

Corralling a Band of Bandit Algorithms

Alekh Agarwal¹, **Haipeng Luo**¹, Behnam Neyshabur², Rob Schapire¹

¹Microsoft Research, New York

²Toyota Technological Institute at Chicago

Contextual Bandits in Practice

Personalized news recommendation in **MSN**:

The screenshot displays the MSN homepage with several personalized news recommendations. At the top left, a large banner features a smartphone displaying a Pokémon GO update announcement: "Pokémon GO announced its biggest update yet, including 80 new Pokémon | ▶". Below this, a weather widget for Redmond, Washington, shows forecasts for Wednesday (53°/50°), Thursday (52°/43°), and Friday (52°/42°). The main content area is a grid of news cards:

- Top Right:** "Why William and Kate rarely hold hands" with a photo of the royal couple.
- Middle Right:** "Wildfire warns visitors to Yosemite's El Capitan" with a photo of a forest fire.
- Bottom Left:** "American Girl's first-ever boy doll gets mixed in controversy" with a photo of a child.
- Bottom Middle:** "Police: 2 bodies found in Indiana are missing girls" with a photo of a woman.
- Bottom Right:** "Winter's 'biggest storm' expected to unleash California floods" with a photo of a storm.
- Far Bottom Left:** "Ashton Kutcher gives emotional testimony at Senate hearing" with a photo of a man.
- Far Bottom Right:** "'Morning Joe' bans Trump aide Kellyanne Conway" with a photo of a woman.

Contextual Bandits in Practice

Personalized news recommendation in **MSN**:

The screenshot displays the MSN news homepage with several personalized recommendations. At the top left, a large banner features a smartphone displaying a Pokémon GO update, with the text: "Pokémon GO announced its biggest update yet, including 80 new Pokémon | ▶". Below this is a weather widget for Redmond, Washington, showing forecasts for Wednesday (53°/50°), Thursday (52°/43°), and Friday (52°/42°). The main content area includes several news cards: "American Girl's first-ever boy doll gets mixed in controversy" (USA TODAY), "Police: 2 bodies found in Indiana are missing girls" (Associated Press), "Winter's 'biggest storm' expected to unleash California floods" (AccuWeather), "Ashton Kutcher gives emotional testimony at Senate hearing" (The Huffington Post), "Morning Joe' bans Trump aide Kellyanne Conway" (New York Daily News), and "Why William and Kate rarely hold hands" (reuters). A small "reuters" logo is visible in the bottom left corner of the "Why William and Kate..." card.

- EXP4
- Epoch-Greedy
- LinUCB
- ILOVETOCONBANDITS
- BISTRO, BISTRO+
- ...

Motivation

So many existing algorithms, *which one should I use ??*

Motivation

So many existing algorithms, *which one should I use ??*

- no one single algorithm is guaranteed to be the best

Motivation

So many existing algorithms, *which one should I use ??*

- no one single algorithm is guaranteed to be the best

Naive approach: try all and pick the best

Motivation

So many existing algorithms, *which one should I use ??*

- no one single algorithm is guaranteed to be the best

Naive approach: try all and pick the best

- inefficient, wasteful, nonadaptive

Motivation

So many existing algorithms, *which one should I use ??*

- no one single algorithm is guaranteed to be the best

Naive approach: try all and pick the best

- inefficient, wasteful, nonadaptive

Hope: create a **master algorithm** that

- selects base algorithms **automatically and adaptively on the fly**

Motivation

So many existing algorithms, *which one should I use ??*

- no one single algorithm is guaranteed to be the best

Naive approach: try all and pick the best

- inefficient, wasteful, nonadaptive

Hope: create a **master algorithm** that

- selects base algorithms **automatically and adaptively on the fly**
- performs **closely to the best** in the long run

A Closer Look

Full information setting:

- “expert” algorithm (e.g. Hedge (Freund and Schapire, 1997)) solves it

A Closer Look

Full information setting:

- “expert” algorithm (e.g. Hedge (Freund and Schapire, 1997)) solves it

Bandit setting:

- use a multi-armed bandit algorithm (e.g. EXP3 (Auer et al., 2002))?

A Closer Look

Full information setting:

- “expert” algorithm (e.g. Hedge (Freund and Schapire, 1997)) solves it

Bandit setting:

- use a multi-armed bandit algorithm (e.g. EXP3 (Auer et al., 2002))?

Serious flaw in this approach:

- regret guarantee is only about the actual performance

A Closer Look

Full information setting:

- “expert” algorithm (e.g. Hedge (Freund and Schapire, 1997)) solves it

Bandit setting:

- use a multi-armed bandit algorithm (e.g. EXP3 (Auer et al., 2002))?

Serious flaw in this approach:

- regret guarantee is only about the actual performance
- but the performance of base algorithms are significantly influenced due to lack of feedback

Difficulties

An example:

Alg_1 : ✓✓✓✓✓X✓XX✓XX✓XX✓✓✓XX✓XX✓X✓✓XX✓X✓X

Alg_2 : XX✓XX✓✓X✓X✓✓✓✓X✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓

when run separately

Related Work and Our Results

Maillard and Munos (2011) studied similar problems

- EXP3 with higher amount of uniform exploration

Related Work and Our Results

Maillard and Munos (2011) studied similar problems

- EXP3 with higher amount of uniform exploration
- if base algorithms have \sqrt{T} regret, master has $T^{2/3}$ regret

Related Work and Our Results

Maillard and Munos (2011) studied similar problems

- EXP3 with **higher amount of uniform exploration**
- if base algorithms have \sqrt{T} regret, master has $T^{2/3}$ regret

Our results:

- a novel algorithm: **more active and adaptive exploration**
 - ▶ almost same regret as base algorithms

Related Work and Our Results

Maillard and Munos (2011) studied similar problems

- EXP3 with **higher amount of uniform exploration**
- if base algorithms have \sqrt{T} regret, master has $T^{2/3}$ regret

Our results:

- a novel algorithm: **more active and adaptive exploration**
 - ▶ almost same regret as base algorithms
- **two major applications:**
 - ▶ exploiting easy environments while keeping worst case robustness

Related Work and Our Results

Maillard and Munos (2011) studied similar problems

- EXP3 with **higher amount of uniform exploration**
- if base algorithms have \sqrt{T} regret, master has $T^{2/3}$ regret

Our results:

- a novel algorithm: **more active and adaptive exploration**
 - ▶ almost same regret as base algorithms
- **two major applications:**
 - ▶ exploiting easy environments while keeping worst case robustness
 - ▶ selecting correct model automatically

Outline

1 Introduction

2 Formal Setup

3 Main Results

4 Conclusion and Open Problems

A General Bandit Problem

for $t = 1$ **to** T **do**

Environment reveals some **side information** $x_t \in \mathcal{X}$

A General Bandit Problem

for $t = 1$ **to** T **do**

Environment reveals some **side information** $x_t \in \mathcal{X}$

Player **picks an action** $\theta_t \in \Theta$

A General Bandit Problem

for $t = 1$ **to** T **do**

Environment reveals some **side information** $x_t \in \mathcal{X}$

Player **picks an action** $\theta_t \in \Theta$

Environment decides a **loss function** $f_t : \Theta \times \mathcal{X} \mapsto [0, 1]$

A General Bandit Problem

for $t = 1$ **to** T **do**

Environment reveals some **side information** $x_t \in \mathcal{X}$

Player **picks an action** $\theta_t \in \Theta$

Environment decides a **loss function** $f_t : \Theta \times \mathcal{X} \mapsto [0, 1]$

Player suffers and observes (only) the loss $f_t(\theta_t, x_t)$

A General Bandit Problem

for $t = 1$ **to** T **do**

Environment reveals some **side information** $x_t \in \mathcal{X}$

Player **picks an action** $\theta_t \in \Theta$

Environment decides a **loss function** $f_t : \Theta \times \mathcal{X} \mapsto [0, 1]$

Player suffers and observes (only) the loss $f_t(\theta_t, x_t)$

Example: **contextual bandits**

- x is **context**, $\theta \in \Theta$ is a **policy**, $f_t(\theta, x) =$ loss of arm $\theta(x)$

A General Bandit Problem

for $t = 1$ **to** T **do**

Environment reveals some **side information** $x_t \in \mathcal{X}$

Player **picks an action** $\theta_t \in \Theta$

Environment decides a **loss function** $f_t : \Theta \times \mathcal{X} \mapsto [0, 1]$

Player suffers and observes (only) the loss $f_t(\theta_t, x_t)$

Example: **contextual bandits**

- x is **context**, $\theta \in \Theta$ is a **policy**, $f_t(\theta, x) =$ loss of arm $\theta(x)$
- environment: i.i.d., adversarial or hybrid

A General Bandit Problem

for $t = 1$ **to** T **do**

Environment reveals some **side information** $x_t \in \mathcal{X}$

Player **picks an action** $\theta_t \in \Theta$

Environment decides a **loss function** $f_t : \Theta \times \mathcal{X} \mapsto [0, 1]$

Player suffers and observes (only) the loss $f_t(\theta_t, x_t)$

Example: **contextual bandits**

- x is **context**, $\theta \in \Theta$ is a **policy**, $f_t(\theta, x) =$ loss of arm $\theta(x)$
- environment: i.i.d., adversarial or hybrid

(Pseudo) Regret:
$$\text{REG} = \sup_{\theta \in \Theta} \mathbb{E} \left[\sum_{t=1}^T f_t(\theta_t, x_t) - f_t(\theta, x_t) \right]$$

Base Algorithms

Suppose given M base algorithms $\mathcal{B}_1, \dots, \mathcal{B}_M$.

Base Algorithms

Suppose given M base algorithms $\mathcal{B}_1, \dots, \mathcal{B}_M$.

At each round t , receive suggestions $\theta_t^1, \dots, \theta_t^M$.

Base Algorithms

Suppose given M base algorithms $\mathcal{B}_1, \dots, \mathcal{B}_M$.

At each round t , receive suggestions $\theta_t^1, \dots, \theta_t^M$.

Hope: create a master s.t.

loss of master \approx loss of best base algorithm *if run on its own*

Base Algorithms

Suppose given M base algorithms $\mathcal{B}_1, \dots, \mathcal{B}_M$.

At each round t , receive suggestions $\theta_t^1, \dots, \theta_t^M$.

Hope: create a master s.t.

loss of master \approx loss of best base algorithm *if run on its own*

How to formally measure?

Goal

Suppose running \mathcal{B}_i alone gives

$$\text{REG}_{\mathcal{B}_i} \leq \mathcal{R}_i(T)$$

Goal

Suppose running \mathcal{B}_i alone gives

$$\text{REG}_{\mathcal{B}_i} \leq \mathcal{R}_i(T)$$

When running master with all base algorithms, want

$$\text{REG}_{\mathcal{M}} \leq \mathcal{O}(\text{poly}(M)\mathcal{R}_i(T))$$

Goal

Suppose running \mathcal{B}_i alone gives

$$\text{REG}_{\mathcal{B}_i} \leq \mathcal{R}_i(T)$$

When running master with all base algorithms, want

$$\text{REG}_{\mathcal{M}} \leq \mathcal{O}(\text{poly}(M)\mathcal{R}_i(T))$$

Example: for contextual bandits

Algorithm	REG	environment
ILOVETOCONBANDITS (Agarwal et al., 2014)	\sqrt{T}	i.i.d.
BISTRO+ (Syrkkanis et al., 2016)	$T^{2/3}$	hybrid

Goal

Suppose running \mathcal{B}_i alone gives

$$\text{REG}_{\mathcal{B}_i} \leq \mathcal{R}_i(T)$$

When running master with all base algorithms, want

$$\text{REG}_{\mathcal{M}} \leq \mathcal{O}(\text{poly}(M)\mathcal{R}_i(T))$$

Example: for contextual bandits

Algorithm	REG	environment
Master	\sqrt{T}	i.i.d.
Master	$T^{2/3}$	hybrid

Goal

Suppose running \mathcal{B}_i alone gives

$$\text{REG}_{\mathcal{B}_i} \leq \mathcal{R}_i(T)$$

When running master with all base algorithms, want

$$\text{REG}_{\mathcal{M}} \leq \mathcal{O}(\text{poly}(M)\mathcal{R}_i(T)) \text{ impossible in general!}$$

Example: for contextual bandits

Algorithm	REG	environment
Master	\sqrt{T}	i.i.d.
Master	$T^{2/3}$	hybrid

A Natural Assumption

Typical strategy:

- sample a base algorithm $i_t \sim \mathbf{p}_t$
- feed **importance-weighted feedback** to all \mathcal{B}_i : $\frac{f_t(\theta_t, x_t)}{p_{t,i}} \mathbf{1}\{i = i_t\}$

A Natural Assumption

Typical strategy:

- sample a base algorithm $i_t \sim \mathbf{p}_t$
- feed **importance-weighted feedback** to all \mathcal{B}_i : $\frac{f_t(\theta_t, x_t)}{p_{t,i}} \mathbf{1}\{i = i_t\}$

Assume \mathcal{B}_i ensures

$$\text{REG}_{\mathcal{B}_i} \leq \mathcal{R}_i(T)?$$

A Natural Assumption

Typical strategy:

- sample a base algorithm $i_t \sim \mathbf{p}_t$
- feed importance-weighted feedback to all \mathcal{B}_i : $\frac{f_t(\theta_t, x_t)}{p_{t,i}} \mathbf{1}\{i = i_t\}$

Assume \mathcal{B}_i ensures

$$\text{REG}_{\mathcal{B}_i} \leq \mathbb{E} \left[\max_t \frac{1}{p_{t,i}} \right] \mathcal{R}_i(T)$$

A Natural Assumption

Typical strategy:

- sample a base algorithm $i_t \sim \mathbf{p}_t$
- feed importance-weighted feedback to all \mathcal{B}_i : $\frac{f_t(\theta_t, x_t)}{p_{t,i}} \mathbf{1}\{i = i_t\}$

Assume \mathcal{B}_i ensures

$$\text{REG}_{\mathcal{B}_i} \leq \mathbb{E} \left[\left(\max_t \frac{1}{p_{t,i}} \right)^{\alpha_i} \right] \mathcal{R}_i(T)$$

A Natural Assumption

Typical strategy:

- sample a base algorithm $i_t \sim \mathbf{p}_t$
- feed importance-weighted feedback to all \mathcal{B}_i : $\frac{f_t(\theta_t, x_t)}{p_{t,i}} \mathbf{1}\{i = i_t\}$

Assume \mathcal{B}_i ensures

$$\text{REG}_{\mathcal{B}_i} \leq \mathbb{E} \left[\left(\max_t \frac{1}{p_{t,i}} \right)^{\alpha_i} \right] \mathcal{R}_i(T)$$

Want

$$\text{REG}_{\mathcal{M}} \leq \mathcal{O}(\text{poly}(M) \mathcal{R}_i(T))$$

A Natural Assumption

Typical strategy:

- sample a base algorithm $i_t \sim \mathbf{p}_t$
- feed **importance-weighted feedback** to all \mathcal{B}_i : $\frac{f_t(\theta_t, x_t)}{p_{t,i}} \mathbf{1}\{i = i_t\}$

Assume \mathcal{B}_i ensures

$$\text{REG}_{\mathcal{B}_i} \leq \mathbb{E} \left[\left(\max_t \frac{1}{p_{t,i}} \right)^{\alpha_i} \right] \mathcal{R}_i(T)$$

Want

$$\text{REG}_{\mathcal{M}} \leq \mathcal{O}(\text{poly}(M) \mathcal{R}_i(T))$$

Note: **EXP3 still doesn't work**

Outline

1 Introduction

2 Formal Setup

3 Main Results

4 Conclusion and Open Problems

A Special OMD

Intuition 1: want $\frac{1}{\rho_{t,i}}$ to be small

A Special OMD

Intuition 1: want $\frac{1}{p_{t,i}}$ to be small

Mirror Map	$1/p_{t,i} \approx$
Shannon Entropy (EXP3)	$\exp(\eta \cdot \text{loss})$

A Special OMD

Intuition 1: want $\frac{1}{p_{t,i}}$ to be small

Mirror Map	$1/p_{t,i} \approx$
Shannon Entropy (EXP3)	$\exp(\eta \cdot \text{loss})$
Log Barrier: $-\frac{1}{\eta} \sum_i \ln p_i$ (Foster et al., 2016)	$\eta \cdot \text{loss}$

A Special OMD

Intuition 1: want $\frac{1}{p_{t,i}}$ to be small

Mirror Map	$1/p_{t,i} \approx$
Shannon Entropy (EXP3)	$\exp(\eta \cdot \text{loss})$
Log Barrier: $-\frac{1}{\eta} \sum_i \ln p_i$ (Foster et al., 2016)	$\eta \cdot \text{loss}$

In some sense, this provides the **least extreme weighting**

An Increasing Learning Rates Schedule

Intuition 2: Need to **learn faster** if a base algorithm has a low sampled probability

An Increasing Learning Rates Schedule

Intuition 2: Need to **learn faster** if a base algorithm has a low sampled probability

Solution:

- **individual learning rates:** $\sum_i \frac{-\ln p_i}{\eta_i}$ as mirror map

An Increasing Learning Rates Schedule

Intuition 2: Need to **learn faster** if a base algorithm has a low sampled probability

Solution:

- **individual learning rates:** $\sum_i \frac{-\ln p_i}{\eta_i}$ as mirror map
- **increase** learning rate η_i when $\frac{1}{p_{t,i}}$ is too large

Our Algorithm: CORRAL

initial learning rates $\eta_i = \eta$, initial thresholds $\rho_i = 2M$

Our Algorithm: CORRAL

initial learning rates $\eta_i = \eta$, initial thresholds $\rho_i = 2M$

for $t = 1$ **to** T **do**

Observe x_t and send x_t to all base algorithms

Receive suggested actions $\theta_t^1, \dots, \theta_t^M$

Sample $i_t \sim \mathbf{p}_t$, predict $\theta_t = \theta_t^{i_t}$, observe loss $f_t(\theta_t, x_t)$

Construct unbiased estimated loss $f_t^i(\theta, x)$ and send it to \mathcal{B}_i

Our Algorithm: CORRAL

initial learning rates $\eta_i = \eta$, initial thresholds $\rho_i = 2M$

for $t = 1$ **to** T **do**

Observe x_t and send x_t to all base algorithms

Receive suggested actions $\theta_t^1, \dots, \theta_t^M$

Sample $i_t \sim \mathbf{p}_t$, predict $\theta_t = \theta_t^{i_t}$, observe loss $f_t(\theta_t, x_t)$

Construct unbiased estimated loss $f_t^i(\theta, x)$ and send it to \mathcal{B}_i

Update $\mathbf{p}_{t+1} \leftarrow \text{LOG-BARRIER-OMD}(\mathbf{p}_t, \frac{f_t(\theta_t, x_t)}{\rho_{t, i_t}} \mathbf{e}_{i_t}, \eta)$

Our Algorithm: CORRAL

initial learning rates $\eta_i = \eta$, initial thresholds $\rho_i = 2M$

for $t = 1$ **to** T **do**

Observe x_t and send x_t to all base algorithms

Receive suggested actions $\theta_t^1, \dots, \theta_t^M$

Sample $i_t \sim \mathbf{p}_t$, predict $\theta_t = \theta_t^{i_t}$, observe loss $f_t(\theta_t, x_t)$

Construct unbiased estimated loss $f_t^i(\theta, x)$ and send it to \mathcal{B}_i

Update $\mathbf{p}_{t+1} \leftarrow \text{LOG-BARRIER-OMD}(\mathbf{p}_t, \frac{f_t(\theta_t, x_t)}{p_{t, i_t}} \mathbf{e}_{i_t}, \eta)$

for $i = 1$ **to** M **do**

if $\frac{1}{p_{t+1, i}} > \rho_i$ **then** update $\rho_i \leftarrow 2\rho_i, \eta_i \leftarrow \beta\eta_i$

Regret Guarantee

Theorem

If for some environment there exists a base algorithm \mathcal{B}_i such that:

$$\text{REG}_{\mathcal{B}_i} \leq \mathbb{E} \left[\rho_{T,i}^{\alpha_i} \right] \mathcal{R}_i(T)$$

then under the same environment CORRAL ensures:

$$\text{REG}_{\mathcal{M}} = \tilde{\mathcal{O}} \left(\frac{M}{\eta} + T\eta - \frac{\mathbb{E}[\rho_{T,i}]}{\eta} + \mathbb{E}[\rho_{T,i}^{\alpha_i}] \mathcal{R}_i(T) \right)$$

Application

Contextual Bandits:

Algorithm	REG	environment
ILOVETOCONBANDITS (Agarwal et al., 2014)	\sqrt{T}	i.i.d.
BISTRO+ (Syrkkanis et al., 2016)	$T^{2/3}$	hybrid

Application

Contextual Bandits:

Algorithm	REG	environment
CORRAL	\sqrt{T}	i.i.d.
CORRAL	$T^{3/4}$	hybrid

Outline

- 1 Introduction
- 2 Formal Setup
- 3 Main Results
- 4 Conclusion and Open Problems**

Conclusion

We resolve the problem of creating a master that is almost as well as the best base algorithm if it was run on its own.

- **least extreme weighting**: LOG-BARRIER-OMD
- **increasing learning rate** to learn faster
- applications for many settings

Conclusion

We resolve the problem of creating a master that is almost as well as the best base algorithm if it was run on its own.

- least extreme weighting: LOG-BARRIER-OMD
- increasing learning rate to learn faster
- applications for many settings

Open problems:

- inherit exactly the regret of base algorithms ?

$$\text{REG}_{\mathcal{M}} \leq \mathcal{O}(\text{poly}(M)\mathcal{R}_i(T))$$

Conclusion

We resolve the problem of creating a master that is almost as well as the best base algorithm if it was run on its own.

- least extreme weighting: LOG-BARRIER-OMD
- increasing learning rate to learn faster
- applications for many settings

Open problems:

- inherit exactly the regret of base algorithms ?

$$\text{REG}_{\mathcal{M}} \leq \mathcal{O}(\text{poly}(M)\mathcal{R}_i(T))$$

- dependence on M : from polynomial to logarithmic?