MLSS2024@Okinawa

An introduction to regret analysis: environment models and best-of-both-worlds

March, 2024

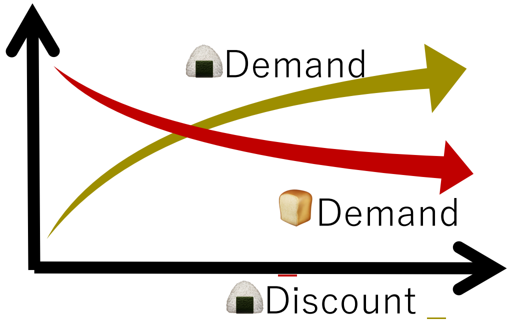
RIKEN AIP / NEC Data Science Laboratory

Shinji Ito

Self-introduction: Shinji Ito (伊藤伸志), PhD

- Affiliation : NEC Corporation, Data Science Laboratory
  RIKEN AIP, Sequential Decision-Making Team (Team Leader)

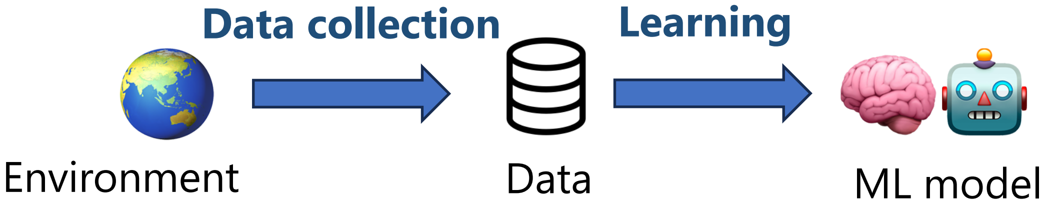
- Short bio.:
  - As a graduate student (~2015), SI worked on research on numerical calculations and inverse problems, and completed my master's degree
  - 2015~ NEC Corporation
    - 2015 - 2017 : Research and development of price optimization
    - 2018 – Present:
      - Research and development of online learning
      - Got a PhD (Information Science and Engineering)



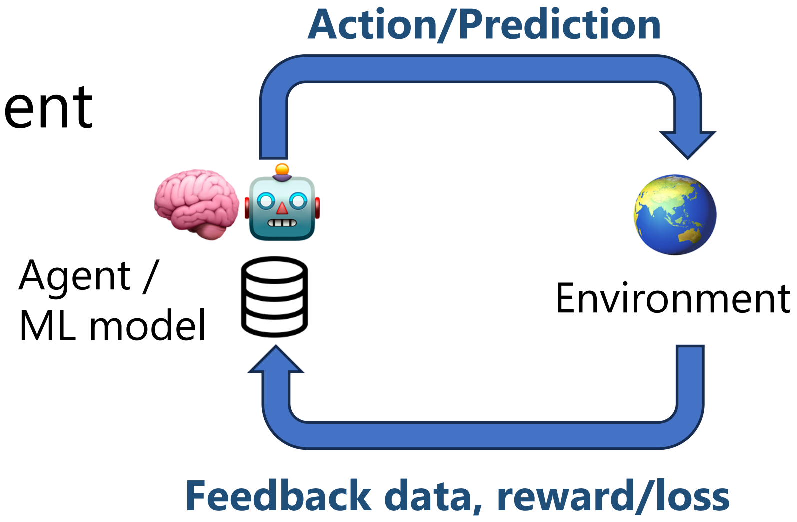
- Research interests:
  Applied mathematics, especially decision-making under uncertainty

Offline learning and online learning

- **Offline** (batch) learning / data-driven decision-making
  - Learning process with batch data
  - 👍 stability and consistency
  - 👎 difficulty in real-time adaptation



- **Online** learning / sequential decision-making:
  - Learning via repetitive interactions with the environment
  - 👍 flexibility, memory efficiency
  - 👎 sensitivity to noise, difficulty in tuning



Scope and goal in this lecture

- Topics in sequential decision-making
  - **Online learning**
  - **Bandit algorithms**
  - **Regret analysis**
  - Reinforcement learning
  - Continual learning
  - Repeated games
  - Competitive analysis
  - …

Scope of this lecture

- Goal:
  - Introduce the basics of online learning and the idea of regret by looking at simple examples, such as the *expert problem* and *multi-armed bandit*
  - Explore the analysis methods and the results of *Best-of-both-worlds* bounds

Outline of the talk

- Problem setup
  - Prediction with expert advice and multi-armed bandit
  - Two models for environments

- Basic results of regret analysis
  - Algorithms and regret analysis for the expert problem
  - Comparison of regrets in stochastic and adversarial environments

- Best-of-both-worlds algorithms and analysis
  - Hedge with adaptive learning rate
  - Analysis between stochastic and adversarial (stochastic environment with adversarial corruption)
  - Other recent developments

Outline of the talk

- **Problem setup**
  - **Prediction with expert advice and multi-armed bandit**
  - Two models for environments

- Basic results of regret analysis
  - Algorithms and regret analysis for the expert problem
  - Comparison of regrets in stochastic and adversarial environments

- Best-of-both-worlds algorithms and analysis
  - Hedge with adaptive learning rate
  - Analysis between stochastic and adversarial (stochastic environment with adversarial corruption)
  - Other recent developments

Expert problem (prediction with expert advice)

N experts

- I 😀 decided to imitate my friends 🤓🐵👺👻…🐶 and try horse racing.
- For T races, I 😀 choose one friend and buy the same betting ticket as that friend (and then disclose all friends' results)

Expert problem (prediction with expert advice)

N experts

- I 😀 decided to imitate my friends $\underbrace{🤓🐵👺👻…🐶}_{}$ and try horse racing.
- For T races, I 😀 choose one friend and buy the same betting ticket as that friend (and then disclose all friends' results)

🤓Tell me

😀 🤓🐵👺👻

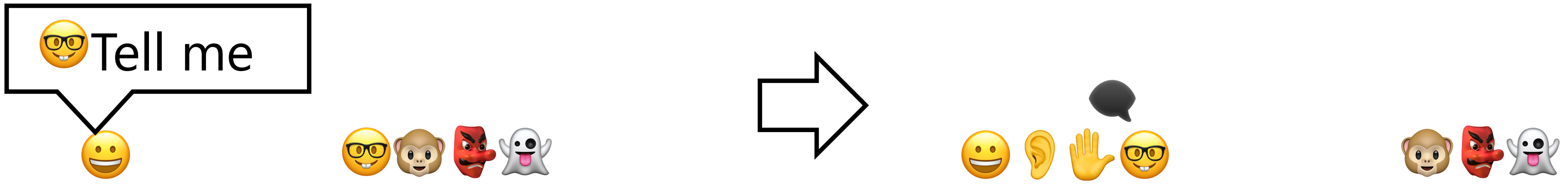
Friends' performance (The numbers in the table represent loss or (-1) x profit)

round	1	2	3	4	…	T	total
🤓	😀				…		
🐵					…		
👺					…		
😀					…		

Expert problem (prediction with expert advice)

N experts

- I 😀 decided to imitate my friends 🤓🐵👺👻…🐶 and try horse racing.
- For T races, I 😀 choose one friend and buy the same betting ticket as that friend (and then disclose all friends' results)



Friends' performance (The numbers in the table represent loss or (-1) x profit)

round	1	2	3	4	…	T	total
🤓	1.0 😀						1.0
🐵	0.5						0.5
👺	0.2						0.2
😀	1.0						1.0

Expert problem (prediction with expert advice)

N experts

- I 😀 decided to imitate my friends 🤓🐵👺👻…🐶 and try horse racing.
- For T races, I 😀 choose one friend and buy the same betting ticket as that friend (and then disclose all friends' results)

👺Tell me

😀 🤓🐵👺👻

Friends' performance (The numbers in the table represent loss or (-1) x profit)

round	1	2	3	4	...	T	total
🤓	1.0 😀						1.0
🐵	0.5						0.5
👺	0.2	😀					0.2
😀	1.0						1.0

Expert problem (prediction with expert advice)

N experts

- I 😀 decided to imitate my friends $\overbrace{\text{🤓🐵👺👻...🐶}}^{\text{N experts}}$ and try horse racing.
- For T races, I 😀 choose one friend and buy the same betting ticket as that friend (and then disclose all friends' results)

👺Tell me

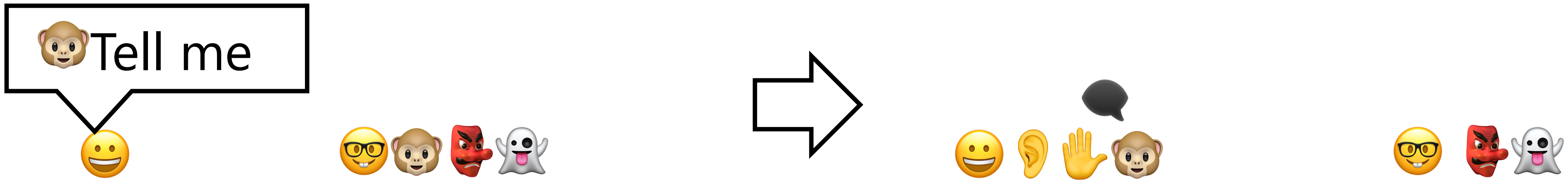
😀 🤓🐵👺👻 ⇨ 😀👂✋🗨👺 🤓🐵 👻

Friends' performance (The numbers in the table represent loss or (-1) x profit)

round	1	2	3	4	…	T	total
🤓	1.0 😀	0.6					1.6
🐵	0.5	0.1					0.6
👺	0.2	0.3 😀					0.5
😀	1.0	0.3					1.3

Expert problem (prediction with expert advice)

- I 😀 decided to imitate my friends $\underbrace{🤓🐵👺👻…🐶}_{\text{N experts}}$ and try horse racing.
- For T races, I 😀 choose one friend and buy the same betting ticket as that friend (and then disclose all friends' results)



Friends' performance (The numbers in the table represent loss or (-1) x profit)

round	1	2	3	4	…	T	total
🤓	1.0 😀	0.6	0.8				2.4
🐵	0.5	0.1	0.6 😀				1.2
👺	0.2	0.3 😀	0.9				1.4
😀	1.0	0.3	0.6				1.9

Expert problem (prediction with expert advice)

N experts

- I 😀 decided to imitate my friends $\underbrace{\text{🤓🐵👺👻…🐶}}$ and try horse racing.
- For T races, I 😀 choose one friend and buy the same betting ticket as that friend (and then disclose all friends' results)

Repeat. . . . .

😀 🤓🐵👺👻

Friends' performance (The numbers in the table represent loss or (-1) x profit)

round	1	2	3	4	…	T	total
🤓	1.0 😀	0.6	0.8	0.1	· · ·	0.2 😀	26.1
🐵	0.5	0.1	0.6 😀	1.0	· · ·	0.2	20.3
👺	0.2	0.3 😀	0.9	0.7 😀	· · ·	0.8	30.6
😀	1.0	0.3	0.6	0.7	· · ·	0.2	27.8

Expert problem (prediction with expert advice)

N experts

- I 😀 decided to imitate my friends $\underbrace{\text{🤓🐵👺👻…🐶}}$ and try horse racing.
- For T races, I 😀 choose one friend and buy the same betting ticket as that friend (and then disclose all friends' results)

I want to find out who is the best and minimize my losses as much as possible…

😀 🤓🐵👺👻

Friends' performance (The numbers in the table represent loss or (-1) x profit)

round	1	2	3	4	…	T	total
🤓	1.0 😀	0.6	0.8	0.1	…	0.2 😀	26.1
🐵	0.5	0.1	0.6 😀	1.0	…	0.2	20.3
👺	0.2	0.3 😀	0.9	0.7 😀	…	0.8	30.6
😀	1.0	0.3	0.6	0.7	…	0.2	27.8

Evaluation measure: Regret R_T

Friends' performance (The numbers in the table represent loss or (-1) x profit)

round	1	2	3	4	...	T	total
🤓	1.0 😀	0.6	0.8	0.1	...	0.2 😀	26.1
🐵	0.5	0.1	0.6 😀	1.0	...	0.2	20.3
👺	0.2	0.3 😀	0.9	0.7 😀	...	0.8	30.6
😀	1.0	0.3	0.6	0.7	...	0.2	27.8

Luckiest friend 🐵

🐵's overall score (cumulative loss) was 20.3.

Me 😀 ∘○○ I wish if I had trusted 🐵 from the beginning. . .

A value that quantifies this *regret*:

$$R_T = \sum_{t=1}^{T} \ell_{ti_t} - \min_{i^* \in [N]} \sum_{t=1}^{T} \ell_{ti^*} = 27.8 - 20.3 = 7.5$$

i_t: 😀's chosen friend

i^*: 🐵 luckiest friend

R_T is small ⇒ The result is close to the result if you continue to take the best option

Problem settings and regrets

round $t = 1,2,\ldots,T$

expert	round	1	2	3	4	...	T	total
$i = 1$	🤓	1.0 😀	0.6	0.8	0.1	...	0.2 😀	26.1
$i = 2$	🐵	0.5	0.1	0.6 😀	1.0	...	0.2	20.3
$i = 3$	👺	0.2	0.3 😀	0.9	0.7 😀	...	0.8	30.6
	😀	1.0	0.3	0.6	0.7	...	0.2	27.8

- ℓ_{ti}: Loss for choosing expert i at round t
  - Assume $\ell_{ti} \in [0,1]$
- i_t: Expert selected by the algorithm 😀 at round t
- $R_T = \sum_{t=1}^{T} \ell_{ti_t} - \min_{i^* \in [N]} \sum_{t=1}^{T} \ell_{ti^*}$: regret
  - If $\underline{R_T = o(T)}$ is achieved, it can be said to be a good algorithm in a sense.

    inferior linear regret , no-regret, vanishing regret, etc. $\left(\text{as } \lim_{T\to\infty} \frac{R_T}{T} = 0\right)$
- Note: Regret R_T is based on comparison with best expert i^* *fixed over all rounds.*
  If we want to track the round-wise best expert i_t^*, we need to use different notion of regret, such as *adaptive regret* and *dynamic regret*

Expert problems : various applications

- (Complete information type) Repeated game

round	1	2	3	4	...	T	total
✊ Rock					...		
✌️ Scissors					...		
🖐️ Paper					...		
Which move will you make?					...		

Expert problems : various applications

- (Complete information type) Repeated game

✋👩

round	1	2	3	4	…	T	total
✊ Rock	1				…		
✌ Scissors	-1				…		
✋ Paper	0				…		
Which move will you make?					…		

Expert problems : various applications

- (Complete information type) Repeated game

✋👩 ✊👩

round	1	2	3	4	…	T	total
✊ Rock	1	0			…		
✌️ Scissors	-1	1			…		
✋ Paper	0	-1			…		
Which move will you make?					…		

Expert problems : various applications

- (Complete information type) Repeated game
- Investment in stocks etc.

Round (monthly)	1	2	3	4	…	T	total
company A's stock	+$100				…		
Reserve in investment trust	-$200				…		
held in cash	$0				…		
What to invest in?					…		

Expert problems : various applications

- (Complete information type) Repeated game
- Investment in stocks etc.
- Selecting the order quantity of the product

round (date)	1	2	3	…	T	total
100 pieces	Opportunity loss ×20			…		
120 pieces	0			…		
140 pieces	Waste loss ×20			…		
How many products should I order?				…		

Expert problems : various applications

- (Complete information type) Repeated game
- Investment in stocks etc.
- Selecting the order quantity of the product
- Model selection/integration in online prediction

Round (test data)	1	2	3	…	T	total
linear model	Prediction error : 0.3			…		
DNN	Prediction error : 0.5			…		
BGDT	Prediction error : 0.2			…		
Which model should I use?				…		

Expert problems : various applications

- (Complete information type) Repeated game
- Investment in stocks etc.
- Selecting the order quantity of the product
- Model selection/integration in online prediction
- Parameter selection in online prediction

Round (test data)	1	2	3	4	…	T	total
Learning rate 0.1, batch size 10					…		
Learning rate 0.3, batch size 10					…		
Learning rate 0.3, batch size 30					…		
Which model should I use?					…		

Multi-armed bandit problem

N experts

- I 😀 decided to imitate my friends 🤓🐵👺👻…🐶 and try horse racing.
- For T races, I 😀 choose one friend and buy the same betting ticket as that friend (**only the chosen friend** will tell you which ticket is bought)

Multi-armed bandit problem

N experts

- I 😀 decided to imitate my friends 🤓🐵👺👻…🐶 and try horse racing.
- For T races, I 😀 choose one friend and buy the same betting ticket as that friend (**only the chosen friend** will tell you which ticket is bought)

🤓Tell me

😀 🤓 🐵 👺👻

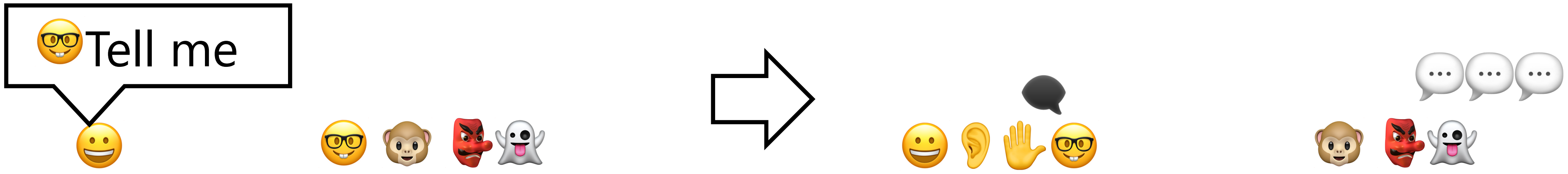
Friends' performance (The numbers in the table represent loss or (-1) x profit)

round	1	2	3	4	…	T	total
🤓	😀				…		
🐵					…		
👺					…		
😀					…		

Multi-armed bandit problem

N experts

- I 😀 decided to imitate my friends 🤓🐵👺👻…🐶 and try horse racing.
- For T races, I 😀 choose one friend and buy the same betting ticket as that friend (**only the chosen friend** will tell you which ticket is bought)



Friends' performance (The numbers in the table represent loss or (-1) x profit)

round	1	2	3	4	…	T	total
🤓	1.0 😀				…		
🐵	?				…		
👺	?				…		
😀	1.0				…		1.0

Multi-armed bandit problem

N experts

- I 😀 decided to imitate my friends $\underbrace{\text{🤓🐵👺👻…🐶}}$ and try horse racing.
- For T races, I 😀 choose one friend and buy the same betting ticket as that friend (**only the chosen friend** will tell you which ticket is bought)

I want to find out who is the best and minimize my losses as much as possible…

😀 🤓🐵👺👻

Friends' performance (The numbers in the table represent loss or (-1) x profit)

round	1	2	3	4	…	T	total
🤓	1.0 😀	?	?	?	…	0.2 😀	
🐵	?	?	0.6 😀	?	…	?	
👺	?	0.3 😀	?	0.7 😀	…	?	
😀	1.0	0.3	0.6	0.7	…	0.2	27.8

Evaluation measure: Regret

Friends' performance (The numbers in the table represent loss or (-1) x profit)

round	1	2	3	4	...	T	total
	1.0	?	?	?	...	0.2	26.1
	?	?	0.6	?	...	?	20.3
	?	0.3	?	0.7	...	?	30.6
	1.0	0.3	0.6	0.7	...	0.2	27.8

$$R_T = \sum_{t=1}^{T} \ell_{ti_t} - \min_{i^* \in [N]} \sum_{t=1}^{T} \ell_{ti^*} = 27.8 - 20.3 = 7.5$$

i_t: 's chosen friend

i^*: luckiest friend

- The definition of regret is the same as the expert problem.
- Although the loss ℓ_{ti} for $i \neq i_t$ cannot be observed, it is assumed that it is generated in advance.
- Note that, i^* or $\min_{i^* \in [N]} \sum_{t=1}^{T} \ell_{ti^*}$ cannot be observed in general even after the process is over.

Therefore, the value of R_T cannot be known (even after the T-th round)

Outline of the talk

- **Problem setup**
  - Prediction with expert advice and multi-armed bandit
  - **Two models for environments**

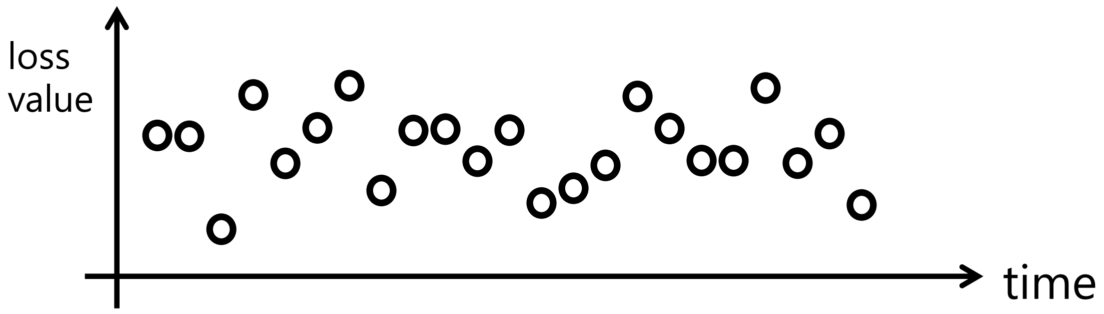
- Basic results of regret analysis
  - Algorithms and regret analysis for the expert problem
  - Comparison of regrets in stochastic and adversarial environments

- Best-of-both-worlds algorithms and analysis
  - Hedge with adaptive learning rate
  - Analysis between stochastic and adversarial (stochastic environment with adversarial corruption)
  - Other recent developments

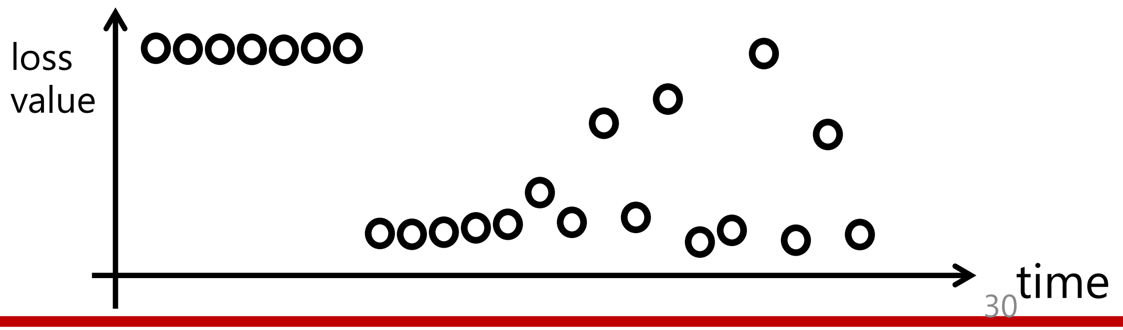
Two models for loss sequences (environments)

round	1	2	3	4	…	T	total
🤓					…		
🐵					…		
👺					…		
😀					…		

1. **Stochastic environment model**:
   Losses and rewards are i.i.d.
   (Values in cells of the same color follow an identical distribution)



2. **Adversarial environment model**:
   Losses and rewards **change arbitrarily**
   (the distribution may changes even in the cells of the same color**)**

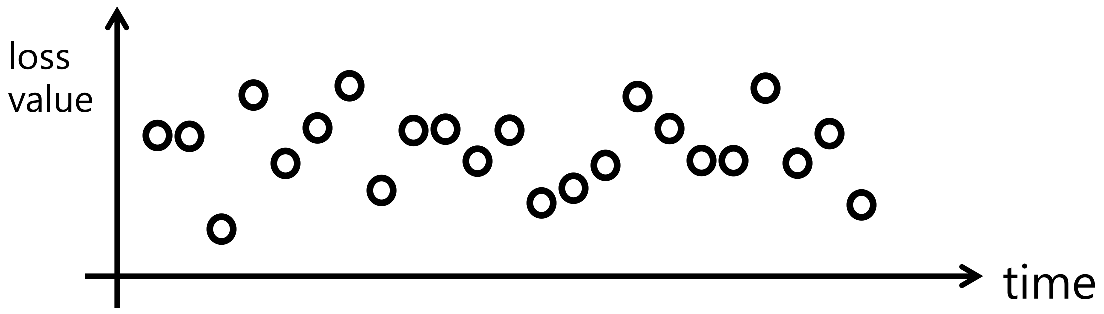


Expert problem : two loss models

Round (test data)	1	2	3	4	...	T	total
linear model					...		
DNN					...		
BGDT					...		
Which model should I use?					...		

model selection

1. **Stochastic environment model**:
   Losses and rewards are i.i.d.
   (Values in cells of the same color follow an identical distribution)



2. **Adversarial environment model**:
   Losses and rewards **change arbitrarily**
   (the distribution may changes even in the cells of the same color)

Expert problem : two loss models

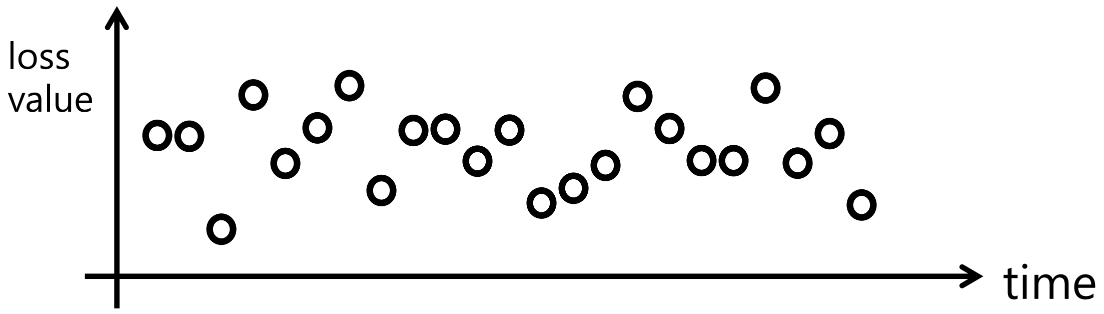
round (date)	1	2	3	4	…	T	total
100 pieces					…		
120 pieces					…		
140 pieces					…		
How many items ordered?					…		

Order quantity optimization

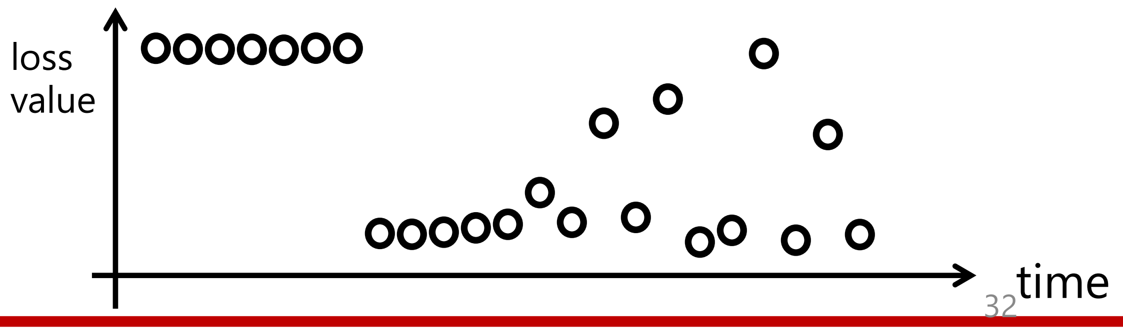
daily necessities, ✏️ 🔋
food etc. 🍙🍞

seasonal products etc.
🍨🍦🌂

1. **Stochastic environment model**:
   Losses and rewards are i.i.d.
   (Values in cells of the same color follow an identical distribution)



2. **Adversarial environment model**:
   Losses and rewards **change arbitrarily**
   (the distribution may changes even in the cells of the same color**)**



Expert problem : two loss models

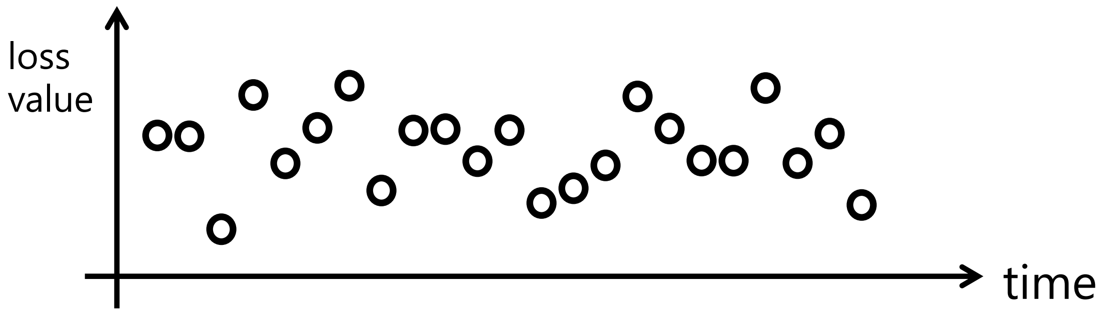
round	1	2	3	4	…	T	total
✊ Rock					…		
✌ Scissors					…		
✋ Paper					…		
Which move will you make?					…		

Rock, paper, scissors

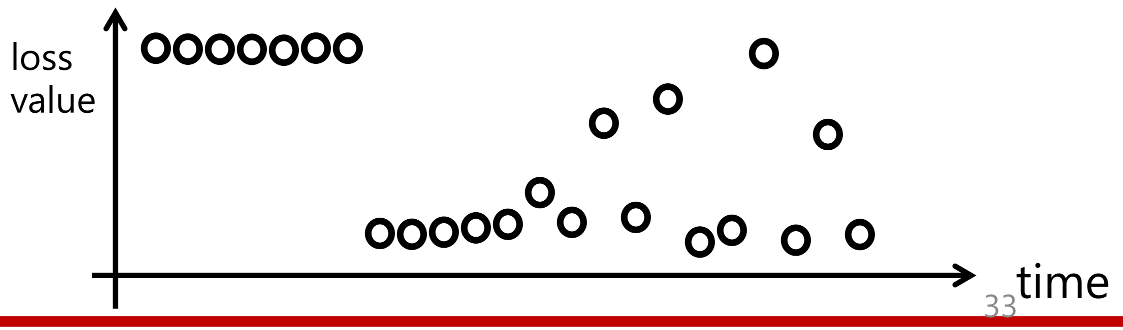
✊ 👩 Random? ?

? ✊ 👩 Strategic?

1. **Stochastic environment model**:
   Losses and rewards are i.i.d.
   (Values in cells of the same color follow an identical distribution)



2. **Adversarial environment model**:
   Losses and rewards **change arbitrarily**
   (the distribution may changes even in the cells of the same color**)**

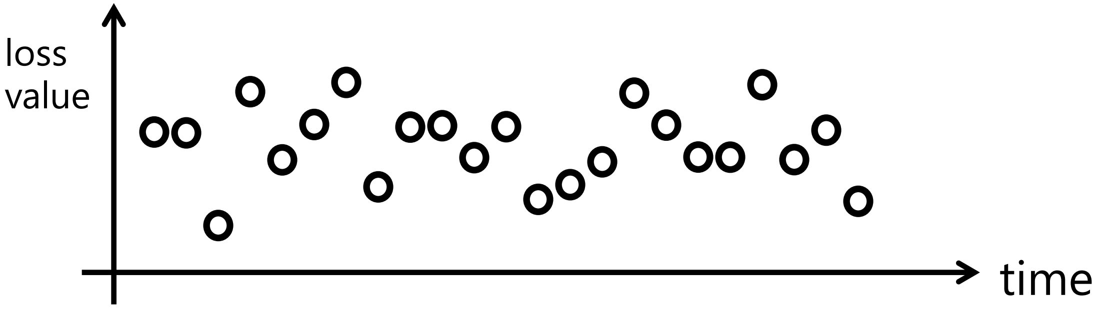


Expert problem : two loss models

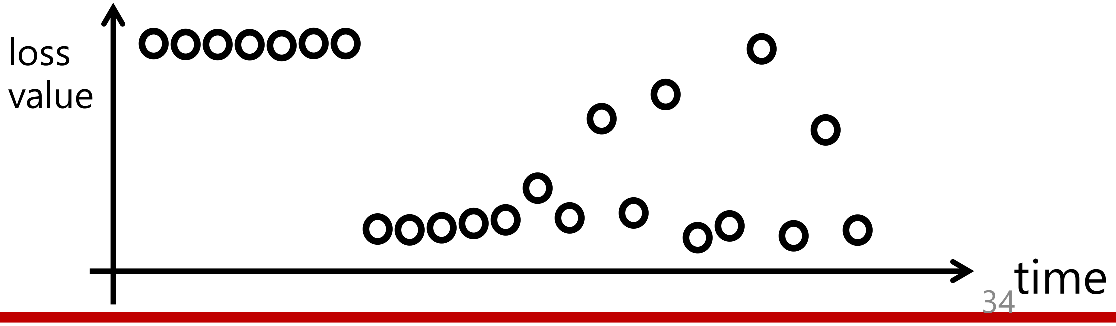
round	1	2	3	4	...	T	total
🤓					...		
🐵					...		
👺					...		
😀					...		

Choice of environmental model is highly non-trivial, which requires expertise in the application

1. **Stochastic environment model**:
   Losses and rewards are i.i.d.
   (Values in cells of the same color follow an identical distribution)



2. **Adversarial environment model**:
   Losses and rewards **change arbitrarily**
   (the distribution may changes even in the cells of the same color**)**



Outline of the talk

- Problem setup
    - Prediction with expert advice and multi-armed bandit
    - Two models for environments

- **Basic results of regret analysis**
    - **Algorithms and regret analysis for the expert problem**
    - Comparison of regrets in stochastic and adversarial environments

- Best-of-both-worlds algorithms and analysis
    - Hedge with adaptive learning rate
    - Analysis between stochastic and adversarial (stochastic environment with adversarial corruption)
    - Other recent developments

Notes on definitions and assumptions

- In the following slides, when the algorithms and/or the environments contains randomness, we consider the regret defined by

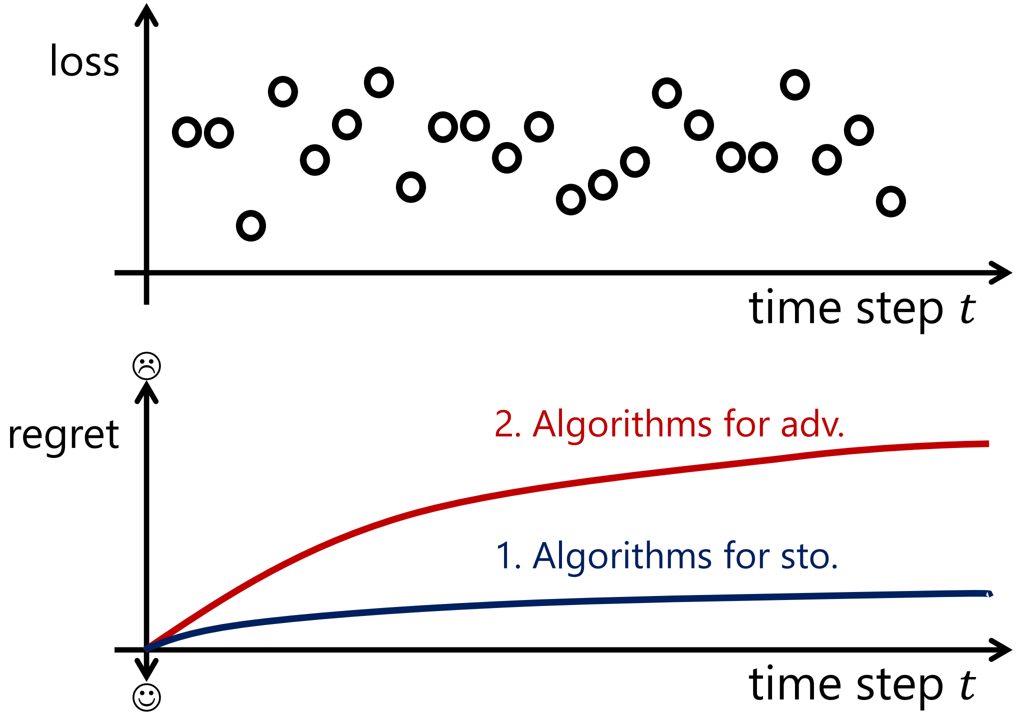
$$R_T = \mathrm{E}\left[\sum_{t=1}^{T} \ell_{t i_t}\right] - \min_{i^* \in [N]} \mathrm{E}\left[\sum_{t=1}^{T} \ell_{t i^*}\right]$$

  - $\mathrm{E}[\cdot]$ means expectation w.r.t. randomness in algorithms and environments
  - This definition allows us to handle standard regret notions in both stochastic and adversarial environment models in a unified way.

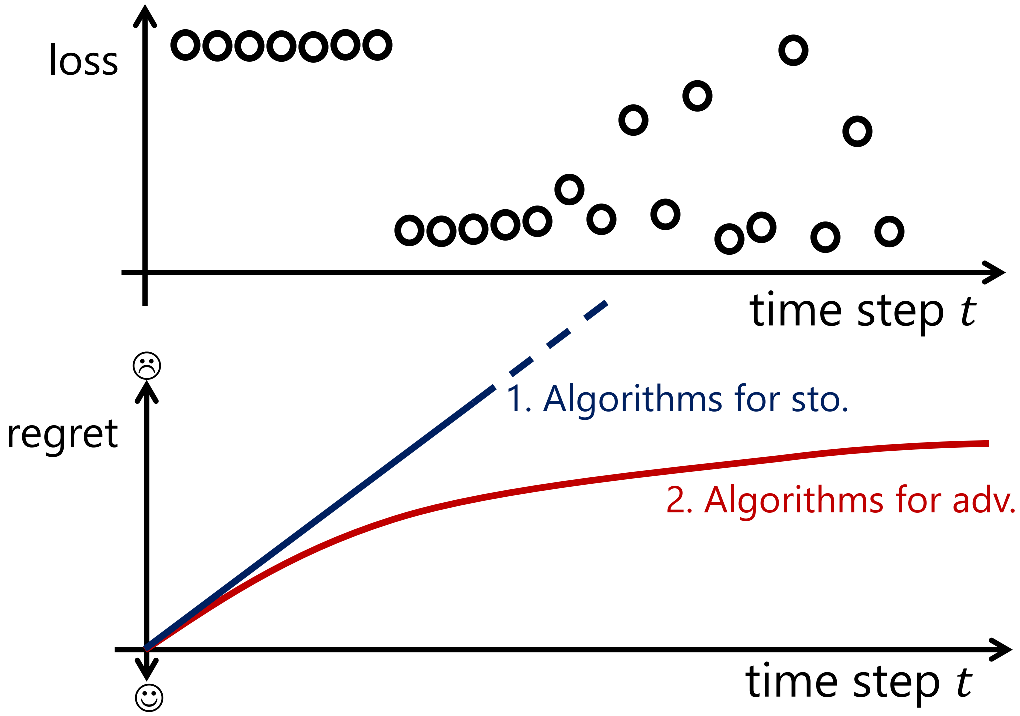
- In the analysis for evaluating R_T, we omit $\mathrm{E}[\cdot]$ for simplicity
  - For example, $R_T \leq \mathrm{E}[A + B + \cdots]$ can be simply written as $R_T \leq A + B + \cdots$
  - This omission is allowed only when there is no problem in doing so, e.g., the case in which we can apply Jensen's inequality.

Two loss models and two algorithms

1. **Stochastic environments**:
loss vectors ℓ^t are **i.i.d.**



2. **Adversarial environments**:
$\{\ell^t\}_{t=1}^T$ is an **arbitrary sequence** in $[0,1]^K$



- It is important to choose the right algorithm for the environments

Follow-the-leader (FTL) algorithm

$t = 1, 2, 3, \dots, T$

	round	1	2	3	4	…	T	total
$i = 1$	🤓	😀				…		0.0
$i = 2$	🐵	😀				…		0.0
$i = 3$	👺	😀				…		0.0
	😀					…		0.0

Follow-the-leader (FTL) algorithm:
At each round t, choose an expert i_t from S_t, the set of experts with the top overall performance until the last round $(t-1)$ so far:

$$i_t \in S_t := \arg\min_{i \in [N]} \left\{ \sum_{s=1}^{t-1} \ell_{si} \right\}$$

Follow-the-leader (FTL) algorithm

$t = 1, 2, 3, \dots, T$

	round	1	2	3	4	...	T	total
$i = 1$	🤓	1.0 😀				...		1.0
$i = 2$	🐵	0.5 😀				...		0.5
$i = 3$	👺	0.2 😀	😀			...		0.2
	😀	1.0				...		1.0

Follow-the-leader (FTL) algorithm:
At each round t, choose an expert i_t from S_t, the set of experts with the top overall performance until the last round $(t-1)$ so far:

$$i_t \in S_t := \arg\min_{i \in [N]} \left\{ \sum_{s=1}^{t-1} \ell_{si} \right\}$$

Follow the leader algorithm

$t = 1, 2, 3, \ldots, T$

	round	1	2	3	4	...	T	total
$i = 1$	🤓	1.0 😀	0.6			...		1.6
$i = 2$	🐵	0.5 😀	0.1			...		0.6
$i = 3$	👺	0.2 😀	0.3 😀	😀		...		0.5
	😀	1.0	0.3			...		1.3

Follow-the-leader (FTL) algorithm:
At each round t, choose an expert i_t from S_t, the set of experts with the top overall performance until the last round $(t-1)$ so far:

$$i_t \in S_t \coloneqq \arg\min_{i \in [N]} \left\{ \sum_{s=1}^{t-1} \ell_{si} \right\}$$

Follow the leader algorithm

$t = 1, 2, 3, \dots, T$

	round	1	2	3	4	...	T	total
$i = 1$	🤓	1.0 😀	0.6	0.8		...		2.4
$i = 2$	🐵	0.5 😀	0.1	0.6	😀	...		1.2
$i = 3$	👺	0.2 😀	0.3 😀	0.9 😀		...		1.4
	😀	1.0	0.3	0.6		...		1.9

Follow-the-leader (FTL) algorithm:
At each round t, choose an expert i_t from S_t, the set of experts with the top overall performance until the last round $(t-1)$ so far:

$$i_t \in S_t := \arg\min_{i \in [N]} \left\{ \sum_{s=1}^{t-1} \ell_{si} \right\}$$

Follow the leader algorithm

$$t = 1, 2, 3, \dots, T$$

	round	1	2	3	4	...	T	total
$i = 1$	🤓	1.0 😀	0.6	0.8	0.1	...	0.2	26.1
$i = 2$	🐵	0.5 😀	0.1	0.6	1.0 😀	...	0.2 😀	20.3
$i = 3$	👺	0.2 😀	0.3 😀	0.9 😀	0.7	...	0.8	30.6
	😀	1.0	0.3	0.6	0.7	...	0.2	???

Follow-the-leader (FTL) algorithm:
At each round t, choose an expert i_t from S_t, the set of experts with the top overall performance until the last round $(t-1)$ so far:

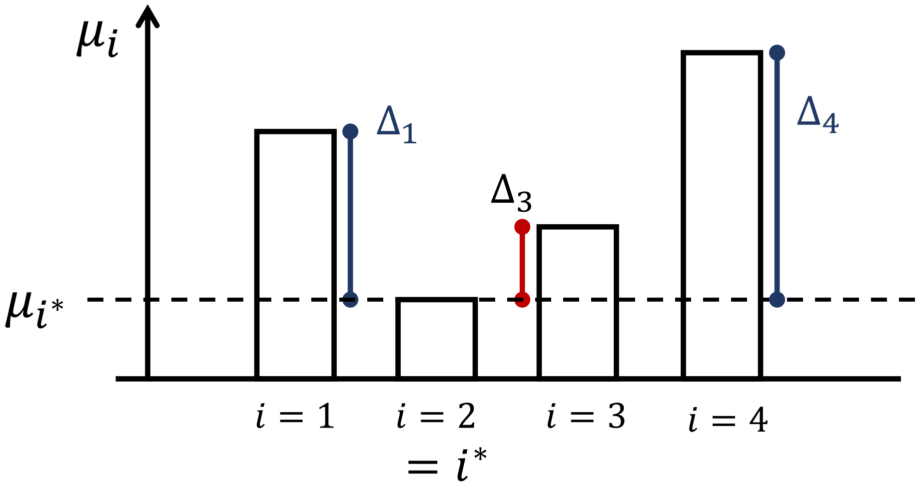
$$i_t \in S_t := \arg\min_{i \in [N]} \left\{ \sum_{s=1}^{t-1} \ell_{si} \right\}$$

Analysis of FTL in stochastic environments

Assumptions (stochastic environment) :
For each $i \in [N]$, there exists a distribution D_i over the interval $[0,1]$ such that ℓ_{ti} follows D_i independently for all $t \in [T]$

expected single-round regret for choosing i

- $\mu_i = \mathrm{E}[\ell_{ti}]$: the expected value of a random variable $\ell_{ti} \sim D_i$
- $i^* \in \arg\min_{i \in [N]} \mu_i$: the optimal expert (in expectation), $\Delta_i := \mu_i - \mu_{i^*}$

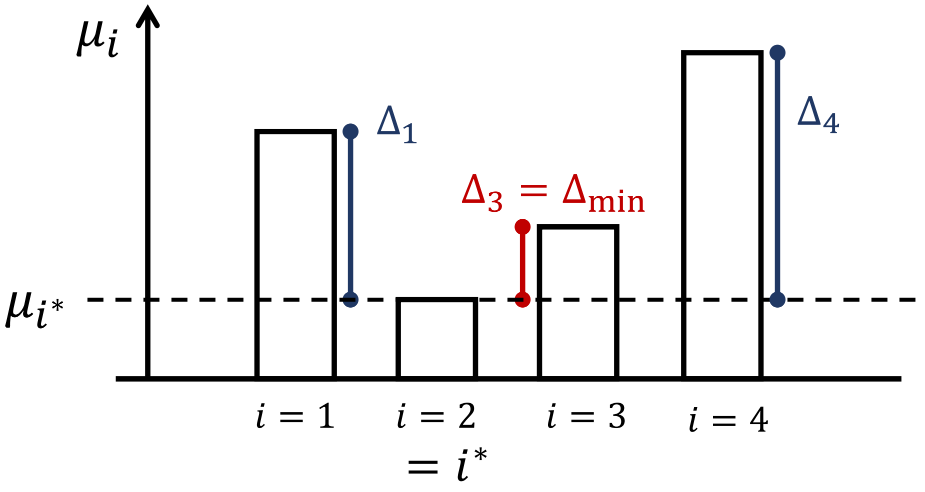


Analysis of FTL in stochastic environments

Assumptions (stochastic environment) :
For each $i \in [N]$, there exists a distribution D_i over the interval $[0,1]$ such that ℓ_{ti} follows D_i independently for all $t \in [T]$

expected single-round regret for choosing i

- $\mu_i = \mathrm{E}[\ell_{ti}]$: the expected value of a random variable $\ell_{ti} \sim D_i$
- $i^* \in \arg\min_{i \in [N]} \mu_i$: the optimal expert (in expectation), $\Delta_i \coloneqq \mu_i - \mu_{i^*}$
- $\Delta_{\min} = \min_{i \in [N] \setminus \{i^*\}} \Delta_i$
- Assume $\Delta_{\min} > 0$ for simplicity



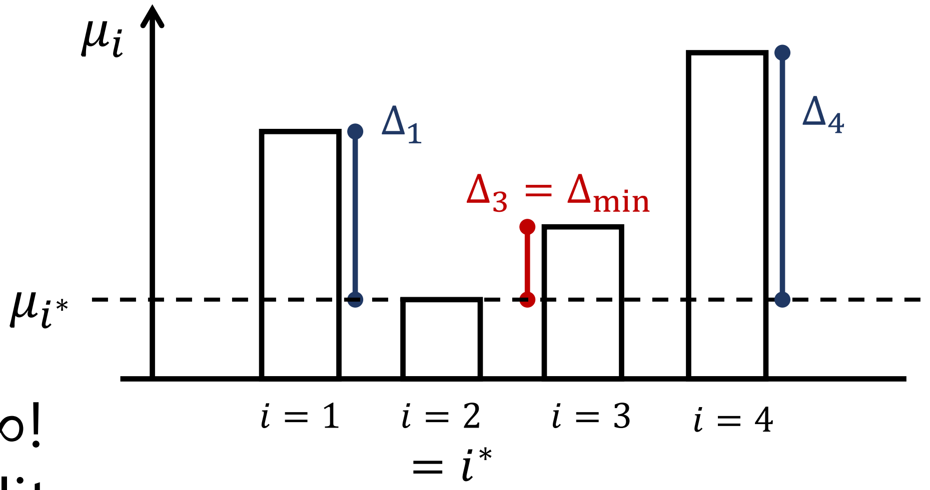
Analysis of FTL in stochastic environments

Assumptions (stochastic environment) :
For each $i \in [N]$, there exists a distribution D_i over the interval $[0,1]$ such that ℓ_{ti} follows D_i independently for all $t \in [T]$

expected single-round regret for choosing i

- $\mu_i = \mathrm{E}[\ell_{ti}]$: the expected value of a random variable $\ell_{ti} \sim D_i$
- $i^* \in \arg\min_{i \in [N]} \mu_i$: the optimal expert (in expectation), $\Delta_i \coloneqq \mu_i - \mu_{i^*}$
- $\Delta_{\min} = \min_{i \in [N] \setminus \{i^*\}} \Delta_i$
- Assume $\Delta_{\min} > 0$ for simplicity

Theorem : In stochastic environment,
FTL achieves $R_T = O\left(\min\left\{\frac{\log N}{\Delta_{\min}}, \sqrt{T \log N}\right\}\right)$



- Regret is bounded even if T approaches ∞!
- This can be shown via Hoeffding's inequality

FTL for adversarial environments

$t = 1, 2, 3, \ldots, T$

N

round	1	2	3	4	5	6	…	T	total
🤓							…		0
😈							…		0
👺							…		0
😀							…		

If 😀 uses FTL in an adversarial environment:

FTL for adversarial environments

$t = 1, 2, 3, \dots, T$

N

round	1	2	3	4	5	6	…	T	total
🤓	😀						…		0
😈	😀						…		0
👺	😀						…		0
😀							…		

If 😀 uses FTL in an adversarial environment:

FTL for adversarial environments

$$t = 1, 2, 3, \ldots, T$$

N

round	1	2	3	4	5	6	…	T	total
🤓	1 😀						…		1
😈	0.5 😀	😀					…		0.5
👺	1 😀						…		1
😀	1						…		

If 😀 uses FTL in an adversarial environment:

FTL for adversarial environments

$t = 1, 2, 3, \dots, T$

N

round	1	2	3	4	5	6	…	T	total
🤓	1 😀	0	😀				…		1
😈	0.5 😀	1 😀					…		1.5
👺	1 😀	1					…		2
😀	1	1					…		

If 😀 uses FTL in an adversarial environment:

FTL for adversarial environments

$t = 1, 2, 3, \dots, T$

N

round	1	2	3	4	5	6	...	T	total
🤓	1 😀	0	1 😀				...		2
😈	0.5 😀	1 😀	0	😀			...		1.5
👺	1 😀	1	1				...		3
😀	1	1	1				...		

If 😀 uses FTL in an adversarial environment:

FTL for adversarial environments

$t = 1, 2, 3, \dots, T$

N

round	1	2	3	4	5	6	...	T	total
🤓	1 😀	0	1 😀	0	😀		...		2
😈	0.5 😀	1 😀	0	1 😀			...		2.5
👺	1 😀	1	1	1			...		4
😀	1	1	1	1			...		

If 😀 uses FTL in an adversarial environment:

FTL for adversarial environments

$t = 1, 2, 3, \ldots, T$

N

round	1	2	3	4	5	6	...	T	total
🤓	1 😀	0	1 😀	0	1 😀		...		3
😈	0.5 😀	1 😀	0	1 😀	0	😀	...		2.5
👺	1 😀	1	1	1	1		...		5
😀	1	1	1	1	1		...		

If 😀 uses FTL in an adversarial environment:

FTL for adversarial environments

$t = 1, 2, 3, \dots, T$

N

round	1	2	3	4	5	6	…	T	total
🤓	1 😀	0	1 😀	0	1 😀	0	…	1 😀	
😈	0.5 😀	1 😀	0	1 😀	0	1 😀	…	0	
👺	1 😀	1	1	1	1	1	…	1	
😀	1	1	1	1	1	1	…	1	

If 😀 uses FTL in an adversarial environment:

FTL for adversarial environments

$t = 1, 2, 3, \ldots, T$

round	1	2	3	4	5	6	…	T	total
🤓	1 😀	0	1 😀	0	1 😀	0	…	1 😀	$\approx T/2$
😈	0.5 😀	1 😀	0	1 😀	0	1 😀	…	0	$\approx T/2$
👺	1 😀	1	1	1	1	1	…	1	$= T$
😀	1	1	1	1	1	1	…	1	$\approx T$

N

If 😀 uses FTL in an adversarial environment:

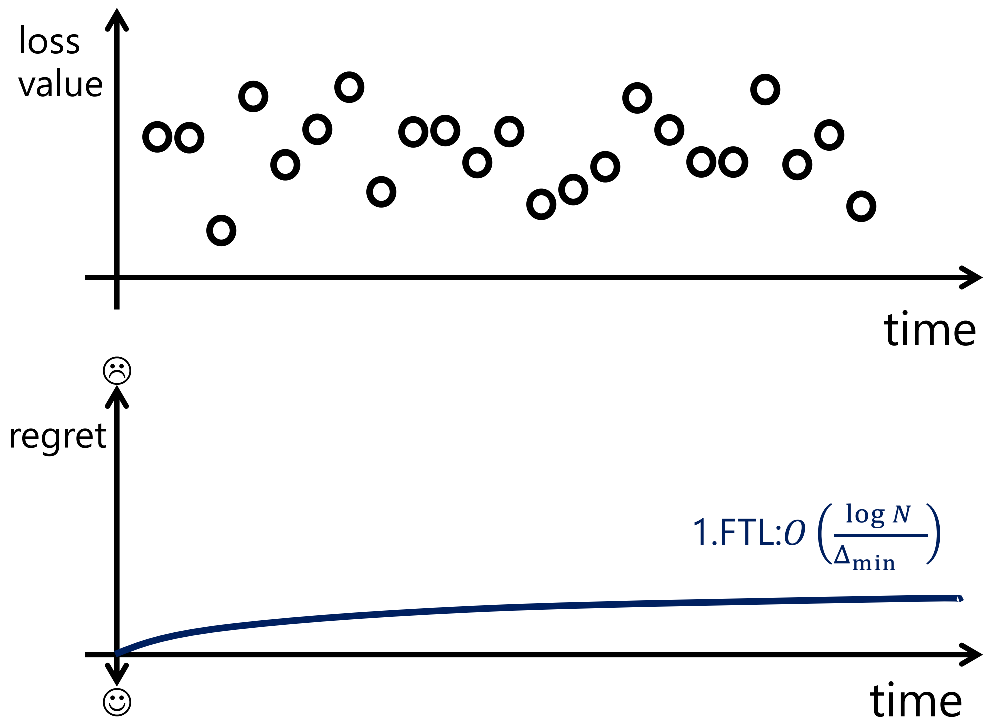
😀's cumulative loss $\approx T$; 😈's cumulative loss of $\approx T/2$

$$R_T = \sum_{t=1}^{T} \ell_{ti_t} - \min_{i^* \in [N]} \sum_{t=1}^{T} \ell_{ti^*} \approx T - \frac{T}{2} = \frac{T}{2} \geq \Omega(T)$$

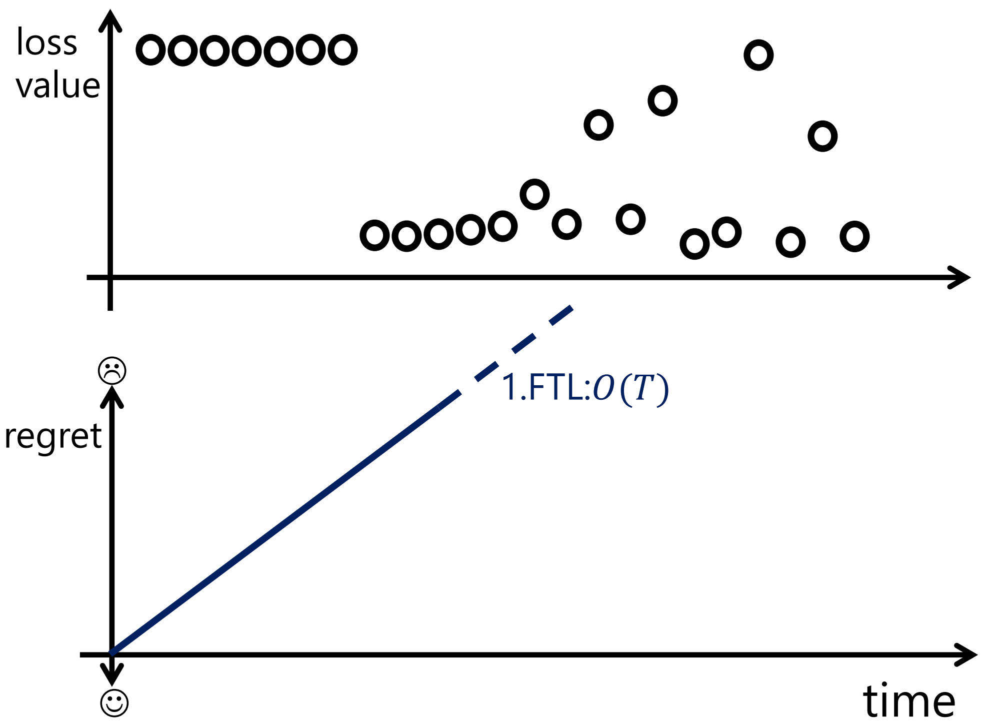
Suffers **linear regret**!

Two environment models and two algorithms

1. **Stochastic environment model**:
   Losses and rewards are i.i.d.



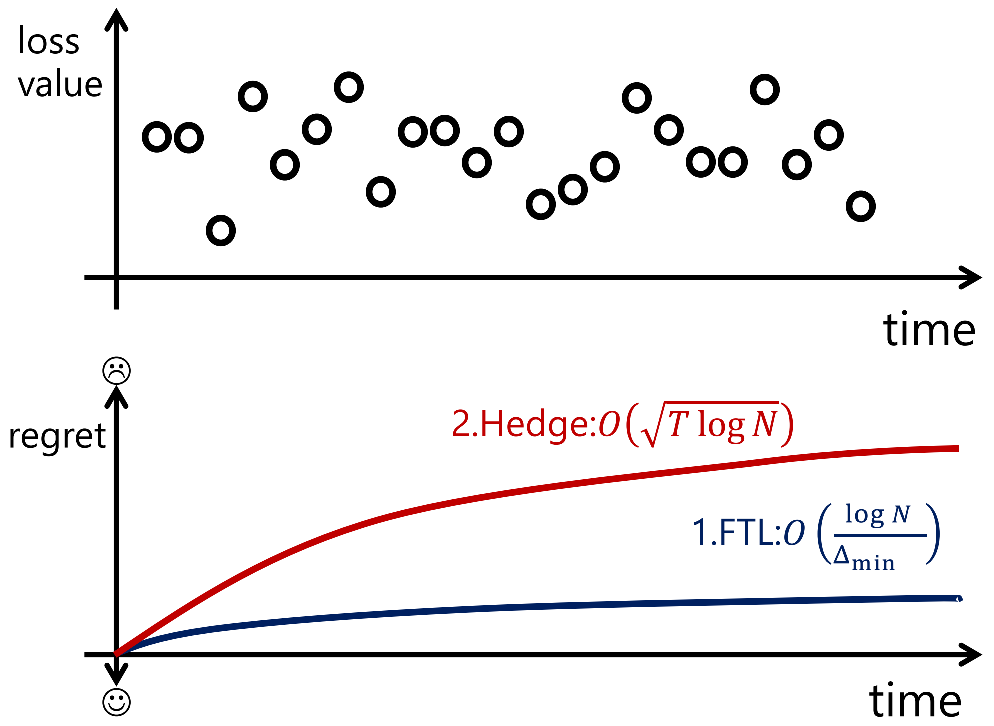
2. **Adversarial environment model**:
   Losses and rewards **change arbitrarily**



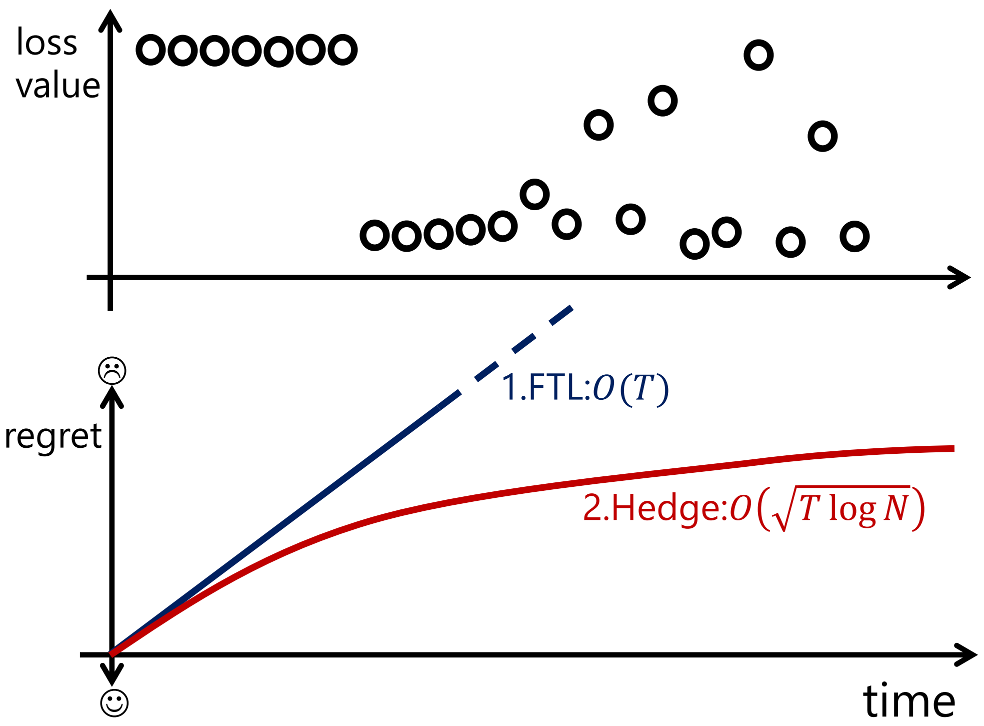
FTL works great in stochastic environments, but it can be terrible in adversarial environment

Two environment models and two algorithms

1. **Stochastic environment model**:
Losses and rewards are i.i.d.



2. **Adversarial environment model**:
Losses and rewards **change arbitrarily**



FTL works great in stochastic environments, but it can be terrible in adversarial environment

Hedge Algorithm

[LW94], [AHK12]

$t = 1, 2, 3, \dots, T$

	round	1	2	3	4	5	6	...	T	total
$i = 1$	🤓	1 😀	0	1	0	1	0	...	1 😀	$\approx T/2$
$i = 2$	😈	0.5 😀	1	0	1	0	1	...	0 😀	$\approx T/2$
$i = 3$	👺	1 😀	1	1	1	1	1	...	1	$= T$
	😀									$\approx T$

Hedge Algorithm:

- Set learning rate $\eta > 0$, initialize the weight (reliability) by $w_{1i} = 1$ for each $i \in [N]$
- In each round t, choose expert with a probability proportional to w_{ti}

  After observing the loss, each weight is updated with $w_{t+1,i} = w_{ti} \exp(-\eta \ell_{ti})$

Hedge Algorithm

[LW94], [AHK12]

$$t = 1, 2, 3, \dots, T$$

	round	1	2	3	4	5	6	...	T	total
$i = 1$	🤓	1 😀	0	1	0	1	0	...	1 😀	$\approx T/2$
$i = 2$	😈	0.5 😀	1	0	1	0	1	...	0 😀	$\approx T/2$
$i = 3$	👺	1 😀	1	1	1	1	1	...	1	$= T$
	😀									$\approx T$

Hedge Algorithm:

- Set learning rate $\eta > 0$, initialize the weight (reliability) by $w_{1i} = 1$ for each $i \in [N]$
- In each round t, choose expert with a probability proportional to w_{ti}
  After observing the loss, each weight is updated with $w_{t+1,i} = w_{ti} \exp(-\eta \ell_{ti})$

If loss ℓ_{ti} is large, the reliability of i is decreased.

Hedge Algorithm

[LW94], [AHK12]

$t = 1, 2, 3, \dots, T$

	round	1	2	3	4	5	6	...	T	total
$i = 1$	🤓	1 😀	0	1	0	1	0	...	1 😀	$\approx T/2$
$i = 2$	😈	0.5 😀	1	0	1	0	1	...	0 😀	$\approx T/2$
$i = 3$	👺	1 😀	1	1	1	1	1	...	1	$= T$
	😀									$\approx T$

Hedge Algorithm:

- Set learning rate $\eta > 0$, initialize the weight (reliability) by $w_{1i} = 1$ for each $i \in [N]$
- In each round t, choose expert with a probability proportional to w_{ti}
  After observing the loss, each weight is updated with $w_{t+1,i} = w_{ti} \exp(-\eta \ell_{ti})$

As a result, the weight is determined as $w_{ti} = \exp\left(-\eta\left(\ell_{1i} + \ell_{2i} + \cdots + \ell_{t-1,i}\right)\right)$

Expert i is chosen with probability $p_{ti} = \frac{w_{ti}}{\sum_{j=1}^{N} w_{tj}}$

Hedge Algorithm

W94], [AHK12]

$\approx$
1/2
1/2
0

$t = 1, 2, 3, \dots, T$

	round	1		2	3	4	5	6	…			total
$i=1$	🤓	1	😀		1	0	1	0	…	1	😀	$\approx T/2$
$i=2$	😈	0.5	😀		0	1	0	1	…	0	😀	$\approx T/2$
$i=3$	👺	1	😀		1	1	1	1	…	1		$= T$
	😀											$\approx T$

1/3
1/3
1/3

Hedge Algorithm:

- Set learning rate $\eta > 0$, initialize the weight (reliability) by $w_{1i} = 1$ for each $i \in [N]$
- In each round t, choose expert with a probability proportional to w_{ti}
  After observing the loss, each weight is updated with $w_{t+1,i} = w_{ti} \exp(-\eta \ell_{ti})$

As a result, the weight is determined as $w_{ti} = \exp\left(-\eta\left(\ell_{1i} + \ell_{2i} + \cdots + \ell_{t-1,i}\right)\right)$

Expert i is chosen with probability $p_{ti} = \frac{w_{ti}}{\sum_{j=1}^{N} w_{tj}}$

Analysis of Hedge Algorithm

- Hedge Algorithm:

$$w_{ti} = \exp\left(-\eta\left(\ell_{1i} + \ell_{2i} + \cdots + \ell_{t-1,i}\right)\right),$$ Expert i is chosen with probability $p_{ti} = \frac{w_{ti}}{\sum_{j=1}^{N} w_{tj}}$

Theorem :

Suppose that $\eta \in [0,1]$. For any loss sequence $(\ell_t)_{t=1}^{T} \in ([0,1]^N)^T$,

Hedge achieves $R_T \leq \frac{1}{\eta}\log N + \frac{\eta}{4}T$

Corollary:

When setting $\eta = \min\left\{1,\ 2\sqrt{\frac{\log N}{T}}\right\}$, Hedge achieves $R_T \leq \sqrt{T \log N}$

Outline of the talk

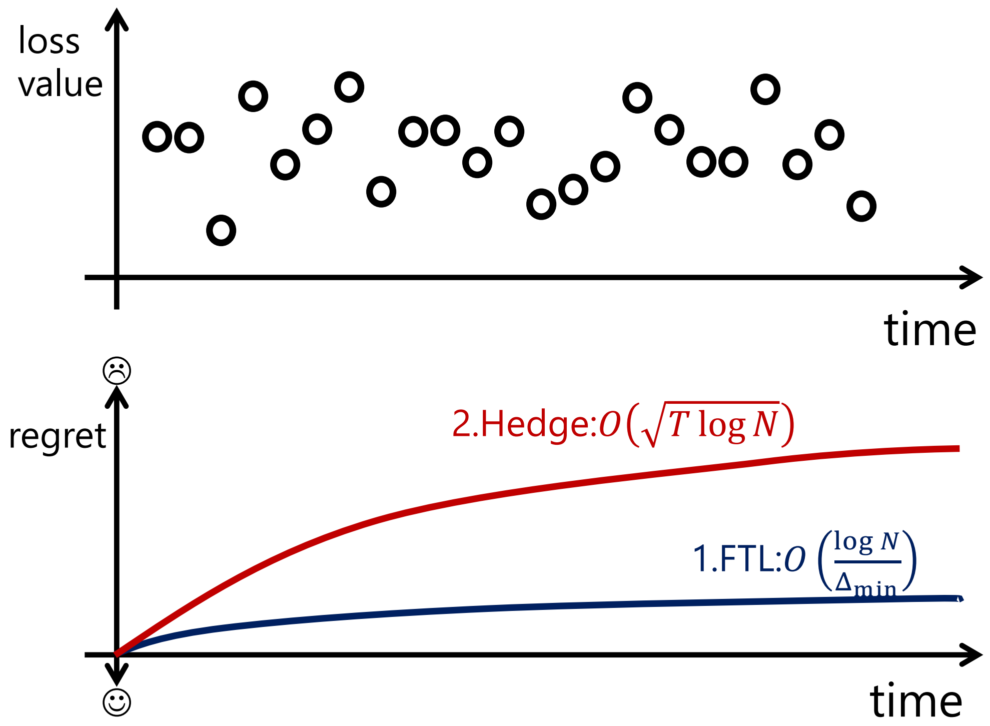
- Problem setup
  - Prediction with expert advice and multi-armed bandit
  - Two models for environments

- Basic results of regret analysis
  - Algorithms and regret analysis for the expert problem
  - Comparison of regrets in stochastic and adversarial environments

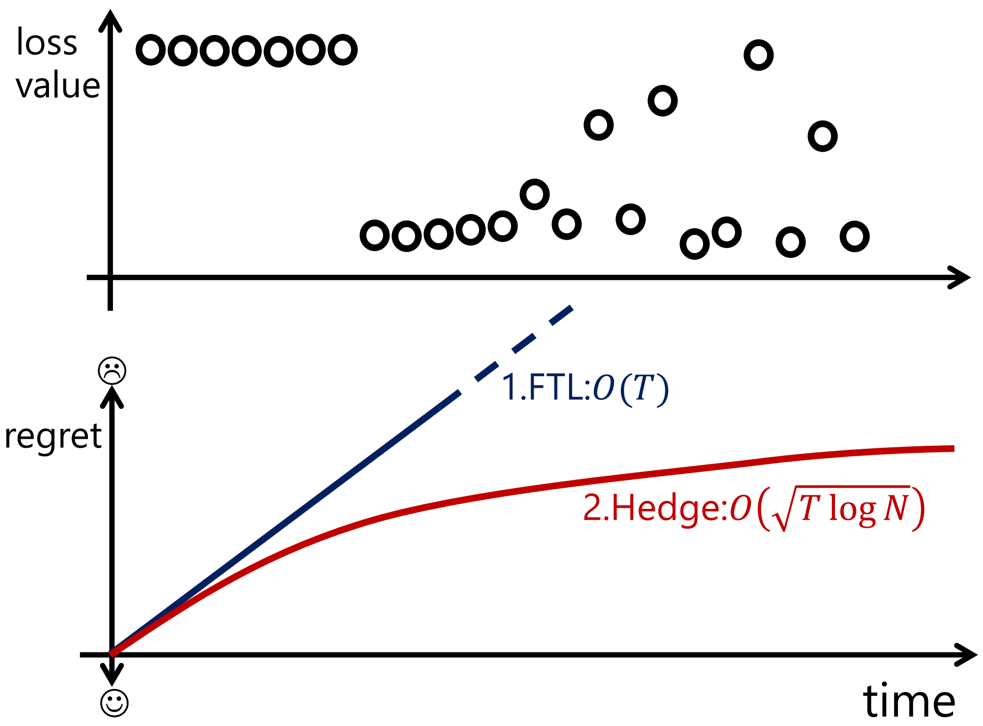
- Best-of-both-worlds algorithms and analysis
  - Hedge with adaptive learning rate
  - Analysis between stochastic and adversarial (stochastic environment with adversarial corruption)
  - Other recent developments

Two loss models and two algorithms

1. **Stochastic environment model**:
   Losses and rewards are **i.i.d.**



2. **Adversarial environment model**:
   Losses and rewards **change arbitrarily**



- It is important to choose the right algorithm for the environment

Expert problem: Summary so far

(almost) tight upper bound

Table 1: Regret bounds for expert problems

	stochastic environment	hostile environment
FTL	$O\left(\frac{\log N}{\Delta_{\min}}\right)$	$O(T)$
Hedge [LW94], [AHK12]	$O(\sqrt{T \log N})$	$O(\sqrt{T \log N})$
regret lower bound	$\Omega\left(\frac{\log N}{\Delta_{\min}}\right)$	$\Omega(\sqrt{T \log N})$

- FTL/Hedge is optimal in each of stochastic and adversarial environments.
- To obtain the best results, it is necessary to choose an algorithm that matches the environment.

(almost) tight upper bound

Table 1: Regret bounds for expert problems

	stochastic environment	hostile environment
FTL	$O\left(\frac{\log N}{\Delta_{\min}}\right)$	$O(T)$
Hedge [LW94], [AHK12]	$O(\sqrt{T \log N})$	$O(\sqrt{T \log N})$
regret lower bound	$\Omega\left(\frac{\log N}{\Delta_{\min}}\right)$	$\Omega(\sqrt{T \log N})$

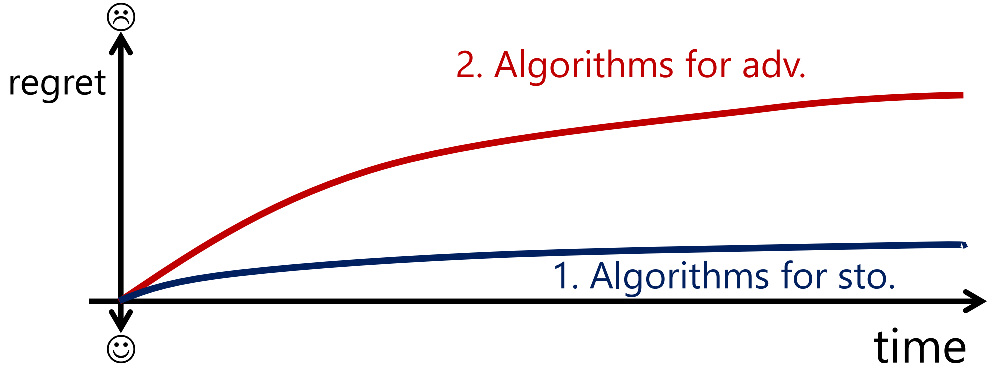
Table 2: Regret bounds for multi-armed bandit problem

	Stochastic setting	adversarial setting
UCB etc. [ACBF02]	$O\left(\sum_{i \neq i^*} \frac{\log T}{\Delta_i}\right)$	$O(T)$
Exp3 [ACBFS02]	$O(\sqrt{TN \log N})$	$O(\sqrt{TN \log N})$
regret lower bound	$\Omega\left(\sum_{i \neq i^*} \frac{\log T}{\Delta_i}\right)$	$\Omega(\sqrt{TN})$

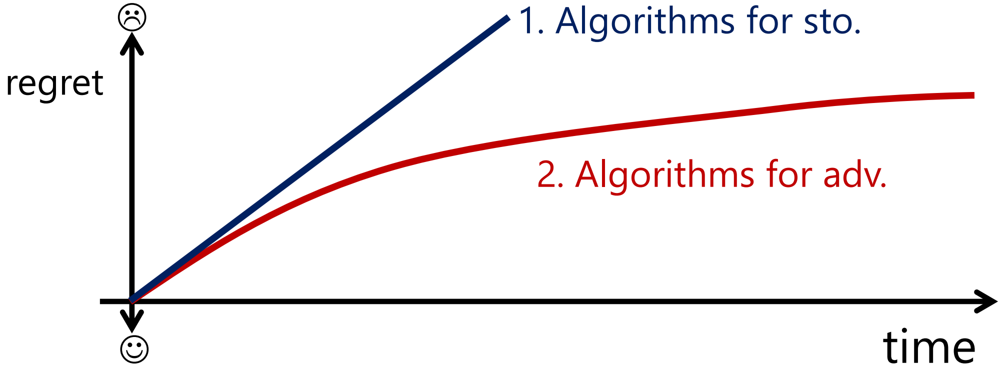
- Similar results have been provided for the multi-armed bandit problem.

After all, which one should we use?

1.**Stochastic environment model** :

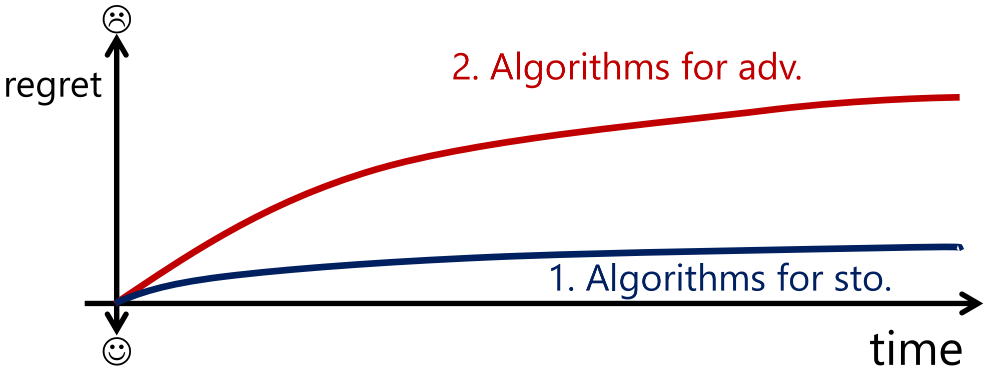


2. **Adversarial environment model** :

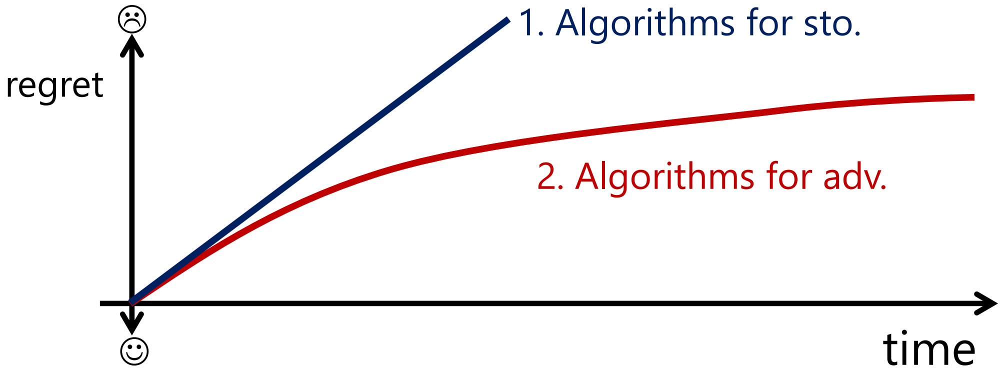


After all, which one should we use?

1.**Stochastic environment model** :



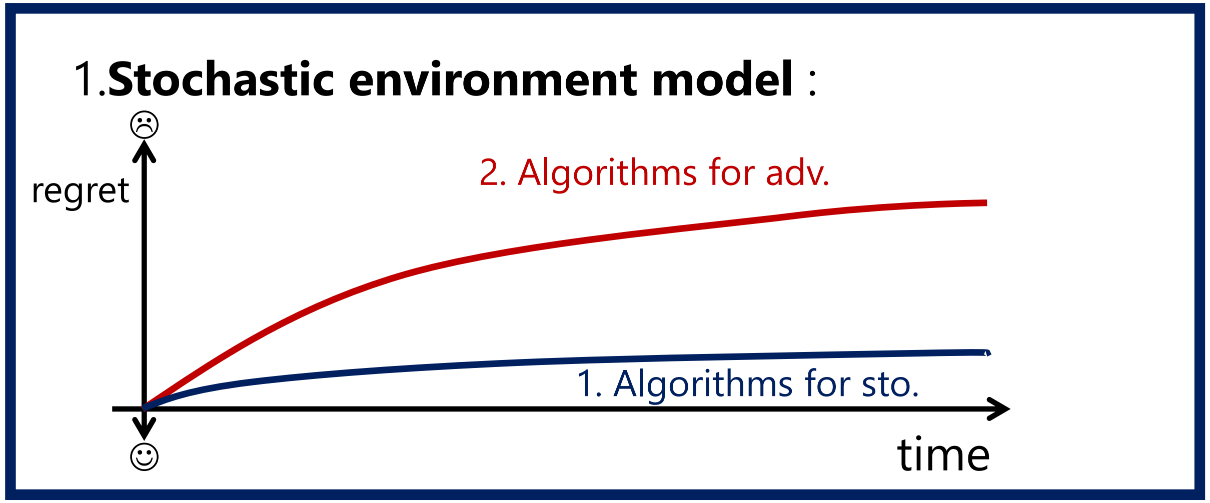
2. **Adversarial environment model** :

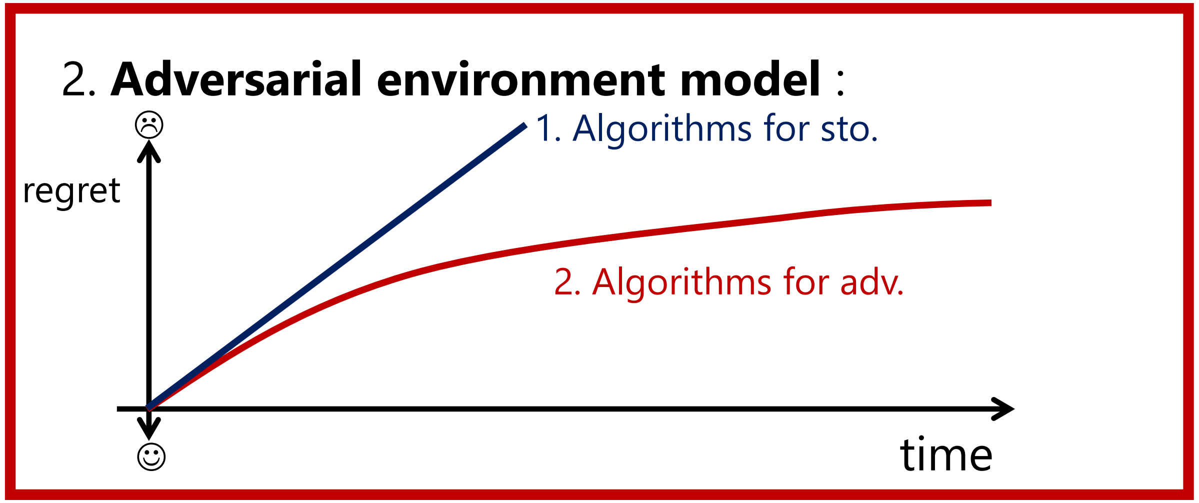


- Arguments supporting stochastic environments
  - The real world does not change that often. It can be approximated sufficiently well by a stochastic model.
  - Considering the worst case in an adversarial model is overly pessimistic and conservative. In reality, situations that correspond to the worst case are rare.

- Arguments supporting adversarial environments
  - Adversarial models include stochastic models and are more general-purpose.
  - Guaranteed worst-case performance is useful because it means stability for any input sequence.
  - In reality, losses and rewards are rarely i.i.d.

After all, which one should we use?





- Arguments supporting stochastic environments
  - The real world does not change that often. It can be approximated sufficiently well by a stochastic model.
  - Considering the worst case in an adversarial model is overly pessimistic and conservative. In reality, situations that correspond to the worst case are rare.

View of statistical learning theory and information theory (?)

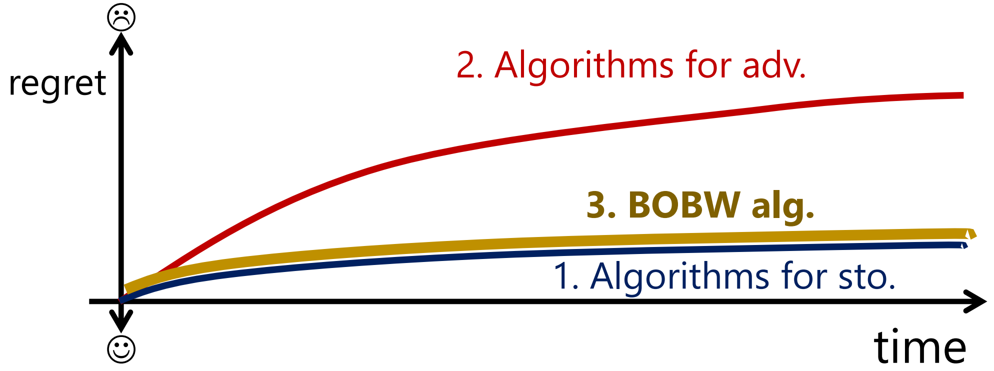
- Arguments supporting adversarial environments
  - Adversarial models include stochastic models and are more general-purpose.
  - Guaranteed worst-case performance is useful because it means stability for any input sequence.
  - In reality, losses and rewards are rarely i.i.d.

View of theoretical computer science, optimization theory, etc. (?)

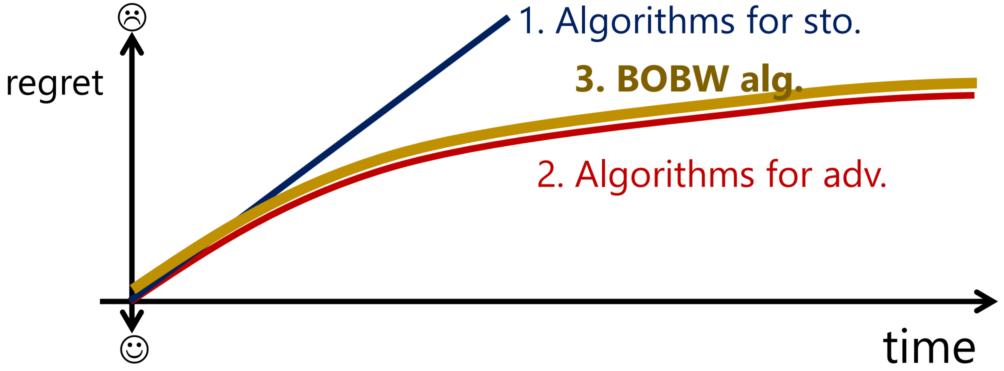
- The basic concepts and research communities seem different. Depending on our standpoint, both can be criticized/justified.

After all, which one should we use?

1.**Stochastic environment model** :



2. **Adversarial environment model** :



- Arguments supporting stochastic environments
  - The real world does not change that often. It can be approximated sufficiently well by a stochastic model.
  - Considering the worst case in an adversarial model is overly pessimistic and conservative. In reality, situations that correspond to the worst case are rare.

View of statistical learning theory and information theory (?)

- Arguments supporting adversarial environments
  - Adversarial models include stochastic models and are more general-purpose.
  - Guaranteed worst-case performance is useful because it means stability for any input sequence.
  - In reality, losses and rewards are rarely i.i.d.

View of theoretical computer science, optimization theory, etc. (?)

- The basic concepts and research communities seem different. Depending on our standpoint, both can be criticized/justified.
- From a practical viewpoint: In any case, we want better performance ⇒ **Best-of-both-worlds algorithm**

Outline of the talk

- Problem setup
  - Prediction with expert advice and multi-armed bandit
  - Two models for environments

- Basic results of regret analysis
  - Algorithms and regret analysis for the expert problem
  - Comparison of regrets in stochastic and adversarial environments

- **Best-of-both-worlds algorithms and analysis**
  - Hedge with adaptive learning rate
  - Analysis between stochastic and adversarial (stochastic environment with adversarial corruption)
  - Other recent developments

Best-of-both-worlds (BOBW) algorithm

(almost) tight upper bound

Table 1: Regret bounds for expert problems

	stochastic environment	hostile environment
FTL	$O\left(\frac{\log N}{\Delta_{\min}}\right)$	$O(T)$
Hedge [LW94], [AHK12]	$O(\sqrt{T \log N})$	$O(\sqrt{T \log N})$
BOBW algorithm	$O\left(\frac{\log N}{\Delta_{\min}}\right)$	$O(\sqrt{T \log N})$
regret lower bound	$\Omega\left(\frac{\log N}{\Delta_{\min}}\right)$	$\Omega(\sqrt{T \log N})$

- Goal: Achieve optimal performance in both stochastic/adversarial environments
- Strategy: Introduce a framework that encompasses both of FTL and Hedge and (adaptively) interpolate them

Outline of the talk

- Problem setup
  - Prediction with expert advice and multi-armed bandit
  - Two models for environments

- Basic results of regret analysis
  - Algorithms and regret analysis for the expert problem
  - Comparison of regrets in stochastic and adversarial environments

- **Best-of-both-worlds algorithms and analysis**
  - **Hedge with adaptive learning rate**
  - Analysis between stochastic and adversarial (stochastic environment with adversarial corruption)
  - Other recent developments

Hedge: Interpretation with entropy regularization

- $\Delta^N = \{p \in [0,1]^N : \|p\|_1 = 1\}$: Probability simplex
- $H(p) = -\sum_{i=1}^N p_i \log p_i$: Shannon entropy
- $\ell_t = \begin{bmatrix} \ell_{t1} \\ \ell_{t2} \\ \vdots \\ \ell_{tN} \end{bmatrix}, \quad p_t = \begin{bmatrix} p_{t1} \\ p_{t2} \\ \vdots \\ p_{tN} \end{bmatrix}$ (p_{ti}: probability of choosing i at round t)

The Hedge algorithm is given by:

$$p_t \in \arg\min_{p \in \Delta^N} \left\{ \left\langle \sum_{s=1}^{t-1} \ell_s, p \right\rangle - \frac{1}{\eta} H(p) \right\}$$

Hedge: Interpretation with entropy regularization

- $\Delta^N = \{p \in [0,1]^N\colon\ \|p\|_1 = 1\}$: Probability simplex
- $H(p) = -\sum_{i=1}^N p_i \log p_i$: Shannon entropy
- $\ell_t = \begin{bmatrix} \ell_{t1} \\ \ell_{t2} \\ \vdots \\ \ell_{tN} \end{bmatrix}, \quad p_t = \begin{bmatrix} p_{t1} \\ p_{t2} \\ \vdots \\ p_{tN} \end{bmatrix}$ (p_{ti}: probability of choosing i at round t)

The Hedge algorithm is given by:

$$p_t \in \arg\min_{p \in \Delta^N} \left\{ \left\langle \sum_{s=1}^{t-1} \ell_s , p \right\rangle - \frac{1}{\eta} H(p) \right\}$$

In fact, from the first-order optimality condition,

$$\sum_{s=1}^{t-1} \ell_s - \frac{1}{\eta} \nabla H(p_t) + \lambda_t \mathbf{1} = 0 \implies \log p_{ti} = -\eta \sum_{s=1}^{t-1} \ell_{si} + \eta \lambda_t - 1 \implies p_{ti} \propto \exp\left(-\eta \sum_{s=1}^{t-1} \ell_{st} \right)$$

Comparison of Hedge and FTL

- $\Delta^N = \{p \in [0,1]^N \colon \|p\|_1 = 1\}$: Probability simplex
- $H(p) = -\sum_{i=1}^N p_i \log p_i$: Shannon entropy
- $\ell_t = \begin{bmatrix} \ell_{t1} \\ \ell_{t2} \\ \vdots \\ \ell_{tN} \end{bmatrix}, \quad p_t = \begin{bmatrix} p_{t1} \\ p_{t2} \\ \vdots \\ p_{tN} \end{bmatrix}$ (p_{ti}: probability of choosing i at round t)

Hedge algorithm:

$$p_t \in \arg\min_{p \in \Delta^N} \left\{ \left\langle \sum_{s=1}^{t-1} \ell_s , p \right\rangle \underline{- \frac{1}{\eta} H(p)} \right\}$$

FTL algorithm:

$$p_t \in \arg\min_{p \in \Delta^N} \left\{ \left\langle \sum_{s=1}^{t-1} \ell_s , p \right\rangle \right\}$$

- Hedge can be interpreted as FTL with regularization that increases entropy
- If η is large enough, the behavior is close to FTL (cf. standard Hedge employs $\eta \approx \sqrt{\frac{\log N}{T}}$)

$\Rightarrow$ By adjusting η adequately (optimizing η itself) depending on observed data, we can interpolates between Hedge and FTL well

Follow the regularized leader (FTRL)

Eg. [Chapter 28, Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*, 2020.]

Hedge algorithm is given by:

$$p_t \in \arg\min_{p \in \Delta^N} \left\{ \left\langle \sum_{s=1}^{t-1} \ell_s , p \right\rangle - \frac{1}{\eta} H(p) \right\} \qquad (H(p) = -\textstyle\sum_{i=1}^{N} p_i \log p_i\text{: Shannon entropy})$$

[LS20] Generalize region Δ^N to any convex set and $H(p)$ to any regularizer

FTRL algorithm: Define x_t using convex function $\psi: X \to \mathbb{R}$ as follows :

$$x_t \in \arg\min_{x \in X} \left\{ \left\langle \sum_{s=1}^{t-1} g_s , x \right\rangle + \frac{1}{\eta} \psi(x) \right\}$$

Example :

- $X = \Delta^d$, $g_t = \ell_t, \psi(x) = -H(x) = \sum_{i=1}^{d} x_i \log x_i$ Hedge
- $f_t: X \to \mathbb{R}$ (convex function), $g_t = \nabla f_t(x_t)$, $\psi(x) = \|x\|_2^2$ (a variant of) gradient descent

Hedge analysis

Hedge algorithm: $p_t \in \arg\min_{p \in \Delta^N} \left\{ \left\langle \sum_{s=1}^{t-1} \ell_s, p \right\rangle - \frac{1}{\eta} H(p) \right\}$

e.g. [Chapter 28, LS20]

Standard analysis method for FTRL decompose regret into the sum of stability and penalty terms

$$R_T \le \underbrace{\sum_{t=1}^{T} \frac{1}{\eta} D_{\mathrm{KL}}(p_t || p_{t+1})}_{\text{Stability term}} + \underbrace{\frac{1}{\eta} H(p_1)}_{\text{Penalty term}}$$

Stability term

- The magnitude of the variation of output distribution p_t

Penalty term

- Corresponding to the bias due to regularization

[LS20] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*, 2020.

Hedge analysis

Hedge algorithm: $p_t \in \arg\min_{p \in \Delta^N} \left\{ \left\langle \sum_{s=1}^{t-1} \ell_s, p \right\rangle - \frac{1}{\eta} H(p) \right\}$

e.g. [Chapter 28, LS20]

Standard analysis method for FTRL decompose regret into the sum of stability and penalty terms

$$R_T \le \sum_{t=1}^{T} \frac{1}{\eta} D_{\mathrm{KL}}(p_t \| p_{t+1}) + \frac{1}{\eta} H(p_1) \le \sum_{t=1}^{T} \eta z_t + \frac{1}{\eta} \log N \le \frac{\eta T}{4} + \frac{\log N}{\eta}$$

Stability term

- The magnitude of the variation of output distribution p_t
- The stronger the regularization (the smaller η), the smaller the value
- z_t: The variance of a random variable that takes value ℓ_{ti} with probability p_{ti}

$$\left(z_t = \sum_{i=1}^{N} p_{ti} \left(\ell_{ti} - \bar{\ell}_t \right)^2 \le \frac{1}{4}, \bar{\ell}_t = \sum_{i=1}^{N} p_{ti}, \right)$$

Penalty term

- Corresponding to the bias due to regularization
- The weaker the regularization (the bigger η), the smaller the value

[LS20] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*, 2020.

Hedge analysis

Hedge algorithm: $p_t \in \arg\min_{p \in \Delta^N} \left\{ \left\langle \sum_{s=1}^{t-1} \ell_s, p \right\rangle - \frac{1}{\eta} H(p) \right\}$

e.g. [Chapter 28, LS20]

Standard analysis method for FTRL decompose regret into the sum of stability and penalty terms

$$R_T \leq \sum_{t=1}^{T} \frac{1}{\eta} D_{\mathrm{KL}}(p_t || p_{t+1}) + \frac{1}{\eta} H(p_1) \leq \sum_{t=1}^{T} \eta z_t + \frac{1}{\eta} \log N \leq \frac{\eta T}{4} + \frac{\log N}{\eta}$$

Stability term

- The magnitude of the variation of output distribution p_t
- The stronger the regularization (the smaller η), the smaller the value
- z_t: The variance of a random variable that takes value ℓ_{ti} with probability p_{ti}

$$\left(z_t = \sum_{i=1}^{N} p_{ti} \left(\ell_{ti} - \bar{\ell}_t \right)^2 \leq \frac{1}{4}, \bar{\ell}_t = \sum_{i=1}^{N} p_{ti}, \right)$$

Penalty term

- Corresponding to the bias due to regularization
- The weaker the regularization (the bigger η), the smaller the value

Minimization of the right-hand side:

$$\eta = 2\sqrt{\frac{\log N}{T}}$$

$$R_T \leq \frac{\eta T}{4} + \frac{\log N}{\eta} = \sqrt{T \log N}$$

The setting of $\eta = \sqrt{\frac{8 \log N}{T}}$ can also be interpreted as balancing the stability and penalty terms: $\left(\frac{\eta T}{8} = \frac{\log N}{\eta} \right)$

[LS20] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*, 2020.

Hedge with adaptive learning rate

Hedge with adaptive learning rate: $p_t \in \arg\min_{p \in \Delta^N} \left\{ \left\langle \sum_{s=1}^{t-1} \ell_s , p \right\rangle - \frac{1}{\eta_t} H(p) \right\}$

- Adaptively adjust the regularization strength (learning rate) parameter η over rounds.
- The strength of regularization is varied monotonically: $\eta_1 \geq \eta_2 \geq \eta_3 \geq \cdots > 0$

Hedge with adaptive learning rate

Hedge with adaptive learning rate: $p_t \in \arg\min_{p \in \Delta^N} \left\{ \left\langle \sum_{s=1}^{t-1} \ell_s , p \right\rangle - \frac{1}{\eta_t} H(p) \right\}$

- Adaptively adjust the regularization strength (learning rate) parameter η over rounds.
- The strength of regularization is varied monotonically: $\eta_1 \geq \eta_2 \geq \eta_3 \geq \cdots > 0$
- Applying the standard analysis method of FTRL: eg [Chapter 28, LS20]

$$R_T \leq \sum_{t=1}^{T} \left(\eta_t z_t + \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right) H(p_{t+1}) \right) + \frac{1}{\eta_1} H(p_1) \leq \sum_{t=1}^{T} \eta_t z_t + \frac{1}{\eta_{T+1}} \log N$$

[LS20] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*, 2020.

Hedge with adaptive learning rate

Hedge with adaptive learning rate: $p_t \in \arg\min_{p \in \Delta^N} \left\{ \left\langle \sum_{s=1}^{t-1} \ell_s , p \right\rangle - \frac{1}{\eta_t} H(p) \right\}$

- Adaptively adjust the regularization strength (learning rate) parameter η over rounds.
- The strength of regularization is varied monotonically: $\eta_1 \geq \eta_2 \geq \eta_3 \geq \cdots > 0$
- Applying the standard analysis method of FTRL: eg [Chapter 28, LS20]

$$R_T \leq \sum_{t=1}^{T} \left(\eta_t z_t + \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right) H(p_{t+1}) \right) + \frac{1}{\eta_1} H(p_1) \leq \sum_{t=1}^{T} \eta_t z_t + \frac{1}{\eta_{T+1}} \log N$$

- Adjusting η_t using the information of z_t: $\eta_t = \sqrt{\frac{\log N}{1 + \sum_{s=1}^{t-1} z_s}}$ [CBMS07]

$$\sum_{t=1}^{T} \eta_t z_t + \frac{1}{\eta_{T+1}} \log N = \sqrt{\log N} \sum_{t=1}^{T} \frac{z_t}{\sqrt{1 + \sum_{s=1}^{t-1} z_s}} + \sqrt{\log N \cdot \left(1 + \sum_{t=1}^{T} z_t\right)} \leq 2 \sqrt{\log N \cdot \left(1 + \sum_{t=1}^{T} z_t\right)}$$

- Similar idea to optimization method AdaGrad.

[LS20] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*, 2020.

[CBMS07] Nicolo Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66:321–352, 2007.

Regret upper bound for the CBMS algorithm

The CBMS algorithm: $\eta_t = \sqrt{\frac{\log N}{1+\sum_{s=1}^{t-1} z_s}}$, $p_t \in \arg\min_{p \in \Delta^N} \left\{ \left\langle \sum_{s=1}^{t-1} \ell_s, p \right\rangle - \frac{1}{\eta_t} H(p) \right\}$

$$\left(z_t = \sum_{i=1}^{N} p_{ti} \left(\ell_{ti} - \bar{\ell}_t \right)^2 \leq \frac{1}{4}, \bar{\ell}_t = \sum_{i=1}^{N} p_{ti}, \right)$$

Theorem :

The CBMS algorithm achieves $R_T = O\left(\sqrt{\log N \cdot \left(1 + \sum_{t=1}^{T} z_t\right)} \right)$

Corollary : The CBMS algorithm has the following regret upper bound

- In adversarial environments, $R_T = O\left(\sqrt{\log N \cdot T} \right)$
- In stochastic environments, $R_T = O\left(\frac{\log N}{\Delta_{\min}} \right)$

Regret upper bound for the CBMS algorithm

The CBMS algorithm: $\eta_t = \sqrt{\frac{\log N}{1+\sum_{s=1}^{t-1} z_s}}$, $p_t \in \arg\min_{p \in \Delta^N} \left\{ \langle \sum_{s=1}^{t-1} \ell_s , p \rangle - \frac{1}{\eta_t} H(p) \right\}$

$$\left(z_t = \sum_{i=1}^{N} p_{ti} \left(\ell_{ti} - \bar{\ell}_t \right)^2 \leq \frac{1}{4} , \bar{\ell}_t = \sum_{i=1}^{N} p_{ti} , \right)$$

Theorem :

The CBMS algorithm achieves $R_T = O\left(\sqrt{\log N \cdot (1 + \sum_{t=1}^{T} z_t)} \right)$

Corollary : The CBMS algorithm has the following regret upper bound

- In adversarial environments, $R_T = O(\sqrt{\log N \cdot T}) \leftarrow$ Clear from $z_t \leq 1/4$
- In stochastic environments, $R_T = O\left(\frac{\log N}{\Delta_{\min}} \right) \leftarrow$ Nontrivial. Proof on next page

Regret upper bound for the CBMS algorithm

Theorem : CBMS achieves $R_T = O\left(\sqrt{\log N \cdot (1 + \sum_{t=1}^{T} z_t)}\right)$

Corollary : CBMS achieves $R_T = O\left(\frac{\log N}{\Delta_{\min}}\right)$ in stochastic environments

(Proof of Corollary)

- In a stochastic environment, $1 + \sum_{t=1}^{T} z_t = O\left(\frac{1}{\Delta_{\min}} R_T\right)$ holds
  - $z_t := \sum_{i=1}^{N} p_{ti}\left(\ell_{ti} - \bar{\ell}_t\right)^2 \leq \sum_{i=1}^{N} p_{ti}(\ell_{ti} - \ell_{ti^*})^2 = \sum_{i \neq i^*} p_{ti}(\ell_{ti} - \ell_{ti^*})^2 \leq 1 - p_{ti^*}$
  - In a stochastic environment, every time any suboptimal expert (other than i^*) is chosen, regret of at least $\Delta_{\min}$ is suffered in expectation:
    $R_T = \sum_{t=1}^{T} \sum_{i=1}^{N} \Delta_i p_{ti} \geq \sum_{t=1}^{T} \sum_{i \neq i^*} \Delta_{\min} p_{ti} = \Delta_{\min} \sum_{t=1}^{T} (1 - p_{ti^*})$
  - Combining the above two points, we obtain $R_T \geq \Delta_{\min} \sum_{t=1}^{T} z_t$
- Substituting this into the result of the theorem, we obtain $R_T = O\left(\sqrt{\frac{\log N}{\Delta_{\min}} R_T}\right)$
- Square both sides: $R_T^2 = O\left(\frac{\log N}{\Delta_{\min}} R_T\right)$. Divide both sides by R_T: $R_T = O\left(\frac{\log N}{\Delta_{\min}}\right)$

Regret upper bound for CBMS

Theorem : CBMS achieves $R_T = O\left(\sqrt{\log N \cdot (1 + \sum_{t=1}^{T} z_t)}\right)$

Corollary : CBMS achieves $R_T = O\left(\frac{\log N}{\Delta_{\min}}\right)$ in stochastic environments

(Proof of Corollary)

- In a stochastic environment, $1 + \sum_{t=1}^{T} z_t = O\left(\frac{1}{\Delta_{\min}} R_T\right)$ holds.
  - $z_t := \sum_{i=1}^{N} p_{ti}\left(\ell_{ti} - \bar{\ell}_t\right)^2 \le \sum_{i=1}^{N} p_{ti}(\ell_{ti} - \ell_{ti^*})^2 = \sum_{i \ne i^*} p_{ti}(\ell_{ti} - \ell_{ti^*})^2 \le 1 - p_{ti^*}$
  - In a stochastic environment, every time any suboptimal expert (other than i^*) is chosen, regret of at least $\Delta_{\min}$ is suffered in expectation:
    $R_T = \sum_{t=1}^{T} \sum_{i=1}^{N} \Delta_i p_{ti} \ge \sum_{t=1}^{T} \sum_{i \ne i^*} \Delta_{\min} p_{ti} = \Delta_{\min} \sum_{t=1}^{T} (1 - p_{ti^*})$
  - Combining the above two points, we obtain $R_T \ge \Delta_{\min} \sum_{t=1}^{T} z_t$
- Substituting this into the result of the theorem, we obtain $R_T = O\left(\sqrt{\frac{\log N}{\Delta_{\min}} R_T}\right)$.
- Square both sides: $R_T^2 = O\left(\frac{\log N}{\Delta_{\min}} R_T\right)$. Divide both sides by R_T: $R_T = O\left(\frac{\log N}{\Delta_{\min}}\right)$

Regret upper bound for CBMS

Theorem : CBMS achieves $R_T = O\left(\sqrt{\log N \cdot (1 + \sum_{t=1}^{T} z_t)}\right)$

Corollary : CBMS achieves $R_T = O\left(\frac{\log N}{\Delta_{\min}}\right)$ in stochastic environments

(Proof sketch of Corollary)

- In a stochastic environment, $1 + \sum_{t=1}^{T} z_t = O\left(\frac{R_T}{\Delta_{\min}}\right)$ holds true.
- Substituting this into the inequality of Theorem, $R_T = O\left(\sqrt{\frac{\log N}{\Delta_{\min}} R_T}\right)$, which implies $R_T = O\left(\frac{\log N}{\Delta_{\min}}\right)$

Regret upper bound for CBMS

Theorem : CBMS achieves $R_T = O\left(\sqrt{\log N \cdot (1 + \sum_{t=1}^{T} z_t)}\right)$

Corollary : CBMS achieves $R_T = O\left(\frac{\log N}{\Delta_{\min}}\right)$ in stochastic environments

(Proof sketch of Corollary)

- In a stochastic environment, $1 + \sum_{t=1}^{T} z_t = O\left(\frac{R_T}{\Delta_{\min}}\right)$ holds true.
- Substituting this into the inequality of Theorem, $R_T = O\left(\sqrt{\frac{\log N}{\Delta_{\min}} R_T}\right)$, which implies $R_T = O\left(\frac{\log N}{\Delta_{\min}}\right)$

Behavior of learning rates η_t:

- In a stochastic environment, $1 + \sum_{t=1}^{T} z_t = O\left(\frac{R_T}{\Delta_{\min}}\right) = O\left(\frac{\log N}{\Delta_{\min}^2}\right)$
- Therefore, $\eta_t \geq \eta_{T+1} \geq \Omega\left(\sqrt{\frac{\log N}{1+\sum_{t=1}^{T} z_t}}\right) \geq \Omega(\Delta_{\min})$
- $\Rightarrow$ learning rate η_t is bounded from below, and hence the algorithm behaves similarly to FTL

Best-of-both-worlds algorithm

(almost) tight upper bound

Table 1: Regret bounds for expert problems

	stochastic environment	Intermediate environment (?)	hostile environment
FTL	$O\left(\frac{\log N}{\Delta_{\min}}\right)$	$O(T)$	$O(T)$
Hedge	$O(\sqrt{T \log N})$	$O(\sqrt{T \log N})$	$O(\sqrt{T \log N})$
CBMS [CBMS07]	$O\left(\frac{\log N}{\Delta_{\min}}\right)$ [GSVE14]	??	$O(\sqrt{T \log N})$
regret lower bound	$\Omega\left(\frac{\log N}{\Delta_{\min}}\right)$	??	$\Omega(\sqrt{T \log N})$

- The CBMS algorithm summary:
  - Adaptively adjusting learning rate η_t (similar to AdaGrad etc.)
  - Achieving optimality for both environments
  - Working to interpolate between FTL and Hedge

[CBMS07] Nicolo Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. 2007.
[GSVE14] Pierre Gaillard, Gilles Stoltz, and Tim Van Erven. A second-order bound with excess losses. In *Conference on Learning Theory.* 2014.

Best-of-both-worlds algorithm

(almost) tight upper bound

Table 1: Regret bounds for expert problems

	stochastic environment	Intermediate environment (?)	hostile environment
FTL	$O\left(\frac{\log N}{\Delta_{\min}}\right)$	$O(T)$	$O(T)$
Hedge	$O(\sqrt{T \log N})$	$O(\sqrt{T \log N})$	$O(\sqrt{T \log N})$
CBMS [CBMS07]	$O\left(\frac{\log N}{\Delta_{\min}}\right)$ [GSVE14]	??	$O(\sqrt{T \log N})$
regret lower bound	$\Omega\left(\frac{\log N}{\Delta_{\min}}\right)$	??	$\Omega(\sqrt{T \log N})$

- The CBMS algorithm summary:
  - Adaptively adjusting learning rate η_t (similar to AdaGrad etc.)
  - Achieving optimality for both environments
  - Working to interpolate between FTL and Hedge

$\Downarrow$

- Does it work well in an intermediate environment between stochastic and adversarial?

Outline of the talk

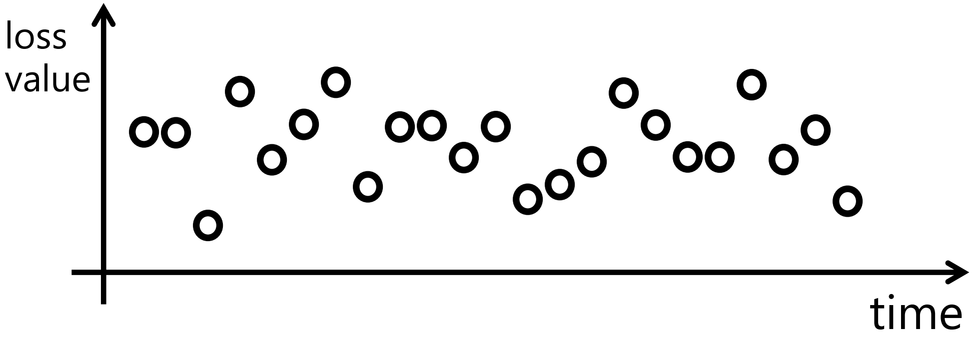
- Problem setup
  - Prediction with expert advice and multi-armed bandit
  - Two models for environments

- Basic results of regret analysis
  - Algorithms and regret analysis for the expert problem
  - Comparison of regrets in stochastic and adversarial environments

- **Best-of-both-worlds algorithms and analysis**
  - Hedge with adaptive learning rate
  - **Analysis between stochastic and adversarial (stochastic environment with adversarial corruption)**
  - Other recent developments

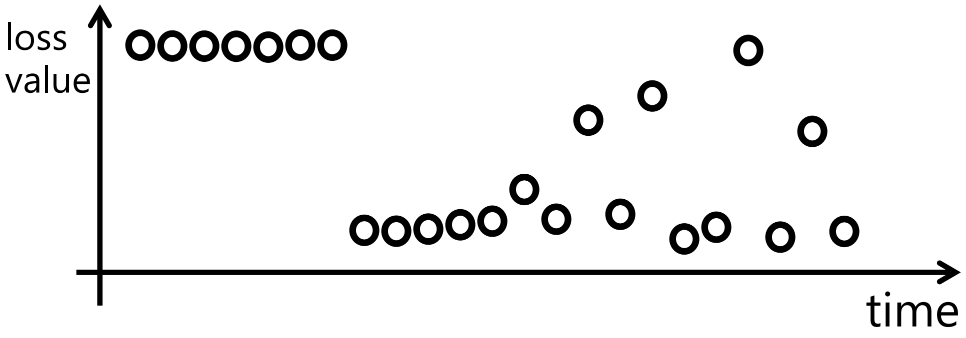
Stochastic + adversarial environment model

1. **Stochastic environment model**:
Losses and rewards are i.i.d.



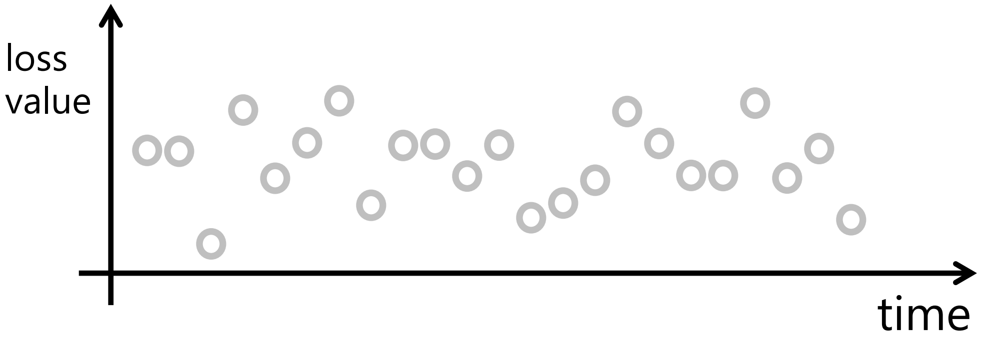
…

2. **Adversarial environment model**:
Losses and rewards **change arbitrarily**



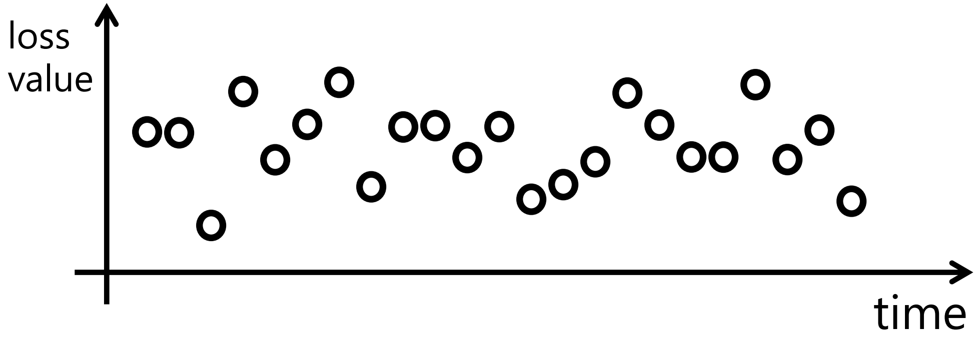
3. **Stochastic environment model with adversarial corruption**:
Between stochastic and adversarial

Generated from a stationary probability distribution

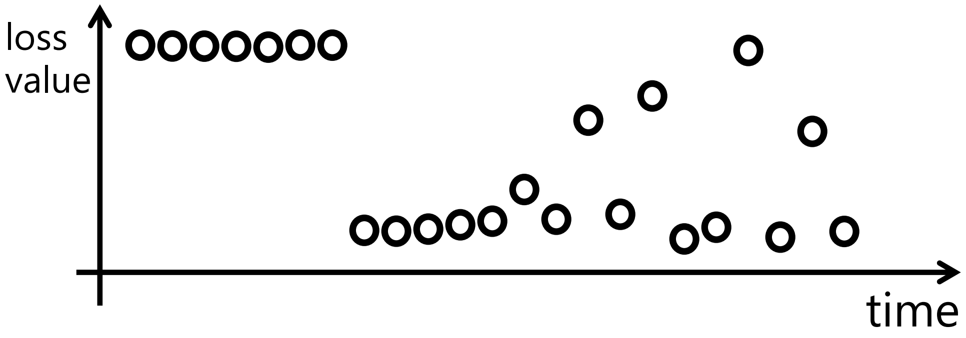


Stochastic + adversarial environment model

1. **Stochastic environment model**:
Losses and rewards are i.i.d.

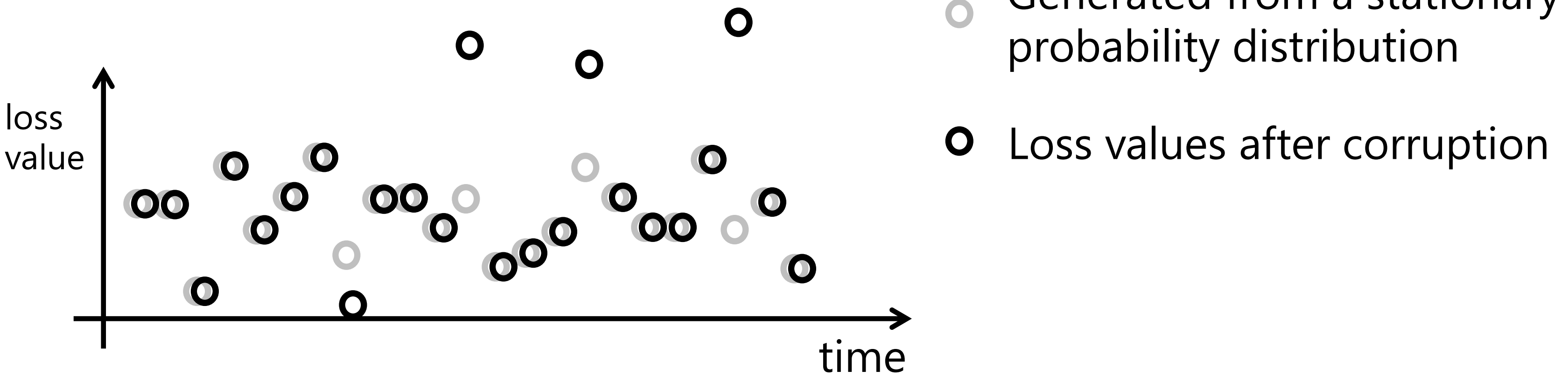


2. **Adversarial environment model**:
Losses and rewards **change arbitrarily**



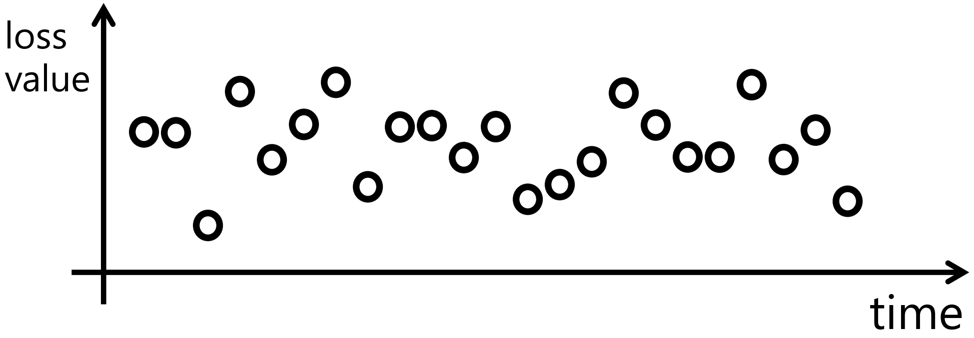
…

3. **Stochastic environment model with adversarial corruption**:
Between stochastic and adversarial



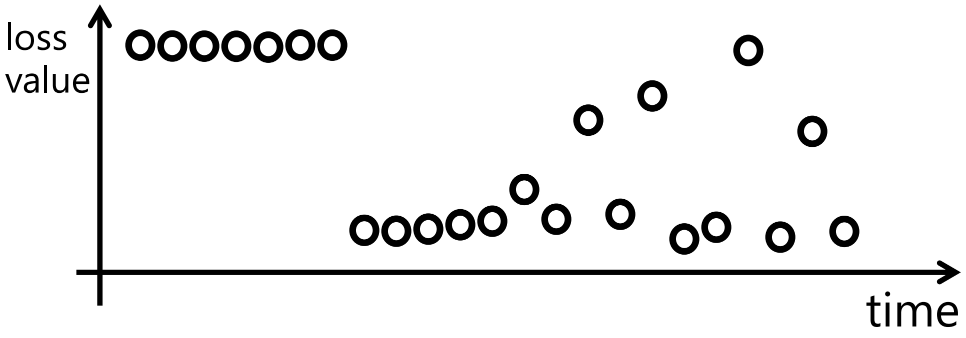
Stochastic + adversarial environment model

1. **Stochastic environment model**:
   Losses and rewards are i.i.d.

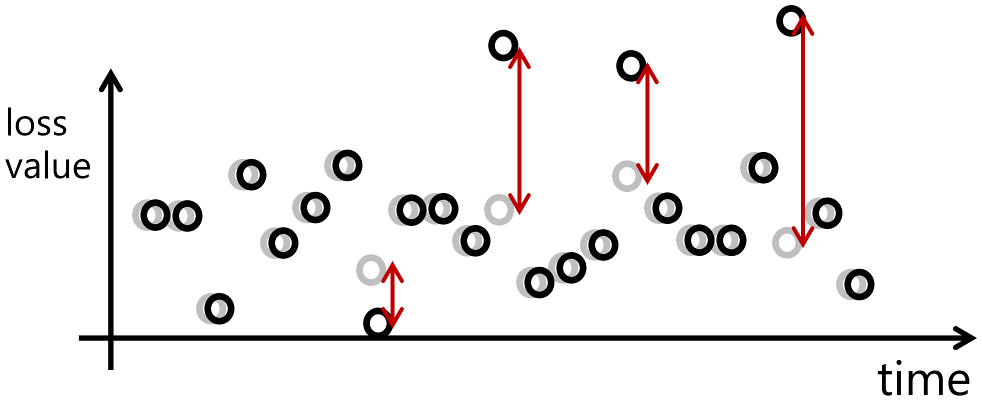


…

2. **Adversarial environment model**:
   Losses and rewards **change arbitrarily**



3. **Stochastic environment model with adversarial corruption**:
   Between stochastic and adversarial



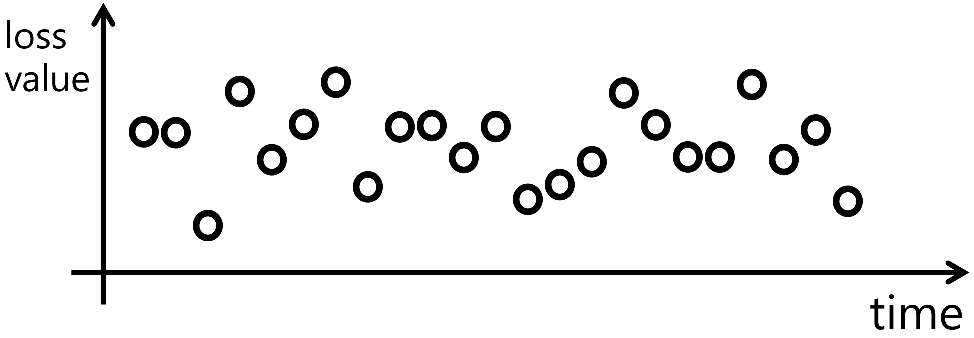
Generated from a stationary probability distribution

Loss values after corruption

Corruption level C:
sum of amount of corruption

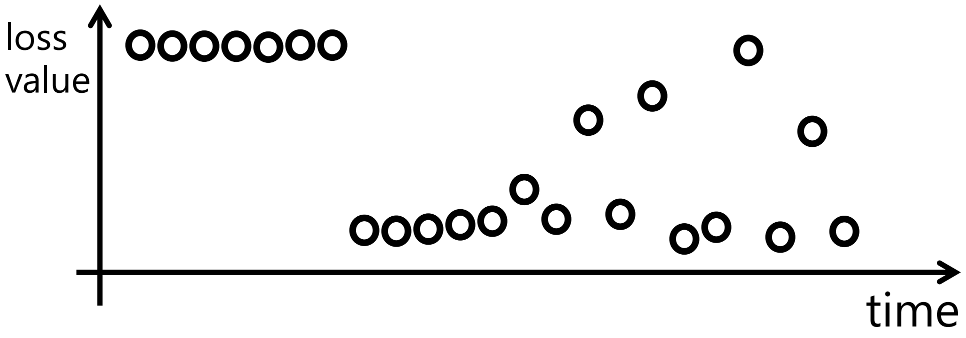
Stochastic + adversarial environment model

1. **Stochastic environment model**:
Losses and rewards are i.i.d.



…

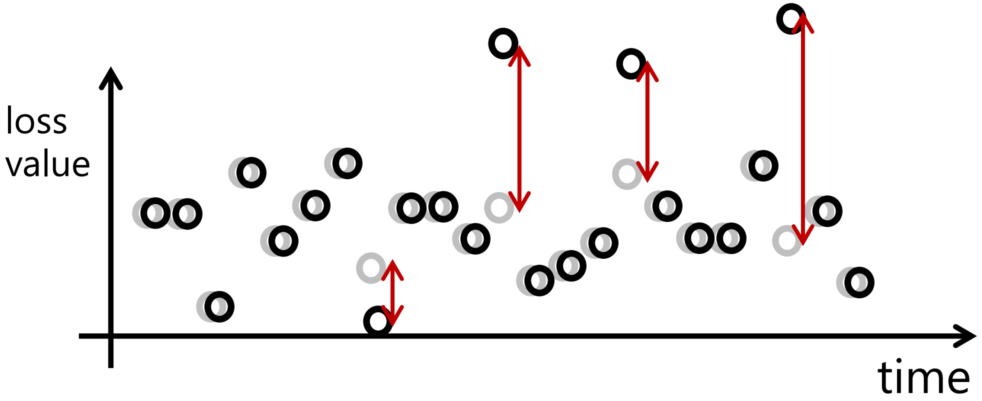
2. **Adversarial environment model**:
Losses and rewards **change arbitrarily**



$C = 0$

$C = O(T)$

3. **Stochastic environment model with adversarial corruption**:
Between stochastic and adversarial

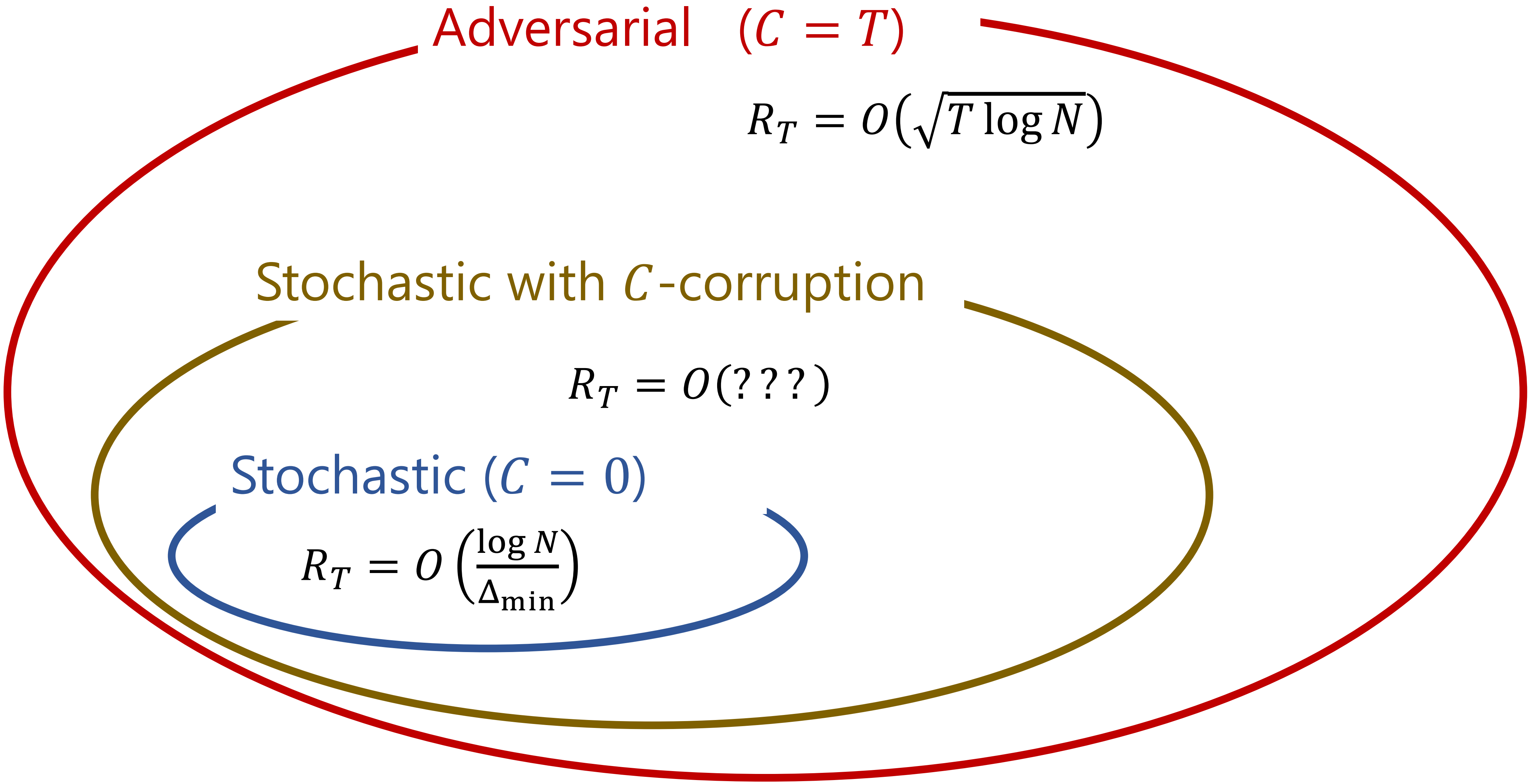


Generated from a stationary probability distribution

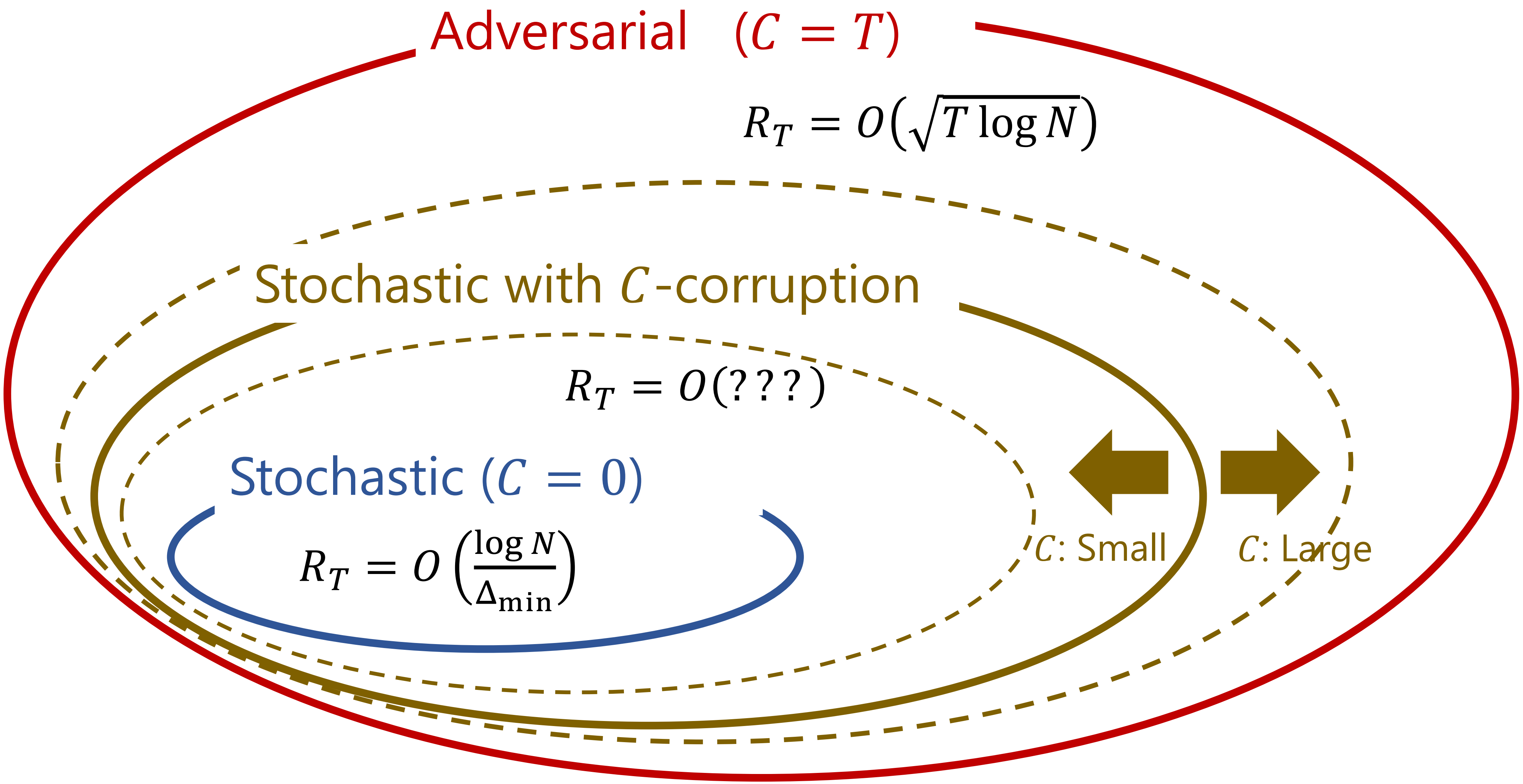
Loss values after corruption

Corruption level C:
sum of amount of corruption

Stochastic + adversarial environment model



Stochastic + adversarial environment model



Analysis for corrupted environments

Assumptions (stochastic environment model with corruption) :

There exists a distribution D over $[0,1]^d$ such that $\ell'_t \sim D$ (iid) for each $t \in [T]$ and the actual loss $\ell_t \in [0,1]^d$ satisfies $\sum_{t=1}^{T} \|\ell_t - \ell'_t\|_\infty \leq C$ for some C

In other words, loss is decomposed as $\ell_t = \ell'_t + c_t$ where $\sum_{t=1}^{T} \|c_t\|_\infty \leq C$

Stochastic **Adversarial**

What we can actually observe is only ℓ_t, and ℓ'_t and c_t **cannot be observed**

Suppose the corruption level C is **not given**

Analysis for corrupted environments

Assumptions (stochastic environment model with corruption) :

There exists a distribution D over $[0,1]^d$ such that $\ell'_t \sim D$ (iid) for each $t \in [T]$ and the actual loss $\ell_t \in [0,1]^d$ satisfies $\sum_{t=1}^{T} \|\ell_t - \ell'_t\|_\infty \leq C$ for some C

- Denote the expected value of $\ell'_t \sim D$ as $\mu = \mathrm{E}[\ell'_t]$
- Optimal expert is denoted as $i^* \in \arg\min_{i \in [N]} \mu_i$
- $\Delta_i \coloneqq \mu_i - \mu_{i^*}$ (Expected regret for choosing i), $\Delta_{\min} \coloneqq \min_{i \in [N] \setminus i^*} \Delta_i > 0$

Theorem: For any C, CBMS achieves $R_T = O\left(\frac{\log N}{\Delta_{\min}} + \sqrt{\frac{C \log N}{\Delta_{\min}}}\right)$

The effects of corruption is bounded by $O(\sqrt{C})$

Analysis for corrupted environments

Assumptions (stochastic environment model with corruption) :

There exists a distribution D over $[0,1]^d$ such that $\ell'_t \sim D$ (iid) for each $t \in [T]$ and the actual loss $\ell_t \in [0,1]^d$ satisfies $\sum_{t=1}^{T} \|\ell_t - \ell'_t\|_\infty \le C$ for some C

Theorem: For any C, CBMS achieves $R_T = O\left(\frac{\log N}{\Delta_{\min}} + \sqrt{\frac{C \log N}{\Delta_{\min}}}\right)$

(Proof sketch)

- Let R'_T denote the regret for loss ℓ'_t **before** corruption. Then $|R_T - R'_T| \le 2C$ from assumptions.
- CBMS achieves $R_T = O\left(\sqrt{\frac{\log N}{\Delta_{\min}} R'_T}\right)$ (similar to the proof for stochastic environments)
- From the above two points, $R_T = O\left(\sqrt{\frac{\log N}{\Delta_{\min}} (R_T + 2C)}\right)$, which implies $R_T^2 = O\left(\frac{\log N}{\Delta_{\min}} (R_T + 2C)\right)$
- This can be seen as a quadratic inequality in variable R_T, leading to $R_T = O\left(\frac{\log N}{\Delta_{\min}} + \sqrt{\frac{2C \log N}{\Delta_{\min}}}\right)$

Best-of-three-worlds algorithm

(almost) tight upper bound

Table 1: Regret bounds for expert problems

	stochastic environment	Stochastic with corruption	hostile environment
FTL	$O\left(\frac{\log N}{\Delta_{\min}}\right)$	$O(T)$	$O(T)$
M.W.U.	$O(\sqrt{T \log N})$	$O(\sqrt{T \log N})$	$O(\sqrt{T \log N})$
CBMS [CBMS07]	$O\left(\frac{\log N}{\Delta_{\min}}\right)$ [GSVE14]	$O\left(\frac{\log N}{\Delta_{\min}} + \sqrt{\frac{C \log N}{\Delta_{\min}}}\right)$ [I21]	$O(\sqrt{T \log N})$
regret lower bound	$\Omega\left(\frac{\log N}{\Delta_{\min}}\right)$	$\Omega\left(\frac{\log N}{\Delta_{\min}} + \sqrt{\frac{C \log N}{\Delta_{\min}}}\right)$ [I21]	$\Omega(\sqrt{T \log N})$

- CBMS is optimal even in stochastic environments with corruption!
    - Together with best-of-both-worlds regret bounds, it is sometimes called best-of-three-worlds (BOTW)

[CBMS07] Nicolo Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. 2007.
[GSVE14] Pierre Gaillard, Gilles Stoltz, and Tim Van Erven. A second-order bound with excess losses. *COLT.* 2014.
[I21] Shinji Ito. On optimal robustness to adversarial corruption in online decision problems. *NeurIPS*. 2021.

Best-of-three-worlds algorithm

(almost) tight upper bound

Table 2: Regret bounds for **multi-armed bandit problem**

	Stochastic setting	Stochastic with corruption	adversarial setting
UCB etc. [ACBF02]	$O\left(\sum_{i \neq i^*} \frac{\log T}{\Delta_i}\right)$	$O(T)$	$O(T)$
Exp3 [ACBFS02]	$O(\sqrt{TN \log N})$	$O(\sqrt{TN \log N})$	$O(\sqrt{TN \log N})$
Tsallis-INF [ZS21]	$O\left(\sum_{i \neq i^*} \frac{\log T}{\Delta_i}\right)$	$O\left(\sum_{i \neq i^*} \frac{\log T}{\Delta_i} + \sqrt{\sum_{i \neq i^*} \frac{C \log T}{\Delta_i}}\right)$	$O(\sqrt{TN})$
regret lower bound	$\Omega\left(\sum_{i \neq i^*} \frac{\log T}{\Delta_i}\right)$	$\Omega\left(\sum_{i \neq i^*} \frac{\log T}{\Delta_i} + \sqrt{\frac{C}{\Delta_{\min}}}\right)$	$\Omega(\sqrt{TN})$

- Tsallis-INF algorithm
  - Best-of-three-worlds for the multi-armed bandit problem
  - Based on the FTRL framework similarly to (adaptive) Hedge
  - Employing Tsallis entropy regularizers instead of Shannon entropy

[ACBF02] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47:235–256, 2002.
[ACBFS02] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.
[ZS21] Julian Zimmert and Yevgeny Seldin. Tsallis-INF: An optimal algorithm for stochastic and adversarial bandits. *Journal of Machine Learning Research*, 22(28):1–49, 2021.

FTRL approach

FTRL: set the distribution of arm selection by $p_t \in \arg\min_{p \in \Delta^N} \left\{ \left\langle \sum_{s=1}^{t-1} \hat{\ell}_s, p \right\rangle + \frac{1}{\eta_t} \psi(p) \right\}$

($\hat{\ell}_t$: unbiased estimator of ℓ_t, $\psi(p)$: regularization function)

- The regret is decomposed into stability z_t and penalty h_t by the standard analysis of FTRL:

$$R_T \leq \sum_{t=1}^{T} \left(\eta_t z_t + \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right) h_{t+1} \right) + \frac{1}{\eta_1} h_1$$

  - z_t, h_t changes depending on the regularization function ψ

FTRL approach

FTRL: set the distribution of arm selection by $p_t \in \arg\min_{p\in\Delta^N} \left\{\left\langle \sum_{s=1}^{t-1} \hat{\ell}_s , p \right\rangle + \frac{1}{\eta_t}\psi(p)\right\}$

($\hat{\ell}_t$: unbiased estimator of ℓ_t, $\psi(p)$: regularization function)

- The regret is decomposed into stability z_t and penalty h_t by the standard analysis of FTRL:

$$R_T \leq \sum_{t=1}^{T} \left(\eta_t z_t + \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right) h_{t+1} \right) + \frac{1}{\eta_1} h_1$$

  - z_t, h_t changes depending on the regularization function ψ

- Exp3 algorithm: [ACBFS02]
  - Defined ψ by $\psi(p) = -H(p)$ similarly to Hedge. Then $z_t \leq N$ and $h_t \leq \log N$ hold.
  - Therefore we have $R_T \leq N \sum_{t=1}^{T} \eta_t + \frac{\log N}{\eta_{T+1}}$. Setting $\eta_t = \sqrt{\frac{\log N}{NT}}$ leads to $R_T = O\left(\sqrt{TN \log N}\right)$

FTRL approach

FTRL: set the distribution of arm selection by $p_t \in \arg\min_{p \in \Delta^N} \left\{ \left\langle \sum_{s=1}^{t-1} \hat{\ell}_s, p \right\rangle + \frac{1}{\eta_t} \psi(p) \right\}$

($\hat{\ell}_t$: unbiased estimator of ℓ_t, $\psi(p)$: regularization function)

- The regret is decomposed into stability z_t and penalty h_t by the standard analysis of FTRL:

$$R_T \leq \sum_{t=1}^{T} \left(\eta_t z_t + \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right) h_{t+1} \right) + \frac{1}{\eta_1} h_1$$

  - z_t, h_t changes depending on the regularization function ψ

- Exp3 algorithm: [ACBFS02]
  - Defined ψ by $\psi(p) = -H(p)$ similarly to Hedge. Then $z_t \leq N$ and $h_t \leq \log N$ hold.
  - Therefore we have $R_T \leq N \sum_{t=1}^{T} \eta_t + \frac{\log N}{\eta_{T+1}}$. Setting $\eta_t = \sqrt{\frac{\log N}{NT}}$ leads to $R_T = O\left(\sqrt{TN \log N}\right)$
  - In expert problems, stability is $z_t \leq (1 - p_{ti^*})$ However, in multi-armed bandit $z_t \leq N$. This is doe to the larger variance of $\hat{\ell}_t$
  - Due to this worsening of the bound on z_t, learning rates depending on z_t is not effective for achieving BOBW

FTRL approach

FTRL: set the distribution of arm selection by $p_t \in \arg\min_{p \in \Delta^N} \left\{ \left\langle \sum_{s=1}^{t-1} \hat{\ell}_s , p \right\rangle + \frac{1}{\eta_t} \psi(p) \right\}$

($\hat{\ell}_t$: unbiased estimator of ℓ_t, $\psi(p)$: regularization function)

- The regret is decomposed into stability z_t and penalty h_t by the standard analysis of FTRL:

$$R_T \leq \sum_{t=1}^{T} \left(\eta_t z_t + \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right) h_{t+1} \right) + \frac{1}{\eta_1} h_1$$

  - z_t, h_t changes depending on the regularization function ψ

- **Tsallis-INF** algorithm: [ZS21]
  - Define regularization function with **1/2-Tsallis entropy :** $\psi(p) = -\sum_{i=1}^{N} \left(\sqrt{p_i} - p_i \right) = -\sum_{i=1}^{N} \sqrt{p_i} + 1$

FTRL approach

FTRL: set the distribution of arm selection by $p_t \in \arg\min_{p \in \Delta^N} \left\{ \left\langle \sum_{s=1}^{t-1} \hat{\ell}_s, p \right\rangle + \frac{1}{\eta_t} \psi(p) \right\}$

($\hat{\ell}_t$: unbiased estimator of ℓ_t, $\psi(p)$: regularization function)

- The regret is decomposed into stability z_t and penalty h_t by the standard analysis of FTRL:

$$R_T \leq \sum_{t=1}^{T} \left(\eta_t z_t + \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right) h_{t+1} \right) + \frac{1}{\eta_1} h_1$$

  - z_t, h_t changes depending on the regularization function ψ

- **Tsallis-INF** algorithm: [ZS21]
  - Define regularization function with **1/2-Tsallis entropy :** $\psi(p) = -\sum_{i=1}^{N} \left(\sqrt{p_i} - p_i \right) = -\sum_{i=1}^{N} \sqrt{p_i} + 1$
  - We then have $z_t = O\left(\sum_{i=1}^{N} \sqrt{p_{ti}} - 1 \right)$ and $h_t = O\left(\sum_{i=1}^{N} \sqrt{p_{ti}} - 1 \right)$

FTRL approach

FTRL: set the distribution of arm selection by $p_t \in \arg\min_{p \in \Delta^N} \left\{ \left\langle \sum_{s=1}^{t-1} \hat{\ell}_s, p \right\rangle + \frac{1}{\eta_t} \psi(p) \right\}$

($\hat{\ell}_t$: unbiased estimator of ℓ_t, $\psi(p)$: regularization function)

- The regret is decomposed into stability z_t and penalty h_t by the standard analysis of FTRL:

$$R_T \leq \sum_{t=1}^{T} \left(\eta_t z_t + \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right) h_{t+1} \right) + \frac{1}{\eta_1} h_1$$

  - z_t, h_t changes depending on the regularization function ψ

- **Tsallis-INF** algorithm: [ZS21]
  - Define regularization function with **1/2-Tsallis entropy :** $\psi(p) = -\sum_{i=1}^{N} \left(\sqrt{p_i} - p_i \right) = -\sum_{i=1}^{N} \sqrt{p_i} + 1$
  - We then have $z_t = O\left(\sum_{i=1}^{N} \sqrt{p_{ti}} - 1 \right)$ and $h_t = O\left(\sum_{i=1}^{N} \sqrt{p_{ti}} - 1 \right)$
  - Set $\eta_t = \frac{1}{\sqrt{t}}$. Then, $\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right) = O\left(\frac{1}{\sqrt{t+1}} \right)$. Hence,

$$R_T \leq \sum_{t=1}^{T} \left(\eta_t z_t + \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right) h_{t+1} \right) + \frac{1}{\eta_1} h_1 = O\left(\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \left(\sum_{i=1}^{N} \sqrt{p_{ti}} - 1 \right) \right) = O\left(\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \sum_{i=i^*} \sqrt{p_{ti}} \right)$$

Regret upper bound for Tsallis-INF

Theorem: Tsallis-INF achieves $R_T = O\left(\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \sum_{i \neq i^*} \sqrt{p_{ti}}\right)$

Corollary 1: Tsallis-INF Achieves $R_T = O\left(\sqrt{NT}\right)$ in adversarial environments

Corollary 2: Tsallis-INF achieves $R_T = O\left(\sqrt{\sum_{i \neq i^*} \frac{\log T}{\Delta_i}}\right)$ in stochastic environments

Regret upper bound for Tsallis-INF

Theorem: Tsallis-INF achieves $R_T = O\left(\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \sum_{i \neq i^*} \sqrt{p_{ti}}\right)$

Corollary 1: Tsallis-INF Achieves $R_T = O\left(\sqrt{NT}\right)$ in adversarial environments

(proof) $\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \sum_{i \neq i^*} \sqrt{p_{ti}} \leq \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \sqrt{(N-1) \sum_{i \neq i^*} p_{ti}} \leq \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \sqrt{(N-1)} = O\left(\sqrt{NT}\right)$

Cauchy–Schwarz

Corollary 2: Tsallis-INF achieves $R_T = O\left(\sqrt{\sum_{i \neq i^*} \frac{\log T}{\Delta_i}}\right)$ in stochastic environments

Regret upper bound for Tsallis-INF

Theorem: Tsallis-INF achieves $R_T = O\left(\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \sum_{i \neq i^*} \sqrt{p_{ti}}\right)$

Corollary 1: Tsallis-INF Achieves $R_T = O\left(\sqrt{NT}\right)$ in adversarial environments

(proof) $\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \sum_{i \neq i^*} \sqrt{p_{ti}} \leq \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \sqrt{(N-1) \sum_{i \neq i^*} p_{ti}} \leq \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \sqrt{(N-1)} = O\left(\sqrt{NT}\right)$

Cauchy–Schwarz

Corollary 2: Tsallis-INF achieves $R_T = O\left(\sqrt{\sum_{i \neq i^*} \frac{\log T}{\Delta_i}}\right)$ in stochastic environments

(proof) $\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \sum_{i \neq i^*} \sqrt{p_{ti}} = \sum_{t=1}^{T} \sum_{i \neq i^*} \frac{1}{\sqrt{t\Delta_i}} \sqrt{\Delta_i p_{ti}}$

$$\leq \sqrt{\sum_{t=1}^{T} \sum_{i \neq i^*} \frac{1}{t\Delta_i}} \cdot \sqrt{\sum_{t=1}^{T} \sum_{i \neq i^*} \Delta_i p_{ti}} = \sqrt{\sum_{i \neq i^*} \frac{1}{\Delta_i} \sum_{t=1}^{T} \frac{1}{t}} \cdot \sqrt{R_T}$$

Cauchy–Schwarz

$$\leq \sqrt{\sum_{i \neq i^*} \frac{\log T}{\Delta_i} \cdot R_T}$$

- This can be extended to corrupted environments as well

Key points in proofs for (corrupted) stochastic environment

- Self-bounding technique
  - If we obtain a bound of $R_T = O\left(\sqrt{A \cdot R_T} + B\right)$ for some A and B, we have $R_T = O(A + B)$
  - This approach is called *self-bounding technique*
  - In recent years, it is often used in the design and analysis of BOBW/BOTW algorithms
  - There are similar analysis techniques in the context of gradient descent in convex optimization (e.g., improving convergence rate for strongly convex functions).

Key points in proofs for (corrupted) stochastic environment

- Self-bounding technique
  - If we obtain a bound of $R_T = O\left(\sqrt{A \cdot R_T} + B\right)$ for some A and B,
    we have $R_T = O(A + B)$
  - This approach is called *self-bounding technique*
  - In recent years, it is often used in the design and analysis of BOBW/BOTW algorithms
  - There are similar analysis techniques in the context of gradient descent in convex optimization (e.g., improving convergence rate for strongly convex functions).

- My own impressions
  - I was surprised that it was possible to obtain a tight upper bound in stochastic environments without using concentration inequalities (e.g., Hoeffding's).

Progress in research on BOBW (1)

Highlight : self-bounding technique

2012 ~ **2017**

- Concept of BOBW in multi-armed bandit [Bubeck & Slivkins , COLT'12]
- Expert problems [de Rooij +, JMLR'14] [Gaillard, Stoltz & Van Erven, COLT'14] [Luo & Schapire , COLT'15]
- Improved BOBW for MAB [Seldin & Slivkins , ICML'14] [Auer & Chiang, COLT'16] [Seldin & Lugosi, COLT'17]

2018

- Multi-armed bandit self-bounding technique [Zimmert & Seldin, AISTATS'18, JMLR'21] [Wei & Luo, COLT'18]
- Best-arm identification [Abbasi- Yadkori , COLT'18]

2019

- Combinatorial semi-bandit [Zimmert , Luo & Wei, ICML'19]

2020

- Decoupled multi-armed bandit [Rouyer & Seldin, COLT'20]
- MDP (known transition model) [Jin & Luo, NeurIPS'20]

2021

- Multi-armed bandit with data-dependent bound [I, COLT'21]
- Multi-armed bandit stochastic/adversarial mixture [Masoudian & Seldin, COLT'21]
- Linear bandit [Lee+, ICML'21]
- Problems with switch cost [Rouyer, Seldin & Cesa-Bianchi, ICML'21]
- MDP (unknown transition model) [Jin , Huang & Luo, NeurIPS'21]
- Graph bandit [Erez & Koren, NeurIPS'21]
- Combination semi-bandit data-dependent bounds [I, NeurIPS'21]
- Expert problems stochastic/adversarial mixture [I, NeurIPS'21]

Progress in research on BOBW (1)

Highlight : self-bounding technique

2022

- Multi-armed bandit variance-dependent bound [I, Tsuchiya & Honda, COLT'22]
- Submodular function minimization [I, ICML'22]
- Dueling bandit [Saha & Gaillard, ICML'22]
- Graph bandit [I, Tsuchiya & Honda, NeurIPS'22] , [Kong, Zhou & Li, ICML'22], [Rouyer +, NeurIPS'22]
- Problem with switching cost [Amir+, NeurIPS'22]
- Delayed feedback MAB [Masoudian, Zimmert & Seldin, NeurIPS'22]

2023

- Partial observation problem [Tsuchiya, I & Honda, ALT'23]
- FTPL analysis [Honda, I & Tsuchiya, ALT'23]
- Combination semi-bandit variance-dependent bound [Tsuchiya, I & Honda, AISTATS'23]
- MDP (policy optimization) [Dann, Wei & Zimmert , COLT'23]
- Linear bandit [I & Takemura , COLT'23] [I & Takemura , NeurIPS'23] , [Kong, Zhao & Li, COLT'23]
- Black-box conversion [Dann, Wei & Zimmert , COLT'23]
- Sparse multi-armed bandit [Tsuchiya, I & Honda, NeurIPS'23]
- Relaxing the optimal solution uniqueness assumption [Jin , Liu & Luo, NeurIP'23]
- MDP (adversarial transition) [Jin +, NeurIPS'23]

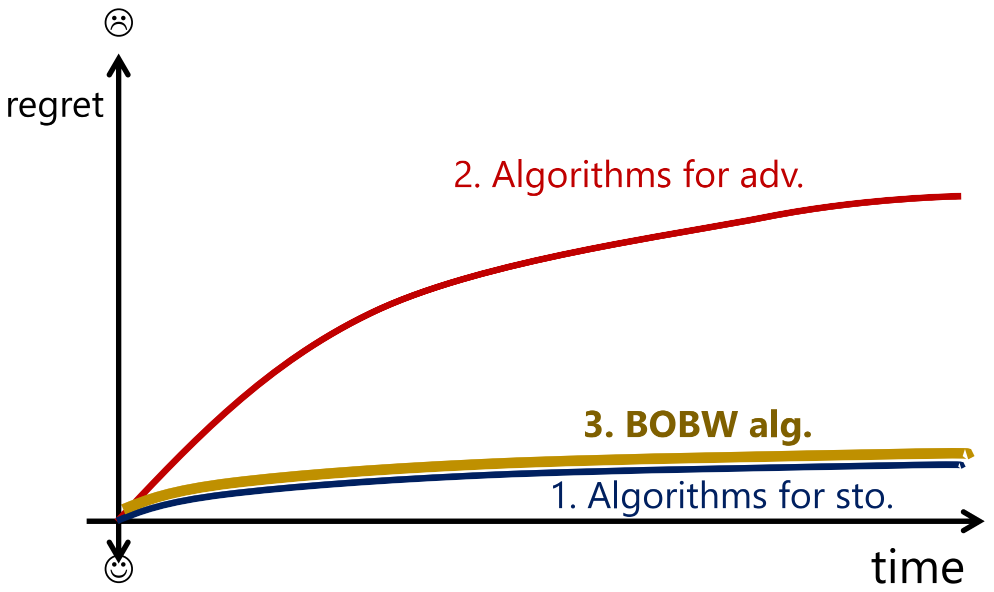
Summary of BOBW/BOTW algorithms

- For the expert problem, CBMS achieves BOTW
- For the multi-armed bandit problem Tsallis-INF achieves BOTW
- For both the expert problem and the multi-armed bandit problem, algorithm design and regret analysis are based on FTRL and self-bounding technique.
- By appropriately designing the regularization function and learning rates, BOTW algorithms can be constructed for various online learning / Bandit problems, including combinatorial semi-bandits, linear bandits, dueling bandits, graph-feedback problems, episodic MDPs, …

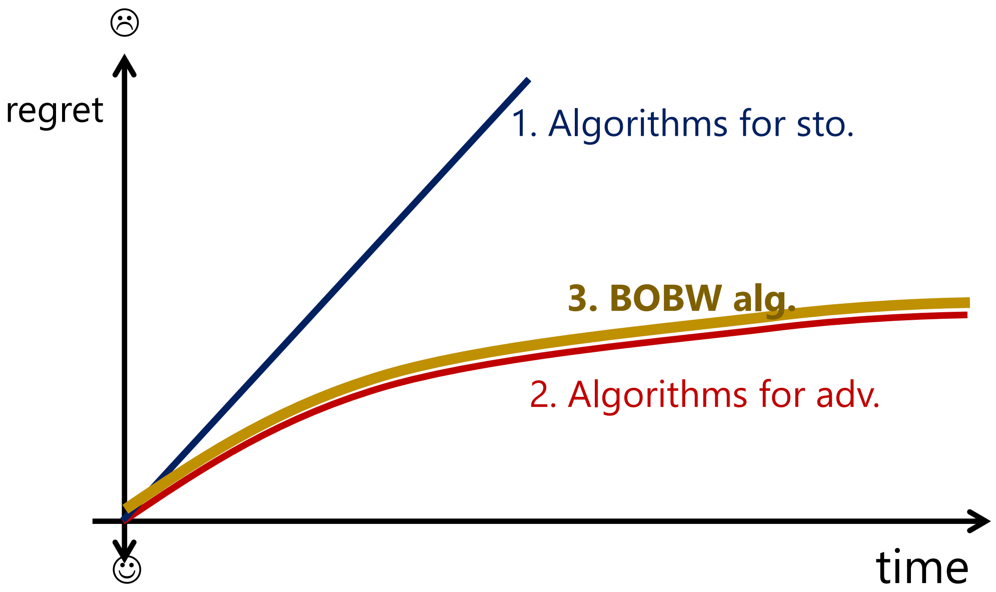
Open question: gaps of constant factors

- Ideal results of BOBW:

1.**Stochastic environment model** :

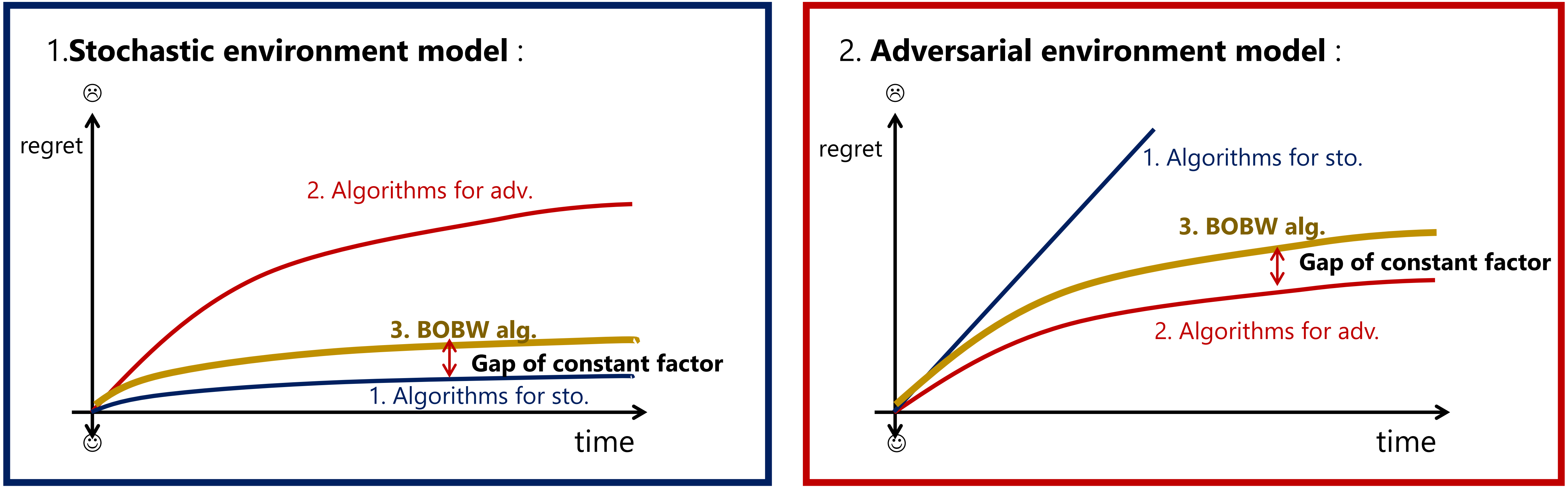


2. **Adversarial environment model** :



Open question: gaps of constant factors

- Real (current state-of-the-art):



Question: Can we remove these gaps of constant factors? How?

Recommended references for further understanding

- Cesa-Bianchi, N., & Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.
- Shalev-Shwartz, S. (2012). Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, *4*(2), 107-194.
- Hazan, E. (2022). *Introduction to online convex optimization*. MIT Press.
- Orabona, F., & Pál, D. (2018). Scale-free online learning. *Theoretical Computer Science*, *716*, 50-69.
- Orabona, F. (2019). A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*.
- Lattimore, T., & Szepesvári, C. (2020). *Bandit algorithms*. Cambridge University Press.

References

[AYBG+18] Yasin Abbasi-Yadkori, Peter Bartlett, Victor Gabillon, Alan Malek, and Michal Valko. Best of both worlds: Stochastic & adversarial best-arm identification. In *Conference on Learning Theory*, pages 918–949. PMLR, 2018.

[AAKL22] Idan Amir, Guy Azov, Tomer Koren, and Roi Livni. Better best of both worlds bounds for bandits with switching costs. *Advances in Neural Information Processing Systems*, 35, 2022.

[AHK12] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[ACBF02] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47:235–256, 2002.

[ACBFS02] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[AK08] Baruch Awerbuch and Robert Kleinberg. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1):97–114, 2008.

[AK04] Baruch Awerbuch and Robert D Kleinberg. Adaptive routing with end-to-end feedback: Dis- tributed learning and geometric approaches. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 45–53, 2004.

[BS12] Sébastien Bubeck and Aleksandrs Slivkins. The best of both worlds: Stochastic and adversarial bandits. In *Conference on Learning Theory*, pages 42–1. PMLR, 2012.

[CBL06] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[CBMS07] Nicolo Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66:321–352, 2007.

[Cov91] Thomas M Cover. Universal portfolios. *Mathematical finance*, 1(1):1–29, 1991.

[DWZ23a] Chris Dann, Chen-Yu Wei, and Julian Zimmert. A blackbox approach to best of both worlds in bandits and beyond. In *Conference on Learning Theory*, pages 5503–5570. PMLR, 2023.

[DWZ23a] Christoph Dann, Chen-Yu Wei, and Julian Zimmert. Best of both worlds policy optimization. In *International Conference on Machine Learning*, pages 6968–7008. PMLR, 2023.

[DRVEGK14] Steven De Rooij, Tim Van Erven, Peter D Grünwald, and Wouter M Koolen. Follow the leader if you can, hedge if you must. *The Journal of Machine Learning Research*, 15(1):1281–1316, 2014.

[DHS11] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[EK21] Liad Erez and Tomer Koren. Towards best-of-all-worlds online learning with feedback graphs. *Advances in Neural Information Processing Systems*, 34, 2021.

[GSVE14] Pierre Gaillard, Gilles Stoltz, and Tim Van Erven. A second-order bound with excess losses. In *Conference on Learning Theory*, pages 176–196. PMLR, 2014.

[Haz16] Elad Hazan. Introduction to online convex optimization. *Found. Trends Optim.*, 2:157–325, 2016.

[HAK06] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69:169–192, 2006.

[HRB07] Elad Hazan, Alexander Rakhlin, and Peter Bartlett. Adaptive online gradient descent. *Advances in Neural Information Processing Systems*, 20, 2007.

[HIT23] Junya Honda, Shinji Ito, and Taira Tsuchiya. Follow-the-perturbed-leader achieves best-of- both-worlds for bandit problems. In *International Conference on Algorithmic Learning Theory*, pages 726–754. PMLR, 2023.

[Ito21a] Shinji Ito. Hybrid regret bounds for combinatorial semi-bandits and adversarial linear bandits. *Advances in Neural Information Processing Systems* , 34:2654–2667, 2021.

Ito . On optimal robustness to adversarial corruption in online decision problems. *Advances in Neural Information Processing Systems* , 34:7409–7420, 2021.

References

[Ito21c] Shinji Ito. Parameter-free multi-armed bandit algorithms with hybrid data-dependent regret bounds. In *Conference on Learning Theory*, pages 2552–2583. PMLR, 2021.
[Ito22] Shinji Ito. Revisiting online submodular minimization: Gap-dependent regret bounds, best of both worlds and adversarial robustness. In *International Conference on Machine Learning*, pages 9678–9694. PMLR, 2022.
[IT23] Shinji Ito and Kei Takemura. Best-of-three-worlds linear bandit algorithm with variance- adaptive regret bounds. In *Conference on Learning Theory*, pages 2653–2677. PMLR, 2023.
[ITH22a] Shinji Ito, Taira Tsuchiya, and Junya Honda. Adversarially robust multi-armed bandit algorithm with variance-dependent regret bounds. In *Conference on Learning Theory*, pages 1421–1422. PMLR, 2022.
[ITH22b] Shinji Ito, Taira Tsuchiya, and Junya Honda. Nearly optimal best-of-both-worlds algorithms for online learning with feedback graphs. In *Advances in Neural Information Processing Systems*, volume 35, 2022.
[JHL21] Tiancheng Jin, Longbo Huang, and Haipeng Luo. The best of both worlds: stochastic and adversarial episodic mdps with unknown transition. *Advances in Neural Information Processing Systems*, 34, 2021.
[JLL23] Tiancheng Jin, Junyan Liu, and Haipeng Luo. Improved best-of-both-worlds guarantees for multi-armed bandits: Ftrl with general regularizers and multiple optimal arms. *arXiv preprint arXiv:2302.13534*, 2023.
[JL20] Tiancheng Jin and Haipeng Luo. Simultaneously learning stochastic and adversarial episodic mdps with known transition. *Advances in Neural Information Processing Systems*, 33:16557– 16566, 2020.
[KZL22] Fang Kong, Yichi Zhou, and Shuai Li. Simultaneously learning stochastic and adversarial bandits with general graph feedback. In *International Conference on Machine Learning*, pages 11473–11482. PMLR, 2022.
[LS20] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.
[LLW+21] Chung-Wei Lee, Haipeng Luo, Chen-Yu Wei, Mengxiao Zhang, and Xiaojin Zhang. Achieving near instance-optimality and minimax-optimality in stochastic and adversarial linear bandits simultaneously. In *International Conference on Machine Learning*, pages 6142–6151. PMLR, 2021.
[LW94] Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.
[LS15] Haipeng Luo and Robert E Schapire. Achieving all with no parameters: Adanormalhedge. In *Conference on Learning Theory*, pages 1286–1304. PMLR, 2015.
[MS21] Saeed Masoudian and Yevgeny Seldin. Improved analysis of the tsallis-inf algorithm in stochastically constrained adversarial bandits and stochastic bandits with adversarial corruptions. In *Conference on Learning Theory*, pages 3330–3350. PMLR, 2021.
[MZS22] Saeed Masoudian, Julian Zimmert, and Yevgeny Seldin. A best-of-both-worlds algorithm for bandits with delayed feedback. *Advances in Neural Information Processing Systems*, 35, 2022.
[MS10] H Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization. *COLT 2010*, page 244, 2010.
[Ora19] Francesco Orabona. A modern introduction to onlinelearning.*arXivpreprintarXiv:1912.13213*, 2019.
[RS20] Chloé Rouyer and Yevgeny Seldin. Tsallis-inf for decoupled exploration and exploitation in multi-armed bandits. In *Conference on Learning Theory*, pages 3227–3249. PMLR, 2020.
[RSCB21] Chloé Rouyer, Yevgeny Seldin, and Nicolò Cesa-Bianchi. An algorithm for stochastic and adversarial bandits with switching costs. In *International Conference on Machine Learning*, pages 9127–9135. PMLR, 2021.
[RvdHCBS22] Chloé Rouyer, Dirk van der Hoeven, Nicolò Cesa-Bianchi, and Yevgeny Seldin. A near-optimal best-of-both-worlds algorithm for online learning with feedback graphs. *Advances in Neural Information Processing Systems*, 35, 2022.

References

[SG22] Aadirupa Saha and Pierre Gaillard. Versatile dueling bandits: Best-of-both world analyses for learning from relative preferences. In *International Conference on Machine Learning*, pages 19011–19026. PMLR, 2022.

[SL17] Yevgeny Seldin and Gábor Lugosi. An improved parametrization and analysis of the EXP3++ algorithm for stochastic and adversarial bandits. In *Conference on Learning Theory*, pages 1743–1759. PMLR, 2017.

[SS14] Yevgeny Seldin and Aleksandrs Slivkins. One practical algorithm for both stochastic and adversarial bandits. In *International Conference on Machine Learning*, pages 1287–1295. PMLR, 2014.

[TIH23a] Taira Tsuchiya, Shinji Ito, and Junya Honda. Best-of-both-worlds algorithms for partial monitoring. In *International Conference on Algorithmic Learning Theory*, pages 1484–1515. PMLR, 2023.

[TIH23b] Taira Tsuchiya, Shinji Ito, and Junya Honda. Further adaptive best-of-both-worlds algorithm for combinatorial semi-bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 8117–8144. PMLR, 2023.

[TIH23c] Taira Tsuchiya, Shinji Ito, and Junya Honda. Stability-penalty-adaptive follow-the-regularized- leader: Sparsity, game-dependency, and best-of-both-worlds. *arXiv preprint arXiv:2305.17301*, 2023.

[Vov01] Volodya Vovk. Competitive on-line statistics. *International Statistical Review*, 69(2):213–248, 2001.

[WL18] Chen-YuWeiandHaipengLuo.Moreadaptivealgorithmsforadversarialbandits.In*Conference On Learning Theory*, pages 1263–1291. PMLR, 2018.

[ZLW19] Julian Zimmert, Haipeng Luo, and Chen-Yu Wei. Beating stochastic and adversarial semi- bandits optimally and simultaneously. In *International Conference on Machine Learning*, pages 7683–7692. PMLR, 2019.

[ZS21] Julian Zimmert and Yevgeny Seldin. Tsallis-INF: An optimal algorithm for stochastic and adversarial bandits. *Journal of Machine Learning Research*, 22(28):1–49, 2021.

[Zin03] Martin Zinkevich . Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning* , pages 928–936, 2003.