No-Regret Online Learning Algorithms

Joseph Chuang-Chieh Lin

Department of Computer Science and Information Engineering,
Tamkang University

Lecture Notes

22 November 2021

The slides are based on the lectures of Prof. Luca Trevisan.
https://lucatrevisan.github.io/40391/index.html

Outline

Outline

Online Convex Optimization

Goal: Design an algorithm such that

- At discrete time steps $t = 1, 2, \ldots,$ output $x_t \in K$, for each t.
  - K: a convex set of feasible solutions.
- After x_t is generated, a convex cost function $f_t : K \mapsto \mathbb{R}$ is revealed.
- Then the algorithm suffers the loss $f_t(x_t)$.

And we want to minimize the cost.

The difficulty

- The cost functions f_t is unknown before t.
- $f_1, f_2, \ldots, f_t, \ldots$ are not necessarily fixed.
  - Can be generated dynamically by an adversary.

What's the regret?

- The offline optimum: After T steps,

$$\min_{x \in K} \sum_{t=1}^{T} f_t(x).$$

- The regret after T steps:

$$\text{regret}_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in K} \sum_{t=1}^{T} f_t(x).$$

What's the regret?

- The offline optimum: After T steps,

$$\min_{x \in K} \sum_{t=1}^{T} f_t(x).$$

- The regret after T steps:

$$\text{regret}_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in K} \sum_{t=1}^{T} f_t(x).$$

- The rescue: $\text{regret}_T \leq o(T)$.

What's the regret?

- The offline optimum: After T steps,

$$\min_{x \in K} \sum_{t=1}^{T} f_t(x).$$

- The regret after T steps:

$$\text{regret}_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in K} \sum_{t=1}^{T} f_t(x).$$

- The rescue: $\text{regret}_T \leq o(T)$. $\Rightarrow$ **No-Regret** in average when $T \to \infty$.
  - For example, $\text{regret}_T / T = \frac{\sqrt{T}}{T} \to 0$ when $T \to \infty$.

Outline

Listen to the experts?

- Let's say we have n experts.
- We want to make best use of the advices coming from the experts.

Listen to the experts?

- Let's say we have n experts.
- We want to make best use of the advices coming from the experts.
- The idea: at each time step, decide the probability distribution (i.e., weights) of the experts to follow their advice.
  - $x_t = (x_t(1), x_t(2), \ldots, x_t(n))$, where $x_t(i) \in [0, 1]$ and $\sum_i x_t(i) = 1$.

Listen to the experts?

- Let's say we have n experts.
- We want to make best use of the advices coming from the experts.
- The idea: at each time step, decide the probability distribution (i.e., weights) of the experts to follow their advice.
  - $x_t = (x_t(1), x_t(2), \ldots, x_t(n))$, where $x_t(i) \in [0, 1]$ and $\sum_i x_t(i) = 1$.
- The loss of following expert i at time t: $\ell_t(i)$.
- The expected loss of the algorithm at time t:

$$\langle x_t, \ell_t \rangle = \sum_{i=1}^{n} x_t(i)\ell_t(i).$$

The regret of listening to the experts...

$$\text{regret}_T^* = \sum_{t=1}^{T} \langle x_t, \ell_t \rangle - \min_i \sum_{t=1}^{T} \ell_t(i).$$

- The set of feasible solutions $K = \Delta \subseteq \mathbb{R}^n$, probability distributions over $\{1, \ldots, n\}$.
- $f_t(x) = \sum_i x(i)\ell_t(i)$: linear function.

⋆ Assume that $|\ell_t(i)| \leq 1$ for all t and i.

The MWU Algorithm

- The spirit: "Hedge".
- Well-known and frequently rediscovered.

The MWU Algorithm

- The spirit: "Hedge".
- Well-known and frequently rediscovered.

Multiplicative Weight Update (MWU)

- Maintain a vector of weights $w_t = (w_t(1), \ldots, w_t(n))$ where $w_1 := (1, 1, \ldots, 1)$.
- Update the weights at time t by
  - $w_t(i) := w_{t-1}(i) \cdot e^{-\beta \ell_{t-1}(i)}$.
  - $x_t := \frac{w_t(i)}{\sum_{j=1}^{n} w_t(j)}$.

β: a parameter which will be optimized later.

The MWU Algorithm

- The spirit: "Hedge".
- Well-known and frequently rediscovered.

Multiplicative Weight Update (MWU)

- Maintain a vector of weights $w_t = (w_t(1), \ldots, w_t(n))$ where $w_1 := (1, 1, \ldots, 1)$.
- Update the weights at time t by
  - $w_t(i) := w_{t-1}(i) \cdot e^{-\beta \ell_{t-1}(i)}$.
  - $x_t := \frac{w_t(i)}{\sum_{j=1}^{n} w_t(j)}$.

β: a parameter which will be optimized later.

The weight of expert i at time t: $e^{-\beta \sum_{k=1}^{t-1} \ell_k(i)}$.

MWU is of no-regret

Theorem 1 (MWU is of no-regret)

Assume that $|\ell_t(i)| \leq 1$ for all t and i. For $\beta \in (0, 1/2)$, the regret of MWU after T steps is bounded as

$$\text{regret}_T^* \leq \beta \sum_{t=1}^{T} \sum_{i=1}^{n} x_t(i)\ell_t^2(i) + \frac{\ln n}{\beta} \leq \beta T + \frac{\ln n}{\beta}.$$

In particular, if $T > 4 \ln n$, then

$$\text{regret}_T^* \leq 2\sqrt{T \ln n}$$

by setting $\beta = \sqrt{\frac{\ln n}{T}}$.

Proof of Theorem 1

Let $W_t := \sum_{i=1}^{n} w_t(i)$.

The idea:

- If the algorithm incurs a large loss after T steps, then W_{T+1} is small.
- And, if W_{T+1} is small, then even the best expert performs quite badly.

Proof of Theorem 1

Let $W_t := \sum_{i=1}^{n} w_t(i)$.

The idea:

- If the algorithm incurs a large loss after T steps, then W_{T+1} is small.
- And, if W_{T+1} is small, then even the best expert performs quite badly.

Let $L^* := \min_i \sum_{t=1}^{T} \ell_t(i)$.

The proof (contd.)

Lemma 1 (W_{T+1} is SMALL $\Rightarrow$ L^* is LARGE)

$W_{T+1} \geq e^{-\beta L^*}$.

Proof.

Let $j = \arg\min L^* = \arg\min_i \sum_{t=1}^{T} \ell_t(i)$.

$$W_{T+1} = \sum_{i=1}^{n} e^{-\beta \sum_{t=1}^{T} \ell_t(i)} \geq e^{-\beta \sum_{t=1}^{T} \ell_t(j)} = e^{-\beta L^*}.$$

□

The proof (contd.)

Lemma 2 (MWU brings large loss ⇒ W_{T+1} is SMALL)

$$W_{T+1} \leq n \prod_{t=1}^{n} (1 - \beta\langle x_t, \ell_t\rangle + \beta^2\langle x_t, \ell_t^2\rangle),$$

Proof.

Note: $W_1 = n$.

$$\frac{W_{t+1}}{W_t} = \sum_{i=1}^{n} \frac{w_{t+1}(i)}{W_t} = \sum_{i=1}^{n} \frac{w_t(i) \cdot e^{-\beta\ell_t(i)}}{W_t}$$

The proof (contd.)

Lemma 2 (MWU brings large loss $\Rightarrow$ W_{T+1} is SMALL)

$$W_{T+1} \leq n \prod_{t=1}^{n} (1 - \beta\langle x_t, \ell_t \rangle + \beta^2 \langle x_t, \ell_t^2 \rangle),$$

Proof.

Note: $W_1 = n$.

$$\begin{aligned} \frac{W_{t+1}}{W_t} &= \sum_{i=1}^{n} \frac{w_{t+1}(i)}{W_t} = \sum_{i=1}^{n} \frac{w_t(i) \cdot e^{-\beta \ell_t(i)}}{W_t} = \sum_{i=1}^{n} x_t(i) \cdot e^{-\beta \ell_t(i)} \\ &\leq \sum_{i=1}^{n} x_t(i) \cdot (1 - \beta \ell_t(i) + \beta^2 \ell_t^2(i)) \end{aligned}$$

The proof (contd.)

Lemma 2 (MWU brings large loss $\Rightarrow W_{T+1}$ is SMALL)

$$W_{T+1} \le n \prod_{t=1}^{n} (1 - \beta\langle x_t, \ell_t\rangle + \beta^2\langle x_t, \ell_t^2\rangle),$$

Proof.

Note: $W_1 = n$.

$$\begin{aligned}
\frac{W_{t+1}}{W_t} &= \sum_{i=1}^{n} \frac{w_{t+1}(i)}{W_t} = \sum_{i=1}^{n} \frac{w_t(i) \cdot e^{-\beta\ell_t(i)}}{W_t} = \sum_{i=1}^{n} x_t(i) \cdot e^{-\beta\ell_t(i)} \\
&\le \sum_{i=1}^{n} x_t(i) \cdot (1 - \beta\ell_t(i) + \beta^2\ell_t^2(i)) \\
&= 1 - \beta\langle x_t, \ell_t\rangle + \beta^2\langle x_t, \ell_t^2\rangle
\end{aligned}$$

The proof (contd.)

Lemma 2 (MWU brings large loss $\Rightarrow W_{T+1}$ is SMALL)

$$W_{T+1} \leq n \prod_{t=1}^{n} (1 - \beta\langle x_t, \ell_t \rangle + \beta^2 \langle x_t, \ell_t^2 \rangle),$$

Proof.

Note: $W_1 = n$.

$$\begin{aligned} \frac{W_{t+1}}{W_t} &= \sum_{i=1}^{n} \frac{w_{t+1}(i)}{W_t} = \sum_{i=1}^{n} \frac{w_t(i) \cdot e^{-\beta \ell_t(i)}}{W_t} = \sum_{i=1}^{n} x_t(i) \cdot e^{-\beta \ell_t(i)} \\ &\leq \sum_{i=1}^{n} x_t(i) \cdot (1 - \beta \ell_t(i) + \beta^2 \ell_t^2(i)) \\ &= 1 - \beta\langle x_t, \ell_t \rangle + \beta^2 \langle x_t, \ell_t^2 \rangle \leq e^{-\beta\langle x_t, \ell_t \rangle + \beta^2 \langle x_t, \ell_t^2 \rangle}. \end{aligned}$$

□

The proof (contd.)

Lemma 2 (MWU brings large loss $\Rightarrow W_{T+1}$ is SMALL)

$$W_{T+1} \le n \prod_{t=1}^{n} e^{-\beta\langle x_t, \ell_t\rangle + \beta^2\langle x_t, \ell_t^2\rangle}.$$

Proof.

Note: $W_1 = n$.

$$\begin{aligned}
\frac{W_{t+1}}{W_t} &= \sum_{i=1}^{n} \frac{w_{t+1}(i)}{W_t} = \sum_{i=1}^{n} \frac{w_t(i)\cdot e^{-\beta\ell_t(i)}}{W_t} = \sum_{i=1}^{n} x_t(i)\cdot e^{-\beta\ell_t(i)} \\
&\le \sum_{i=1}^{n} x_t(i)\cdot(1 - \beta\ell_t(i) + \beta^2\ell_t^2(i)) \\
&= 1 - \beta\langle x_t, \ell_t\rangle + \beta^2\langle x_t, \ell_t^2\rangle \le e^{-\beta\langle x_t, \ell_t\rangle + \beta^2\langle x_t, \ell_t^2\rangle}.
\end{aligned}$$

□

The proof (contd.)

Hence

$$\ln W_{T+1} \le \ln n - \left(\sum_{i=1}^{T} \beta \langle \ell_t, x_t \rangle \right) + \left(\sum_{i=1}^{T} \beta^2 \langle \ell_t^2, x_t \rangle \right)$$

and $\ln W_{T+1} \ge -\beta L^*$.

The proof (contd.)

Hence

$$\ln W_{T+1} \leq \ln n - \left(\sum_{i=1}^{T} \beta \langle \ell_t, x_t \rangle\right) + \left(\sum_{i=1}^{T} \beta^2 \langle \ell_t^2, x_t \rangle\right)$$

and $\ln W_{T+1} \geq -\beta L^*$.

Thus,

$$\left(\sum_{t=1}^{T} \langle \ell_t, x_t \rangle\right) - L^* \leq \frac{\ln n}{\beta} + \beta \sum_{t=1}^{T} \langle \ell_t^2, x_t \rangle.$$

The proof (contd.)

Hence

$$\ln W_{T+1} \leq \ln n - \left(\sum_{i=1}^{T} \beta \langle \ell_t, x_t \rangle \right) + \left(\sum_{i=1}^{T} \beta^2 \langle \ell_t^2, x_t \rangle \right)$$

and $\ln W_{T+1} \geq -\beta L^*$.

Thus,

$$\left(\sum_{t=1}^{T} \langle \ell_t, x_t \rangle \right) - L^* \leq \frac{\ln n}{\beta} + \beta \sum_{t=1}^{T} \langle \ell_t^2, x_t \rangle.$$

Take $\beta = \sqrt{\frac{\ln n}{T}}$, we have $\text{regret}_T \leq 2\sqrt{T \ln n}$.

Outline

Why so complicated?

- How about just *following the one with best performance*?

Why so complicated?

- How about just *following the one with best performance*?
  - Follow The Leader (FTL) Algorithm.

Why so complicated?

- How about just *following the one with best performance*?
    - Follow The Leader (FTL) Algorithm.

- First, we assume to make no assumptions on K and $\{f_t : L \mapsto \mathbb{R}\}$.
- At time t, we are given previous cost functions $f_1, \ldots, f_{t-1}$, and then give the solution

$$x_t := \arg\min_{x \in K} \sum_{k=1}^{t-1} f_k(x).$$

Why so complicated?

- How about just *following the one with best performance*?
    - Follow The Leader (FTL) Algorithm.

- First, we assume to make no assumptions on K and $\{f_t : L \mapsto \mathbb{R}\}$.
- At time t, we are given previous cost functions $f_1, \dots, f_{t-1}$, and then give the solution

$$x_t := \arg\min_{x \in K} \sum_{k=1}^{t-1} f_k(x).$$

That is, the best solution for the previous $t-1$ steps.

Why so complicated?

- How about just *following the one with best performance*?
  - Follow The Leader (FTL) Algorithm.

- First, we assume to make no assumptions on K and $\{f_t : L \mapsto \mathbb{R}\}$.
- At time t, we are given previous cost functions $f_1, \ldots, f_{t-1}$, and then give the solution

$$x_t := \arg\min_{x \in K} \sum_{k=1}^{t-1} f_k(x).$$

  That is, the best solution for the previous $t-1$ steps.

- It seems reasonable and makes sense, doesn't it?

FTL leads to "overfitting"

t:	1
x_t:	$(0.5, 0.5)$
ℓ_t:	$(0, 0.5)$
$f_t(x_t)$:	0.25
$\arg\min_x \sum_{k=1}^{t} f_k(x)$:	$(1, 0)$

FTL leads to "overfitting"

t:	1	2
x_t:	$(0.5, 0.5)$	$(1, 0)$
ℓ_t:	$(0, 0.5)$	$(1, 0)$
$f_t(x_t)$:	0.25	1
$\arg\min_x \sum_{k=1}^{t} f_k(x)$:	$(1, 0)$	$(0, 1)$

FTL leads to "overfitting"

t:	1	2	3
x_t:	$(0.5, 0.5)$	$(1, 0)$	$(0, 1)$
ℓ_t:	$(0, 0.5)$	$(1, 0)$	$(0, 1)$
$f_t(x_t)$:	0.25	1	1
$\arg\min_x \sum_{k=1}^{t} f_k(x)$:	$(1, 0)$	$(0, 1)$	$(1, 0)$

FTL leads to "overfitting"

t:	1	2	3	4
x_t:	$(0.5, 0.5)$	$(1, 0)$	$(0, 1)$	$(1, 0)$
ℓ_t:	$(0, 0.5)$	$(1, 0)$	$(0, 1)$	$(1, 0)$
$f_t(x_t)$:	0.25	1	1	1
$\arg\min_x \sum_{k=1}^{t} f_k(x)$:	$(1, 0)$	$(0, 1)$	$(1, 0)$	$(0, 1)$

FTL leads to "overfitting"

t:	1	2	3	4	5
x_t:	$(0.5, 0.5)$	$(1, 0)$	$(0, 1)$	$(1, 0)$	$(0, 1)$
ℓ_t:	$(0, 0.5)$	$(1, 0)$	$(0, 1)$	$(1, 0)$	$(0, 1)$
$f_t(x_t)$:	0.25	1	1	1	1
$\arg\min_x \sum_{k=1}^{t} f_k(x)$:	$(1, 0)$	$(0, 1)$	$(1, 0)$	$(0, 1)$	$(1, 0)$

FTL leads to "overfitting"

t:	1	2	3	4	5	...
x_t:	$(0.5, 0.5)$	$(1, 0)$	$(0, 1)$	$(1, 0)$	$(0, 1)$	...
ℓ_t:	$(0, 0.5)$	$(1, 0)$	$(0, 1)$	$(1, 0)$	$(0, 1)$	...
$f_t(x_t)$:	0.25	1	1	1	1	...
$\arg\min_x \sum_{k=1}^{t} f_k(x)$:	$(1, 0)$	$(0, 1)$	$(1, 0)$	$(0, 1)$	$(1, 0)$	...

FTL leads to "overfitting"

t:	1	2	3	4	5	...
x_t:	$(0.5, 0.5)$	$(1, 0)$	$(0, 1)$	$(1, 0)$	$(0, 1)$	...
ℓ_t:	$(0, 0.5)$	$(1, 0)$	$(0, 1)$	$(1, 0)$	$(0, 1)$	...
$f_t(x_t)$:	0.25	1	1	1	1	...
$\arg\min_x \sum_{k=1}^t f_k(x)$:	$(1, 0)$	$(0, 1)$	$(1, 0)$	$(0, 1)$	$(1, 0)$	...

optimum loss: $\approx T/2$.

FTL's loss: $\approx T$.

regret: $\approx T/2$ (linear).

Analysis of FTL

Theorem 2 (Analysis of FTL)

For any sequence of cost functions $f_1, \ldots, f_t$ and any number of time steps T, the FTL algorithm satisfies

$$\text{regret}_T \leq \sum_{t=1}^{T} (f_t(x_t) - f_t(x_{t+1})).$$

Analysis of FTL

Theorem 2 (Analysis of FTL)

For any sequence of cost functions $f_1, \ldots, f_t$ and any number of time steps T, the FTL algorithm satisfies

$$\text{regret}_T \leq \sum_{t=1}^{T} (f_t(x_t) - f_t(x_{t+1})).$$

Implication: If $f_t(\cdot)$ is Lipschitz w.r.t. to some distance function $||\cdot||$, then x_t and x_{t+1} are close $\Rightarrow ||f_t(x_t) - f_t(x_{t+1})||$ can't be too large.

Modify FTL: x_t's should't change too much from step by step.

Proof of Theorem 2

Recall that

$$\text{regret}_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in K} \sum_{t=1}^{T} f_t(x)$$

Proof of Theorem 2

Recall that

$$\text{regret}_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in K} \sum_{t=1}^{T} f_t(x) \leq \sum_{t=1}^{T} (f_t(x_t) - f_t(x_{t+1})).$$

The theorem $\Leftrightarrow \sum_{t=1}^{T} f_t(x_{t+1}) \leq \min_{x \in K} \sum_{t=1}^{T} f_t(x)$.

Proof of Theorem 2

Recall that

$$\text{regret}_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x\in K} \sum_{t=1}^{T} f_t(x) \leq \sum_{t=1}^{T} (f_t(x_t) - f_t(x_{t+1})).$$

The theorem $\Leftrightarrow \sum_{t=1}^{T} f_t(x_{t+1}) \leq \min_{x\in K} \sum_{t=1}^{T} f_t(x)$.

Prove by induction. $T = 1$: The definition of x_2.

Proof of Theorem 2

Recall that

$$\text{regret}_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in K} \sum_{t=1}^{T} f_t(x) \leq \sum_{t=1}^{T} (f_t(x_t) - f_t(x_{t+1})).$$

The theorem $\Leftrightarrow \sum_{t=1}^{T} f_t(x_{t+1}) \leq \min_{x \in K} \sum_{t=1}^{T} f_t(x)$.

Prove by induction. $T = 1$: The definition of x_2.

Assume that it holds up to T. Then:

Proof of Theorem 2

Recall that

$$\text{regret}_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in K} \sum_{t=1}^{T} f_t(x) \leq \sum_{t=1}^{T} (f_t(x_t) - f_t(x_{t+1})).$$

The theorem $\Leftrightarrow \sum_{t=1}^{T} f_t(x_{t+1}) \leq \min_{x \in K} \sum_{t=1}^{T} f_t(x)$.

Prove by induction. $T = 1$: The definition of x_2.

Assume that it holds up to T. Then:

$$\sum_{t=1}^{T+1} f_t(x_{t+1}) = \sum_{t=1}^{T} f_t(x_{t+1}) + f_{T+1}(x_{T+2}) \leq \sum_{t=1}^{T+1} f_t(x_{T+2}) = \min_{x \in K} \sum_{t=1}^{T+1} f_t(x),$$

Proof of Theorem 2

Recall that

$$\text{regret}_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in K} \sum_{t=1}^{T} f_t(x) \leq \sum_{t=1}^{T} (f_t(x_t) - f_t(x_{t+1})).$$

The theorem $\Leftrightarrow \sum_{t=1}^{T} f_t(x_{t+1}) \leq \min_{x \in K} \sum_{t=1}^{T} f_t(x)$.

Prove by induction. $T = 1$: The definition of x_2.

Assume that it holds up to T. Then:

$$\sum_{t=1}^{T+1} f_t(x_{t+1}) = \sum_{t=1}^{T} f_t(x_{t+1}) + f_{T+1}(x_{T+2}) \leq \sum_{t=1}^{T+1} f_t(x_{T+2}) = \min_{x \in K} \sum_{t=1}^{T+1} f_t(x),$$

where

$$\sum_{t=1}^{T} f_t(x_{t+1}) \leq \min_{x \in K} \sum_{t=1}^{T} f_t(x) \leq \sum_{t=1}^{T} f_t(x_{T+2}).$$

Outline

Introducing REGULARIZATION

- You might have already been using regularization for quite a long time.

Introducing REGULARIZATION

```
from keras import regularizers
model.add(Dense(64, input_dim=64,
                kernel_regularizer=regularizers.l2(0.01)
```

Introducing REGULARIZATION

```
# L1 data (only 5 informative features)
X_1, y_1 = datasets.make_classification(n_samples=n_samples,
                                        n_features=n_features, n_informative=5,
                                        random_state=1)

# L2 data: non sparse, but less features
y_2 = np.sign(.5 - rnd.rand(n_samples))
X_2 = rnd.randn(n_samples, n_features // 5) + y_2[:, np.newaxis]
X_2 += 5 * rnd.randn(n_samples, n_features // 5)

clf_sets = [(LinearSVC(penalty='l1', loss='squared_hinge', dual=False,
                       tol=1e-3),
             np.logspace(-2.3, -1.3, 10), X_1, y_1),
            (LinearSVC(penalty='l2', loss='squared_hinge', dual=True),
             np.logspace(-4.5, -2, 10), X_2, y_2)]
```

The regularizer

At each step, we compute the solution

$$x_t := \arg\min_{x \in K} \left(R(x) + \sum_{k=1}^{t-1} f_k(x) \right).$$

This is called Follow the Regularized Leader (FTRL).

In short,

$$\text{FTRL} = \text{FTL} + \text{Regularizer}.$$

Analysis of FTRL

Theorem 3 (Analysis of FTRL)

For

- every sequence of cost function $\{f_t(\cdot)\}_{t\geq 1}$ and
- every regularizer function $R(\cdot)$,

for every x, the regret with respect to x after T steps of the FTRL algorithm is bounded as

$$\text{regret}_T(x) \leq \left(\sum_{t=1}^{T} f_t(x_t) - f_t(x_{t+1}) \right) + R(x) - R(x_1),$$

where $\text{regret}_T(x) := \sum_{t=1}^{T} (f_t(x_t) - f_t(x))$.

Proof of Theorem 3

- Consider a *mental* experiment:

Proof of Theorem 3

- Consider a *mental* experiment:
  - We run the FTL algorithm for $T + 1$ steps.
  - The sequence of cost functions: $R, f_1, f_2, \ldots, f_T$.
    - Use x_1 as the first solution.
  - The solutions: $x_1, x_1, x_2, \ldots, x_T$.

Proof of Theorem 3

- Consider a *mental* experiment:
  - We run the FTL algorithm for $T+1$ steps.
  - The sequence of cost functions: $R, f_1, f_2, \ldots, f_T$.
    - Use x_1 as the first solution.
  - The solutions: $x_1, x_1, x_2, \ldots, x_T$.
- The regret:

$$R(x_1) - R(x) + \sum_{t=1}^{T}(f_t(x_t) - f_t(x))$$

Proof of Theorem 3

- Consider a *mental* experiment:
  - We run the FTL algorithm for $T + 1$ steps.
  - The sequence of cost functions: $R, f_1, f_2, \ldots, f_T$.
    - Use x_1 as the first solution.
  - The solutions: $x_1, x_1, x_2, \ldots, x_T$.
- The regret:

$$R(x_1) - R(x) + \sum_{t=1}^{T}(f_t(x_t) - f_t(x)) \leq R(x_1) - R(x_1) + \sum_{t=1}^{T}(f_t(x_t) - f_t(x_{t+1}))$$

minimizer of $R(\cdot)$

Proof of Theorem 3

- Consider a *mental* experiment:
  - We run the FTL algorithm for $T + 1$ steps.
  - The sequence of cost functions: $R, f_1, f_2, \ldots, f_T$.
    - Use x_1 as the first solution.
  - The solutions: $x_1, x_1, x_2, \ldots, x_T$.
- The regret:

$$R(x_1) - R(x) + \sum_{t=1}^{T}(f_t(x_t) - f_t(x)) \leq R(x_1) - R(x_1) + \sum_{t=1}^{T}(f_t(x_t) - f_t(x_{t+1}))$$

output of FTRL at $t + 1$

Outline

Using negative-entropy regularization

- We have seen an example that FTL tends to put all probability mass on one expert (it's bad!)

Using negative-entropy regularization

- We have seen an example that FTL tends to put all probability mass on one expert (it's bad!)
- **Idea:** penalize over "concentralized" distributions.
  - *negative*-entropy: a good measure of how centralized a distribution is.

Using negative-entropy regularization

- We have seen an example that FTL tends to put all probability mass on one expert (it's bad!)
- **Idea:** penalize over "concentralized" distributions.
  - *negative*-entropy: a good measure of how centralized a distribution is.

$$R(x) := c \cdot \sum_{i=1}^{n} x(i) \ln x(i).$$

Using negative-entropy regularization

- We have seen an example that FTL tends to put all probability mass on one expert (it's bad!)
- **Idea:** penalize over "concentralized" distributions.
    - *negative*-entropy: a good measure of how centralized a distribution is.

$$R(x) := c \cdot \sum_{i=1}^{n} x(i) \ln x(i).$$

- So our FTRL gives

$$x_t = \arg\min_{x \in \Delta} \left(\sum_{k=1}^{t-1} \langle \ell_k, x \rangle + c \cdot \sum_{i=1}^{n} x(i) \ln x(i) \right).$$

Using negative entropy regularization

$$x_t = \arg\min_{x \in \Delta} \left(\sum_{k=1}^{t-1} \langle \ell_k, x \rangle + c \cdot \sum_{i=1}^{n} x(i) \ln x(i) \right).$$

- The constraint $x \in \Delta \Rightarrow \sum_i x_i = 1$.
- So we use Lagrange multiplier to solve

$$\mathcal{L} = \left(\sum_{k=1}^{t-1} \langle \ell_k, x \rangle \right) + c \cdot \left(\sum_{i=1}^{n} x(i) \ln x(i) \right) + \lambda \cdot (\langle x, \mathbf{1} \rangle - 1).$$

Using negative entropy regularization

$$x_t = \arg\min_{x \in \Delta} \left(\sum_{k=1}^{t-1} \langle \ell_k, x \rangle + c \cdot \sum_{i=1}^{n} x(i) \ln x(i) \right).$$

- The constraint $x \in \Delta \Rightarrow \sum_i x_i = 1$.
- So we use Lagrange multiplier to solve

$$\mathcal{L} = \left(\sum_{k=1}^{t-1} \langle \ell_k, x \rangle \right) + c \cdot \left(\sum_{i=1}^{n} x(i) \ln x(i) \right) + \lambda \cdot (\langle x, \mathbf{1} \rangle - 1).$$

- The partial derivative $\frac{\partial \mathcal{L}}{\partial x(i)}$:

$$\left(\sum_{k=1}^{t-1} \ell_k(i) \right) + c \cdot (1 + \ln x_i) + \lambda$$

Rediscover MWU?

$$\frac{\partial \mathcal{L}}{\partial x(i)} = 0 \quad \Rightarrow \quad x(i) = \exp\left(-1 - \frac{\lambda}{c} - \frac{1}{c}\sum_{k=1}^{t-1} \ell_k(i)\right)$$

Rediscover MWU?

$$\frac{\partial \mathcal{L}}{\partial x(i)} = 0 \quad \Rightarrow \quad x(i) = \exp\left(-1 - \frac{\lambda}{c} - \frac{1}{c}\sum_{k=1}^{t-1} \ell_k(i)\right)$$

Take the value of λ to make the solution a probability distribution. Thus,

Rediscover MWU?

$$\frac{\partial \mathcal{L}}{\partial x(i)} = 0 \quad \Rightarrow \quad x(i) = \exp\left(-1 - \frac{\lambda}{c} - \frac{1}{c}\sum_{k=1}^{t-1} \ell_k(i)\right)$$

Take the value of λ to make the solution a probability distribution. Thus,

$$x(i) = \frac{\exp\left(-\frac{1}{c}\sum_{k=1}^{t-1} \ell_k(i)\right)}{\sum_j \exp\left(-\frac{1}{c}\sum_{k=1}^{t-1} \ell_k(j)\right)}.$$

Rediscover MWU?

$$\frac{\partial \mathcal{L}}{\partial x(i)} = 0 \quad \Rightarrow \quad x(i) = \exp\left(-1 - \frac{\lambda}{c} - \frac{1}{c}\sum_{k=1}^{t-1} \ell_k(i)\right)$$

Take the value of λ to make the solution a probability distribution. Thus,

$$x(i) = \frac{\exp\left(-\frac{1}{c}\sum_{k=1}^{t-1} \ell_k(i)\right)}{\sum_j \exp\left(-\frac{1}{c}\sum_{k=1}^{t-1} \ell_k(j)\right)}.$$

Exactly the solution of MWU if we take $c = 1/\beta$!

Rediscover MWU?

$$\frac{\partial \mathcal{L}}{\partial x(i)} = 0 \quad \Rightarrow \quad x(i) = \exp\left(-1 - \frac{\lambda}{c} - \frac{1}{c}\sum_{k=1}^{t-1} \ell_k(i)\right)$$

Take the value of λ to make the solution a probability distribution. Thus,

$$x(i) = \frac{\exp\left(-\frac{1}{c}\sum_{k=1}^{t-1} \ell_k(i)\right)}{\sum_j \exp\left(-\frac{1}{c}\sum_{k=1}^{t-1} \ell_k(j)\right)}.$$

Exactly the solution of MWU if we take $c = 1/\beta$!

- Now it remains to bound the deviation of each step.

Regret of FTRL + Negative-Entropy Regularization

- At each step,
  $f_t(x_t) - f_t(x_{t+1}) = \langle \ell_t, x_t - x_{t+1} \rangle$.
- Let's go back to use the notation of MWU.
  - $w_1(i) = 1$ (initialization).
  - $w_{t+1}(i) = w_t(i) \cdot e^{-\ell_t(i)/c}$.

Regret of FTRL + Negative-Entropy Regularization

- At each step,
  $f_t(x_t) - f_t(x_{t+1}) = \langle \ell_t, x_t - x_{t+1} \rangle$.
- Let's go back to use the notation of MWU.
  - $w_1(i) = 1$ (initialization).
  - $w_{t+1}(i) = w_t(i) \cdot e^{-\ell_t(i)/c}$.
- So, $x_t = \frac{w_t(i)}{\sum_j w_t(j)}$.
- Then,

$$\begin{aligned} x_{t+1}(i) &= \frac{w_{t+1}(i)}{\sum_j w_{t+1}(j)} = \frac{w_t(i) e^{-\ell_t(i)/c}}{\sum_j w_{t+1}(j)} \geq \frac{w_t(i) e^{-\ell_t(i)/c}}{\sum_j w_t(j)} \\ &\geq x_t(i) \cdot e^{-1/c} \geq (1 - 1/c) x_t(i). \end{aligned}$$

Regret of FTRL + Negative-Entropy Regularization

- At each step,
  $f_t(x_t) - f_t(x_{t+1}) = \langle \ell_t, x_t - x_{t+1} \rangle$.
- Let's go back to use the notation of MWU.
  - $w_1(i) = 1$ (initialization).
  - $w_{t+1}(i) = w_t(i) \cdot e^{-\ell_t(i)/c}$.
- So, $x_t = \frac{w_t(i)}{\sum_j w_t(j)}$.
- Then,

$$x_{t+1}(i) = \frac{w_{t+1}(i)}{\sum_j w_{t+1}(j)} = \frac{w_t(i)e^{-\ell_t(i)/c}}{\sum_j w_{t+1}(j)} \geq \frac{w_t(i)e^{-\ell_t(i)/c}}{\sum_j w_t(j)}$$
$$\geq x_t(i) \cdot e^{-1/c} \geq (1 - 1/c)x_t(i).$$

$\because$ weights are non-increasing

Regret of FTRL + Negative-Entropy Regularization

- At each step,
  $f_t(x_t) - f_t(x_{t+1}) = \langle \ell_t, x_t - x_{t+1} \rangle$.
- Let's go back to use the notation of MWU.
  - $w_1(i) = 1$ (initialization).
  - $w_{t+1}(i) = w_t(i) \cdot e^{-\ell_t(i)/c}$.
- So, $x_t = \frac{w_t(i)}{\sum_j w_t(j)}$.
- Then,

$$\begin{aligned} x_{t+1}(i) &= \frac{w_{t+1}(i)}{\sum_j w_{t+1}(j)} = \frac{w_t(i)e^{-\ell_t(i)/c}}{\sum_j w_{t+1}(j)} \geq \frac{w_t(i)e^{-\ell_t(i)/c}}{\sum_j w_t(j)} \\ &\geq x_t(i) \cdot e^{-1/c} \geq (1 - 1/c)x_t(i). \end{aligned}$$

assume $0 \leq \ell_t(i) \leq 1$

Regret of FTRL + Negative-Entropy Regularization

- At each step,
  $f_t(x_t) - f_t(x_{t+1}) = \langle \ell_t, x_t - x_{t+1} \rangle \leq \sum_i \ell_t(i) \cdot \frac{1}{c} x_t(i) \leq \frac{1}{c}$.
- Let's go back to use the notation of MWU.
  - $w_1(i) = 1$ (initialization).
  - $w_{t+1}(i) = w_t(i) \cdot e^{-\ell_t(i)/c}$.
- So, $x_t = \frac{w_t(i)}{\sum_j w_t(j)}$.
- Then,

$$\begin{aligned} x_{t+1}(i) &= \frac{w_{t+1}(i)}{\sum_j w_{t+1}(j)} = \frac{w_t(i) e^{-\ell_t(i)/c}}{\sum_j w_{t+1}(j)} \geq \frac{w_t(i) e^{-\ell_t(i)/c}}{\sum_j w_t(j)} \\ &\geq x_t(i) \cdot e^{-1/c} \geq (1 - 1/c) x_t(i). \end{aligned}$$

Regret of FTRL + Negative-Entropy Regularization

- By Theorem 3, for any x,

$$\text{regret}_T(x) \leq \sum_{t=1}^{T} (f_t(x_t) - f_t(x_{t+1})) + R(x) - R(x_1) \leq \frac{T}{c} + c \ln n.$$

Regret of FTRL + Negative-Entropy Regularization

- By Theorem 3, for any x,

$$\text{regret}_T(x) \leq \sum_{t=1}^{T} (f_t(x_t) - f_t(x_{t+1})) + R(x) - R(x_1) \leq \frac{T}{c} + c \ln n.$$

$\because$ max entropy for uniform distribution

Regret of FTRL + Negative-Entropy Regularization

- By Theorem 3, for any x,

$$\text{regret}_T(x) \leq \sum_{t=1}^{T} (f_t(x_t) - f_t(x_{t+1})) + R(x) - R(x_1) \leq \frac{T}{c} + c \ln n.$$

Again, we have $\text{regret}_T \leq 2\sqrt{T \ln n}$ by choosing $c = \sqrt{\frac{T}{\ln n}}$.

Regret of FTRL + Negative-Entropy Regularization

- By Theorem 3, for any x,

$$\text{regret}_T(x) \leq \sum_{t=1}^{T} (f_t(x_t) - f_t(x_{t+1})) + R(x) - R(x_1) \leq \frac{T}{c} + c \ln n.$$

Again, we have $\text{regret}_T \leq 2\sqrt{T \ln n}$ by choosing $c = \sqrt{\frac{T}{\ln n}}$.

- Note the slight difference b/w regret and regret*.

Outline

L2 Regularization

- Let's try to apply the FTRL to the case that the regularizer is of L2 norm!
- Consider also linear cost functions but $K = \mathbb{R}^n$ first.
- What kind of problem we might encounter?

L2 Regularization

- Let's try to apply the FTRL to the case that the regularizer is of L2 norm!
- Consider also linear cost functions but $K = \mathbb{R}^n$ first.
- What kind of problem we might encounter?
- The offline optimum could be $-\infty$.
- FTL will also tend to find a solution of "big" size, too.

L2 Regularization

- Let's try to apply the FTRL to the case that the regularizer is of L2 norm!
- Consider also linear cost functions but $K = \mathbb{R}^n$ first.
- What kind of problem we might encounter?
- The offline optimum could be $-\infty$.
- FTL will also tend to find a solution of "big" size, too.
- To fight this tendency, it makes sense to use a regularizer which penalizes the size of a solution.

$$R(x) := c||x||^2.$$

The regularizer of 2-norm tells us...

- $x_1 = \mathbf{0}$.
- $x_{t+1} = \arg\min_{x \in \mathbb{R}^n} c||x||^2 + \sum_{k=1}^{t} \langle \ell_k, x \rangle$.

- Compute the gradient:

$$\begin{aligned} & 2cx + \sum_{k=1}^{t} \ell_k = 0 \\ \Rightarrow \quad & x = -\frac{1}{2c} \sum_{k=1}^{t} \ell_k. \end{aligned}$$

Hence, $x_1 = \mathbf{0}, x_{t+1} = x_t - \frac{1}{2c} \ell_t$.

The regularizer of 2-norm tells us...

- $x_1 = \mathbf{0}$.
- $x_{t+1} = \arg\min_{x \in \mathbb{R}^n} c||x||^2 + \sum_{k=1}^{t} \langle \ell_k, x \rangle$.
  convex
- Compute the gradient:

$$2cx + \sum_{k=1}^{t} \ell_k = 0$$

$$\Rightarrow \quad x = -\frac{1}{2c} \sum_{k=1}^{t} \ell_k.$$

Hence, $x_1 = \mathbf{0}, x_{t+1} = x_t - \frac{1}{2c}\ell_t$.

The regularizer of 2-norm tells us...

- $x_1 = \mathbf{0}$.
- $x_{t+1} = \arg\min_{x \in \mathbb{R}^n} c||x||^2 + \sum_{k=1}^{t} \langle \ell_k, x \rangle$.

- Compute the gradient:

$$2cx + \sum_{k=1}^{t} \ell_k = 0$$

$$\Rightarrow \quad x = -\frac{1}{2c} \sum_{k=1}^{t} \ell_k.$$

Hence, $x_1 = \mathbf{0}, x_{t+1} = x_t - \frac{1}{2c} \ell_t$.

The regularizer of 2-norm tells us...

- $x_1 = \mathbf{0}$.
- $x_{t+1} = \arg\min_{x \in \mathbb{R}^n} c||x||^2 + \sum_{k=1}^{t} \langle \ell_k, x \rangle$.

- Compute the gradient:

$$2cx + \sum_{k=1}^{t} \ell_k = 0$$

$$\Rightarrow \quad x = -\frac{1}{2c} \sum_{k=1}^{t} \ell_k.$$

Hence, $x_1 = \mathbf{0}, x_{t+1} = x_t - \frac{1}{2c}\ell_t$.
$\rightarrow$ penalize the experts that performed badly in the past!

The regret of FTRL with 2-norm regularization

- First, we have

$$f_t(x_t) - f_t(x_{t+1}) = \langle \ell_t, x_t - x_{t+1} \rangle = \left\langle \ell_t, \frac{1}{2c}\ell_t \right\rangle = \frac{1}{2c}||\ell_t||^2.$$

- So, with respect to a solution x,

$$\begin{aligned} \text{regret}_T(x) &\leq R(x) - R(x_1) + \sum_{t=1}^{T} f_t(x_t) - f_t(x_{t+1}) \\ &= c||x||^2 + \frac{1}{2c}\sum_{t=1}^{T}||\ell_t||^2. \end{aligned}$$

- Suppose that $||\ell_t|| \leq L$ for each t and $||x|| \leq D$. Then by optimizing $c = \sqrt{\frac{T}{2D^2L^2}}$, we have

$$\text{regret}_T(x) \leq DL\sqrt{2T}.$$

Dealing with constraints

- Let's deal with the constraint that K is an arbitrary convex set instead of $\mathbb{R}^n$.
- Using the same regularizer, we have our FTRL which gives

$$x_1 = \arg\min_{x \in K} c||x||^2,$$

$$x_{t+1} = \arg\min_{x \in K} c||x||^2 + \sum_{k=1}^{t} \langle \ell_t, x \rangle.$$

Dealing with constraints

- Let's deal with the constraint that K is an arbitrary convex set instead of $\mathbb{R}^n$.
- Using the same regularizer, we have our FTRL which gives

$$x_1 = \arg\min_{x \in K} c||x||^2,$$

$$x_{t+1} = \arg\min_{x \in K} c||x||^2 + \sum_{k=1}^{t} \langle \ell_t, x \rangle.$$

- **The idea:** First solve the unconstrained optimization and then project the solution on K.

Unconstrained optimization + projection

$$y_{t+1} = \arg\min_{y \in \mathbb{R}^n} c||y||^2 + \sum_{k=1}^{t} \langle \ell_t, y \rangle.$$

$$x'_{t+1} = \prod_{K} (y_{t+1}) = \arg\min_{x \in K} ||x - y_{t+1}||.$$

Unconstrained optimization + projection

$$y_{t+1} = \arg\min_{y \in \mathbb{R}^n} c||y||^2 + \sum_{k=1}^{t} \langle \ell_t, y \rangle.$$

$$x'_{t+1} = \prod_{K}(y_{t+1}) = \arg\min_{x \in K} ||x - y_{t+1}||.$$

- **Claim:** $x'_{t+1} = x_{t+1}$.

Proof of the claim: $x'_{t+1} = x_{t+1}$

- First, we already have that $y_{t+1} = -\frac{1}{2c}\sum_{k=1}^{t} \ell_t$.
- Then,

$$\begin{aligned} x'_{t+1} &= \arg\min_{x \in K} ||x - y_{t+1}|| = \arg\min_{x \in K} ||x - y_{t+1}||^2 \\ &= \arg\min_{x \in K} ||x||^2 - 2\langle x, y_{t+1}\rangle + ||y_{t+1}||^2 \end{aligned}$$

Proof of the claim: $x'_{t+1} = x_{t+1}$

- First, we already have that $y_{t+1} = -\frac{1}{2c}\sum_{k=1}^{t} \ell_t$.
- Then,

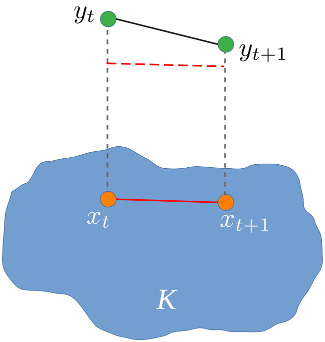
$$\begin{aligned}
x'_{t+1} &= \arg\min_{x\in K} ||x - y_{t+1}|| = \arg\min_{x\in K} ||x - y_{t+1}||^2 \\
&= \arg\min_{x\in K} ||x||^2 - 2\langle x, y_{t+1}\rangle + ||y_{t+1}||^2 \\
&= \arg\min_{x\in K} ||x||^2 - 2\langle x, y_{t+1}\rangle \\
&= \arg\min_{x\in K} ||x||^2 - 2\left\langle x, -\frac{1}{2c}\sum_{k=1}^{t} \ell_t \right\rangle \\
&= \arg\min_{x\in K} c||x||^2 + \left\langle x, \sum_{k=1}^{t} \ell_t \right\rangle \\
&= x_{t+1}.
\end{aligned}$$

To bound the regret

$$f_t(x_t) - f_t(x_{t+1}) = \langle \ell_t, x_t - x_{t+1} \rangle \quad \leq \quad ||\ell_t|| \cdot ||x_t - x_{t+1}||$$

To bound the regret

$$f_t(x_t) - f_t(x_{t+1}) = \langle \ell_t, x_t - x_{t+1} \rangle \leq ||\ell_t|| \cdot ||x_t - x_{t+1}|| \leq ||\ell_t|| \cdot ||y_t - y_{t+1}||.$$



To bound the regret

$$f_t(x_t) - f_t(x_{t+1}) = \langle \ell_t, x_t - x_{t+1} \rangle \leq ||\ell_t|| \cdot ||x_t - x_{t+1}|| \leq ||\ell_t|| \cdot ||y_t - y_{t+1}|| \leq \frac{1}{2c}||\ell_t||^2.$$

So, assume $\max_{x \in K} ||x|| \leq D$ and $||\ell_t|| \leq L$ for all t, we have

$$\text{regret}_T \leq c||x^*||^2 - c||x_1||^2 + \frac{1}{2c}\sum_{t=1}^{T}||\ell_t||^2 \leq cD^2 + \frac{1}{2c}TL^2$$

To bound the regret

$$
\begin{aligned}
f_t(x_t) - f_t(x_{t+1}) = \langle \ell_t, x_t - x_{t+1} \rangle &\leq ||\ell_t|| \cdot ||x_t - x_{t+1}|| \\
&\leq ||\ell_t|| \cdot ||y_t - y_{t+1}|| \\
&\leq \frac{1}{2c} ||\ell_t||^2.
\end{aligned}
$$

So, assume $\max_{x \in K} ||x|| \leq D$ and $||\ell_t|| \leq L$ for all t, we have

$$
\begin{aligned}
\text{regret}_T &\leq c||x^*||^2 - c||x_1||^2 + \frac{1}{2c} \sum_{t=1}^{T} ||\ell_t||^2 \\
&\leq cD^2 + \frac{1}{2c} TL^2 \leq DL\sqrt{2T}.
\end{aligned}
$$

Outline

Multi-Armed Bandit

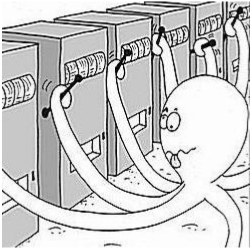


Fig.: Image credit: Microsoft Research

This part of slides are based on the lectures of Prof. Shipra Agrawal.

The setting

- We can see N arms as N experts.
- Arms give are independent.
- We can only pull an arm and observe the reward of it.
    - It's NOT possible to observe the reward of pulling the other arms...
- Each arm i has its own reward $r_i \in [0, 1]$.

The setting

- We can see N arms as N experts.
- Arms give are independent.
- We can only pull an arm and observe the reward of it.
  - It's NOT possible to observe the reward of pulling the other arms...
- Each arm i has its own reward $r_i \in [0, 1]$.
  - μ_i: the mean of reward of arm i
    - $\hat{\mu}_i$: the empirical mean of reward of arm i
  - μ^*: the mean of reward of the BEST arm.
  - $\Delta_i : \mu^* - \mu_i$.
  - Index of the best arm: $I^* := \arg\max_{i \in \{1,\ldots,N\}} \mu_i$.
  - The associated highest expected reward: $\mu^* = \mu_{I^*}$.

The regret formulation for MAB

Let I_t be the arm played by the algorithm at time t.
The regret of the algorithm in T rounds is

$$\text{regret}_T = \sum_{t=1}^{T} (\mu^* - \mu_{I_t})$$

The regret formulation for MAB

Let I_t be the arm played by the algorithm at time t.
The regret of the algorithm in T rounds is

$$\text{regret}_T = \sum_{t=1}^{T}(\mu^* - \mu_{I_t}) = \sum_{i=1}^{N}\sum_{t:I_t=i}(\mu^* - \mu_i)$$

The regret formulation for MAB

Let I_t be the arm played by the algorithm at time t.
The regret of the algorithm in T rounds is

$$\begin{aligned} \text{regret}_T &= \sum_{t=1}^{T} (\mu^* - \mu_{I_t}) = \sum_{i=1}^{N} \sum_{t:I_t=i} (\mu^* - \mu_i) \\ &= \sum_{i=1}^{N} n_{i,T} \Delta_i \end{aligned}$$

The regret formulation for MAB

Let I_t be the arm played by the algorithm at time t.
The regret of the algorithm in T rounds is

$$\begin{aligned} \text{regret}_T &= \sum_{t=1}^{T}(\mu^* - \mu_{I_t}) = \sum_{i=1}^{N} \sum_{t:I_t=i} (\mu^* - \mu_i) \\ &= \sum_{i=1}^{N} n_{i,T}\Delta_i \\ &= \sum_{i:\mu_i<\mu^*} n_{i,T}\Delta_i . \end{aligned}$$

Outline

The upper confidence bound algorithm (UCB)

- At each time step (round), we simply pull the arm with the highest "empirical reward estimate + high-confidence interval size".
- The empirical reward estimate of arm i at time t:

$$\hat{\mu}_{i,t} = \frac{\sum_{s=1}^{t} I_{s,i} \cdot r_s}{n_{i,t}}.$$

$n_{i,t}$: the number of times arm i is played.
$I_{s,i}$: 1 if the choice of arm is i at time s and 0 otherwise.

- Reward estimate + confidence interval:

$$\mathrm{UCB}_{i,t} := \hat{\mu}_{i,t} + \sqrt{\frac{\ln t}{n_{i,t}}}.$$

Algorithm UCB

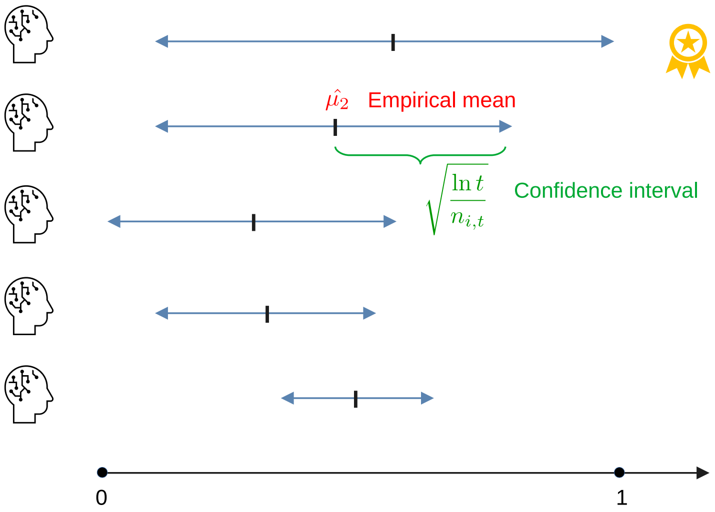
UCB Algorithm

N arms, T rounds such that $T \geq N$.

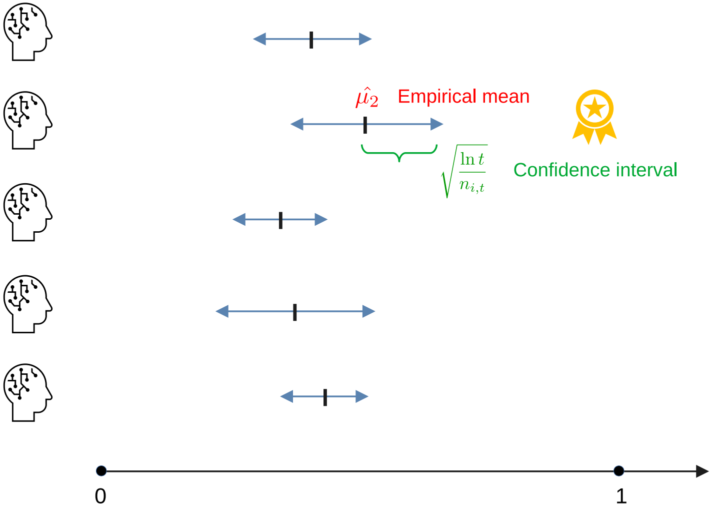
1. For $t = 1, \ldots, N$, play arm t.
2. For $t = N + 1, \ldots, T$, play arm

$$A_t = \arg\max_{i \in \{1,\ldots,N\}} \mathrm{UCB}_{i,t-1}.$$

Algorithm UCB



Algorithm UCB (after more time steps...)



From the Chernoff bound (proof skipped)

For each arm i at time t, we have

$$|\hat{\mu}_{i,t} - \mu_i| < \sqrt{\frac{\ln t}{n_{i,t}}}$$

with probability $\geq 1 - 2/t^2$.

Immediately, we know that

- with prob. $\geq 1 - 2/t^2$, $\text{UCB}_{i,t} := \hat{\mu}_{i,t} + \sqrt{\frac{\ln t}{n_{i,t}}} > \mu_i$.
- with prob. $\geq 1 - 2/t^2$, $\hat{\mu}_{i,t} < \mu_i + \frac{\Delta_i}{2}$ when $n_{i,t} \geq \frac{4 \ln t}{\Delta_i^2}$.

From the Chernoff bound (proof skipped)

For each arm i at time t, we have

$$|\hat{\mu}_{i,t} - \mu_i| < \sqrt{\frac{\ln t}{n_{i,t}}}$$

with probability $\geq 1 - 2/t^2$.

To understand why, please take my Randomized Algorithms course. :)
Immediately, we know that

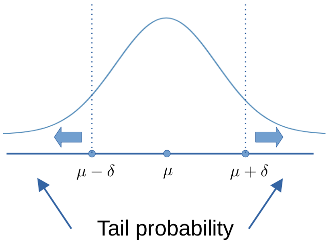
- with prob. $\geq 1 - 2/t^2$, $\text{UCB}_{i,t} := \hat{\mu}_{i,t} + \sqrt{\frac{\ln t}{n_{i,t}}} > \mu_i$.
- with prob. $\geq 1 - 2/t^2$, $\hat{\mu}_{i,t} < \mu_i + \frac{\Delta_i}{2}$ when $n_{i,t} \geq \frac{4 \ln t}{\Delta_i^2}$.

Appendix: Tail probability by the Chernoff/Hoeffding bound

The Chernoff/Hoeffding bound

For independent and identically distributed (i.i.d.) samples $x_1, \ldots, x_n \in [0, 1]$ with $\mathbb{E}[x_i] = \mu$, we have

$$\Pr\left[\left|\frac{\sum_{i=1}^{n} x_i}{n} - \mu\right| \geq \delta\right] \leq 2e^{-2n\delta^2}.$$



Very unlikely to play a suboptimal arm

Lemma 3

At any time step t, if a suboptimal arm i (i.e., $\mu_i < \mu^*$) has been played for $n_{i,t} \geq \frac{4 \ln t}{\Delta_i^2}$ times, then $\text{UCB}_{i,t} < \text{UCB}_{I^*,t}$ with probability $\geq 1 - 4/t^2$. Therefore, for any t,

$$\Pr\left[I_{t+1,i} = 1 \,\middle|\, n_{i,t} \geq \frac{4 \ln t}{\Delta_i^2} \right] \leq \frac{4}{t^2}.$$

Proof of Lemma 3

With probability $< 2/t^2 + 2/t^2$ (union bound) that

$$\begin{aligned}\mathrm{UCB}_{i,t} = \hat{\mu}_{i,t} + \sqrt{\frac{\ln t}{n_{i,t}}} &\leq \hat{\mu}_{i,t} + \frac{\Delta_i}{2} \\ &< \left(\mu_i + \frac{\Delta_i}{2}\right) + \frac{\Delta_i}{2} \\ &= \mu^* < \mathrm{UCB}_{i^*,t}\end{aligned}$$

does NOT hold.

Playing suboptimal arms for very limited number of times

Lemma 4

For any arm i with $\mu_i < \mu^*$,

$$\mathbb{E}[n_{i,T}] \leq \frac{4 \ln T}{\Delta_i} + 8.$$

$$\begin{aligned} \mathbb{E}[n_{i,T}] &= 1 + \mathbb{E}\left[\sum_{t=N}^{T} \mathbb{1}\left\{I_{t+1,i} = 1\right\}\right] \\ &= 1 + \mathbb{E}\left[\sum_{t=N}^{T} \mathbb{1}\left\{I_{t+1,i} = 1, n_{i,t} < \frac{4 \ln t}{\Delta_i^2}\right\}\right] \\ &\quad + \mathbb{E}\left[\sum_{t=N}^{T} \mathbb{1}\left\{I_{t+1,i} = 1, n_{i,t} \geq \frac{4 \ln t}{\Delta_i^2}\right\}\right] \end{aligned}$$

Proof of Lemma 4 (contd.)

$$\begin{aligned}
\mathbb{E}[n_{i,T}] &\leq \frac{4\ln T}{\Delta_i^2} + \mathbb{E}\left[\sum_{t=N}^{T} \mathbb{1}\left\{ I_{t+1,i} = 1, n_{i,t} \geq \frac{4\ln t}{\Delta_i^2} \right\}\right] \\
&= \frac{4\ln T}{\Delta_i^2} + \sum_{t=N}^{T} \Pr\left[I_{t+1,i} = 1, n_{i,t} \geq \frac{4\ln t}{\Delta_i^2} \right] \\
&= \frac{4\ln T}{\Delta_i^2} + \sum_{t=N}^{T} \Pr\left[I_{t+1,i} = 1 \,\middle|\, n_{i,t} \geq \frac{4\ln t}{\Delta_i^2} \right] \cdot \Pr\left[n_{i,t} \geq \frac{4\ln t}{\Delta_i^2} \right] \\
&\leq \frac{4\ln T}{\Delta_i^2} + \sum_{t=N}^{T} \frac{4}{t^2} \\
&\leq \frac{4\ln T}{\Delta_i^2} + 8.
\end{aligned}$$

The regret bound for the UCB algorithm

Theorem 4

For all $T \geq N$, the (expected) regret by the UCB algorithm in round T is

$$\mathbb{E}[\text{regret}_T] \leq 5\sqrt{NT \ln T} + 8N.$$

Proof of Theorem 4

- Divide the arms into two groups:
  1. Group ONE (G_1): "almost optimal arms" with $\Delta_i < \sqrt{\frac{N}{T}\ln T}$.
  2. Group TWO (G_2): "bad" arms with $\Delta_i \geq \sqrt{\frac{N}{T}\ln T}$.

$$\sum_{i\in G_1} n_{i,T}\Delta_i \leq \left(\sqrt{\frac{N}{T}\ln T}\right)\sum_{i\in G_1} n_{i,T} \leq T\cdot\sqrt{\frac{N}{T}\ln T} = \sqrt{NT\ln T}.$$

By Lemma 4,

$$\begin{aligned}\sum_{i\in G_2}\mathbb{E}[n_{i,T}]\Delta_i \leq \sum_{i\in G_2}\frac{4\ln T}{\Delta_i}+8\Delta_i &\leq \sum_{i\in G_2}4\sqrt{\frac{T\ln T}{N}}+8\\ &\leq 4\sqrt{NT\ln T}+8N.\end{aligned}$$

Thank you.