

We can avoid the assumption that T is known in advance, at the cost of a constant factor, by using the so-called doubling trick.

Anytime algorithm: the doubling trick

Whenever we reach a time step t which is a power of 2, we restart the algorithm (forgetting all the information gained in the past) setting η to $\sqrt{\log K/t}$. Let us denote EWA-doubling this algorithm.

Theorem 2 (Anytime bound on the regret)

For all $T \geqslant 1$, the regret of EWA-doubling is then upper-bounded as:

$$R_T \leqslant 7\sqrt{T \log K}\,.$$

The same trick can be used to turn most online algorithms into anytime algorithms (even in more general settings: bandits, general loss,. . .).

We can use the doubling trick whenever we have an algorithm with a regret of order $\mathcal{O}(T^\alpha)$ for some $\alpha > 0$ with a known horizon T to turn it into an algorithm with a regret $\mathcal{O}(T^\alpha)$ for all $T \geqslant 1$.

Proof

For simplicity we assume $T = 2^{M+1} - 1$. The regret of EWA-doubling is then upper-bounded as:

$$\begin{aligned}
R_T &= \sum_{t=1}^{T} f_t(p_t) - \min_{p \in \Delta_K} \sum_{t=1}^{T} f_t(p) \\
&\leqslant \sum_{t=1}^{T} f_t(p_t) - \sum_{m=0}^{M} \min_{p \in \Delta_K} \sum_{t=2^m}^{2^{m+1}-1} f_t(p) \\
&= \sum_{m=0}^{M} \underbrace{\sum_{t=2^m}^{2^{m+1}-1} f_t(p_t) - \min_{p \in \Delta_K} \sum_{t=2^m}^{2^{m+1}-1} f_t(p)}_{R_m} .
\end{aligned}$$

Now, we remark that each term R_m corresponds to the expected regret of an instance of EWA over the 2^m rounds $t = 2^m, \ldots, 2^{m+1} - 1$ and run with the optimal parameter $\eta = \sqrt{\log K / 2^m}$. Therefore, using Theorem 1, we get $R_m \leqslant 2\sqrt{2^m \log K}$, which yields:

$$R_T \leqslant \sum_{m=0}^{M} 2\sqrt{2^m \log K} \leqslant 2(1 + \sqrt{2})\sqrt{2^{M+1} \log K} \leqslant 7\sqrt{T \log K} .$$

Anytime algorithm: time-varying parameter

Another solution is to use time-varying parameters η_t replacing T with the current value of t. The analysis is however less straightforward.

Exercise: Prove a regret bound for the time-varying choice $\eta_t = \sqrt{\log K/t}$ in EWA.

Introduction: What is online learning?

Online Linear Optimization

- The exponentially weighted average forecaster (EWA)
- Application to prediction with expert advice

Online Convex Optimization

Adversarial bandits

Stochastic bandits

Reminder of the setting of prediction with expert advice

At each time step $t \geqslant 1$

- the environment reveals experts forecasts $c_t(k)$ for $k = 1, \dots, K$
- the player chooses a weight vector $p_t \in \Delta_K \overset{\text{def}}{=} \{p \in [0,1]^K : \sum_{k=1}^K p_k = 1\}$ (here x_t is denoted p_t and $\mathcal{K} = \Delta_K$)
- the player forecasts $\widehat{y}_t = \sum_{k=1}^K p_t(k) c_t(k)$
- the environment reveals $y_t \in [0,1]$ and the player suffers loss $f_t(p_t) = f(\widehat{y}_t, y_t)$ where $f : [0,1]^2 \to [0,1]$ is a loss function.

The goal is to minimize the regret with respect to the best expert

$$R_T^{\text{expert}} \overset{\text{def}}{=} \sum_{t=1}^{T} f(\widehat{y}_t, y_t) - \min_{1 \leqslant k \leqslant K} \sum_{t=1}^{T} f(c_t(k), y_t)\,,$$

where $\widehat{y}_t = p_t \cdot c_t$ are the prediction of the algorithm and y_t the observations to be predicted sequentially.

Reminder of the setting of prediction with expert advice

At each time step $t \geqslant 1$

- the environment reveals experts forecasts $c_t(k)$ for $k = 1, \dots, K$
- the player chooses a weight vector $p_t \in \Delta_K \overset{\text{def}}{=} \{p \in [0,1]^K : \sum_{k=1}^K p_k = 1\}$ (here x_t is denoted p_t and $\mathcal{K} = \Delta_K$)
- the player forecasts $\widehat{y}_t = \sum_{k=1}^K p_t(k) c_t(k)$
- the environment reveals $y_t \in [0,1]$ and the player suffers loss $f_t(p_t) = f(\widehat{y}_t, y_t)$ where $f : [0,1]^2 \to [0,1]$ is a loss function.

Player's performance is then measured via a loss function $f_t(p_t) = f(\widehat{y}_t, y_t)$ which measures the distance between the prediction $\widehat{y}_t$ and the output y_t:

- squared loss $f(\widehat{y}_t, y_t) = (\widehat{y}_t - y_t)^2$
- absolute loss $f(\widehat{y}_t, y_t) = |\widehat{y}_t - y_t|$
- absolute percentage of error $f(\widehat{y}_t, y_t) = |\widehat{y}_t - y_t| / |y_t|$
- pinball loss.

All these loss functions are convex, how can we apply our analysis for linear losses?

Prediction with expert advice with convex loss function f.

We state bellow a corrolary to Theorem 1 when the loss functions $f(\cdot,\cdot)$ are convex in there first argument.

Corollary 1 (Regret of EWA for prediction with expert advice and convex loss)

Let $T \geqslant 1$. Assume that the loss function $f : (x, y) \in \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ is convex and takes values in $[-1, 1]$. Then, EWA applied with the vector vectors $g_t = \big(f(c_t(1), y_t), \ldots, f(c_t(K), y_t)\big) \in [-1, 1]^K$ has a regret upper-bounded by

$$R_T^{\text{expert}} \stackrel{\text{def}}{=} \sum_{t=1}^{T} f(\widehat{y}_t, y_t) - \min_{1 \leqslant k \leqslant K} \sum_{t=1}^{T} f(c_t(k), y_t) \leqslant 2\sqrt{T \log K}$$

where $\widehat{y}_t = p_t \cdot c_t$ and were $\eta > 0$ is well-tuned.

Therefore, the average error of the algorithm will converge to the average error of the best expert. This is the case for the square loss, the absolute loss or the absolute percentage of error.

Proof

It suffices to remark that by convexity of $f(\cdot,\cdot)$ in its first argument

$$\begin{aligned} R_T^{\text{expert}} &= \sum_{t=1}^{T} f(p_t \cdot c_t, y_t) - \min_{1 \leqslant k \leqslant K} \sum_{t=1}^{T} f(c_t(k), y_t) \\ &\leqslant \sum_{t=1}^{T} p_t \cdot g_t - \min_{1 \leqslant k \leqslant K} \sum_{t=1}^{T} g_t(k) \overset{\text{def}}{=} R_T . \end{aligned}$$

The result is then obtained by Theorem 1. □

Introduction: What is online learning?

Online Linear Optimization

Online Convex Optimization

Adversarial bandits

Stochastic bandits

Introduction: What is online learning?

Online Linear Optimization

Online Convex Optimization

- The Gradient Trick and EG
- Online Gradient Descent
- Online Mirror Descent

Adversarial bandits

Stochastic bandits

From linear to convex losses

Setting: simplex decision set $\mathcal{K} = \Delta_K$, convex and differentiable loss functions

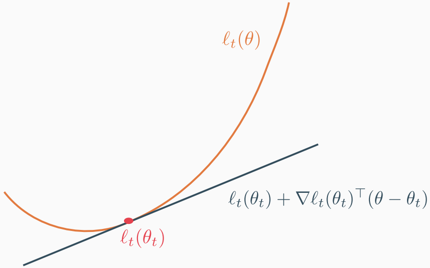
Assumptions and notations: Actions are denoted by p_t (instead of x_t). The losses are assumed to be convex and Lipschitz

$$\forall p_t \in \mathcal{K}, \qquad \big\|\nabla f_t(p_t)\big\|_\infty \leqslant G.$$

We will see a simple trick, so-called the gradient trick that allows to extend the results we saw for linear losses to convex losses.

The resulted algorithm is called the Exponentiated Gradient forecaster (EG). It consists in playing EWA with the gradients $g_t = \nabla f_t \in [-G, G]^K$ as loss vectors.

The gradient trick


$\ell_t(\theta)$
$\ell_t(\theta_t) + \nabla \ell_t(\theta_t)^\top(\theta - \theta_t)$
$\ell_t(\theta_t)$

For $g_t = \nabla f_t(x_t)$, the linear loss $\tilde{f}_t(x) = g_t^\top x$ satisfies for any $x \in \mathcal{K}$

$$f_t(x_t) - f_t(x) \leqslant g_t^\top(x_t - x) \leqslant \tilde{f}_t(x_t) - \tilde{f}_t(x)\,.$$

To prevent infinite regret, need finite $|\tilde{f}_t(x)|$ and hence bounds on the dual norms of the domain and gradients

$$|\tilde{f}_t(x)| \leqslant \|g_t\|_p \|x\|_q, \qquad \frac{1}{p} + \frac{1}{q} = 1\,.$$

Algorithm

The Exponentiated Gradient forecaster (EG)

Parameter: $\eta > 0$

Initialize: $p_1 = \left(\frac{1}{K}, \dots, \frac{1}{K}\right)$

For $t = 1, \dots, T$

- select p_t; incur loss $f_t(p_t)$ and observe $f_t : \mathcal{K} \to [0, 1]$;
- compute the gradient $g_t = \nabla f_t(p_t) \in [-G, G]^K$
- update for all $k \in \{1, \dots, K\}$

$$p_{t+1}(k) = \frac{e^{-\eta \sum_{s=1}^{t} g_s(k)}}{\sum_{j=1}^{K} e^{-\eta \sum_{s=1}^{t} g_s(j)}} \,.$$

Regret bound of EG

Theorem 3

Let $T \geqslant 1$. For all sequences of convex differentiable losses $f_1, \dots, f_T : \mathcal{K} \to \mathbb{R}$ with bounded gradient $\max_{p \in \mathcal{K}} \|\nabla f_t(p)\|_\infty \leqslant G$, EWA applied with $g_t = \nabla f_t$ achieves the regret bound

$$R_T \stackrel{\text{def}}{=} \sum_{t=1}^{T} f_t(p_t) - \min_{p \in \mathcal{K}} \sum_{t=1}^{T} f_t(p) \leqslant \eta G^2 T + \frac{\log K}{\eta} \,. \tag{4}$$

Therefore, for the choice $\eta = \frac{1}{G}\sqrt{\frac{\log K}{T}}$, EWA satisfies the regret bound $R_T \leqslant 2G\sqrt{T \log K}$.

Proof

1. **Apply the regret bound of EWA with g_t** (see Theorem 1 of last class):

$$\sum_{t=1}^{T} p_t \cdot g_t - \min_{p \in \Delta_K} \sum_{t=1}^{T} p \cdot g_t \leqslant \eta \sum_{t=1}^{T} \sum_{k=1}^{K} p_t(k) g_t(k)^2 + \frac{\log K}{\eta} .$$

Remark that the theorem also holds for loss vectors $g_t \in [-G, G]^K$ as soon as $\eta \leqslant 1/G$.
Upper-bounding $g_t(j)^2 \leqslant \|\nabla f_t(p_t)\|_\infty^2 \leqslant G^2$, substituting $g_t = \nabla f_t(p_t)$, this yields for all $p \in \Delta_K$

$$\sum_{t=1}^{T} p_t \cdot \nabla f_t(p_t) - p \cdot \nabla f_t(p_t) \leqslant \eta T G^2 + \frac{\log K}{\eta} .$$

2. **Gradient inequality**: by convexity of the losses

$$f_t(p_t) - f_t(p) \leqslant (p_t - p) \cdot \nabla f_t(p_t) ,$$

which yields

$$\sum_{t=1}^{T} f_t(p_t) - f_t(p) \leqslant \eta T G^2 + \frac{\log K}{\eta} .$$

3. **Optimize η:** $\eta = \frac{1}{G}\sqrt{\frac{\log K}{T}}$. □

Example: Prediction with expert advice (continued)

Setting: A sequence of observations $y_1, \dots, y_T \in [0,1]$ is to be predicted with the help of K expert advice $c_t(k) \in [0,1]$ for $1 \leqslant k \leqslant K$. The learner predict $\widehat{y}_t = \sum_{k=1}^{K} p_t(k) c_t(k)$ and suffers a loss $f(\widehat{y}_t, y_t)$.

If the loss function is convex and Lipschitz in its first argument, we can apply Theorem 3 with $f_t : p \mapsto f(p \cdot c_t, y_t)$.

For instance, with the absolute loss, $G = 1$ and EG satisfies:

$$\sum_{t=1}^{T} |\widehat{y}_t - y_t| - \min_{p \in \mathcal{K}} \sum_{t=1}^{T} \Big| p \cdot c_t - y_t \Big| \leqslant 2\sqrt{T \log K}\,.$$

Hence, on the long run we perform as good as the best convex combination of the experts which may outperform the best expert.

Introduction: What is online learning?

Online Linear Optimization

Online Convex Optimization

- The Gradient Trick and EG
- Online Gradient Descent
- Online Mirror Descent

Adversarial bandits

Stochastic bandits

Online Gradient Descent (OGD) [2]

Setting: convex differentiable Lipschitz loss function, convex and compact decision set $\mathcal{K}$

Online Gradient Descent (OGD)

```
Parameter: $\eta > 0$
Initialize: $x_1 \in \mathcal{K}$ arbitrarily chosen

For $t = 1, \dots, T$
  - select $x_t$; incur loss $f_t(x_t)$ and observe $f_t : \mathcal{K} \to [0, 1]$;
  - compute the gradient $\nabla f_t(x_t)$
  - update
        $x_{t+1} = \mathrm{Proj}_{\mathcal{K}} \left( x_t - \eta \nabla f_t(x_t) \right) .$
    where $\mathrm{Proj}_{\mathcal{K}}$ is the Euclidean projection onto $\mathcal{K}$.
```

Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent", 2003.

Regret bound for OGD

Online Gradient Descent

$$x_{t+1} \leftarrow \mathrm{Proj}_{\mathcal{K}}\left(x_t - \eta \nabla f_t(x_t)\right)$$

Theorem 4 (Regret of OGD)

Let $D, G, \eta > 0$. Assume that $\max_{x,x' \in \mathcal{K}} \|x - x'\| \leqslant D$ and. Then for any sequence $f_1, \dots, f_T$ of convex differentiable loss functions such that $\max_{x \in \mathcal{K}} \|\nabla f_t(x)\| \leqslant G$, the regret of OGD satisfies

$$\sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x) \leqslant \frac{D^2}{2\eta} + \frac{\eta}{2} G^2 T \,.$$

In particular, for $\eta = \frac{D}{G\sqrt{T}}$, we have $R_T \leqslant DG\sqrt{T}$.

Comparison of EG and OGD

Assume that $\mathcal{K} = \Delta_K$ is the simplex and the loss functions are sub-differentiable convex functions with $\|\nabla f_t\|_\infty \leqslant G_\infty$. Then both EG and OGD are possible algorithms (see Theorems 3 and 10).

We saw in Theorem 3 that EG has a regret bound $R_T \leqslant 2G_\infty\sqrt{T\log K}$. In this case, for all $p, p' \in \Delta_K$

$$\|p - p'\| = \sum_{k=1}^{K} \big(p(i) - p'(i)\big)^2 \leqslant \sum_{i=1}^{K} \big|p(i) - p'(i)\big| \leqslant \sum_{i=1}^{K} p(i) + p'(i) = 2\,,$$

and $\|\nabla f_t(p)\| \leqslant \sqrt{K}\|\nabla f_t(p)\|_\infty \leqslant \sqrt{K}G_\infty$. Therefore, the regret of OGD is upper-bounded by $R_t \leqslant G_\infty\sqrt{2KT}$. Thus

$$\boxed{\text{EG:} \quad R_T \leqslant 2G_\infty\sqrt{T\log K} \qquad \text{and} \qquad \text{OGD:} \quad R_T \leqslant \sqrt{2KT}\,.}$$

The dependence on K of OGD is suboptimal in this case. This is solved by OMD, a generalization of both algorithms.

Regret bound for OGD

Online Gradient Descent

$$x_{t+1} \leftarrow \mathrm{Proj}_{\mathcal{K}}\left(x_t - \eta \nabla f_t(x_t)\right)$$

Theorem 4 (Regret of OGD)

Let $D, G, \eta > 0$. Assume that $\max_{x,x' \in \mathcal{K}} \|x - x'\| \leqslant D$ and. Then for any sequence $f_1, \dots, f_T$ of convex differentiable loss functions such that $\max_{x \in \mathcal{K}} \|\nabla f_t(x)\| \leqslant G$, the regret of OGD satisfies

$$\sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x) \leqslant \frac{D^2}{2\eta} + \frac{\eta}{2} G^2 T\,.$$

In particular, for $\eta = \frac{D}{G\sqrt{T}}$, we have $R_T \leqslant DG\sqrt{T}$.

Proof (Step 1)

Recall the update of OGD:

$$\boxed{\text{OGD}: \quad x_{t+1} \leftarrow \text{Proj}_{\mathcal{K}}\big(\underbrace{x_t - \eta \nabla f_t(x_t)}_{z_t}\big)}$$

1. **Upper-bound the regret with gradient inequality:** by convexity

$$R_T \overset{\text{def}}{=} \sum_{t=1}^{T} f_t(x_t) - f_t(x^*) \overset{\text{Convexity}}{\leqslant} \sum_{t=1}^{T} \langle \nabla f_t(x_t), x_t - x^* \rangle$$

Proof (Step 2)

2. Get a telescoping sum:

$$\begin{aligned}\left\|x_{t+1}-x^*\right\|^2 &\overset{\text{Projection}}{\leqslant} \left\|z_t-x^*\right\|^2 \\ &= \left\|x_t-\eta\nabla f_t(x_t)-x^*\right\|^2 \\ &= \left\|x_t-x^*\right\|^2+\eta^2\left\|\nabla f_t(x_t)\right\|^2-2\eta\langle\nabla f_t(x_t),x_t-x^*\rangle\end{aligned}$$

$$x_{t+1} \leftarrow \mathrm{Proj}_{\mathcal{K}}\left(\underbrace{x_t-\eta\nabla f_t(x_t)}_{z_t}\right)$$

Thus,

$$\langle\nabla f_t(x_t),x_t-x^*\rangle \leqslant \frac{1}{2\eta}\left(\left\|x_t-x^*\right\|^2-\left\|x_{t+1}-x^*\right\|^2\right)+\frac{\eta}{2}\left\|\nabla f_t(x_t)\right\|^2$$

Summing over $t=1,\ldots,T$ and it telescopes

$$\begin{aligned}R_T &\leqslant \frac{1}{2\eta}\left(\left\|x_1-x^*\right\|^2-\cancel{\left\|x_{T+1}-x^*\right\|^2}\right)+\frac{\eta}{2}G^2T \\ &\leqslant \frac{D^2}{2\eta}+\frac{\eta G^2T}{2}\end{aligned}$$

Exercises

Exercise: Prove an upper-bound on the regret of OGD

a) when η is calibrated with a doubling trick.

b) when η is calibrated using a time-varying parameter $\eta_t = D/(G\sqrt{t})$

Exercise: Prove an upper-bound on the regret of OGD with respect to any sequence of points $x_1^*, \dots, x_t^* \in \mathcal{K}$ such that $\sum_{t=2}^T \|x_t^* - x_{t-1}^*\| \leqslant X$

$$\sum_{t=1}^T f_t(x_t) - \sum_{t=1}^T f_t(x_t^*) \leqslant \quad \dots$$

Logarithmic regret under strong-convexity

Online Gradient Descent:

$$\boxed{x_{t+1} \leftarrow \mathrm{Proj}_{\mathcal{K}}\left(x_t - \eta_t \nabla f_t(x_t)\right)}$$

Theorem 5 (Regret of OGD under strong-convexity)

Let $D, G, \gamma > 0$. Assume that $\max_{x,x' \in \mathcal{K}} \|x - x'\| \leqslant D$ and. Then for any sequence $f_1, \dots, f_T$ of γ-strongly convex differentiable loss functions such that $\max_{x \in \mathcal{K}} \|\nabla f_t(x)\| \leqslant G$, the regret of OGD with $\eta_t = 1/(\gamma t)$ satisfies

$$R_T \overset{\text{def}}{=} \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x) \leqslant \frac{G^2}{2\gamma}\left(1 + \log T\right).$$

Proof

1. **Upper-bound the regret with strong convexity:**

$$R_T \overset{\text{def}}{=} \sum_{t=1}^{T} f_t(x_t) - f_t(x^*) \quad \overset{\text{Strong Convexity}}{\leqslant} \quad \sum_{t=1}^{T} \langle \nabla f_t(x_t), x_t - x^* \rangle - \frac{\gamma}{2} \left\| x_t - x^* \right\|^2$$

2. **Upper-bound the gradient term as for OGD analysis**

$$\langle \nabla f_t(x_t), x_t - x^* \rangle \leqslant \frac{1}{2\eta_t} \left(\left\| x_t - x^* \right\|^2 - \left\| x_{t+1} - x^* \right\|^2 \right) + \frac{\eta_t}{2} \left\| \nabla f_t(x_t) \right\|^2$$

3. **Substitute in the previous inequality and conclude**

$$\begin{aligned} R_T &\leqslant \sum_{t=1}^{T} \frac{1}{2\eta_t} \left(\left\| x_t - x^* \right\|^2 - \left\| x_{t+1} - x^* \right\|^2 \right) + \frac{\eta_t G^2}{2} - \frac{\gamma}{2} \left\| x_t - x^* \right\|^2 \\ &= \frac{1}{2} \sum_{t=1}^{T} \cancel{\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \gamma \right)} \left\| x_t - x^* \right\|^2 + \frac{G^2}{2} \sum_{t=1}^{T} \frac{1}{\gamma t} \\ &\leqslant \frac{G^2}{\gamma} (1 + \log T) \end{aligned}$$

Introduction: What is online learning?

Online Linear Optimization

Online Convex Optimization

- The Gradient Trick and EG
- Online Gradient Descent
- Online Mirror Descent

Adversarial bandits

Stochastic bandits

Online Mirror Descent (OMD)

Generalization of OGD to better exploit the geometry of the decision space $\mathcal{K}$.

OMD is the online counterpart of the Mirror Descent algorithm from convex optimization.

Updates are performed into a dual space defined by a convex differentiable function $R : \mathcal{K} \to \mathbb{R}$.

Definition (Bregman divergence)

For any continuously differentiable convex function R, the Bregman divergence with respect to R is defined as

$$D_R(x||y) \leqslant R(x) - R(y) - \nabla R(y) \cdot (x - y) \quad \forall x, y \in \mathcal{K} .$$

It is the difference between the value of the regularization function at x and the value of its first order Taylor approximation.

Online Mirror Descent (OMD)

Online Mirror Descent (OMD)

Parameters: $\eta > 0$, regularization function R

Initialize: $z_1 \in \mathbb{R}^d$ such that $\nabla R(z_1) = 0$ and $x_1 = \arg\min_{x \in \mathcal{K}} B_R(x||y_1)$

For $t = 1, \dots, T$

- select x_t; incur loss $f_t(x_t)$ and observe $f_t : \mathcal{K} \to [0, 1]$;
- compute the gradient $\nabla f_t(x_t)$
- update z_t such that

$$\nabla R(z_{t+1}) = \nabla R(x_t) - \eta \nabla f_t(x_t)\,.$$

- project according to the Bregman divergence

$$x_{t+1} \in \arg\min_{x \in \mathcal{K}} D_R(x||z_{t+1})\,.$$

Regret of OMD

Theorem 6

Let $t \geqslant 1$. Let $\mathcal{K}$ be a compact and convex set. Then, for all sequences of convex subdifferentiable loss functions $f_1, \ldots, f_T : \mathcal{K} \to [0,1]$, the regret of OMD is upper-bounded as

$$\sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x) \leqslant \frac{D}{\eta} + \frac{1}{\eta} \sum_{t=1}^{T} D_{R^*}\left(\nabla R(x_t) - \eta \nabla f_t(x_t) || \nabla R(x_t)\right)$$

where $D \geqslant \max_{x \in \mathcal{K}} |R(x)|$ and R^* is the Fenchel conjugate of R defined as $R^*(z) \overset{\text{def}}{=} \max_{x \in \mathcal{K}} \left\{x \cdot z - R(x)\right\}$.

The proof can be found for instance in Bubeck, Cesa-Bianchi, et al., "Regret analysis of stochastic and nonstochastic multi-armed bandit problems", 2012. EG and OGD are two particular cases of Online Mirror Descent.

Example: OMD with Balls in $\mathbb{R}^d$ = OGD

Recall the update of OGD and OMD:

OGD : $x_{t+1} \leftarrow \mathrm{Proj}_{\mathcal{K}}\left(x_t - \eta \nabla f_t(x_t)\right)$ **OMD** : $\begin{array}{l} \nabla R(z_{t+1}) = \nabla R(x_t) - \eta \nabla f_t(x_t) \\ x_{t+1} \in \arg\min_{x \in \mathcal{K}} D_R(x||z_{t+1}) \end{array}$

If $\mathcal{K} \subset \mathbb{R}^d$, we can choose $R(x) = \frac{1}{2}\|x\|^2$.

Then

$$\nabla R(x) = x \quad \text{and} \quad D_R(x||y) = \frac{1}{2}\|x - y\|^2 .$$

Therefore, the update of OMD becomes $z_{t+1} = x_t - \eta \nabla f_t(x_t)$ and $x_{t+1} = \mathrm{Proj}_{\mathcal{K}}(z_{t+1})$.

We recover the online gradient descent algorithm.

OMD in the Simplex = EG

Recall the update of EG and OMD:

EG : $g_t = \nabla f_t(x_t)$, $x_{t+1}(k) = \frac{x_t(k)e^{-\eta g_t(k)}}{\sum_{j=1}^K x_t(j)e^{-\eta g_t(j)}}$

OMD : $\nabla R(z_{t+1}) = \nabla R(x_t) - \eta \nabla f_t(x_t)$, $x_{t+1} \in \arg\min_{x\in\mathcal{K}} D_R(x||z_{t+1})$

If $\mathcal{K} = \Delta_K$. We can choose the negative entropy

$$R(x) = \sum_{i=1}^{K} x(i) \log x(i) \, .$$

In this case, $\nabla R(x)_i = 1 + \log x(i)$ and the Bregman Divergence is $D_R(x||y) = \sum_{i=1}^K x(i) \log(x(i)/y(i))$ also known as the Kullback-Leibler divergence. The update of OMD is then

$$1 + \log(z_{t+1}(i)) = 1 + \log x_t(i) - \eta g_t(i) \, ,$$

where $g_t = \nabla f_t(x_t) \in \mathbb{R}^K$. This can be rewritten

$$z_{t+1}(i) = x_t(i) e^{-\eta g_t(i)} .$$

The projection to the simplex is a simple renormalization (exercise), we thus recover EG.

Introduction: What is online learning?

Online Linear Optimization

Online Convex Optimization

Adversarial bandits

Stochastic bandits

Setting of an online learning problem/online convex optimization

At each time step $t = 1, \ldots, T$

- the player observes a context $c_t \in \mathcal{X}$ (optional step)
- the player chooses an action $x_t \in \mathcal{K}$ (compact decision/parameter set);
- the environment chooses a loss function $f_t : \mathcal{K} \to [0, 1]$;
- the player suffers loss $f_t(x_t)$ and observes
  - the losses of every actions: $f_t(x)$ for all $x \in \mathcal{K}$ → full-information feedback
  - the loss of the chosen action only: $f_t(x_t)$ → bandit feedback.

The goal of the player is to minimize his cumulative loss:

$$\widehat{L}_T \overset{\mathrm{def}}{=} \sum_{t=1}^{T} f_t(x_t) \, .$$

Previous results

The Exponentially Weighted Average (EWA) forecaster

$$\boxed{p_t(k) = \frac{e^{-\eta \sum_{s=1}^{t-1} g_s(k)}}{\sum_{j=1}^{K} e^{-\eta \sum_{s=1}^{t-1} g_s(j)}}} \tag{EWA}$$

achieves a cumulative regret $R_T \lesssim \sqrt{T \log K}$ when the set of actions is the K-dimensional simplex and for linear losses $f_t(p) = p^\top g_t$ with $g_t \in [-1, 1]^K$.

In particular, we saw the intermediate regret-bound if $-\eta g_t(k) \leqslant 1$

$$\boxed{\sum_{t=1}^{T} p_t \cdot g_t - \min_{1 \leqslant j \leqslant K} \sum_{t=1}^{T} g_t(j) \leqslant \eta \sum_{t=1}^{T} \sum_{k=1}^{K} p_t(k) g_t(k)^2 + \frac{\log K}{\eta}\,.} \tag{$*$}$$

Note that the loss vectors g_t may depend on past information $p_1, g_1, \dots, g_{t-1}, p_t$.

This lesson

We will see what we can do with bandit feedback.

At each time step $t = 1, \dots, T$

- the player observes a context $c_t \in \mathcal{X}$ (optional step)
- the player chooses an action $x_t \in \mathcal{K}$ (compact decision/parameter set);
- the environment chooses a loss function $f_t : \mathcal{K} \to [0, 1]$;
- the player suffers loss $f_t(x_t)$ and observes
  - the losses of every actions: $f_t(x)$ for all $x \in \mathcal{K}$ → full-information feedback
  - the loss of the chosen action only: $f_t(x_t)$ → bandit feedback.

The goal of the player is to minimize his cumulative loss:

$$\widehat{L}_T \overset{\text{def}}{=} \sum_{t=1}^{T} f_t(x_t)\,.$$

Introduction: What is online learning?

Online Linear Optimization

Online Convex Optimization

Adversarial bandits

- The exponentially weighted average algorithm for bandits
- Adversarial bandits with experts
- OGD without Gradients

Stochastic bandits

Adversarial multi-armed bandit and pseudo-regret

Setting: $\mathcal{K} = \{1, \ldots, K\}$. At round t, the player chooses an action $k_t \in \{1, \ldots, K\}$ and suffers and observes the loss $f_t(k_t) \in [0, 1]$ only.

Regret with respect to action $k \in [K]$ by

$$R_T(k) \overset{\text{def}}{=} \sum_{t=1}^{T} f_t(k_t) - \sum_{t=1}^{T} f_t(k)\,.$$

Instead of minimizing the expected regret $\mathbb{E}[R_T] = \mathbb{E}[\max_k R_T(k)]$, we will start with an easier objective, the pseudo-regret.

Definition (Pseudo-regret)

$$\bar{R}_T \overset{\text{def}}{=} \max_{k \in [K]} \mathbb{E}\big[R_T(k)\big] = \max_{k \in [K]} \mathbb{E}\left[\sum_{t=1}^{T} f_t(k_t) - \sum_{t=1}^{T} f_t(k)\right]. \tag{pseudo regret}$$

Oblivious vs adaptive adversary

$$\bar{R}_T \overset{\text{def}}{=} \max_{k \in [K]} \mathbb{E}\big[R_T(k)\big] = \max_{k \in [K]} \mathbb{E}\left[\sum_{t=1}^{T} f_t(k_t) - \sum_{t=1}^{T} f_t(k)\right]$$

The expectation is taken with respect to the randomness of the algorithm: the decisions k_t are random.

We can distinguish two types of adversaries:

- oblivious adversary: all the loss functions $f_1, \dots, f_t$ are chosen in advance before the game starts and do not depend on the past player decisions $k_1, \dots, k_T$. In this case, the losses $f_t(k)$ are determinist and there is thus equality: $\bar{R}_T = \mathbb{E}[R_T]$.
- adaptive adversary: the loss function f_t at round $t \geqslant 1$ may depend on past information $\sigma(k_1, \dots, k_{t-1})$. It is thus random. By Jensen's inequality $\max_{k \in [K]} \mathbb{E}\big[R_T(k)\big] \leqslant \mathbb{E}\big[\max_{k \in [K]} R_T(k)\big]$ and thus $\bar{R}_T \leqslant \mathbb{E}[R_T]$.

How to use EWA for bandits?

The Exponentially Weighted Average (EWA) forecaster

$$\boxed{p_t(k) = \frac{e^{-\eta \sum_{s=1}^{t-1} g_s(k)}}{\sum_{j=1}^{K} e^{-\eta \sum_{s=1}^{t-1} g_s(j)}}} \qquad \text{(EWA)}$$

Question: Can we use directly $p_t(k)$ as defined by EWA with $g_t = (f_t(1), \dots, f_t(K))$ and sample $k_t \sim p_t$ as we did for random EWA?

□ Yes □ No

How to use EWA for bandits?

The Exponentially Weighted Average (EWA) forecaster

$$p_t(k) = \frac{e^{-\eta \sum_{s=1}^{t-1} g_s(k)}}{\sum_{j=1}^{K} e^{-\eta \sum_{s=1}^{t-1} g_s(j)}} \tag{EWA}$$

Question: Can we use directly $p_t(k)$ as defined by EWA with $g_t = (f_t(1), \dots, f_t(K))$ and sample $k_t \sim p_t$ as we did for random EWA?

Answer: No, since the player does not observe $f_t(k)$ for $k \neq k_t$ and cannot compute p_t.

How to use EWA for bandits?

The Exponentially Weighted Average (EWA) forecaster

$$\boxed{p_t(k) = \frac{e^{-\eta \sum_{s=1}^{t-1} g_s(k)}}{\sum_{j=1}^{K} e^{-\eta \sum_{s=1}^{t-1} g_s(j)}}} \tag{EWA}$$

Question: What about setting using $f_t(k)$ if we observe it and 0 otherwise:

$$g_t(k) = \begin{cases} f_t(k) & \text{if } k = k_t \quad \leftarrow \text{i.e., decision } k \text{ is observed} \\ 0 & \text{otherwise} \end{cases} \quad ?$$

□ Yes □ No

How to use EWA for bandits?

The Exponentially Weighted Average (EWA) forecaster

$$\boxed{p_t(k) = \frac{e^{-\eta \sum_{s=1}^{t-1} g_s(k)}}{\sum_{j=1}^{K} e^{-\eta \sum_{s=1}^{t-1} g_s(j)}}} \tag{EWA}$$

Question: What about setting using $f_t(k)$ if we observe it and 0 otherwise:

$$g_t(k) = \begin{cases} f_t(k) & \text{if } k = k_t \quad \leftarrow \text{i.e., decision } k \text{ is observed} \\ 0 & \text{otherwise} \end{cases} \quad ?$$

Answer: No, because this estimate would be biased:

$$\mathbb{E}_{k_t \sim p_t}\big[g_t(k_t)\big] = p_t(k) f_t(k) \neq f_t(k)\,.$$

In other words, the actions that are less likely to be chosen by the algorithm (small weight $p_t(k)$) are more likely to be unobserved and incur 0 loss. We need to correct this phenomenon.

How to use EWA for bandits?

The Exponentially Weighted Average (EWA) forecaster

$$\boxed{p_t(k) = \frac{e^{-\eta \sum_{s=1}^{t-1} g_s(k)}}{\sum_{j=1}^{K} e^{-\eta \sum_{s=1}^{t-1} g_s(j)}}} \tag{EWA}$$

Therefore, we choose

$$\boxed{g_t(k) = \frac{f_t(k)}{p_t(k)} \mathbb{1}\{k = k_t\}\,,}$$

which leads to the algorithm EXP3 detailed below.

Exponential Weights for bandits

EXP3

Parameter: $\eta > 0$

Initialize: $p_1 = \left(\frac{1}{K}, \dots, \frac{1}{K}\right)$

For $t = 1, \dots, T$

- draw $k_t \sim p_t$; incur loss $f_t(k_t)$ and observe $f_t(k_t) \in [0, 1]$;
- update for all $k \in \{1, \dots, K\}$

$$p_{t+1}(k) = \frac{e^{-\eta \sum_{s=1}^{t} g_s(k)}}{\sum_{j=1}^{K} e^{-\eta \sum_{s=1}^{t} g_s(j)}}, \quad \text{where } g_s(k) = \frac{f_s(k)}{p_s(k)} \mathbb{1}\{k = k_s\}$$

Pseudo-Regret bound for EXP3

$$p_{t+1}(k) = \frac{e^{-\eta \sum_{s=1}^{t} g_s(k)}}{\sum_{j=1}^{K} e^{-\eta \sum_{s=1}^{t} g_s(j)}}, \quad \text{where } g_s(k) = \frac{f_s(k)}{p_s(k)} \mathbb{1}\{k = k_s\} \tag{EXP3}$$

Theorem 7

Let $T \geqslant 1$. The pseudo-regret of EXP3 run with $\eta = \sqrt{\frac{\log K}{KT}}$ is upper-bounded as:

$$\bar{R}_T \overset{\text{def}}{=} \max_{k \in [K]} \mathbb{E}\left[\sum_{t=1}^{T} f_t(k_t) - \sum_{t=1}^{T} f_t(k)\right] \leqslant 2\sqrt{KT \log K}\,.$$

Proof

Applying EWA to the estimated losses $g_t(j)$ that are completely observed and taking the expectation:

$$\mathbb{E}\left[\sum_{t=1}^{T} p_t \cdot g_t - \min_{j \in [K]} \sum_{t=1}^{T} g_t(j)\right] \leqslant \frac{\log K}{\eta} + \eta \sum_{t=1}^{T} \mathbb{E}\Big[p_t \cdot g_t^2\Big] \,. \tag{$*$}$$

The rest of the proof consists in computing the expectations:

$$\mathbb{E}\big[p_t \cdot g_t\big] = \mathbb{E}\big[f_t(k_t)\big]\,, \qquad \mathbb{E}\big[g_t(j)\big] = \mathbb{E}\big[f_t(j)\big] \qquad \text{and} \qquad \mathbb{E}\big[p_t \cdot g_t^2\big] \leqslant K \tag{5}$$

Proof

Denote by $\mathcal{F}_{t-1} \overset{\text{def}}{=} \sigma(p_1, f_1, k_1, \ldots, k_{t-1}, p_t, f_t)$ the past information available at round t for the adversary (which cannot use the randomness of k_t but can use p_t).

Note that f_t and p_t are $\mathcal{F}_{t-1}$-measurable by assumption.

1) Proof that $\mathbb{E}\big[g_t(j)\big] = \mathbb{E}\big[f_t(j)\big]$

$$\forall j \in [K] \qquad \mathbb{E}\left[g_t(j)\middle|\mathcal{F}_{t-1}\right] = \mathbb{E}\left[\frac{f_t(j)}{p_t(j)}\mathbb{1}\{j = k_t\}\middle|\mathcal{F}_{t-1}\right] = \sum_{k=1}^{K} p_t(k)\frac{f_s(j)}{p_t(j)}\mathbb{1}\{j = k\} = f_t(j)$$

2) Proof that $\mathbb{E}\big[p_t \cdot g_t\big] = \mathbb{E}\big[f_t(k_t)\big]$

$$\begin{aligned}\mathbb{E}\Big[p_t \cdot g_t\Big] &= \mathbb{E}\left[\sum_{j=1}^{K} p_t(j) g_t(j)\right] = \mathbb{E}\left[\sum_{j=1}^{K} p_t(j)\mathbb{E}\left[g_t(j)\middle|\mathcal{F}_{t-1}\right]\right] \\ &= \mathbb{E}\left[\sum_{j=1}^{K} p_t(j) f_t(j)\right] = \mathbb{E}\Big[\mathbb{E}\big[f_t(k_t)\big|\mathcal{F}_{t-1}\big]\Big] = \mathbb{E}\big[f_t(k_t)\big]\,.\end{aligned}$$

Proof

Therefore, using

$$\mathbb{E}\big[p_t \cdot g_t\big] = \mathbb{E}\big[f_t(k_t)\big] \qquad \text{and} \qquad \mathbb{E}\big[g_t(j)\big] = \mathbb{E}\big[f_t(j)\big] \tag{6}$$

we have

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T} p_t \cdot g_t - \min_{j\in[K]} \sum_{t=1}^{T} g_t(j)\right] &\geqslant \max_{j\in[K]} \mathbb{E}\left[\sum_{t=1}^{T} p_t \cdot g_t - \sum_{t=1}^{T} g_t(j)\right] \\ &= \max_{j\in[K]} \mathbb{E}\left[\sum_{t=1}^{T} f_t(k_t) - \sum_{t=1}^{T} f_t(j)\right] = \bar{R}_T\,.\end{aligned}$$

Proof

3) Proof that $\mathbb{E}\left[p_t \cdot g_t^2\right] \leqslant K$

$$\begin{aligned}\mathbb{E}\left[p_t \cdot g_t^2\right] &= \mathbb{E}\left[\sum_{j=1}^{K} p_t(j) g_t(j)^2\right] = \mathbb{E}\left[\sum_{j=1}^{K} p_t(j)\, \mathbb{E}\left[g_t(j)^2 \,\middle|\, \mathcal{F}_{t-1}\right]\right] \\ &= \mathbb{E}\left[\sum_{j=1}^{K}\sum_{k=1}^{K} p_t(j) p_t(k) \left(\frac{f_t(j)}{p_t(j)} \mathbb{1}\{j = k\}\right)^2\right] \\ &= \mathbb{E}\left[\sum_{j=1}^{K}\sum_{k=1}^{K} p_t(k) \frac{f_t(j)^2}{p_t(j)} \mathbb{1}\{j = k\}\right] \\ &= \mathbb{E}\left[\sum_{j=1}^{K} f_t(j)^2\right] \leqslant K\,.\end{aligned}$$

4) Conclusion. Substituting into Inequality $(*)$ yields

$$\bar{R}_T \leqslant \frac{\log K}{\eta} + \eta K T\,.$$

and optimizing $\eta = \sqrt{KT/(\log K)}$ concludes.

Limit of the result

The issue with the above regret bound is that it bounds the pseudo-regret and not the expected regret. This is because we have

$$\mathbb{E}\left[\min_j \sum_{t=1}^{T} g_t(j)\right] \leqslant \min_j \mathbb{E}\left[\sum_{t=1}^{T} g_t(j)\right] = \min_{j\in[K]} \mathbb{E}\left[\sum_{t=1}^{T} f_t(j)\right]$$

but not

$$\mathbb{E}\left[\min_j \sum_{t=1}^{T} g_t(j)\right] \not\leqslant \mathbb{E}\left[\min_j \sum_{t=1}^{T} f_t(j)\right]. \tag{7}$$

Hence, controlling the cumulative loss agains the best estimated action only controls the pseudo regret and not the true regret.

EXP3.P

EXP3.P

```
Parameters: $\eta > 0, \beta \in (0,1), \gamma \in (0,1)$
Initialize: $p_1 = \left(\frac{1}{K}, \dots, \frac{1}{K}\right)$

For $t = 1, \dots, T$
  - draw $k_t \sim p_t$; receive reward $r_t(k_t) = 1 - f_t(k_t)$ and observe $r_t(k_t) \in [0,1]$;
  - update for all $k \in \{1, \dots, K\}$

      $p_{t+1}(k) = (1-\gamma) \frac{e^{\eta \sum_{s=1}^{t} g_s(k)}}{\sum_{j=1}^{K} e^{\eta \sum_{s=1}^{t} g_s(j)}} + \frac{\gamma}{K}$,

    where $g_s(k) = \frac{r_s(k) \mathbb{1}\{k = k_s\} + \beta}{p_s(k)}$.
```

The weights $p_t(k)$ of EXP3.P are necessary larger than γ/K and thus $|\eta g_t(j)| \leqslant 1$ as soon as $\eta(1+\beta)K/\gamma \leqslant 1$.

Regret bound for Exp3.P

Theorem 8

For well-chosen parameters $\gamma \in (0,1)$, $\beta \in (0,1)$ and $\eta > 0$ satisfying $\eta(1+\beta)K/\gamma \leqslant 1$, for any $\delta > 0$, the EXP3.P algorithm achieves

$$R_T \leqslant 6\sqrt{TK \log K} + \sqrt{\frac{TK}{\log K}} \log(1/\delta)\,.$$

with probability at least $1-\delta$.

With the choice $\delta = 1/T$ it yields

$$\mathbb{E}\left[R_T\right] \leqslant 6\sqrt{TK \log K} + \sqrt{\frac{TK}{\log K}} \log(T) + 1$$

Introduction: What is online learning?

Online Linear Optimization

Online Convex Optimization

Adversarial bandits

- The exponentially weighted average algorithm for bandits
- Adversarial bandits with experts
- OGD without Gradients

Stochastic bandits

Setting of adversarial bandits with experts

Setting

At each time step $t = 1, \dots, T$

- N experts propose recommendations $h_t(i) \in [K]$ for $i \in [N]$
- the environment chooses a loss function $f_t : \mathcal{K} \to [0,1]$;
- the player chooses an action $k_t \in [K]$
- the player suffers loss $f_t(k_t)$
- the player observes the loss of the chosen action only: $f_t(k_t)$

Goal: compete with the best expert, i.e., minimize

$$R_T^{\text{exp}} \overset{\text{def}}{=} \max_{i=1,\dots,N} \mathbb{E}\left[\sum_{t=1}^{T} f_t(k_t) - \sum_{t=1}^{T} f_t\big(h_t(i)\big)\right]$$

with respect to the experts.

EXP3 solution

By using EXP3 on the set of experts instead of the set of actions, we would get

$$\bar{R}_T \leqslant \sqrt{TN \log N}\,.$$

However it does not take into account the information on the reward of all experts that choose the same action $h_t(i) = k_t$.

EXP4

EXP4

Parameter: $\eta > 0$ Initialize: $q_1 = \left(\frac{1}{N}, \dots, \frac{1}{N}\right)$.

For each round $t = 1, \dots, n$

1. Get expert advice $h_t(1), \dots, h_t(N) \in [K]$
2. Draw an expert i_t with probability distribution $q_t \in \Delta_N$
3. Choose decision $k_t = h_t(i_t)$
4. Compute the estimated loss for each decision

$$g_t(k) = \frac{f_t(k)}{p_t(k)} \mathbb{1}\{k = k_t\},$$

   where $p_t \overset{\text{def}}{=} \sum_{i=1}^N q_t(i)\delta_{f_t(i)} \in \Delta_K$.

5. Compute the estimated loss of the experts component-wise $g_t(h_t(i))$
6. Update the probability distribution over the experts component-wise

$$q_{t+1}(i) = \frac{\exp\Big(-\eta \sum_{s=1}^t g_s\big(h_s(i)\big)\Big)}{\sum_{j=1}^N \exp\Big(\eta \sum_{s=1}^t g_s\big(h_s(j)\big)\Big)}, \qquad \forall 1 \leqslant i \leqslant N.$$

Regret of EXP4

Theorem 9

EXP4 with $\eta = \sqrt{\log N/(KT)}$ *satisfies* $R_T^{exp} \leqslant 2\sqrt{TK \log N}$.

Proof left as exercise.

Introduction: What is online learning?

Online Linear Optimization

Online Convex Optimization

Adversarial bandits

- The exponentially weighted average algorithm for bandits
- Adversarial bandits with experts
- OGD without Gradients

Stochastic bandits

Beyond finite set of actions?

At each time step $t = 1, \ldots, T$

- the player observes a context $c_t \in \mathcal{X}$ (optional step)
- the player chooses an action $x_t \in \mathcal{K}$ (compact decision/parameter set);
- the environment chooses a loss function $f_t : \mathcal{K} \to [0,1]$;
- the player suffers loss $f_t(x_t)$ and observes
  - the losses of every actions: $f_t(x)$ for all $x \in \mathcal{K}$ → full-information feedback
  - the loss of the chosen action only: $f_t(x_t)$ → bandit feedback.

The goal of the player is to minimize his cumulative loss:

$$\widehat{L}_T \overset{\text{def}}{=} \sum_{t=1}^{T} f_t(x_t)\,.$$

This lecture: we saw variants of EXP3 when $\mathcal{K}$ is finite.

What if the losses f_t are convex but $\mathcal{K}$ is any bounded convex set in $\mathbb{R}^d$?

Online Gradient Descent

In the full information setting (when gradient can be observed), we saw OGD algorithm:

$$x_{t+1} \leftarrow \mathrm{Proj}_{\mathcal{K}}\left(x_t - \eta \nabla f_t(x_t)\right)$$

Theorem 10 (Regret of OGD)

Let $D, G, \eta > 0$. Assume that $\mathcal{K}$ has diameter bounded by D and the convex losses have sub-Gradients bounded by G in f_2-norm, the regret of OGD satisfies

$$\sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x) \leqslant DG\sqrt{T}.$$

How to adapt this algorithm to the bandit setting? That is, when only $f_t(x_t)$ are observed and not $\nabla f_t(x_t)$?

Point-wise gradient estimators

$$\boxed{x_{t+1} \leftarrow \mathrm{Proj}_{\mathcal{K}}\left(x_t - \eta \nabla f_t(x_t)\right)}$$

Similarly to EXP3, the idea is to replace the gradient in OGD with unbiased estimators. That is try to find an observable random variable $\widehat{g}_t$ that satisfies

$$\mathbb{E}[\widehat{g}_t] \approx \nabla f_t(x_t)$$

Example: one-dimensional gradient estimate

$$f'(x) = \lim_{\delta \to 0} \frac{f(x+\delta) - f(x-\delta)}{2\delta} .$$

Thus we can define

$$\widehat{g}(x) = \begin{cases} \frac{f(x+\delta)}{\delta} & \text{with proba } \frac{1}{2} \\ -\frac{f(x-\delta)}{\delta} & \text{with proba } \frac{1}{2} \end{cases} \quad \text{which yields} \quad \mathbb{E}[\widehat{g}(x)] = \frac{f(x+\delta) - f(x-\delta)}{2\delta} .$$

Thus in expectation, for small δ, $\widehat{g}(x)$ approximates $f'(x)$.

Point-wise gradient estimators: multi-dimensional case

We show here how the one-dimensional pointwise gradient estimator can be extended to the multi-dimensional case.

We define $\widehat{f_t}$ to be a smoothed version of the loss:

$$\widehat{f_t}(x) = \mathbb{E}_v\left[f_t(x + \delta v)\right]$$

where $v \sim \mathit{Unif}(\mathbb{B})$. If δ is small, $\widehat{f_t}$ is a good approximation of f_t.

Lemma 1

Let $\widehat{f_t}(x) = \mathbb{E}\left[f_t(x + \delta v)\right]$ where $v \sim \mathit{Unif}(\mathbb{B})$ be a smoothed version of the loss, then

$$\mathbb{E}_u\left[\frac{d}{\delta} f_t(x_t + \delta u) u\right] = \nabla \widehat{f_t}(x)\,.$$

Proof.

Left as exercise. See Lem. 6.7, Hazan et al., "Introduction to online convex optimization", 2016. □

OGD without Gradients

Similarly to EXP3, the idea is to replace the gradient in OGD with unbiased estimators.

OGD without gradients

```
For $t = 1, \dots, T$
    - Draw $u_t \in \mathbb{S}$ uniformly at random in the unit sphere
    - Set $\widehat{x}_t = x_t + \delta u_t$ a random perturbation of the current point $x_t$
    - Play $\widehat{x}_t$
    - Estimate the gradient in $x_t$ with
          $\widehat{g}_t = \frac{d}{\delta} f_t(\widehat{x}_t) u_t$
    - Update
          $x_{t+1} \leftarrow \mathrm{Proj}_{\mathcal{K}_\delta}(x_t - \eta \widehat{g}_t)$
    where $\mathcal{K}_\delta = \{x \in \mathcal{K} \quad \text{s.t} \quad x + \delta u \in \mathcal{K} \quad \forall u \in \mathbb{S}\}$
```

Regret of OGD without gradients

OGD without gradients:

$$x_{t+1} \leftarrow \mathrm{Proj}_{\mathcal{K}_\delta}\left(x_t - \eta \widehat{g}_t\right) \qquad \text{where} \quad \widehat{g}_t = \frac{d}{\eta} f_t(\widehat{x}_t) u_t \text{ and } \widehat{x}_t = x_t + \delta u_t$$

Theorem 11

If the losses are in $[-1,1]$ *and* G*-Lipschitz, OGD without gradients with parameters* $\delta = \min\{D, (1/2)\sqrt{Dd/G}\,T^{-1/4}\}$ *and* $\eta = D\delta/(dT^{1/2})$ *satisfies the expected regret bound*

$$\sum_{t=1}^{T} \mathbb{E}\left[f_t(\widehat{x}_t)\right] - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x) \leqslant 2d\sqrt{T} + 2\sqrt{GDd}\,T^{3/4}\,.$$

Proof (Step 1)

Denote

$$x^* \in \arg\min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x) \qquad \text{and} \qquad x_\delta^* = \mathrm{Proj}_{\mathcal{K}_\delta}(x^*)\,.$$

Then,

$$\left\|x^* - x_\delta^*\right\| \leqslant \delta$$

Thus, if the losses are G-Lipschitz

$$\begin{aligned} R_T := \sum_{t=1}^{T} \mathbb{E}\big[f_t(\widehat{x}_t)\big] - \sum_{t=1}^{T} f_t(x^*) &\leqslant \sum_{t=1}^{T} \mathbb{E}\big[f_t(\widehat{x}_t)\big] - \sum_{t=1}^{T} f_t(x_\delta^*) \\ &\leqslant \sum_{t=1}^{T} \mathbb{E}\big[f_t(x_t)\big] - \sum_{t=1}^{T} f_t(x_\delta^*) + \delta TG \\ &\leqslant \sum_{t=1}^{T} \mathbb{E}\big[\widehat{f}_t(x_t)\big] - \sum_{t=1}^{T} \widehat{f}_t(x_\delta^*) + 3\delta TG \qquad (*) \end{aligned}$$

where $\widehat{f}_t(x) = \mathbb{E}_v\big[f_t(x + \delta v)\big]$ with $v \sim \mathit{Unif}(\mathbb{B})$ are the smoothed versions of the losses.

Proof (Step 2)

Now, recall that the algorithm runs OGD with $\widehat{g}_t$ in place of the gradients:

$$x_{t+1} \leftarrow \mathrm{Proj}_{\mathcal{K}_\delta}\left(x_t - \eta \widehat{g}_t\right)$$

Defining the pseudo-loss $h_t(x) = \widehat{f}_t(x) + \left(\widehat{g}_t - \nabla \widehat{f}_t(x_t)\right)^\top x$, we can see that

$$\nabla h_t(x_t) = \nabla \widehat{f}_t(x_t) + \widehat{g}_t - \nabla \widehat{f}_t(x_t) = \widehat{g}_t .$$

Therefore, the algorithm actually runs OGD on the losses h_t and thus satisfies the OGD regret bound (see Lecture 2)

$$\sum_{t=1}^{T} h_t(x_t) - \sum_{t=1}^{T} h_t(x_\delta^*) \leqslant \frac{D^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\widehat{g}_t\|^2 .$$

Furthermore, by construction of the gradient estimator, we have $\mathbb{E}_{u_t}\left[\widehat{g}_t\right] = \nabla \widehat{f}_t(x_t)$, which yields

$$\mathbb{E}_{u_t}\left[h_t(x_t)\right] = \widehat{f}_t(x_t) \quad \text{and} \quad \mathbb{E}_{u_t}\left[h_t(x_\delta^*)\right] = \widehat{f}_t(x_\delta^*)$$

Thus taking the expectation in the previous regret bound entails

$$\sum_{t=1}^{T} \mathbb{E}\left[\widehat{f}_t(x_t)\right] - \sum_{t=1}^{T} \widehat{f}_t(x_\delta^*) = \mathbb{E}\left[\sum_{t=1}^{T} h_t(x_t) - \sum_{t=1}^{T} h_t(x_\delta^*)\right] \leqslant \frac{D^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \mathbb{E}\left[\|\widehat{g}_t\|^2\right] \qquad (**)$$

Proof (Step 3)

Combining the two bounds (*) and (**) that we have proved, we get

$$R_T \leqslant \frac{D^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \mathbb{E}\left[\|\widehat{g}_t\|^2\right] + 3\delta T G$$

Then, since $|f_t(x)| \leqslant 1$ for all $x \in \mathcal{K}$,

$$\|\widehat{g}_t\|^2 = \left(\frac{d}{\delta} f_t(\widehat{x}_t)\right)^2 \leqslant \frac{d^2}{\delta^2}$$

This finally yields the regret

$$R_T \leqslant \frac{D^2}{2\eta} + \frac{\eta d^2 T}{2\delta^2} + 3\delta T G \leqslant 2d\sqrt{T} + 2\sqrt{GDd}\, T^{3/4}$$

for the choices of δ and η.

More on convex bandits

Convex bandits is still an active research area with many open problems.

The above regret bound of order $O(T^{3/4})$ is suboptimal.

More complicated methods can achieve $O(\sqrt{T})$ regret but with sub-optimal dependence on d and worst computational complexities.

More information can be found in Hazan et al., “Introduction to online convex optimization”, 2016.

Introduction: What is online learning?

Online Linear Optimization

Online Convex Optimization

Adversarial bandits

Stochastic bandits

Introduction: What is online learning?

Online Linear Optimization

Online Convex Optimization

Adversarial bandits

Stochastic bandits

- The stochastic bandit problem
- Consequences of the stochasticity
- Exploration and exploitation
- Optimism in face of uncertainty: UCB
- Stochastic Linear Bandits

Online learning / adversarial bandit

At each time step $t = 1, \ldots, T$

- the player observes a context $c_t \in \mathcal{X}$ (optional step)
- the player chooses an action $x_t \in \mathcal{K}$ (compact decision/parameter set);
- the environment chooses a loss function $f_t : \mathcal{K} \to [0, 1]$;
- the player suffers loss $f_t(x_t)$ and observes
  - the losses of every actions: $f_t(x)$ for all $x \in \mathcal{K}$ → full-information feedback
  - the loss of the chosen action only: $f_t(x_t)$ → bandit feedback.

The goal of the player is to minimize his cumulative loss:

$$\widehat{L}_T \overset{\text{def}}{=} \sum_{t=1}^{T} f_t(x_t)\,.$$

Stochastic bandit

At each time step $t = 1, \dots, T$

- the player observes a context $c_t \in \mathcal{X}$ (optional step)
- the player chooses an arm $k_t \in \mathcal{K}$ (compact decision/parameter set, most often $\{1, \dots, K\}$);
- the player observes
  - the rewards of every arm: $X_t^k \sim \nu_k$ for all $k \in \mathcal{K}$ → full-information feedback
  - the reward of the chosen arm only: $X_t^{k_t} \sim \nu_{k_t}$ → bandit feedback.

The goal of the player is to maximize their cumulative reward.

Regret?

We could use the definition of the regret from adversarial bandits:

Definition (Regret, attempt 1)

$$R_T = \max_k \sum_{t=1}^{T} X_t^k - \sum_{t=1}^{T} X_t^{k_t} \ .$$

Let's see why we don't use that definition.

Notations and assumptions:

- The arm set is $[K] = \{1, \ldots, K\}$.
- $\mu^k = \mathbb{E}_{X \sim \nu_k}[X]$, assumed finite for all arms k.
- $\mu^* = \max_{k \in [K]} \mu^k$.

The first notion of regret is inadequate

$$R_T = \max_k \sum_{t=1}^{T} X_t^k - \sum_{t=1}^{T} X_t^{k_t} .$$

ν_k Bernoulli(1/2) for all $k \in [K]$. $\mu^k = 1/2$ for all k.

All arms are the same → there is no bad choice and no bad algorithm.

But:

$$\begin{aligned}
\mathbb{E} R_T &= \mathbb{E}[\max_{k \in [K]} \sum_{t=1}^{T} X_t^k] - T/2 \\
&= \mathbb{E}[\max_{k \in [K]} \sum_{t=1}^{T} (X_t^k - 1/2)] \\
&\approx \sqrt{T \log K}
\end{aligned}$$

(See any course/book/wikipedia article on symmetric random walks).

Regret definition

We want a regret notion that does not blow up with stochastic fluctuations.

Definition ((Pseudo)-Regret)

The regret is defined as

$$R_T = \max_k \sum_{t=1}^{T} \mu^k - \sum_{t=1}^{T} \mu^{k_t} = T\mu^* - \sum_{t=1}^{T} \mu^{k_t} .$$

Recall that $\mu^k = \mathbb{E}_{X\sim\nu_k}[X]$.

Most often, we bound the expected regret $\mathbb{E}[R_T]$.

Note that the expectation here is over the random rewards and the randomness of the algorithm, if there is any.

Regret decomposition

Suppose that the set of arms is finite: $[K]$.

Define the gap of arm $k \in [K]$ by $\Delta_k = \mu^* - \mu^k$.

$$R_T = T\mu^* - \sum_{t=1}^{T} \mu^{k_t} = \sum_{t=1}^{T} (\mu^* - \mu^{k_t}) = \sum_{t=1}^{T} \Delta_{k_t} = \sum_{k=1}^{K} N_T^k \Delta_k \ ,$$

where $N_T^k = \sum_{t=1}^{T} \mathbb{I}\{k_t = k\}$ is the number of pulls of arm k up to time T.

Bounding the regret ⇔ bounding the number of pulls of bad arms

Stochastic bandit

At each time step $t = 1, \ldots, T$

- the player observes a context $c_t \in \mathcal{X}$ (optional step)
- the player chooses an arm $k_t \in \mathcal{K}$ (compact decision/parameter set, most often $\{1, \ldots, K\}$);
- the player observes the reward of the chosen arm only: $X_t^{k_t} \sim \nu_{k_t}$ (independent of other rewards).

The goal of the player is to minimize their expected regret: $\mathbb{E}[R_T] = \sum_{k=1}^{K} \mathbb{E}[N_T^k]\Delta_k$.

Variants and extensions

Setting variants:

- Contextual bandit: $X_t^{k_t} \sim \nu_{k_t}(c_t)$, for a known context c_t
- Linear bandit: $\nu_{k_t} = \mathcal{N}(x^\top c_{k_t}, 1)$
- Structured bandit: the algorithm knows constraints on $(\mu^k)_{k\in[K]}$, e.g. Lipschitz, linear, monotone. . .

Goal variants: instead of minimizing the regret, we want to

- Minimize the simple regret: return an arm at time T, and minimize its expected gap.
- Identify the best arm: return an arm at time T, and minimize the probability that its not one of the best ones.

Relaxed assumptions: rewards not independent, distributions changing with time, etc.

Introduction: What is online learning?

Online Linear Optimization

Online Convex Optimization

Adversarial bandits

Stochastic bandits

- The stochastic bandit problem
- Consequences of the stochasticity
- Exploration and exploitation
- Optimism in face of uncertainty: UCB
- Stochastic Linear Bandits

Convergence to the mean

Main idea: we can estimate the mean of the arms with the empirical mean.

Let $(X_s)_{s \in \mathbb{N}}$ be iid random variables with $\mathbb{E}[|X_1|] < \infty$ and expectated value $\mathbb{E}[X_1] = \mu$.

Let $\bar{X}_t = \sum_{s=1}^{t} X_s$.

Theorem 12 (Strong law of large numbers)

$\bar{X}_t \xrightarrow{a.s.} \mu$, that is $\mathbb{P}(\bar{X}_t \to \mu) = 1$.

Theorem 13 (Central limit theorem)

If $\mathbb{V}[X] = \sigma^2 < \infty$, then $\sqrt{t}(\bar{X}_t - \mu) \xrightarrow{d} \mathcal{N}(\mu, \sigma^2)$.

Problem: those are asymptotic results.

Main question: if I have 15 samples of arm k, how reliable is my estimate for μ^k ?

Concentration inequalities

Our main tools are concentration inequalities: bounds on the probability that the empirical mean (or another statistic) is far from its expected value.

Theorem 14 (Hoeffding's inequality)

If $X_1, \ldots, X_t$ are independent random variables almost surely in $[a, b]$ then for all $\delta \in (0, 1)$ we have

$$\mathbb{P}\left(\sum_{s=1}^{t} X_s - \mathbb{E}\left[\sum_{s=1}^{t} X_s\right] \geq (b-a)\sqrt{\frac{t}{2}\log\frac{1}{\delta}}\right) \leq \delta\,.$$

Equivalently, for all $\varepsilon \geq 0$,

$$\mathbb{P}\left(\sum_{s=1}^{t} X_s - \mathbb{E}\left[\sum_{s=1}^{t} X_s\right] \geq \varepsilon\right) \leq \exp\left(-\frac{2\varepsilon^2}{t(b-a)^2}\right)\,.$$

Proof

Proof under a sub-Gaussian assumption. Exercise: bounded implies sub-Gaussian.

Assumption: for all s, X_s is σ^2-sub-Gaussian, which means that for all $\lambda \in \mathbb{R}$,

$$\mathbb{E}[e^{\lambda(X_s - \mu_s)}] \leq e^{\frac{1}{2}\sigma^2\lambda^2}.$$

Proof

Warning: random number of samples

In the analysis of bandit algorithms, we want to bound $\widehat{\mu}_t^k - \mu^k$, where $\widehat{\mu}_t^k = \frac{1}{N_t^k} \sum_{s=1}^t X_s^{k_s} \mathbb{I}\{k_s = k\}$.

k_s is a random variable that depends on all previous rewards.

Issue: $\widehat{\mu}_t^k$ is a sum of a random number of random variables which are not independent.

- $\widehat{\mu}_t^k$ is not unbiased.
- $\widehat{\mu}_t^k$ is not a sum of a fixed number of independent random variables.
- Hoeffding's inequality does not apply.

How to avoid the difficulty: union bounds, or martingale arguments (see proofs later in the course).

Introduction: What is online learning?

Online Linear Optimization

Online Convex Optimization

Adversarial bandits

Stochastic bandits

- The stochastic bandit problem
- Consequences of the stochasticity
- Exploration and exploitation
- Optimism in face of uncertainty: UCB
- Stochastic Linear Bandits

Follow the leader

Goal: minimize $\mathbb{E}[R_T] = T\mu^* - \sum_{t=1}^{T} \mu^{k_t}$.

Since the empirical mean of an arm concentrate around its expected value, can we simply pull the arm with highest empirical mean?

Definition (Follow-The-Leader)

The FTL algorithm first explores each arm once $k_t = t$ for $k \leqslant K$ and then pulls arm $k_t = \arg\max_{k \in [K]} \widehat{\mu}_{t-1}^{k}$ for all $t \geqslant K + 1$.

Full information: yes, FTL is optimal.

Bandit: answer is no, FTL does not work. It has linear expected regret in most settings.

FTL still fails

Explore then commit

Need to not only exploit, but also explore.

Explore-Then-Commit

Parameter: $m \geqslant 1$.

1. Exploration

- For rounds $t = 1, \dots, mK$ explore by drawing each arm m times.
- Compute for each arm k its empirical mean of rewards obtained by pulling arm k m times

$$\widehat{\mu}^k_{mK} = \frac{1}{m} \sum_{s=1}^{Km} \mathbb{I}\{k_s = k\} X^k_s \, .$$

2. Exploitation: keep playing the best arm $\arg\max_k \widehat{\mu}^k_{mK}$ for the remaining rounds $t = mK + 1, \dots, T$.

Regret of ETC

Theorem 15 (Thm 6.1, Lattimore and Szepesvári, "Bandit algorithms", 2019)

If all distributions are bounded in $[0,1]$ *and* $1 \leqslant m \leqslant T/K$ *then ETC has expected regret*

$$\mathbb{E}[R_T] \leqslant m \sum_{k=1}^{K} \Delta_k + (T - mK) \sum_{k=1}^{K} \Delta_k \exp\left(- m\Delta_k^2 \right) .$$

- m too large $\Rightarrow$ too much exploration, linear regret.
- m too small $\Rightarrow$ too little exploration, linear regret.
- What m should we choose?

Proof

Proof

Finding the right trade-off

Two arms bandit: arm 1 is the best arm, arm 2 has gap Δ.

ETC verifies

$$\mathbb{E}[R_T] \leqslant m\Delta + (T - 2m)\Delta e^{-m\Delta^2} .$$

Theorem 16

If $K = 2$ and $m = \max\{1, \left\lceil \frac{\log(T\Delta^2)}{\Delta^2} \right\rceil\}$, then

$$\mathbb{E}[R_T] \leq \Delta + \frac{1 + \log(T\Delta^2)}{\Delta} .$$

This is a distribution dependent bound, meaning that it depends on the gap.

Issue with those bounds: meaningless if Δ is small.

Worst case bound

ETC verifies

$$\mathbb{E}[R_T] \leqslant m\Delta + (T - 2m)\Delta e^{-m\Delta^2} .$$

Theorem 17

If $K = 2$ *and* $m = \max\left\{1, \left\lceil \frac{\log(T\Delta^2)}{\Delta^2} \right\rceil \right\}$, *then*

$$\mathbb{E}[R_T] \le \min\left\{\Delta + \frac{1 + \log(T\Delta^2)}{\Delta},\ T\Delta\right\} \lesssim \sqrt{T \log T} .$$

This is close to optimal: we can prove a lower bound of order $\sqrt{T}$.

Problems:

- m depends on Δ, which is unknown.
- What can we do for $K > 2$?

Introduction: What is online learning?

Online Linear Optimization

Online Convex Optimization

Adversarial bandits

Stochastic bandits

- The stochastic bandit problem
- Consequences of the stochasticity
- Exploration and exploitation
- Optimism in face of uncertainty: UCB
- Stochastic Linear Bandits

Homework

The homework is available on my webpage:

http://pierre.gaillard.me/teaching.html

It is due by **Dec. 22nd 2023**.

Upload your notebook using the form on my webpage:

http://pierre.gaillard.me/teaching/online_learning_uga.php

Being optimistic

The UCB (Upper-Confidence-Bound) algorithm is a very popular bandit algorithm that has several advantages over ETC:

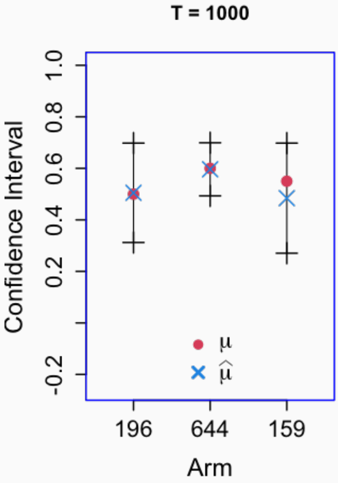
- It does not rely on an initial exploration phase but explores on the fly as rewards are observed.
- Unlike ETC, it does not require knowledge of gaps and behaves well when there are more than two arms.

For each arm k, it builds a confidence interval on its expected reward based on past observation

$$I_t^k = \left[L_t^k, U_t^k\right] .$$

It is optimistic, acting as if the best possible rewards are the real rewards:

$$k_t \in \underset{k \in \{1,\dots,K\}}{\arg\max} \; U_t^k .$$



Being optimistic

The UCB (Upper-Confidence-Bound) algorithm is a very popular bandit algorithm that has several advantages over ETC:

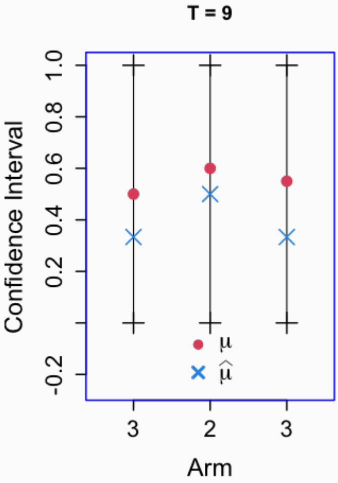
- It does not rely on an initial exploration phase but explores on the fly as rewards are observed.
- Unlike ETC, it does not require knowledge of gaps and behaves well when there are more than two arms.

For each arm k, it builds a confidence interval on its expected reward based on past observation

$$I_t^k = \left[L_t^k, U_t^k\right] .$$

It is optimistic, acting as if the best possible rewards are the real rewards:

$$k_t \in \underset{k \in \{1,\dots,K\}}{\arg\max} \; U_t^k .$$



Being optimistic

The UCB (Upper-Confidence-Bound) algorithm is a very popular bandit algorithm that has several advantages over ETC:

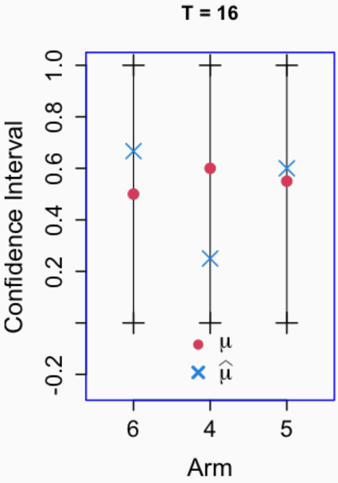
- It does not rely on an initial exploration phase but explores on the fly as rewards are observed.
- Unlike ETC, it does not require knowledge of gaps and behaves well when there are more than two arms.

For each arm k, it builds a confidence interval on its expected reward based on past observation

$$I_t^k = \left[L_t^k, U_t^k\right] .$$

It is optimistic, acting as if the best possible rewards are the real rewards:

$$k_t \in \underset{k \in \{1,\dots,K\}}{\arg\max} \; U_t^k .$$



Being optimistic

The UCB (Upper-Confidence-Bound) algorithm is a very popular bandit algorithm that has several advantages over ETC:

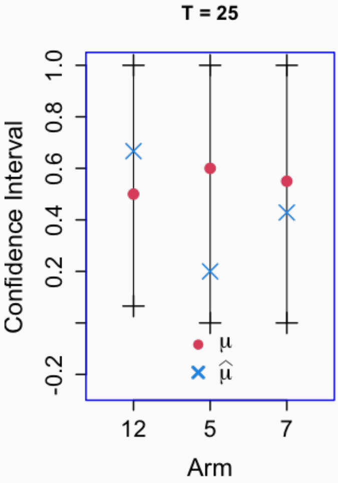
- It does not rely on an initial exploration phase but explores on the fly as rewards are observed.
- Unlike ETC, it does not require knowledge of gaps and behaves well when there are more than two arms.

For each arm k, it builds a confidence interval on its expected reward based on past observation

$$I_t^k = \left[L_t^k, U_t^k\right] .$$

It is optimistic, acting as if the best possible rewards are the real rewards:

$$k_t \in \underset{k\in\{1,\dots,K\}}{\arg\max} \; U_t^k .$$



Being optimistic

The UCB (Upper-Confidence-Bound) algorithm is a very popular bandit algorithm that has several advantages over ETC:

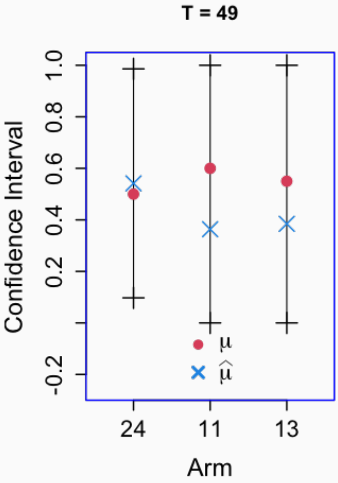
- It does not rely on an initial exploration phase but explores on the fly as rewards are observed.
- Unlike ETC, it does not require knowledge of gaps and behaves well when there are more than two arms.

For each arm k, it builds a confidence interval on its expected reward based on past observation

$$I_t^k = \left[L_t^k, U_t^k\right] .$$

It is optimistic, acting as if the best possible rewards are the real rewards:

$$k_t \in \underset{k \in \{1,\ldots,K\}}{\arg\max} \; U_t^k .$$



Being optimistic

The UCB (Upper-Confidence-Bound) algorithm is a very popular bandit algorithm that has several advantages over ETC:

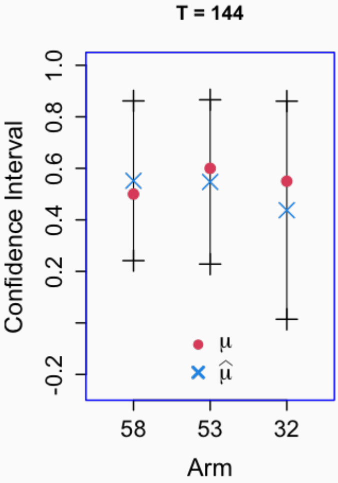
- It does not rely on an initial exploration phase but explores on the fly as rewards are observed.
- Unlike ETC, it does not require knowledge of gaps and behaves well when there are more than two arms.

For each arm k, it builds a confidence interval on its expected reward based on past observation

$$I_t^k = \left[L_t^k, U_t^k\right] .$$

It is optimistic, acting as if the best possible rewards are the real rewards:

$$k_t \in \underset{k \in \{1,\dots,K\}}{\arg\max} \; U_t^k .$$



Being optimistic

The UCB (Upper-Confidence-Bound) algorithm is a very popular bandit algorithm that has several advantages over ETC:

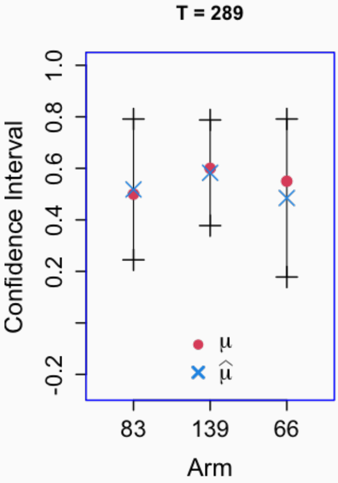
- It does not rely on an initial exploration phase but explores on the fly as rewards are observed.
- Unlike ETC, it does not require knowledge of gaps and behaves well when there are more than two arms.

For each arm k, it builds a confidence interval on its expected reward based on past observation

$$I_t^k = \left[L_t^k, U_t^k\right] .$$

It is optimistic, acting as if the best possible rewards are the real rewards:

$$k_t \in \underset{k\in\{1,\ldots,K\}}{\arg\max}\ U_t^k .$$



Being optimistic

The UCB (Upper-Confidence-Bound) algorithm is a very popular bandit algorithm that has several advantages over ETC:

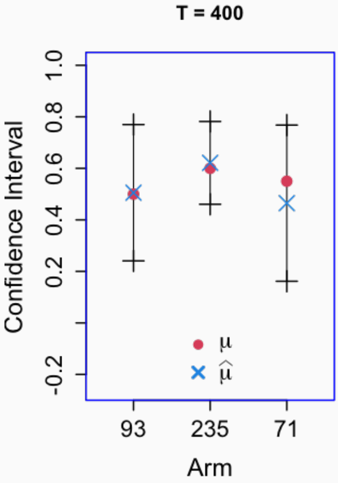
- It does not rely on an initial exploration phase but explores on the fly as rewards are observed.
- Unlike ETC, it does not require knowledge of gaps and behaves well when there are more than two arms.

For each arm k, it builds a confidence interval on its expected reward based on past observation

$$I_t^k = \left[L_t^k, U_t^k\right] .$$

It is optimistic, acting as if the best possible rewards are the real rewards:

$$k_t \in \underset{k\in\{1,\dots,K\}}{\arg\max} \; U_t^k .$$



Being optimistic

The UCB (Upper-Confidence-Bound) algorithm is a very popular bandit algorithm that has several advantages over ETC:

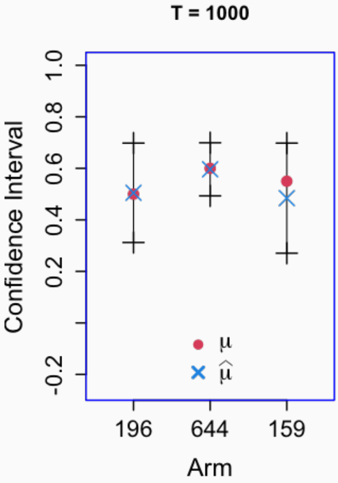
- It does not rely on an initial exploration phase but explores on the fly as rewards are observed.
- Unlike ETC, it does not require knowledge of gaps and behaves well when there are more than two arms.

For each arm k, it builds a confidence interval on its expected reward based on past observation

$$I_t^k = \left[L_t^k, U_t^k\right] .$$

It is optimistic, acting as if the best possible rewards are the real rewards:

$$k_t \in \underset{k\in\{1,\ldots,K\}}{\arg\max} \; U_t^k .$$



Confidence intervals

How to design the upper confidence bounds?

→ concentration inequalities. Here Hoeffding's inequality.

Theorem 18 (Hoeffding's inequality)

If $X_1, \dots, X_t$ are independent random variables almost surely in $[a, b]$ with same mean μ then for all $\delta \in (0, 1)$ we have

$$\mathbb{P}\left(\frac{1}{t} \sum_{s=1}^{t} X_s - \mu \geq \sqrt{\frac{(b-a)^2}{2t} \log \frac{1}{\delta}} \right) \leq \delta \, .$$

Careful: UCB is adaptive, hence $\widehat{\mu}_t$ is not exactly a sum of independent random variables. But we will make it work.

For rewards in $[0, 1]$: $U_t^k = \widehat{\mu}_{t-1}^k + \sqrt{\frac{2 \log t}{N_{t-1}^k}}$

UCB

Initialization For rounds $t = 1, \ldots, K$ pull arm $k_t = t$.

For $t = K+1, \ldots, T$, choose

$$k_t \in \arg\max_{k \in [K]} \left\{ \widehat{\mu}_{t-1}^k + \sqrt{\frac{2 \log t}{N_{t-1}^k}} \right\} ,$$

and get reward $X_t^{k_t}$.

Regret Bound

Theorem 19

If the distributions ν_k have supports all included in $[0,1]$ then for all k such that $\Delta_k > 0$

$$\mathbb{E}\left[N_T^k\right] \leqslant \frac{8 \log T}{\Delta_k^2} + 2\,.$$

In particular, this implies that the expected regret of UCB is upper-bounded as

$$\mathbb{E}[R_T] \leqslant 2K + \sum_{k:\Delta_k>0} \frac{8 \log T}{\Delta_k}\,.$$

Remarks :

- we can also prove $\mathbb{E}[R_T] \lesssim \sqrt{KT \log(T)}$. Close to the optimal $O(\sqrt{KT})$.
- Deals with multiple gaps, without any knowledge of the gaps, unlike ETC.
- Bounded can be replaced by sub-Gaussian.

Proof start

Idea: if the means belong to the confidence intervals and the arms are pulled enough, the algorithm cannot pull a suboptimal arm.

We prove that if $k_t = k \neq *$, then one of these inequalities must be false:

$$\mu^* \leq \widehat{\mu}^*_{t-1} + \sqrt{\frac{2\log t}{N^*_{t-1}}} \qquad \leftarrow \mu^* \text{ smaller than UCB} \tag{i}$$

$$\mu^k \geq \widehat{\mu}^k_{t-1} - \sqrt{\frac{2\log t}{N^k_{t-1}}} \qquad \leftarrow \mu_k \text{ larger than LCB} \tag{ii}$$

$$N^k_{t-1} \geq \frac{8\log t}{\Delta_k^2} \qquad \leftarrow k \text{ played enough} \tag{iii}$$

Proof 2

$$\mu^* \leq \widehat{\mu}^*_{t-1} + \sqrt{\frac{2\log t}{N^*_{t-1}}} \quad \text{and} \quad \mu^k \geq \widehat{\mu}^k_{t-1} - \sqrt{\frac{2\log t}{N^k_{t-1}}} \quad \text{and} \quad N^k_{t-1} \geq \frac{8\log t}{\Delta_k^2}$$

Prove that if k is pulled at t, then there is a contradiction.

Proof 3: decomposition wrt events

One of these is false:

$$\mu^* \leq \widehat{\mu}^*_{t-1} + \sqrt{\frac{2\log t}{N^*_{t-1}}} \quad ; \quad \mu^k \geq \widehat{\mu}^k_{t-1} - \sqrt{\frac{2\log t}{N^k_{t-1}}} \quad ; \quad N^k_{t-1} \geq \frac{8\log t}{\Delta_k^2}$$

Then: $\mathbb{E}\big[N^k_T\big] \leqslant u + \sum_{t=u+1}^{T} \Big(\mathbb{P}\big\{\text{(i) is false}\big\} + \mathbb{P}\big\{\text{(ii) is false}\big\}\Big)$ for $u = \left\lceil \frac{8\log T}{\Delta_k^2} \right\rceil$.

Proof 4: probability of the concentration event

We show: $\mathbb{P}(\mu^k < \widehat{\mu}^k_{t-1} - \sqrt{\frac{2 \log t}{N^k_{t-1}}}) \leq t^{-3}$.

Proof summary

For $u = \frac{8\log T}{\Delta_k^2}$, $\mathbb{E}\left[N_T^k\right] \leqslant u + \sum_{t=u+1}^{T} \left(\mathbb{P}\left\{\mu^* > \widehat{\mu}_{t-1}^* + \sqrt{\frac{2\log t}{N_{t-1}^*}}\right\} + \mathbb{P}\left\{\mu^k < \widehat{\mu}_{t-1}^k - \sqrt{\frac{2\log t}{N_{t-1}^k}}\right\}\right)$.

Each of these probabilities is smaller than t^{-3}.

$$\mathbb{E}[R_T] \leq \frac{8\log T}{\Delta_k^2} + 2\sum_{t=u+1}^{T} \frac{1}{t^3} \leq \frac{8\log T}{\Delta_k^2} + 2\,.$$

The bound of the regret then comes from $\mathbb{E}[R_T] = \sum_k \mathbb{E}[N_T^k]\Delta_k$.

Other Algorithms: ε-greedy

ε-greedy

First choose a parameter $\varepsilon \in (0, 1)$, then at each round, select the arm with the highest empirical mean with probability ε (i.e., be greedy), and explore by playing a random arm with probability ε.

Works quite well in practice and is used in many application because of its simple implementation (in particular in reinforcement learning).

Choosing $\varepsilon \approx K/(\Delta^2 T)$ yields to an upper-bound of order $K \log T/\Delta^2$. However it requires the knowledge of Δ.

Other Algorithms: Thompson Sampling

Thompson Sampling

Thomson sampling was the first algorithm proposed for bandits by Thomson in 1933. It assumes a uniform prior over the expected rewards $\mu_i \in (0,1)$, then at each round $t \geqslant 1$, it

- computes $\widehat{\nu}_{k,t}$ the posterior distribution of the rewards of each arm k given the rewards observed so far;
- samples $x_{k,t} \sim \widehat{\nu}_{k,t}$ independently;
- selects $k_t \in \arg\max_{k\in\{1,\dots,K\}} x_{k,t}$.

Thomson sampling has a similar upper-bound of order $O(K \log T/\Delta)$ than the one achieved by UCB. Somewhat different proof techniques.

An advantage over UCB is the possibility of incorporating easily prior knowledge on the arms.

UCB proved easier to adapt to structured bandits (it can be hard to sample a posterior conditioned on structural information).

Introduction: What is online learning?

Online Linear Optimization

Online Convex Optimization

Adversarial bandits

Stochastic bandits

- The stochastic bandit problem
- Consequences of the stochasticity
- Exploration and exploitation
- Optimism in face of uncertainty: UCB
- Stochastic Linear Bandits

Stochastic Linear Bandits - Motivation

Main motivation: use contexts.

Unknown parameter: $\mu^* \in \mathbb{R}^d$.

At each time step $t = 1, \dots, T$

- the environment chooses $\mathcal{K}_t \subseteq \mathbb{R}^d$, the decision set;
- the player chooses an action $x_t \in \mathcal{K}_t$;
- given x_t, the environment draws the reward

$$X_t = x_t^\top \mu^* + \varepsilon_t$$

  where ε_t is i.i.d. 1-subgaussian noise. $(\forall \lambda > 0,\ \mathbb{E}\big[\exp(\lambda \varepsilon_t)\big] \leqslant \exp(\lambda^2/2))$
- the player only observes the feedback X_t.

The player wants to minimize its expected regret defined as

$$\mathbb{E}R_T \overset{\text{def}}{=} \mathbb{E}\left[\sum_{t=1}^{T} \max_{x \in \mathcal{K}_t} x^\top \mu^* - \sum_{t=1}^{T} x_t^\top \mu^*\right].$$

Examples

- Finite-armed bandit: if $\mathcal{K}_t = (e_1, \dots, e_d)$, unit vectors in $\mathbb{R}^d$ and $\mu^* = (\mu_1, \dots, \mu_d)$, we recover the setting of finite-armed bandit (with d arms).
- Contextual linear bandit: if $c_t \in \mathcal{X}$ is a context observed by the player and the reward function μ is of the form

$$\mu(x, x) = \psi(x, x)^\top \mu^*, \qquad \forall (x, x) \in [K] \times \mathcal{X}\,,$$

  for some unknown parameter $\mu^* \in \mathbb{R}^d$ and feature map $\psi : [K] \times \mathcal{X} \to \mathbb{R}^d$.
- Combinatorial bandit: $\mathcal{K}_t \subseteq \{0, 1\}^d \to$ combinatorial bandit problems. Example: decision set $=$ possible paths in a graph, the vector μ^* assigns to each edge a reward corresponding to its cost and the goal is to find the smallest path with smallest cost.

Algorithmic principle: optimism

Algorithm LinUCB - UCB for linear bandits.

- Build confidence region for the parameter: $\mathcal{C}_t$ such that $\mu^* \in \mathcal{C}_t$ with high probability.
- Build confidence bounds for the arm means: $U_t^x = \max_{\mu \in \mathcal{C}_t} x^\top \mu$.
- Be optimistic: pull $x_t = \arg\max_x U_t^x$.

Main question: how do we get $\mathcal{C}_t$?

Confidence region

After time t, the algorithm observed:

$$
\begin{aligned}
X_1 &= x_1^\top \mu^* + \varepsilon_1 \\
X_2 &= x_2^\top \mu^* + \varepsilon_2 \\
&\dots \\
X_t &= x_t^\top \mu^* + \varepsilon_t
\end{aligned}
$$

The unknown parameter we want to estimate is μ^*.

Denoting by I_d the $d \times d$ identity matrix and picking $\lambda > 0$, we can estimate μ^* with regularized least square

$$\widehat{\mu}_t \overset{\text{def}}{=} \arg\min_{\mu \in \mathbb{R}^d} \left\{ \sum_{s=1}^{t} \left(X_s - x_s^\top \mu\right)^2 + \lambda \|\mu\|^2 \right\} = V_t^{-1} \sum_{s=1}^{t} x_s X_s \,,$$

where $V_t \overset{\text{def}}{=} \lambda I_d + \sum_{s=1}^{t} x_s x_s^\top$.

Confidence region

Lemma 2

Let $\delta \in (0,1)$. Then, with probability at least $1-\delta$, if $\max_{x \in \mathcal{K}_t} \|x\|_2 \leqslant 1$, for all $t \geqslant 1$

$$\left\|\widehat{\mu}_t - \mu^*\right\|_{V_t} \leqslant \sqrt{\lambda}\|\mu^*\| + \sqrt{2\log(1/\delta) + d\log\left(1 + \frac{T}{\lambda}\right)} \overset{\text{def}}{=} \beta(\delta)\,,$$

where $\|\mu\|_{V_t}^2 = \mu^\top V_t \mu$.

Conclusion: with probability $1-\delta$, for all $t \geqslant 1$,

$$\mu^* \in C_t, \quad \text{where} \quad C_t \overset{\text{def}}{=} \left\{x \in \mathbb{R}^d : \left\|\mu - \widehat{\mu}_{t-1}\right\|_{V_{t-1}} \leqslant \beta(\delta/T)\right\}. \tag{8}$$

Regret Bound

Theorem 20

Let $T \geqslant 1$ *and* $\mu^* \in \mathbb{R}^d$. *Assume that for all* $x \in \cup_{t=1}^{T} \mathcal{K}_t$, $|x^\top \mu^*| \leqslant 1$, $\|\mu^*\| \leqslant 1$ *and* $\|x\| \leqslant 1$, *then LinUCB with* C_t *defined as above satisfies the regret bound*

$$\mathbb{E} R_T \leqslant \square_\lambda d \sqrt{T} \log(T)\,,$$

where $\square_\lambda$ *is a constant that may depend on* λ.

Remark:

- $O(\sqrt{T})$: the exponent does not depend on d.

Proof

With probability $1 - 1/T$, for all $t \geqslant 1$,

$$\mu^* \in C_t, \quad \text{where} \quad C_t \overset{\text{def}}{=} \left\{ x \in \mathbb{R}^d : \left\| \mu - \widehat{\mu}_{t-1} \right\|_{V_{t-1}} \leqslant \beta(1/T^2) \right\}.$$

Proof 2

Proof 3

Summary

LinUCB with C_t defined as above satisfies the regret bound

$$\mathbb{E} R_T \lesssim d\sqrt{T}\log(T)\,,$$

To prove it, we assumed the following lemma:

Lemma 3

Let $\delta \in (0,1)$. Then, with probability at least $1-\delta$, if $\max_{x\in\mathcal{K}_t}\|x\|_2 \leqslant 1$, for all $t \geqslant 1$

$$\left\|\widehat{\mu}_t - \mu^*\right\|_{V_t} \leqslant \sqrt{\lambda}\|\mu^*\| + \sqrt{2\log(1/\delta) + d\log\left(1+\frac{T}{\lambda}\right)} \overset{\text{def}}{=} \beta(\delta)\,,$$

where $\|\mu\|_{V_t}^2 = \mu^\top V_t \mu$.

Improvements

Under additional assumptions, it is possible to improve the regret bound $O(d\sqrt{T}\log T)$.

- If the set of available actions at time t is fixed and finite; i.e., $x_t \in \mathcal{K}$ where $|\mathcal{K}| = K$. Then, it is possible to achieve

$$\mathbb{E}R_T \leqslant \square\sqrt{Td\log(TK)}\,,$$

  which improves the previous bound by a factor $\sqrt{d}/\log(K)$ and improves the classical bound of UCB $O(\sqrt{TK\log T})$ by a factor $K/\sqrt{d}$.
- Another possible improvement when $d \gg 1$ is to assume that μ^* is m_0-sparse (i.e., most of its components are zero). Then under assumptions, one can get a regret of order $\tilde{O}(\sqrt{dm_0T})$.

References

Thank you!

- Bubeck, Sébastien, Nicolo Cesa-Bianchi, et al. **“Regret analysis of stochastic and nonstochastic multi-armed bandit problems”.** In: Foundations and Trends® in Machine Learning 5.1 (2012), pp. 1–122.
- Cesa-Bianchi, Nicolo and Gábor Lugosi. **Prediction, learning, and games.** Cambridge university press, 2006.
- Hazan, Elad et al. **“Introduction to online convex optimization”.** In: Foundations and Trends® in Optimization 2.3-4 (2016), pp. 157–325.
- Lattimore, Tor and Csaba Szepesvári. **“Bandit algorithms”.** In: preprint (2019).
- Shalev-Shwartz, Shai et al. **“Online learning and online convex optimization”.** In: Foundations and Trends® in Machine Learning 4.2 (2012), pp. 107–194.
- Zinkevich, Martin. **“Online convex programming and generalized infinitesimal gradient ascent”.** In: Proceedings of the 20th International Conference on Machine Learning (ICML-03). 2003, pp. 928–936.