NJUAI 南京大学 人工智能学院
SCHOOL OF ARTIFICIAL INTELLIGENCE, NANJING UNIVERSITY

Lecture 11. Adversarial Bandits

Advanced Optimization (Fall 2022)

Peng Zhao

zhaop@lamda.nju.edu.cn

Nanjing University

Outline

- Problem Setup
- Multi-Armed Bandits
  - Exp3 Algorithm
  - Upper Bound
  - Lower Bound
- Advanced Topics

Online Convex Optimization

At each round $t = 1, 2, \cdots$

(1) the player first picks a model $\mathbf{x}_t$ from a convex set $\mathcal{X} \subseteq \mathbb{R}^d$;

(2) and environments pick an online convex function $f_t : \mathcal{X} \to \mathbb{R}$;

(3) the player suffers loss $f_t(\mathbf{x}_t)$, observes some information about f_t and updates the model.

Problem	**Domain**	**Loss Functions**
General OCO	convex set $\mathcal{X} \subset \mathbb{R}^d$	convex function $f_t(\cdot)$
OCO with Strongly Convex Functions	convex set $\mathcal{X} \subset \mathbb{R}^d$	$\nabla^2 f_t(\mathbf{x}) \succeq \alpha \mathbf{I}$
OCO with Exp-concave Functions	convex set $\mathcal{X} \subset \mathbb{R}^d$	$\nabla^2 f_t(\mathbf{x}) \succeq \beta \nabla f_t(\mathbf{x}) \nabla f_t(\mathbf{x})^\top$
Prediction with Experts' Advice	$\Delta_d = \{\mathbf{x} \in \mathbb{R}_+ \mid \sum_{i=1}^d x_i = 1\}$	$f_t(\mathbf{x}) = \langle \boldsymbol{\ell}_t, \mathbf{x} \rangle$

OCO Algorithms learned so far

- Given *first-order* information oracle: *worst-case* bound

Problem(Abbr.)	Domain	Loss Functions	*Algorithms*	Bounds
OCO	$\mathcal{X}$	Convex Function $f_t(\cdot)$	OGD: $\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}[\mathbf{x}_t - \eta \nabla f_t(\mathbf{x}_t)]$	$\mathcal{O}(\sqrt{T})$
Strongly Convex	$\mathcal{X}$	$\nabla^2 f_t(\mathbf{x}) \succeq \alpha \mathbf{I}$	OGD: $\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}[\mathbf{x}_t - \eta_t \nabla f_t(\mathbf{x}_t)]$	$\mathcal{O}(\frac{1}{\alpha} \log T)$
Exp-concave	$\mathcal{X}$	$\nabla^2 f_t(\mathbf{x}) \succeq \frac{1}{\beta} \nabla f_t(\mathbf{x}) \nabla f_t(\mathbf{x})^\top$	ONS: $\mathbf{x}_{t+1} = \Pi_{A_t}[\mathbf{x}_t - \eta A_t^{-1} \nabla f_t(\mathbf{x}_t)]$	$\mathcal{O}(\frac{d}{\beta} \log T)$
PEA	$\Delta(d)$	$f_t(\mathbf{x}) = \langle \mathbf{p}_t, \mathbf{x} \rangle$	Hedge: $x_{t+1,i} \propto \exp(-\eta \sum_{s=1}^{t} \ell_{s,i})$	$\mathcal{O}(\sqrt{T \log d})$

Online Mirror Descent

At each round $t = 1, 2, \cdots$

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x} \in \mathcal{X}} \eta_t \langle \mathbf{x}, \nabla f_t(\mathbf{x}_t) \rangle + \mathcal{D}_\psi(\mathbf{x}, \mathbf{x}_t)$$

where $\mathcal{D}_\psi(\mathbf{x}, \mathbf{y}) = \psi(\mathbf{x}) - \psi(\mathbf{y}) - \langle \nabla \psi(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle$ is the Bregman divergence.

OCO Algorithms learned so far

- Given *first-order* information oracle: *adaptive* bound

$$\mathbf{x}_t = \arg\min_{\mathbf{x}\in\mathcal{X}} \eta \langle M_t, \mathbf{x}\rangle + \frac{1}{2}\|\mathbf{x} - \widehat{\mathbf{x}}_t\|_2^2$$

$$\widehat{\mathbf{x}}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \eta \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}\rangle + \frac{1}{2}\|\mathbf{x} - \widehat{\mathbf{x}}_t\|_2^2$$

	Assumption(s)	**Setting of Optimism**	**Setting of η_t**	**Adaptive Regret Bound**
Small-loss Bound	L-smooth + non-negative	$M_t = \mathbf{0}$	$\approx \frac{D}{\sqrt{1+\widetilde{G}_t}}$	$\mathcal{O}\left(\sqrt{1+F_T}\right)$
Variance Bound	–	$M_t = \widetilde{\mu}_{t-1}$	$\approx \frac{D}{\sqrt{1+\mathrm{Var}_{t-1}}}$	$\mathcal{O}\left(\sqrt{1+\mathrm{Var}_T}\right)$
Variation Bound	L-smooth	$M_t = \nabla f_{t-1}(\mathbf{x}_{t-1})$	$\approx \frac{D}{\sqrt{1+\widetilde{V}_{t-1}}}$	$\mathcal{O}\left(\sqrt{1+V_T}\right)$

Online Convex Optimization

At each round $t = 1, 2, \cdots$

(1) the player first picks a model $\mathbf{x}_t$ from a convex set $\mathcal{X} \subseteq \mathbb{R}^d$;

(2) and environments pick an online convex function $f_t : \mathcal{X} \to \mathbb{R}$;

(3) the player suffers loss $f_t(\mathbf{x}_t)$, observes some information about f_t and updates the model.

on the feedback information:

- ***full information***: observe entire f_t (or at least gradient $\nabla f_t(\mathbf{x}_t)$)

- ***partial information (bandits)***: observe function value $f_t(\mathbf{x}_t)$ only

less information

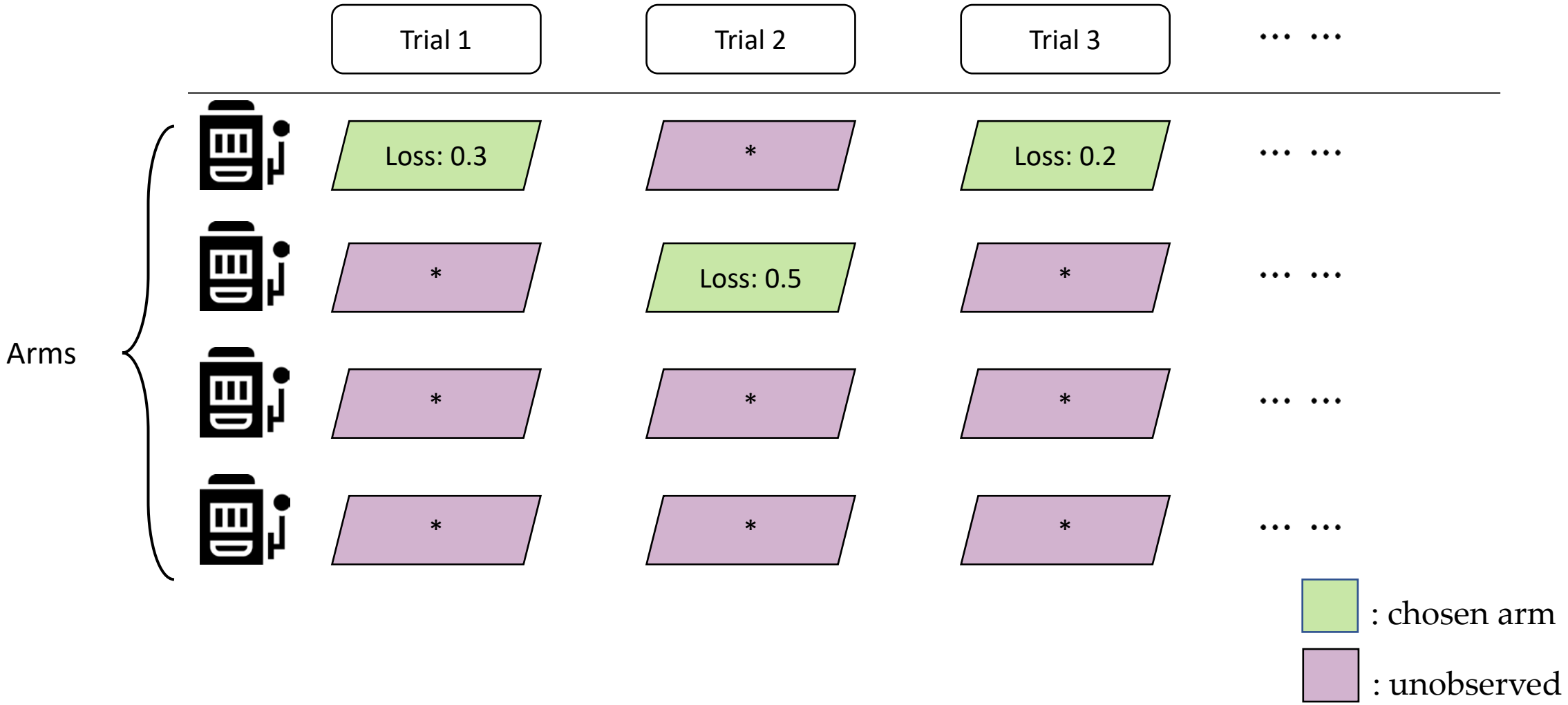
full information

horse racing

partial information

multi-armed bandits

Multi-Armed Bandit



Formulation

At each round $t = 1, 2, \cdots$

(1) the player first picks an arm $a_t \in [K]$ from K candidate arms;

(2) and simultaneously environments pick a loss vector $\boldsymbol{\ell}_t \in [0, 1]^K$;

(3) the player suffers and only observes loss $\ell_t(a_t)$, then updates the model.

on the difficulty of environments:

- ***adversarial*** setting
  - ***oblivious***: $\{\boldsymbol{\ell}_t\}_{t=1}^T$ are chosen before the game starts.
  - ***non-oblivious***: $\boldsymbol{\ell}_t(a_1, \ell_1(a_1), \ldots, a_{t-1}, \ell_{t-1}(a_{t-1}))$ can depend on past history.

- ***stochastic*** setting: $\boldsymbol{\ell}_t \overset{\text{i.i.d.}}{\sim} \mathcal{D}$, where $\mathcal{D}$ is fixed unknown distribution.

Formulation

At each round $t = 1, 2, \cdots$

(1) the player first picks an arm $a_t \in [K]$ from K candidate arms;

(2) and simultaneously environments pick a loss vector $\boldsymbol{\ell}_t \in [0, 1]^K$;

(3) the player suffers and only observes loss $\ell_t(a_t)$, then updates the model.

Goal: to minimize ***expected regret***

$$\mathbb{E}[\text{Regret}_T] = \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(a_t)\right] - \min_{a \in [K]} \sum_{t=1}^{T} \ell_t(a),$$

where the expectation is taken over the ***randomness of algorithms***.

deterministic algorithms will suffer $\Omega(T)$ regret in the worst case under bandit setting!

Comparison

Full-Information Problem	Domain	Loss Functions	Feedback
Prediction with Experts' Advice	Δ_d	$f_t(\boldsymbol{p}_t) = \langle \boldsymbol{\ell}_t, \boldsymbol{p}_t \rangle$	$f_t(\boldsymbol{p}_t),\ \boldsymbol{\ell}_t$
Online Convex Optimization	$\mathcal{X}$	$f_t(\cdot)$	$f_t(\mathbf{x}_t),\ \nabla f_t(\mathbf{x}_t), \ldots$

Bandit Problem	Domain	Loss Functions	Feedback
Multi-Armed Bandits	$\{\mathbf{e}_1, \ldots, \mathbf{e}_K\}$	$f_t(\mathbf{e}_{a_t}) = \langle \boldsymbol{\ell}_t, \mathbf{e}_{a_t} \rangle$	$f_t(\mathbf{e}_{a_t}) = \ell_t(a_t)$
Bandit Convex Optimization	$\mathcal{X}$	$f_t(\cdot)$	$f_t(\mathbf{x}_t)$

Notation: $\mathbf{e}_i \in \mathbb{R}^K$ is the one-hot vector, with i-th entry being 1.

(simplex is the convex hull of $\{\mathbf{e}_1, \ldots, \mathbf{e}_K\}$)

Comparison

Full-Information Problem	Domain	Loss Functions	Feedback
Prediction with Experts' Advice	Δ_d	$f_t(\boldsymbol{p}_t) = \langle \boldsymbol{\ell}_t, \boldsymbol{p}_t \rangle$	$f_t(\boldsymbol{p}_t),\ \boldsymbol{\ell}_t$
Online Convex Optimization	$\mathcal{X}$	$f_t(\cdot)$	$f_t(\mathbf{x}_t),\ \nabla f_t(\mathbf{x}_t), \ldots$

Bandit Problem	Domain	Loss Functions	Feedback
Multi-Armed Bandits	$\{\mathbf{e}_1, \ldots, \mathbf{e}_K\}$	$f_t(\mathbf{e}_{a_t}) = \langle \boldsymbol{\ell}_t, \mathbf{e}_{a_t} \rangle$	$f_t(\mathbf{e}_{a_t}) = \ell_t(a_t)$
Bandit Convex Optimization	$\mathcal{X}$	$f_t(\cdot)$	$f_t(\mathbf{x}_t)$

Notation: $\mathbf{e}_i \in \mathbb{R}^K$ is the one-hot vector, with i-th entry being 1.

(simplex is the convex hull of $\{\mathbf{e}_1, \ldots, \mathbf{e}_K\}$)

A Natural Solution for MAB

- MAB bares much similarity with the PEA problem (except for the amount feedback information).

⇨ Deploying Hedge to MAB problem.

Hedge for PEA

At each round $t = 1, 2, \cdots$

(1) compute $\boldsymbol{p}_t \in \Delta_K$ such that $p_t(i) \propto \exp\left(-\eta L_{t-1}(i)\right)$ for $i \in [K]$

(2) the player submits $\boldsymbol{p}_t$, suffers loss $\langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle$, and observes loss $\boldsymbol{\ell}_t \in \mathbb{R}^K$

(3) update $L_t = L_{t-1} + \boldsymbol{\ell}_t$

A Natural Solution for MAB

- However, Hedge does not fit for MAB setting due to *limited feedback*.

Hedge requires $\ell_t(i)$ for all $i \in [K]$, but only $\ell_t(a_t)$ is available in MAB.

Hedge for PEA

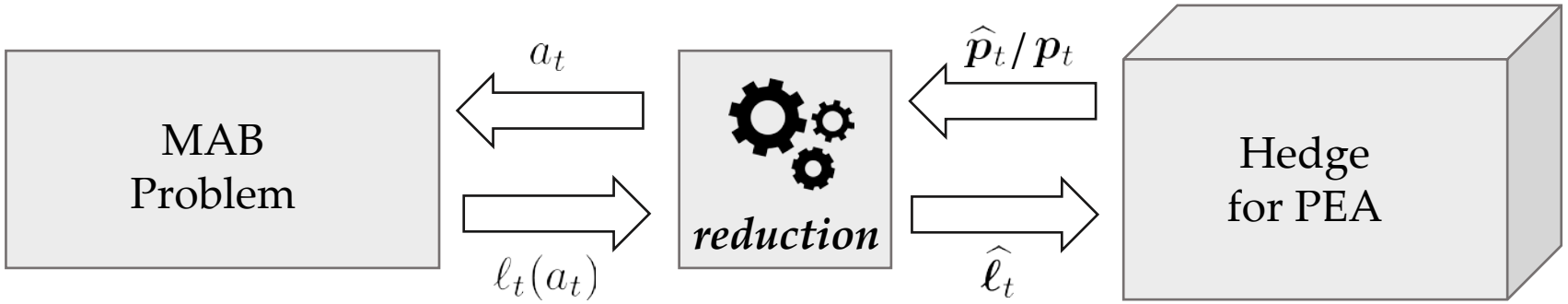
At each round $t = 1, 2, \cdots$

(1) compute $\boldsymbol{p}_t \in \Delta_K$ such that $p_t(i) \propto \exp\left(-\eta L_{t-1}(i)\right)$ for $i \in [K]$

(2) the player submits $\boldsymbol{p}_t$, suffers loss $\langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle$, and observes loss $\boldsymbol{\ell}_t \in \mathbb{R}^K$

(3) update $L_t = L_{t-1} + \boldsymbol{\ell}_t$

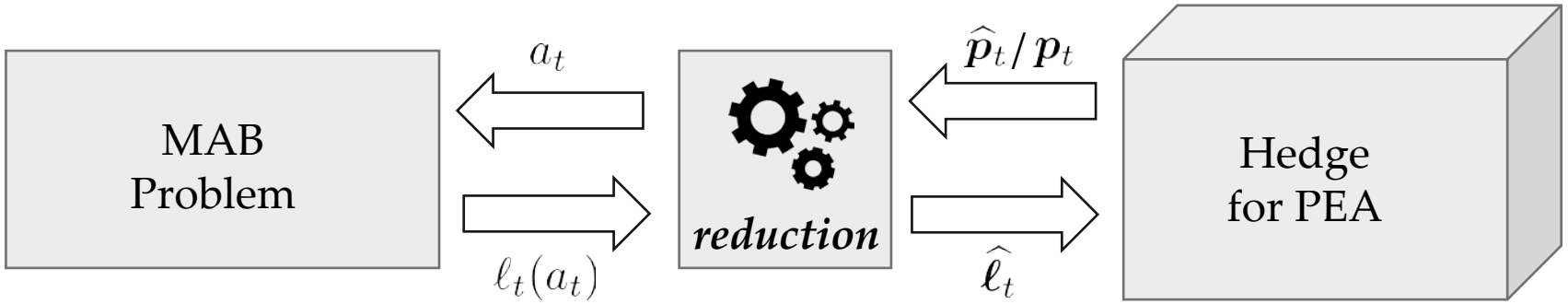
Reduction for MAB



$$\mathrm{Regret}_T^{\mathrm{MAB}} \overset{\text{by reduction}}{\sim} \mathrm{Regret}_T^{\mathrm{PEA}} = \sum_{t=1}^{T} \langle \widehat{\boldsymbol{p}}_t, \widehat{\boldsymbol{\ell}}_t \rangle - \sum_{t=1}^{T} \widehat{\ell}_t(i) \leq \mathcal{O}(\sqrt{T})$$

- $\boldsymbol{p}_t \in \Delta_K$ denotes the distribution over arms — *sampling an arm* $a_t \sim \boldsymbol{p}_t$
- $\widehat{\boldsymbol{\ell}}_t \in \mathbb{R}_+^K$ is the estimated loss fed to Hedge — *how to construct loss estimator?*

Loss Estimator



Idea: ensure $\ell_t(a_t) = \langle \boldsymbol{p}_t, \widehat{\boldsymbol{\ell}}_t \rangle$ in order to re-use Hedge's regret guarantee

Importance-Weighted (IW) Loss Estimator

$$\widehat{\ell}_t(i) = \frac{\ell_t(a_t)}{p_t(i)} \mathbb{1}\{i = a_t\} = \begin{cases} \frac{\ell_t(a_t)}{p_t(a_t)} & \text{if } i = a_t; \\ 0 & \text{else.} \end{cases}$$

Loss Estimator

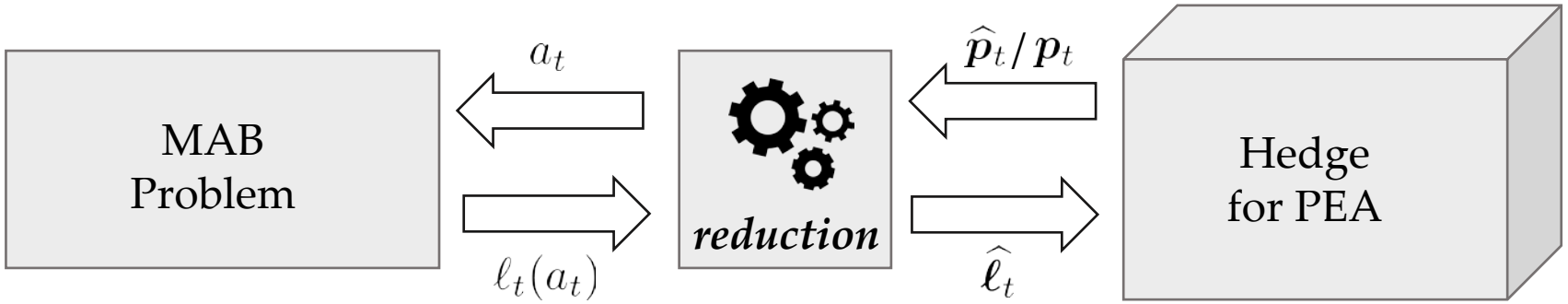
IW Loss Estimator $$\widehat{\ell}_t(i) = \frac{\ell_t(a_t)}{p_t(i)} \mathbb{1}\{i = a_t\} = \begin{cases} \frac{\ell_t(a_t)}{p_t(a_t)} & \text{if } i = a_t; \\ 0 & \text{else.} \end{cases}$$

- Property (1). $\ell_t(a_t) = \langle \boldsymbol{p}_t, \widehat{\boldsymbol{\ell}}_t \rangle$
- Property (2). $\mathbb{E}_{a_t \sim \boldsymbol{p}_t}[\widehat{\ell}_t(i)] = \ell_t(i), \ \forall i \in [K]$ ***unbiasedness***

Proof. $$\mathbb{E}_{a_t \sim \boldsymbol{p}_t}[\widehat{\ell}_t(i)] = \mathbb{E}_{a_t \sim \boldsymbol{p}_t}\left[\frac{\ell_t(a_t)}{p_t(i)} \mathbb{1}\{i = a_t\}\right] = \mathbb{E}_{a_t \sim \boldsymbol{p}_t}\left[\frac{\ell_t(i)}{p_t(i)} \mathbb{1}\{i = a_t\}\right]$$

$$= \frac{\ell_t(i)}{p_t(i)} \mathbb{E}_{a_t \sim \boldsymbol{p}_t}[\mathbb{1}\{i = a_t\}] = \frac{\ell_t(i)}{p_t(i)} p_t(i) = \ell_t(i). \qquad \square$$

Other Choice



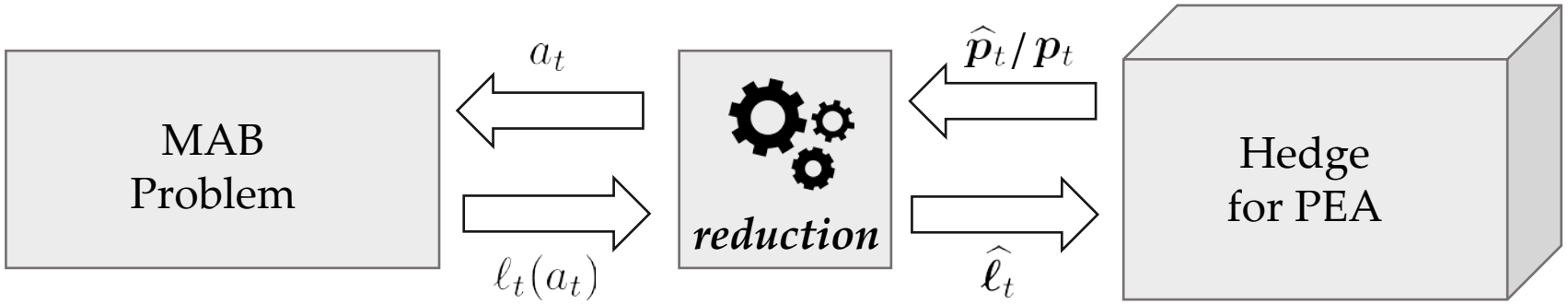
- Other estimators coming to mind,

$$\widehat{\boldsymbol{\ell}}_t = [0, \ldots, 0, \underbrace{\ell_t(a_t)}_{a_t\text{-th entry}}, 0, \ldots, 0]^\top$$

$$\Longrightarrow \quad \underbrace{\langle \boldsymbol{p}_t, \widehat{\boldsymbol{\ell}}_t \rangle}_{\text{loss for Hedge}} = p_t(a_t)\ell_t(a_t) \neq \underbrace{\ell_t(a_t)}_{\text{loss for MAB}}$$

cannot apply Hedge

Importance-Weighted Loss Estimator



- Importance weighting estimator,

$$\widehat{\boldsymbol{\ell}}_t = [0, \ldots, 0, \underbrace{\frac{\ell_t(a_t)}{p_t(a_t)}}_{a_t\text{-th entry}}, 0, \ldots, 0]^\top$$

$\Rightarrow$ balancing exploitation $\ell_t(a_t)$ and exploration $p_t(a_t)$

Exp3 Algorithm

Exp3 (Exponential-weight for Exploration and Exploitation)

At each round $t = 1, 2, \cdots$

(1) compute $\boldsymbol{p}_t \in \Delta_K$ such that $p_t(i) \propto \exp\left(-\eta \widehat{L}_{t-1}(i)\right)$ for $i \in [K]$

(2) chooses $a_t \sim \boldsymbol{p}_t$, suffers and observe loss $\ell_t(a_t)$, and construct loss estimator $\widehat{\boldsymbol{\ell}}_t \in \mathbb{R}^K$ as

$$\widehat{\ell}_t(i) = \frac{\ell_t(a_t)}{p_t(i)} \mathbb{1}\{i = a_t\} = \begin{cases} \frac{\ell_t(a_t)}{p_t(a_t)} & \text{if } i = a_t; \\ 0 & \text{else.} \end{cases}$$

(3) update $\widehat{L}_t = \widehat{L}_{t-1} + \widehat{\boldsymbol{\ell}}_t$

Exp3: Regret Bound

Theorem 1. *Suppose that* $\forall t \in [T]$ *and* $i \in [K], 0 \le \ell_t(i) \le 1$, *then Exp3 with learning rate* $\eta = \sqrt{(\ln K)/(TK)}$ *guarantees*

$$\mathbb{E}[\text{Regret}_T] = \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(a_t)\right] - \min_{a\in[K]} \sum_{t=1}^{T} \ell_t(a) \le \mathcal{O}\left(\sqrt{TK \ln K}\right),$$

where the expectation is taken over the randomness of the algorithm.

Comparision:

Hedge for PEA	***Exp3 for MAB***
full-information feedback	bandit feedback
$\text{Regret}_T \le \mathcal{O}(\sqrt{T \ln K})$	$\mathbb{E}[\text{Regret}_T] \le \mathcal{O}(\sqrt{TK \ln K})$

suffer a larger arm dependence

Proof of Exp3 Regret Bound

Proof.

Recall that (Lecture 7), Hedge under PEA setting guarantees,

$$\sum_{t=1}^{T}\langle \boldsymbol{p}_t, \widehat{\boldsymbol{\ell}}_t\rangle - \sum_{t=1}^{T}\widehat{\ell}_t(i) \le \frac{\ln K}{\eta} + \eta\sum_{t=1}^{T}\sum_{i=1}^{K} p_t(i)\left(\widehat{\ell}_t(i)\right)^2,\ \forall i\in[K]$$

(potential-based analysis allows $\widehat{\boldsymbol{\ell}}_t \in \mathbb{R}_+^K$.)

Note that our previous reduction ensures $\ell_t(a_t) = \langle \boldsymbol{p}_t, \widehat{\boldsymbol{\ell}}_t\rangle$,

$$\Rightarrow \quad \sum_{t=1}^{T}\ell_t(a_t) - \sum_{t=1}^{T}\widehat{\ell}_t(i) \le \frac{\ln K}{\eta} + \eta\sum_{t=1}^{T}\sum_{i=1}^{K} p_t(i)\left(\widehat{\ell}_t(i)\right)^2$$

Proof of Exp3 Regret Bound

Proof. $$\sum_{t=1}^{T} \ell_t(a_t) - \sum_{t=1}^{T} \widehat{\ell}_t(i) \leq \frac{\ln K}{\eta} + \eta \sum_{t=1}^{T} \sum_{i=1}^{K} p_t(i) \left(\widehat{\ell}_t(i)\right)^2$$

Denoted by $\mathbb{E}_t[\,\cdot\,] \triangleq \mathbb{E}_{a_t \sim \boldsymbol{p}_t}[\,\cdot\,|a_1, \ldots, a_{t-1}]$ the conditional expectation given past actions $a_1, \ldots, a_{t-1}$.

We have,

$$\mathbb{E}_t\left[\left(\widehat{\ell}_t(i)\right)^2\right] = \mathbb{E}_t\left[\left(\frac{\ell_t(a_t)}{p_t(i)}\right)^2 \mathbb{1}\{i = a_t\}\right] = \mathbb{E}_t\left[\left(\frac{\ell_t(i)}{p_t(i)}\right)^2 \mathbb{1}\{i = a_t\}\right]$$

$$= \left(\frac{\ell_t(i)}{p_t(i)}\right)^2 \mathbb{E}_t\left[\mathbb{1}\{i = a_t\}\right] = \frac{\left(\ell_t(i)\right)^2}{p_t(i)}.$$

Proof of Exp3 Regret Bound

Proof. $$\sum_{t=1}^{T} \ell_t(a_t) - \sum_{t=1}^{T} \widehat{\ell}_t(i) \leq \frac{\ln K}{\eta} + \eta \sum_{t=1}^{T} \sum_{i=1}^{K} p_t(i) \left(\widehat{\ell}_t(i)\right)^2$$

By the Law of total expectation and the above inequality,

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(a_t) - \sum_{t=1}^{T} \widehat{\ell}_t(i)\right] = \sum_{t=1}^{T} \mathbb{E}\left[\mathbb{E}_t\left[\ell_t(a_t) - \widehat{\ell}_t(i)\right]\right] = \sum_{t=1}^{T} \mathbb{E}\left[\mathbb{E}_t\left[\ell_t(a_t)\right] - \ell_t(i)\right] \quad (\mathbb{E}_t[\widehat{\ell}_t(i)] = \ell_t(i))$$

$$= \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(a_t)\right] - \sum_{t=1}^{T} \ell_t(i)$$

$$\leq \frac{\ln K}{\eta} + \eta \sum_{t=1}^{T} \sum_{i=1}^{K} \mathbb{E}\left[\mathbb{E}_t\left[p_t(i) \cdot \left(\widehat{\ell}_t(i)\right)^2\right]\right]$$

regret bound

Proof of Exp3 Regret Bound

Proof.

$$\mathbb{E}\left[\sum_{t=1}^{T}\ell_t(a_t)\right]-\sum_{t=1}^{T}\ell_t(i)$$

$$\leq \frac{\ln K}{\eta}+\eta\sum_{t=1}^{T}\sum_{i=1}^{K}\mathbb{E}\left[\mathbb{E}_t\left[p_t(i)\cdot\left(\widehat{\ell}_t(i)\right)^2\right]\right]$$

$$= \frac{\ln K}{\eta}+\eta\sum_{t=1}^{T}\sum_{i=1}^{K}p_t(i)\cdot\frac{\ell_t(i)^2}{p_t(i)} \qquad \left(\mathbb{E}_t\left[p_t(i)\cdot\widehat{\ell}_t(i)^2\right]=p_t(i)\cdot\mathbb{E}_t\left[\widehat{\ell}_t(i)^2\right]=p_t(i)\cdot\tfrac{\ell_t(i)^2}{p_t(i)}\right)$$

$$= \frac{\ln K}{\eta}+\eta\sum_{t=1}^{T}\sum_{i=1}^{K}\ell_t(i)^2$$

$$\leq \frac{\ln K}{\eta}+\eta TK\leq\mathcal{O}(\sqrt{TK\ln K}) \qquad \left(\ell_t(i)^2\leq 1,\eta=\sqrt{(\ln K)/TK}\right)$$

□

Lower Bound for MAB

- As above, we have proved the regret upper bound for Exp3:

$$\mathbb{E}\left[\mathrm{Regret}_T\right] \leq \mathcal{O}\left(\sqrt{TK\ln K}\right)$$

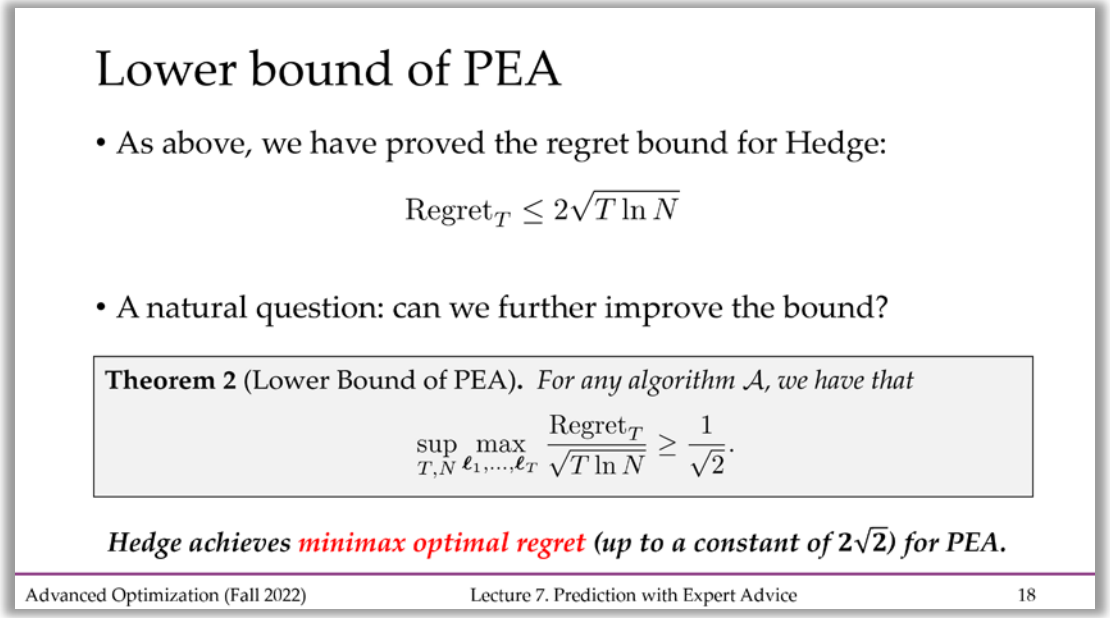
- A natural question: can we further improve the bound?

  Maybe? Exp3 doesn't achieve minimax optimal regret for MAB.

Lower Bound for MAB

Theorem 2 (Lower Bound for MAB). *For any algorithm $\mathcal{A}$, there exists a sequence of loss vectors $\ell_1, \ell_2, \cdots, \ell_T$ constituing an MAB problem such that*

$$\inf_{\mathcal{A}} \sup_{\ell_1,\ldots,\ell_T} \mathbb{E}\left[\text{Regret}_T\right] = \Omega(\sqrt{TK})$$



Lower bound of PEA

- As above, we have proved the regret bound for Hedge:

$$\text{Regret}_T \le 2\sqrt{T \ln N}$$

- A natural question: can we further improve the bound?

Theorem 2 (Lower Bound of PEA). *For any algorithm $\mathcal{A}$, we have that*

$$\sup_{T,N} \max_{\ell_1,\ldots,\ell_T} \frac{\text{Regret}_T}{\sqrt{T \ln N}} \ge \frac{1}{\sqrt{2}}.$$

Hedge achieves minimax optimal regret (up to a constant of $2\sqrt{2}$) for PEA.

Advanced Optimization (Fall 2022) Lecture 7. Prediction with Expert Advice 18

MAB Problem $\Omega(\sqrt{TK})$

PEA Problem $\Omega(\sqrt{T \ln K})$

Proof Sketch

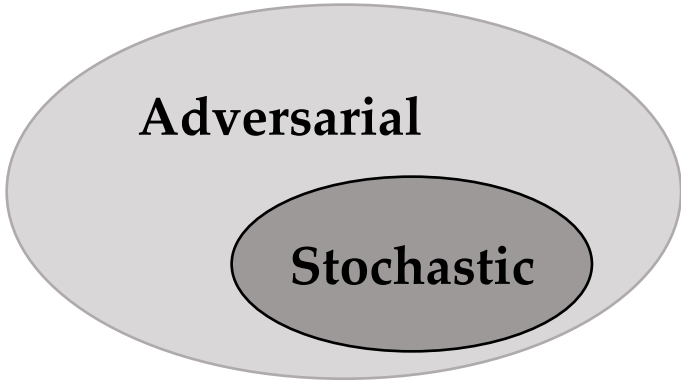
Proof (Sketch).

We prove the theorem under stochastic MAB setting, since the stochastic setting is strictly easier than the adversarial one.

We construct two hard distributions over arms $\mathcal{D}_1, \mathcal{D}_2$ and show that,

$\forall A \in \mathcal{A}$, the following holds

$$\max\left\{\mathbb{E}[\mathrm{Regret_T(A)}; \mathcal{D}_1], \mathbb{E}[\mathrm{Regret_T(A)}; \mathcal{D}_2]\right\} = \Omega(\sqrt{TK})$$



Proof Sketch

- For $\mathcal{D}_1$, each arm's loss is drawn from Gaussian distribution,

$$\ell_t(i) \sim \mathcal{N}(\mu_i, 1), \mu_i = \begin{cases} 1-\Delta & \text{if } i = 1 \\ 1 & \text{else.} \end{cases}$$

  where the first arm is better than others by Δ to be tuned.

- Denoted by $\mathbb{E}[n^{\mathcal{D}_1}(i)]$ the expected times choosing arm i under distribution $\mathcal{D}_1$ and $i_{\min} = \arg\min_{i \in [K]} \mathbb{E}[n^{\mathcal{D}_1}(i)]$ index of the least-played arm.

Proof Sketch

For $\mathcal{D}_1$,

$$\ell_t(i) \sim \mathcal{N}(\mu_i, 1), \mu_i = \begin{cases} 1 - \Delta & \text{if } i = 1 \\ 1 & \text{else.} \end{cases}$$

For $\mathcal{D}_2$, each arm's loss is drawn from Gaussian distribution,

$$\ell_t(i) \sim \mathcal{N}(\mu_i', 1), \mu_i' = \begin{cases} 1 - 2\Delta & \text{if } i = i_{\min} \\ \mu_i & \text{else.} \end{cases}$$

Proof Sketch

For $\mathcal{D}_1$, $\quad \ell_t(i) \sim \mathcal{N}(\mu_i, 1), \mu_i = \begin{cases} 1-\Delta & \text{if } i = 1 \\ 1 & \text{else.} \end{cases}$

For $\mathcal{D}_2$, $\quad \ell_t(i) \sim \mathcal{N}(\mu_i', 1), \mu_i' = \begin{cases} 1-2\Delta & \text{if } i = i_{\min} \\ \mu_i & \text{else.} \end{cases}$

We have

$$\mathbb{E}[\mathrm{Regret_T(A)}; \mathcal{D}_1] = \Delta \cdot (T - \mathbb{E}[n^{\mathcal{D}_1}(1)])$$

$$\mathbb{E}[\mathrm{Regret_T(A)}; \mathcal{D}_2] = \Delta \cdot \mathbb{E}[n^{\mathcal{D}_2}(1)] + \sum_{j \neq 1, i_{\min}} 2\Delta \cdot \mathbb{E}[n^{\mathcal{D}_2}(j)] \geq \Delta \cdot \mathbb{E}[n^{\mathcal{D}_2}(1)]$$

Proof Sketch

$$\mathbb{E}[\text{Regret}_\text{T}(\text{A}); \mathcal{D}_1] = \Delta \cdot (T - \mathbb{E}[n^{\mathcal{D}_1}(1)])$$

$$\mathbb{E}[\text{Regret}_\text{T}(\text{A}); \mathcal{D}_2] \geq \Delta \cdot \mathbb{E}[n^{\mathcal{D}_2}(1)]$$

Taking $\Delta = \sqrt{(K-1)/T}$ yields the following result.

- If $\mathbb{E}[n^{\mathcal{D}_1}(1)] < T/2$,

$$\mathbb{E}[\text{Regret}_\text{T}(\text{A}); \mathcal{D}_1] = \Delta \cdot (T - \mathbb{E}[n^{\mathcal{D}_1}(1)]) \geq \frac{T}{2} \cdot \sqrt{\frac{K-1}{T}} = \frac{1}{2}\sqrt{T(K-1)}$$

- If $\mathbb{E}[n^{\mathcal{D}_1}(1)] \geq T/2$, we assume $\mathbb{E}[n^{\mathcal{D}_2}(1)] \approx \mathbb{E}[n^{\mathcal{D}_1}(1)]$

$$\mathbb{E}[\text{Regret}_\text{T}(\text{A}); \mathcal{D}_2] = \Delta \cdot \mathbb{E}[n^{\mathcal{D}_2}(1)] \geq \frac{T}{2} \cdot \sqrt{\frac{K-1}{T}} = \frac{1}{2}\sqrt{T(K-1)}$$

Proof Sketch

Why can we assume $\mathbb{E}[n^{\mathcal{D}_2}(1)] \approx \mathbb{E}[n^{\mathcal{D}_1}(1)]$?

- The two distributions are similar except for one arm, thus intuitively the algorithm cannot distinguish them.

- The proof reduces to ***hypothesis testing***, i.e. the minimum costs to distinguish two distributions, and $\Delta = \sqrt{(K-1)/T}$ helps to make the difference not so big.

□

Upper and Lower Bounds for MAB

Theorem 1 (Upper Bound for Exp3). *Suppose that* $\forall t \in [T]$ *and* $i \in [K]$, $0 \le \ell_t(i) \le 1$, *then Exp3 with learning rate* $\eta = \sqrt{(\ln K)/(TK)}$ *guarantees*

$$\mathbb{E}[\text{Regret}_T] = \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(a_t)\right] - \min_{a \in [K]} \sum_{t=1}^{T} \ell_t(a) \le \mathcal{O}\left(\sqrt{TK \ln K}\right),$$

where the expectation is taken over the randomness of the algorithm.

Theorem 2 (Lower Bound for MAB). *For any algorithm* $\mathcal{A}$, *there exists a sequence of loss vectors* $\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \cdots, \boldsymbol{\ell}_T$ *constituing an MAB problem such that*

$$\inf_{\mathcal{A}} \sup_{\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_T} \mathbb{E}\left[\text{Regret}_T\right] = \Omega(\sqrt{TK}).$$

Advanced Topics

- How to shave off the extra $\ln K$ factor?

⇨ Using OMD with ***Tsallis entropy*** regularizer, also using the IW estimator

$$\psi(p) = \frac{1 - \sum_{a=1}^{K} p(a)^\beta}{1 - \beta}$$

which is actually a *generalization* of negative-entropy used in Hedge, as we have the following fact due to the L'Hôpital'srule

$$\lim_{\beta \to 1} \frac{1 - \sum_a p(a)^\beta}{1 - \beta} = \sum_a p(a) \ln(p(a)).$$

Reference: Jean-Yves Audibert and S´ebastien Bubeck. Regret bounds and minimax policies under partial monitoring. Journal of Machine Learning Research, 11(Oct):2785–2836, 2010.

Advanced Topics

- How to boost from expected guarantee to a ***high-probability*** one?

⇨ Using an improved estimator: **Implicit eXploration (IX) Loss Estimator**

IW Loss Estimator

$$\widehat{\ell}_t(i) = \frac{\ell_t(a_t)}{p_t(i)} \mathbb{1}\{i = a_t\} = \begin{cases} \frac{\ell_t(a_t)}{p_t(a_t)} & \text{if } i = a_t; \\ 0 & \text{else.} \end{cases}$$

IX Loss Estimator

$$\widehat{\ell}_t(i) = \frac{\ell_t(a_t)}{p_t(i) + \gamma} \mathbb{1}\{i = a_t\} = \begin{cases} \frac{\ell_t(a_t)}{p_t(a_t) + \gamma} & \text{if } i = a_t; \\ 0 & \text{else.} \end{cases}$$

Reference: Gergely Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. NIPS 2015.

SIAM J. COMPUT.
Vol. 32, No. 1, pp. 48–77

THE NONSTOCHASTIC MULTIARMED BANDIT PROBLEM*

PETER AUER†, NICOLÒ CESA-BIANCHI‡, YOAV FREUND§, AND ROBERT E. SCHAPIRE¶

Abstract. In the multiarmed bandit problem, a gambler must decide which arm of K non-identical slot machines to play in a sequence of trials so as to maximize his reward. This classical problem has received much attention because of the simple model it provides of the trade-off between exploration (trying out each arm to find the best one) and exploitation (playing the arm believed to give the best payoff). Past solutions for the bandit problem have almost always relied on assumptions about the statistics of the slot machines.

In this work, we make no statistical assumptions whatsoever about the nature of the process generating the payoffs of the slot machines. We give a solution to the bandit problem in which an adversary, rather than a well-behaved stochastic process, has complete control over the payoffs. In a sequence of T plays, we prove that the per-round payoff of our algorithm approaches that of the best arm at the rate $O(T^{-1/2})$. We show by a matching lower bound that this is the best possible.

We also prove that our algorithm approaches the per-round payoff of *any* set of strategies at a similar rate: if the best strategy is chosen from a pool of N strategies, then our algorithm approaches the per-round payoff of the strategy at the rate $O((\log N)^{1/2}T^{-1/2})$. Finally, we apply our results to the problem of playing an unknown repeated matrix game. We show that our algorithm approaches the minimax payoff of the unknown game at the rate $O(T^{-1/2})$.

Key words. adversarial bandit problem, unknown matrix games

AMS subject classifications. 68Q32, 68T05, 91A20

PII. S0097539701398375

1. Introduction. In the multiarmed bandit problem, originally proposed by Robbins [17], a gambler must choose which of K slot machines to play. At each time step, he pulls the arm of one of the machines and receives a reward or payoff (possibly zero or negative). The gambler's purpose is to maximize his return, i.e., the sum of the rewards he receives over a sequence of pulls. In this model, each arm is assumed to deliver rewards that are independently drawn from a fixed and unknown distribution. As reward distributions differ from arm to arm, the goal is to find the arm with the highest expected payoff as early as possible and then to keep gambling using that best arm.

The problem is a paradigmatic example of the trade-off between exploration and exploitation. On the one hand, if the gambler plays exclusively on the machine that he thinks is best ("exploitation"), he may fail to discover that one of the other arms actually has a higher expected payoff. On the other hand, if he spends too much time

*Received by the editors November 18, 2001; accepted for publication (in revised form) July 7, 2002; published electronically November 19, 2002. An early extended abstract of this paper appeared in *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, 1995, IEEE Computer Society, pp. 322–331.
http://www.siam.org/journals/sicomp/32-1/39837.html
†Institute for Theoretical Computer Science, Graz University of Technology, A-8010 Graz, Austria (pauer@igi.tu-graz.ac.at). This author gratefully acknowledges the support of ESPRIT Working Group EP 27150, Neural and Computational Learning II (NeuroCOLT II).
‡Department of Information Technology, University of Milan, I-26013 Crema, Italy (cesa-bianchi@dti.unimi.it). This author gratefully acknowledges the support of ESPRIT Working Group EP 27150, Neural and Computational Learning II (NeuroCOLT II).
§Banter Inc. and Hebrew University, Jerusalem, Israel (yoavf@cs.huji.ac.il).
¶AT&T Labs – Research, Shannon Laboratory, Florham Park, NJ 07932-0971 (schapire@research.att.com).

The non-stochastic multi-armed bandit problem*

Peter Auer
Institute for Theoretical Computer Science
Graz University of Technology
A-8010 Graz (Austria)
pauer@igi.tu-graz.ac.at

Nicolò Cesa-Bianchi
Department of Computer Science
Università di Milano
I-20135 Milano (Italy)
cesabian@dsi.unimi.it

Yoav Freund **Robert E. Schapire**
AT&T Labs
180 Park Avenue
Florham Park, NJ 07932-0971
{yoav, schapire}@research.att.com

November 18, 2001

TITLE	CITED BY	YEAR
Finite-time analysis of the multiarmed bandit problem P Auer, N Cesa-Bianchi, P Fischer Machine learning 47 (2), 235-256	6816	2002
The nonstochastic multiarmed bandit problem P Auer, N Cesa-Bianchi, Y Freund, RE Schapire SIAM Journal on Computing 32 (1), 48-77	2519	2003
Using confidence bounds for exploitation-exploration trade-offs P Auer Journal of Machine Learning Research 3 (Nov), 397-422	1787	2002
Near-optimal regret bounds for reinforcement learning T Jaksch, R Ortner, P Auer The Journal of Machine Learning Research 11, 1563-1600	1117 *	2010

The Nonstochastic Multiarmed Bandit Problem. SIAM Journal on Computing (SICOMP). 2002.

Bandit Convex Optimization

Full-Information Problem	Domain	Loss Functions	Feedback
Prediction with Experts' Advice	Δ_d	$f_t(\boldsymbol{p}_t) = \langle \boldsymbol{\ell}_t, \boldsymbol{p}_t \rangle$	$f_t(\boldsymbol{p}_t),\ \boldsymbol{\ell}_t$
Online Convex Optimization	$\mathcal{X}$	$f_t(\cdot)$	$f_t(\mathbf{x}_t),\ \nabla f_t(\mathbf{x}_t), \ldots$

Bandit Problem	Domain	Loss Functions	Feedback
Multi-Armed Bandits	$\{\mathbf{e}_1, \ldots, \mathbf{e}_K\}$	$f_t(\mathbf{e}_{a_t}) = \langle \boldsymbol{\ell}_t, \mathbf{e}_{a_t} \rangle$	$f_t(\mathbf{e}_{a_t}) = \ell_t(a_t)$
Bandit Convex Optimization	$\mathcal{X}$	$f_t(\cdot)$	$f_t(\mathbf{x}_t)$

Formulation

At each round $t = 1, 2, \cdots$

(1) the player first picks a decision $\mathbf{x}_t \in \mathcal{X}$;

(2) and simultaneously environments chooses a loss function $f_t(\cdot) : \mathcal{X} \mapsto \mathbb{R}$;

(3) the player suffers and only observes loss $f_t(\mathbf{x}_t)$, then updates the model.

Goal: to optimize ***expected regret***,

$$\mathbb{E}[\text{Regret}_T] = \mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t)\right] - \min_{\mathbf{u}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{u}),$$

where the expectation is taken over the ***randomness of algorithms***.

A Natural Solution for BCO

- BCO bares much similarity with the OCO problem.

⇨ Deploying OGD to BCO problem.

Online Gradient Descent

At each round $t = 1, 2, \cdots$

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}} \left[\mathbf{x}_t - \eta \nabla f_t(\mathbf{x}_t) \right],$$

where $\Pi_{\mathcal{X}}[\cdot]$ denotes the projection onto the feasible domain $\mathcal{X}$.

We actually don't have the gradient information due to the *limited feedback*.

Gradient Estimator

Definition 1 (Gradient Estimator). The gradient estimator is defined as

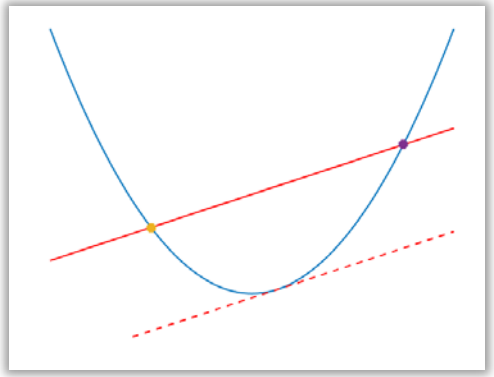
$$\widetilde{\mathbf{g}}_t = \frac{d}{\delta} f_t(\mathbf{y}_t + \delta \mathbf{s}_t) \cdot \mathbf{s}_t$$

where $\mathbf{s}_t$ is sampled from unit sphere $\mathbb{S} = \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|_2 = 1\}$.

- Consider the 1-dim case ($d = 1$).

$$\mathbb{E}_{\mathbf{s}\in\mathbb{S}}\left[\frac{\delta}{d} f(\mathbf{x} + \delta \mathbf{s}) \cdot \mathbf{s}\right] = \frac{1}{2\delta} f(x + \delta) - \frac{1}{2\delta} f(x - \delta)$$

$$\Longrightarrow \lim_{\delta \to 0} \frac{1}{2\delta} f(x + \delta) - \frac{1}{2\delta} f(x - \delta) = f'(x)$$



Gradient Estimator

Definition 1 (Gradient Estimator). The gradient estimator is defined as

$$\widetilde{\mathbf{g}}_t = \frac{d}{\delta} f_t(\mathbf{y}_t + \delta \mathbf{s}_t) \cdot \mathbf{s}_t$$

where $\mathbf{s}_t$ is sampled from unit sphere $\mathbb{S} = \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|_2 = 1\}$.

Lemma 1 *For any convex (but not necessarily differentiable) function $f : \mathcal{X} \mapsto \mathbb{R}$, define its smoothed version $\widehat{f}(\mathbf{x}) = \mathbb{E}_{\mathbf{v} \in \mathbb{B}}[f(\mathbf{v})]$. Then for any $\delta > 0$, we have*

$$\mathbb{E}_{\mathbf{s} \in \mathbb{S}} \left[\frac{d}{\delta} f(\mathbf{x} + \delta \mathbf{s}) \cdot \mathbf{s} \right] = \nabla \widehat{f}(\mathbf{x}),$$

where $\mathbb{B}$ is the unit ball $\mathbb{B} = \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|_2 \leq 1\}$ and $\mathbb{S}$ is the unit sphere.

Bandit Gradient Descent

- Deploy **OGD** to BCO problem using the gradient estimator.

At each round $t = 1, 2, \cdots$

(1) sample a unit vector $\mathbf{s}_t \in \mathbb{S}$;

(2) submit $\mathbf{x}_t = \mathbf{y}_t + \delta \mathbf{s}_t$;

(3) receive feedback $f_t(\mathbf{x}_t)$;

(4) construct gradient estimator $\widetilde{\mathbf{g}}_t = \frac{\delta}{d} f_t(\mathbf{y}_t + \delta \mathbf{s}_t) \cdot \mathbf{s}_t$;

(5) $\mathbf{y}_{t+1} = \Pi_{(1-\alpha)\mathcal{X}}[\mathbf{y}_{t+1} - \eta \widetilde{\mathbf{g}}_t]$.

where $(1-\alpha)\mathcal{X} \triangleq \{\mathbf{x} \in \mathbb{R}^d \mid \frac{1}{1-\alpha}\mathbf{x} \in \mathcal{X}\}$.

Bandit Gradient Descent

Theorem 3 *Assume f_t is L-Lipschitz, $\max_{\mathbf{x}\in\mathcal{X}} |f_t(\mathbf{x})| \le C$, and $r\cdot\mathbb{B} \subseteq \mathcal{X} \subseteq R\cdot\mathbb{B}$. The BGD algorithm satisfies,*

$$\mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t)\right] - \sum_{t=1}^{T} f_t(\mathbf{u}) \le \mathcal{O}\left(\frac{R^2}{\eta} + \frac{\eta d^2 C^2 T}{\delta^2} + \delta\frac{R}{r}LT\right)$$
$$\le \mathcal{O}\left(T^{3/4}\right)$$

by choosing $\eta = \mathcal{O}((R^2/T)^{3/4})$ and $\delta = \eta^{1/3}$.

Proof Sketch

$$\mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t)\right] - \sum_{t=1}^{T} f_t(\mathbf{u}) \text{ (for simplicity we assume } \mathbf{u} \in (1-\alpha)\mathcal{X})$$

: exploitation cost

: exploration cost

$$= \mathbb{E}\left[\sum_{t=1}^{T} \widehat{f}_t(\mathbf{y}_t) - \widehat{f}_t(\mathbf{u})\right] + \mathbb{E}\left[\sum_{t=1}^{T} \widehat{f}_t(\mathbf{x}_t) - \widehat{f}_t(\mathbf{y}_t)\right] + \mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \widehat{f}_t(\mathbf{x}_t)\right] + \mathbb{E}\left[\sum_{t=1}^{T} \widehat{f}_t(\mathbf{u}) - f_t(\mathbf{u})\right]$$

$$\leq \mathcal{O}\left(\frac{\eta T}{\delta^2} + \frac{1}{\eta}\right) + \mathcal{O}(\delta T) + \mathcal{O}(\delta T) + \mathcal{O}(\delta T)$$

Beyond

- Can we further improve the dependence on T?

⇨ If loss function is ***linear***, then using FTRL with self-concordant barrier on $\mathcal{X}$

$$\mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t)\right] - \sum_{t=1}^{T} f_t(\mathbf{u}) \leq \widetilde{\mathcal{O}}\left(T^{1/2}\right)$$

⇨ If loss function is ***smooth***, then use FTRL with self-concordant barrier on $\mathcal{X}$

$$\mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t)\right] - \sum_{t=1}^{T} f_t(\mathbf{u}) \leq \widetilde{\mathcal{O}}\left(T^{2/3}\right)$$

Online convex optimization in the bandit setting: gradient descent without a gradient

Abraham D. Flaxman * Adam Tauman Kalai † H. Brendan McMahan ‡

November 30, 2004

Abstract

We study a general online convex optimization problem. We have a convex set S and an unknown sequence of cost functions $c_1, c_2, \ldots$, and in each period, we choose a feasible point x_t in S, and learn the cost $c_t(x_t)$. If the function c_t is also revealed after each period then, as Zinkevich shows in [25], gradient descent can be used on these functions to get regret bounds of $O(\sqrt{n})$. That is, after n rounds, the total cost incurred will be $O(\sqrt{n})$ more than the cost of the best single feasible decision chosen with the benefit of hindsight, $\min_x \sum c_t(x)$.

We extend this to the "bandit" setting, where, in each period, only the cost $c_t(x_t)$ is revealed, and bound the expected regret as $O(n^{3/4})$.

Our approach uses a simple approximation of the gradient that is computed from evaluating c_t at a single (random) point. We show that this biased estimate is sufficient to approximate gradient descent on the sequence of functions. In other words, it is possible to use gradient descent without seeing anything more than the value of the functions at a single point. The guarantees hold even in the most general case: online against an adaptive adversary.

For the online linear optimization problem [15], algorithms with low regrets in the bandit setting have recently been given against oblivious [1] and adaptive adversaries [19]. In contrast to these algorithms, which distinguish between explicit *explore* and *exploit* periods, our algorithm can be interpreted as doing a small amount of exploration in each period.

1 Introduction

Consider three optimization settings where one would like to minimize a convex function (equivalently maximize a concave function). In all three settings, gradient descent is one of the most popular methods.

1. Offline: Minimize a fixed convex cost function $c\colon \mathbb{R}^d \to \mathbb{R}$. In this case, gradient descent is $x_{t+1} = x_t - \eta \nabla c(x_t)$.
2. Stochastic: Minimize a fixed convex cost function c given only "noisy" access to c. For example, at time $T = t$, we may only have access to $c_t(x) = c(x) + \epsilon_t(x)$, where $\epsilon_t(x)$ is a random sampling error. Here, stochastic gradient descent is $x_{t+1} = x_t - \eta \nabla c_t(x_t)$. (The intuition is that the expected gradient is correct; $\mathbf{E}[\nabla c_t(x)] = \nabla \mathbf{E}[c_t(x)] = \nabla c(x)$.) In non-convex cases, the additional randomness may actually help avoid local minima [3], in a manner similar to Simulated Annealing [13].
3. Online: Minimize an adversarially generated sequence of convex functions, $c_1, c_2, \ldots$. This requires that we choose a sequence $x_1, x_2, \ldots$ where each x_t is selected based only on $x_1, x_2, \ldots, x_{t-1}$ and $c_1, c_2, \ldots, c_{t-1}$. The goals is to have low *regret* $\sum c_t(x_t) - \min_{x \in S} \sum c_t(x)$ for not using the best single point, chosen with the benefit of hindsight. In this setting, Zinkevich analyzes the regret of gradient descent given by $x_{t+1} = x_t - \eta \nabla c_t(x_t)$.

We will focus on gradient descent in a "bandit" version of the online setting. As a motivating example, consider a company that has to decide, every week, how much to spend advertising on each of d different channels, represented as a vector $x_t \in \mathbb{R}^d$. At the end of each week, they calculate their total profit $p_t(x_t)$. In the offline case, one might assume that each week the function $p_1, p_2, \ldots$ are identical. In the stochastic case, one might assume that in different weeks the profit functions $p_t(x)$ will be noisy realizations of some underlying "true" profit function, for example $p_t(x) = p(x) + \epsilon_t(x)$, where $\epsilon_t(x)$ has mean 0. In the online case, *no assumptions* are made about a distribution over convex profit functions and instead they are modeled as the malicious choices of an adversary. This allows, for example, for more complicated time-dependent random noise or the effects of a bad economy, or even an environment that responds to the choices we make (an adaptive adversary).

*http://www.math.cmu.edu/~adf, Department of Mathematical Sciences, Carnegie Mellon University.

†http://people.cs.uchicago.edu/~kalai, Toyota Technical Institute at Chicago.

‡http://www.cs.cmu.edu/~mcmahan, Department of Computer Science, Carnegie Mellon University.

Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. SODA, 2004.

Competing in the Dark: An Efficient Algorithm for Bandit Linear Optimization

Jacob Abernethy
Computer Science Division
UC Berkeley
jake@cs.berkeley.edu
(eligible for best student paper award)

Elad Hazan
IBM Almaden
hazan@us.ibm.com

Alexander Rakhlin
Computer Science Division
UC Berkeley
rakhlin@cs.berkeley.edu

Abstract

We introduce an *efficient* algorithm for the problem of online linear optimization in the bandit setting which achieves the optimal $O^*(\sqrt{T})$ regret. The setting is a natural generalization of the non-stochastic multi-armed bandit problem, and the existence of an efficient optimal algorithm has been posed as an open problem in a number of recent papers. We show how the difficulties encountered by previous approaches are overcome by the use of a self-concordant potential function. Our approach presents a novel connection between online learning and interior point methods.

1 Introduction

One's ability to learn and make decisions rests heavily on the availability of feedback. Indeed, an agent may only improve itself when it can reflect on the outcomes of its own taken actions. In many environments feedback is readily available: a gambler, for example, can observe entirely the outcome of a horse race regardless of where he placed his bet. But such perspective is not always available in hindsight. When the same gambler chooses his route to travel to the race track, perhaps at a busy hour, he will likely never learn the outcome of possible alternatives. When betting on horses, the gambler has thus the benefit (or perhaps the detriment) to muse *"I should have done..."*, yet when betting on traffic he can only think *"the result was..."*.

This problem of sequential decision making was stated by Robbins [19] in 1952 and was later termed "the multi-armed bandit problem". The name inherits from the model whereby, on each of a sequence of rounds, a gambler must pull the arm on one of several slot machines ("one-armed bandits") that each returns a reward chosen stochastically from a fixed distribution. Of course, an ideal strategy would simply be to pull the arm of the machine with the greatest rewards. However, as the gambler does not know the best arm a priori, his goal is then to maximize the reward of his strategy relative to reward he would receive had he known the optimal arm. This problem has gained much interest over the past 20 years in a number of fields, as it presents a very natural model of an agent seeking to simultaneously explore the world while exploiting high-reward actions.

As early as 1990 [8, 13] the sequential decision problem was studied under *adversarial* assumptions, where we assume the environment may even try to hurt the learner. The multi-armed bandit problem was brought into the adversarial learning model in 2002 by Auer et al [1], who showed that one may obtain nontrivial guarantees on the gambler's performance relative to the best arm *even when the arm values are chosen by adversary*! In particular, Auer et al [1] showed that the gambler's *regret*, i.e. the difference between the gain of the best arm minus the gain of the gambler, can be bounded by $O(\sqrt{NT})$ where N is the number of bandit arms, and T is the length of the game. In comparison to the game where the gambler is given full information about alternative arms (such as the horse racing example mentioned above), it is possible to obtain $O(\sqrt{T \log N})$, which scales better in N but identically in T.

One natural and well studied problem which escapes the Auer et al result, is online shortest path. In this problem the decision set is exponentially large (i.e. set of all paths in a given graph), and the straightforward reduction of modeling each path as an arm for the multi-armed bandit problem suffers from both efficiency issues as well as exponential regret. To cope with these issues, several authors [2, 9, 14] have recently proposed a very natural generalization of the multi-armed bandit problem to field of Convex Optimization, and we will call this "bandit linear optimization". In this setting we imagine that, on each round t, an adversary chooses some linear function $f_t(\cdot)$ which is not revealed to the player. The player then chooses a point $\mathbf{x}_t$ within some given convex set[1] $\mathcal{K} \subset \mathbb{R}^n$. The player then suffers $f_t(\mathbf{x}_t)$ and this quantity is revealed to him. This process continues for T rounds, and at the end the learner's payoff is his *regret*:

$$R_T = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}^* \in \mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}^*).$$

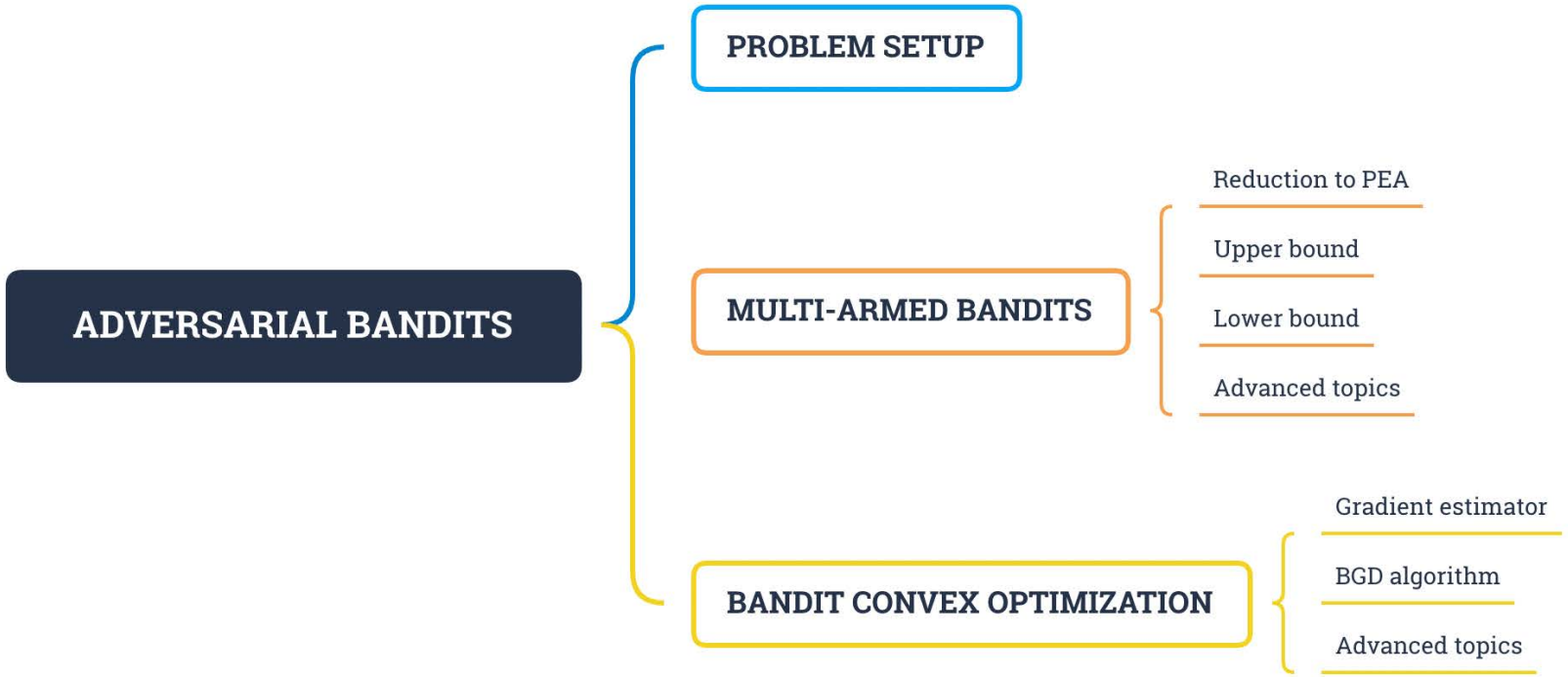
Online linear optimization has been often considered, yet primarily in the full-information setting where the learner sees all of $f_t(\cdot)$ rather than just $f_t(\mathbf{x}_t)$. In the full-information model, it has been known for some time that the optimal regret bound is $O(\sqrt{T})$, and it had been conjectured that the same should hold for the bandit setting as well. Nevertheless, several initially proposed algorithms were shown only

[1] In the case of online shortest path, the convex set can be represented as a set of vectors in $\mathbb{R}^{|E|}$. Hence, the dependence on number of paths in the graph can be circumvented.

COLT 2008
best paper award

Jacob Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. COLT, 2008.

Summary



Q & A

Thanks!