Online learning using limited feedback

Yoav Freund

February 26, 2025

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

The one armed bandit

The multiple arm bandit problem

Given

Play these machines

The multiple arm bandit problem

Given

Play these machines

Limited Feedback: Only the reward/loss from chosen arm is observed.

The multiple arm bandit problem

Given

Play these machines

Limited Feedback: Only the reward/loss from chosen arm is observed. Goal: Maximize expected wealth.

The multiple arm bandit problem

Given

Play these machines

Limited Feedback: Only the reward/loss from chosen arm is observed. Goal: Maximize expected wealth.
Mathematical formulation for common Exploration vs. Exploitation dilemma.

The multiple arm bandit problem

Given

Play these machines

Limited Feedback: Only the reward/loss from chosen arm is observed. Goal: Maximize expected wealth.
Mathematical formulation for common Exploration vs. Exploitation dilemma.
single-iteration reward is in the range $[0, 1]$

Applications of MAB

- Choosing lunch.

Applications of MAB

- Choosing lunch.
- Routing packets through the internet.

Applications of MAB

- Choosing lunch.
- Routing packets through the internet.
- Reinforcement learning.

Classical analysis

- Rewards generated independently at random

Classical analysis

- Rewards generated independently at random
- Each machine has a different distribution of rewards.

Classical analysis

- Rewards generated independently at random
- Each machine has a different distribution of rewards.
- Update upper and lower bounds of the expected reward for each arm.

Classical analysis

- Rewards generated independently at random
- Each machine has a different distribution of rewards.
- Update upper and lower bounds of the expected reward for each arm.
- Choose the arm with the highest upper bound.

Classical analysis

- Rewards generated independently at random
- Each machine has a different distribution of rewards.
- Update upper and lower bounds of the expected reward for each arm.
- Choose the arm with the highest upper bound.
- Good outcome: Upper bound remains highest - stick with action.

Classical analysis

- Rewards generated independently at random
- Each machine has a different distribution of rewards.
- Update upper and lower bounds of the expected reward for each arm.
- Choose the arm with the highest upper bound.
- Good outcome: Upper bound remains highest - stick with action.
- dissapointing outcome: Upper bound is no longer highest - switch to a different action.

Classical analysis

- Rewards generated independently at random
- Each machine has a different distribution of rewards.
- Update upper and lower bounds of the expected reward for each arm.
- Choose the arm with the highest upper bound.
- Good outcome: Upper bound remains highest - stick with action.
- dissapointing outcome: Upper bound is no longer highest - switch to a different action.
- Optimistic algorithm - always chooses action that might be best.

Playing in a Rigged casino

- The casino operator watches you and changes rewards of the machines to confuse you!

Playing in a Rigged casino

- The casino operator watches you and changes rewards of the machines to confuse you!
- Can you still find the best machine?

Playing in a Rigged casino

- The casino operator watches you and changes rewards of the machines to confuse you!
- Can you still find the best machine?
- What does “best machine” mean?

Example adversarial MAB game

Example adversarial MAB game

action1
action2
action3
action4
action5
action6
action7
action8

Example adversarial MAB game

	P_1
action1	$1/8$
action2	$1/8$
action3	$1/8$
action4	$1/8$
action5	$1/8$
action6	$1/8$
action7	$1/8$
action8	$1/8$

Example adversarial MAB game

	P_1	i_1
action1	$1/8$	
action2	$1/8$	
action3	$1/8$	
action4	$1/8$	$\Rightarrow$
action5	$1/8$	
action6	$1/8$	
action7	$1/8$	
action8	$1/8$	

Example adversarial MAB game

	P_1	i_1	$\boldsymbol{x}(1)$
action1	$1/8$		$.1$
action2	$1/8$		$.8$
action3	$1/8$		$.3$
action4	$1/8$	$\Rightarrow$	$.5$
action5	$1/8$		$.9$
action6	$1/8$		0
action7	$1/8$		1
action8	$1/8$		$.8$

Example adversarial MAB game

	P_1	i_1	$\boldsymbol{x}(1)$	p_2
action1	$1/8$		.1	.12
action2	$1/8$		.8	.12
action3	$1/8$		.3	.12
action4	$1/8$	$\Rightarrow$	.5	.16
action5	$1/8$		.9	.12
action6	$1/8$		0	.12
action7	$1/8$		1	.12
action8	$1/8$		.8	.12

Example adversarial MAB game

	P_1	i_1	$\boldsymbol{x}(1)$	p_2	i_2
action1	$1/8$		.1	.12	
action2	$1/8$		.8	.12	
action3	$1/8$		.3	.12	
action4	$1/8$	$\Rightarrow$	.5	.16	
action5	$1/8$		.9	.12	
action6	$1/8$		0	.12	
action7	$1/8$		1	.12	$\Rightarrow$
action8	$1/8$		.8	.12	

Example adversarial MAB game

	P_1	i_1	$\boldsymbol{x}(1)$	p_2	i_2	$\boldsymbol{x}(2)$
action1	$1/8$		.1	.12		.1
action2	$1/8$		.8	.12		.5
action3	$1/8$		.3	.12		.2
action4	$1/8$	$\Rightarrow$	.5	.16		.7
action5	$1/8$		.9	.12		1
action6	$1/8$		0	.12		.1
action7	$1/8$		1	.12	$\Rightarrow$	.7
action8	$1/8$		.8	.12		.2

Example adversarial MAB game

	P_1	i_1	$\boldsymbol{x}(1)$	p_2	i_2	$\boldsymbol{x}(2)$	p^3
action1	$1/8$		.1	.12		.1	0.11
action2	$1/8$		.8	.12		.5	0.11
action3	$1/8$		.3	.12		.2	0.11
action4	$1/8$	$\Rightarrow$	.5	.16		.7	0.15
action5	$1/8$		.9	.12		1	0.11
action6	$1/8$		0	.12		.1	0.11
action7	$1/8$		1	.12	$\Rightarrow$	.7	0.19
action8	$1/8$		.8	.12		.2	0.11

Example adversarial MAB game

	P_1	i_1	$\boldsymbol{x}(1)$	p_2	i_2	$\boldsymbol{x}(2)$	p^3	i_3
action1	$1/8$		.1	.12		.1	0.11	
action2	$1/8$		.8	.12		.5	0.11	$\Rightarrow$
action3	$1/8$		.3	.12		.2	0.11	
action4	$1/8$	$\Rightarrow$	.5	.16		.7	0.15	
action5	$1/8$		.9	.12		1	0.11	
action6	$1/8$		0	.12		.1	0.11	
action7	$1/8$		1	.12	$\Rightarrow$	.7	0.19	
action8	$1/8$		.8	.12		.2	0.11	

Example adversarial MAB game

	P_1	i_1	$\boldsymbol{x}(1)$	p_2	i_2	$\boldsymbol{x}(2)$	p^3	i_3	$\boldsymbol{x}(3)$
action1	$1/8$		.1	.12		.1	0.11		0
action2	$1/8$		.8	.12		.5	0.11	$\Rightarrow$	.2
action3	$1/8$		.3	.12		.2	0.11		.2
action4	$1/8$	$\Rightarrow$	.5	.16		.7	0.15		.8
action5	$1/8$		.9	.12		1	0.11		.8
action6	$1/8$		0	.12		.1	0.11		.2
action7	$1/8$		1	.12	$\Rightarrow$	.7	0.19		.4
action8	$1/8$		.8	.12		.2	0.11		.6

Example adversarial MAB game

	P_1	i_1	$\boldsymbol{x}(1)$	p_2	i_2	$\boldsymbol{x}(2)$	p^3	i_3	$\boldsymbol{x}(3)$	total
action1	$1/8$		.1	.12		.1	0.11		0	.2
action2	$1/8$		.8	.12		.5	0.11	$\Rightarrow$	.2	1.5
action3	$1/8$		.3	.12		.2	0.11		.2	.7
action4	$1/8$	$\Rightarrow$	.5	.16		.7	0.15		.8	2.0
action5	$1/8$		.9	.12		1	0.11		.8	2.7
action6	$1/8$		0	.12		.1	0.11		.2	.3
action7	$1/8$		1	.12	$\Rightarrow$	.7	0.19		.4	2.1
action8	$1/8$		.8	.12		.2	0.11		.6	1.6

The goal

- Total reward be close to total reward of best action.

The goal

- Total reward be close to total reward of best action.
- Weak: in expectation, Strong: With high probability.

The goal

- Total reward be close to total reward of best action.
- Weak: in expectation, Strong: With high probability.
- Why reward instead of loss?

The goal

- Total reward be close to total reward of best action.
- Weak: in expectation, Strong: With high probability.
- Why reward instead of loss?
- Because regret bounds that depend on the loss of the best action (rather than T) are impossible.

The basic algorithm

EXP3 = Exponential weights for Exploration and Exploitation

For each $t = 1, 2, \ldots$

1. Set

$$p_i(t) = (1-\gamma)\frac{w_i^t}{\sum_{j=1}^{K} w_j^t} + \frac{\gamma}{K} \qquad i = 1, \ldots, K.$$

The basic algorithm

EXP3 = Exponential weights for Exploration and Exploitation

For each $t = 1, 2, \ldots$

1. Set
$$p_i(t) = (1-\gamma)\frac{w_i^t}{\sum_{j=1}^K w_j^t} + \frac{\gamma}{K} \qquad i = 1, \ldots, K.$$
2. Draw i_t randomly accordingly to $p_1(t), \ldots, p_K(t)$

The basic algorithm

EXP3 = Exponential weights for Exploration and Exploitation

For each $t = 1, 2, \ldots$

1. Set
$$p_i(t) = (1-\gamma)\frac{w_i^t}{\sum_{j=1}^K w_j^t} + \frac{\gamma}{K} \qquad i = 1, \ldots, K.$$
2. Draw i_t randomly accordingly to $p_1(t), \ldots, p_K(t)$
3. Receive reward $x_{i_t}(t) \in [0, 1]$

The basic algorithm

EXP3 = Exponential weights for Exploration and Exploitation

For each $t = 1, 2, \ldots$

1. Set
$$p_i(t) = (1-\gamma)\frac{w_i^t}{\sum_{j=1}^K w_j^t} + \frac{\gamma}{K} \qquad i = 1, \ldots, K.$$
2. Draw i_t randomly accordingly to $p_1(t), \ldots, p_K(t)$
3. Receive reward $x_{i_t}(t) \in [0, 1]$
4. For $j = 1, \ldots, K$ set
$$\begin{aligned} \hat{x}_j(t) &= \begin{cases} x_j(t)/p_j(t) & \text{if } j = i_t \\ 0 & \text{otherwise,} \end{cases} \\ w_j^{t+1} &= w_t^j \exp\left(\gamma \hat{x}_j(t)/K\right) . \end{aligned}$$

Basic bound

- Let T be the number of iterations and that algorithm **Exp3** is run with

$$\gamma = \min\left\{1, \sqrt{\frac{K \ln K}{(e-1)T}}\right\}.$$

Basic bound

- Let T be the number of iterations and that algorithm **Exp3** is run with

$$\gamma = \min\left\{1, \sqrt{\frac{K \ln K}{(e-1)T}}\right\}.$$

- $G_{\max}$ = Total gain of best Arm.
  $G_{\mathbf{Exp3}}$ = total gain of Algorith (RV)

Basic bound

- Let T be the number of iterations and that algorithm **Exp3** is run with

$$\gamma = \min\left\{1, \sqrt{\frac{K \ln K}{(e-1)T}}\right\}.$$

- $G_{\max}$ = Total gain of best Arm.
  $G_{\textbf{Exp3}}$ = total gain of Algorith (RV)
- Then

$$G_{\max} - \mathbf{E}[G_{\textbf{Exp3}}] \leq 2\sqrt{e-1}\sqrt{TK \ln K} \leq 2.63\sqrt{TK \ln K}$$

Ideas of proof

1. Setting

$$\hat{x}_j(t) = \begin{cases} x_j(t)/p_j(t) & \text{if } j = i_t \\ 0 & \text{otherwise,} \end{cases}$$

guarantees that $\mathbf{E}\left(\sum_{t=1}^{t} \hat{x}_j(t)\right) = \sum_{t=1}^{T} x_j(t)$ i.e. estimate of total gain is Unbiased.

Ideas of proof

1. Setting
$$\hat{x}_j(t) = \begin{cases} x_j(t)/p_j(t) & \text{if } j = i_t \\ 0 & \text{otherwise,} \end{cases}$$
guarantees that $\mathbf{E}\left(\sum_{t=1}^{t} \hat{x}_j(t)\right) = \sum_{t=1}^{T} x_j(t)$ i.e. estimate of total gain is Unbiased.
2. Setting $\gamma = O(\sqrt{\frac{K \log K}{T}})$ guarantees variance of estimator is not too large.

Ideas of proof

1. Setting
$$\hat{x}_j(t) = \begin{cases} x_j(t)/p_j(t) & \text{if } j = i_t \\ 0 & \text{otherwise,} \end{cases}$$
guarantees that $\mathbf{E}\left(\sum_{t=1}^{t} \hat{x}_j(t)\right) = \sum_{t=1}^{T} x_j(t)$ i.e. estimate of total gain is Unbiased.
2. Setting $\gamma = O(\sqrt{\frac{K \log K}{T}})$ guarantees variance of estimator is not too large.
3. **Exp3** mimicks **Hedge** sufficiently well.

Lower bound

- Choose all gains independently at random to be 0 or 1.

Lower bound

- Choose all gains independently at random to be 0 or 1.
- $K - 1$ actions use probs $(1/2, 1/2)$.

Lower bound

- Choose all gains independently at random to be 0 or 1.
- $K - 1$ actions use probs $(1/2, 1/2)$.
- One action (chosen at random) uses probs $1/2 + \epsilon, 1/2 - \epsilon$.

Lower bound

- Choose all gains independently at random to be 0 or 1.
- $K-1$ actions use probs $(1/2, 1/2)$.
- One action (chosen at random) uses probs $1/2+\epsilon, 1/2-\epsilon$.
- The Bayes optimal algorithm has expected regret at least

$$\frac{1}{20}\min\left(\sqrt{KT}, T\right)$$

Tuning γ online

Algorithm Exp3.1
Initialization: Let $t = 1$, and $\hat{G}_i(1) = 0$ for $i = 1, \ldots, K$

Repeat for $r = 0, 1, 2, \ldots$

1. Let $g_r = (K \ln K)/(e-1)\, 4^r$.
2. Restart **Exp3** choosing $\gamma_r = \min\left\{1, \sqrt{\frac{K \ln K}{(e-1)g_r}}\right\}$.
3. **While** $\max_i \hat{G}_i(t) \leq g_r - K/\gamma_r$ **do:**
   (a) Let i_t be the random action chosen by **Exp3** and $x_{i_t}(t)$ the corresponding reward.
   (b) $\hat{G}_i(t+1) = \hat{G}_i(t) + \hat{x}_i(t)$ for $i = 1, \ldots, K$.
   (c) $t := t + 1$

Bound for Exp3.1

$$\begin{aligned} G_{\max} - \mathbf{E}[G_{\mathbf{Exp3.1}}] &\leq 8\sqrt{e-1}\sqrt{G_{\max} K \ln K} + 8(e-1)K + 2K \ln K \\ &= O(\sqrt{G_{\max} K \ln K} \end{aligned}$$

Allowing switching actions

Algorithm Exp3.S
Parameters: Reals $\gamma \in (0,1]$ and $\alpha > 0$.
Initialization: $w_i(1) = 1$ for $i = 1, \ldots, K$.

For each $t = 1, 2, \ldots$

1. Set
$$p_i(t) = (1-\gamma)\frac{w_i(t)}{\sum_{j=1}^{K} w_j(t)} + \frac{\gamma}{K} \qquad i = 1, \ldots, K.$$
2. Draw i_t according to the probabilities $p_1(t), \ldots, p_K(t)$.
3. Receive reward $x_{i_t}(t) \in [0,1]$.
4. For $j = 1, \ldots, K$ set
$$\begin{aligned} \hat{x}_j(t) &= \begin{cases} x_j(t)/p_j(t) & \text{if } j = i_t \\ 0 & \text{otherwise,} \end{cases} \\ w_j(t+1) &= w_j(t)\exp\left(\gamma \hat{x}_j(t)/K\right) + \frac{e\alpha}{K}\sum_{i=1}^{K} w_i(t)\,. \end{aligned}$$

Bound for Exp3.S

- Hardness of sequence = number of switches offline is allowed:

$$S \geq \mathrm{H}(j_1, \ldots, j_T) \stackrel{\text{def}}{=} 1 + |\{1 \leq \ell < T \; : \; j_\ell \neq j_{\ell+1}\}| \; .$$

Bound for Exp3.S

- Hardness of sequence = number of switches offline is allowed:

$$S \geq \mathrm{H}(j_1, \ldots, j_T) \stackrel{\text{def}}{=} 1 + |\{1 \leq \ell < T \ : \ j_\ell \neq j_{\ell+1}\}| \ .$$

- Assume $\alpha = 1/T$ and $\gamma = \min\left\{1, \sqrt{\frac{K(S\ln(KT)+e)}{(e-1)T}}\right\}$.

Bound for Exp3.S

- Hardness of sequence = number of switches offline is allowed:

$$S \geq \mathrm{H}(j_1, \ldots, j_T) \stackrel{\text{def}}{=} 1 + |\{1 \leq \ell < T \; : \; j_\ell \neq j_{\ell+1}\}| \; .$$

- Assume $\alpha = 1/T$ and $\gamma = \min\left\{1, \sqrt{\frac{K(S\ln(KT)+e)}{(e-1)T}}\right\}$.
- Then

$$\begin{aligned} G_S - \mathbf{E}\left[G_{\mathbf{Exp3.S}}\right] &\leq 2\sqrt{e-1}\sqrt{KT\left(S\ln(KT)+e\right)} \\ &= O(\sqrt{KTS\ln(KT)}) \end{aligned}$$

Combining strategies

- K possible actions and N prediction strategies or experts.

Combining strategies

- K possible actions and N prediction strategies or experts.
- $N \gg K$

Combining strategies

- K possible actions and N prediction strategies or experts.
- $N \gg K$
- Expert i predicts with a distribution over actions $\xi^i(t) \in [0, 1]^K$

Combining strategies

- K possible actions and N prediction strategies or experts.
- $N \gg K$
- Expert i predicts with a distribution over actions $\xi^i(t) \in [0,1]^K$
- Reward of expert i is $\xi^i(t) \cdot \boldsymbol{x}(t)$

Combining strategies

- K possible actions and N prediction strategies or experts.
- $N \gg K$
- Expert i predicts with a distribution over actions $\xi^i(t) \in [0, 1]^K$
- Reward of expert i is $\xi^i(t) \cdot \boldsymbol{x}(t)$
- Considering experts as actions, we get a bound $O(\sqrt{G_{\max} N \log N})$ on the regret.

Combining strategies

- K possible actions and N prediction strategies or experts.
- $N \gg K$
- Expert i predicts with a distribution over actions $\xi^i(t) \in [0, 1]^K$
- Reward of expert i is $\xi^i(t) \cdot \boldsymbol{x}(t)$
- Considering experts as actions, we get a bound $O(\sqrt{G_{\max} N \log N})$ on the regret.
- By acting smarter, we can get a bound $O(\sqrt{G_{\max} K \log N})$

Exponential Exploration and Explotation using Experts

For each $t = 1, 2, \ldots$

1. Get advice vectors $\boldsymbol{\xi}^1(t), \ldots, \boldsymbol{\xi}^N(t)$.
2. Set $W_t = \sum_{i=1}^{N} w_i(t)$ and for $j = 1, \ldots, K$ set

$$p_j(t) = (1-\gamma) \sum_{i=1}^{N} \frac{w_i(t)\xi_j^i(t)}{W_t} + \frac{\gamma}{K} \, .$$

3. Draw action i_t randomly according to the probabilities $p_1(t), \ldots, p_K(t)$.
4. Receive reward $x_{i_t}(t) \in [0, 1]$.
5. For $j = 1, \ldots, K$ set

$$\hat{x}_j(t) = \begin{cases} x_j(t)/p_j(t) & \text{if } j = i_t \\ 0 & \text{otherwise,} \end{cases}$$

6. For $i = 1, \ldots, N$ set

$$\begin{aligned} \hat{y}_i(t) &= \boldsymbol{\xi}^i(t) \cdot \hat{\boldsymbol{x}}(t) \\ w_i(t+1) &= w_i(t) \exp\left(\gamma \hat{y}_i(t)/K\right) \, . \end{aligned}$$

Summary

- We can achieve diminishing regret even when only gain of chosen action is observable.

Summary

- We can achieve diminishing regret even when only gain of chosen action is observable.
- The increase in the regret is a result of the limited information. $O(\sqrt{TK \log K})$ instead of $O(\sqrt{T \log K})$.

Summary

- We can achieve diminishing regret even when only gain of chosen action is observable.
- The increase in the regret is a result of the limited information. $O(\sqrt{TK \log K})$ instead of $O(\sqrt{T \log K})$.
- We can handle sequences with S switches: $O(\sqrt{KTS \ln(KT)})$

Summary

- We can achieve diminishing regret even when only gain of chosen action is observable.
- The increase in the regret is a result of the limited information. $O(\sqrt{TK \log K})$ instead of $O(\sqrt{T \log K})$.
- We can handle sequences with S switches: $O(\sqrt{KTS \ln(KT)})$
- If we have many strategies N but only few actions K we can achieve bounds of the form $O(\sqrt{TK \log N})$.