Corralling a Band of Bandit Algorithms

Alekh Agarwal\textsuperscript{1}, Haipeng Luo\textsuperscript{1}, Behnam Neyshabur\textsuperscript{2}, Rob Schapire\textsuperscript{1}

\textsuperscript{1}Microsoft Research, New York
\textsuperscript{2}Toyota Technological Institute at Chicago
Contextual Bandits in Practice

Personalized news recommendation in **MSN:**

- Pokémon GO announced its biggest update yet, including 80 new Pokémon
- Why William and Kate rarely hold hands
- "Firefall" wows visitors to Yosemite's El Capitan
Contextual Bandits in Practice

Personalized news recommendation in **MSN:**

- **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

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
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:

when run separately
Difficulties

An example:

When run separately:

$\text{Alg}_1$: ✔✔✔✔✔✘✘✘✘✘✘✘✘✘✘✘✘✘✘✘✘✘✘

$\text{Alg}_2$: ✘✘✔✘✔✘✘✘✘✘✘✘✘✘✘✘✘✘✘✘✘✘

When run with a master:

$\text{Alg}_1$: ✔✔✔✔✔✘✘✘✘✘✘✘✘✘✘✘✘✘✘✘✘✘✘

$\text{Alg}_2$: ✘✘✔✘✔✘✔✘✘✘✘✘✘✘✘✘✘✘✘✘✘✘✘
Difficulties

An example:

Right objective:

*perform almost as well as the best base algorithm* if it was run on its own
Difficulties

An example:

<table>
<thead>
<tr>
<th>Alg₁</th>
<th>Alg₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓✓✓✓✗✗✗✗✗✗✗✗✗✗✗✗✗✗✓✓✓✓</td>
<td>X✗✗✗✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓</td>
</tr>
<tr>
<td>✓✓✓✓✗✗✗✗✗✗✗✗✗✗✗✗✗✗✓✓✓✓</td>
<td>X✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓</td>
</tr>
</tbody>
</table>

when run separately

<table>
<thead>
<tr>
<th>Alg₁</th>
<th>Alg₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓✓✓✓✗✗✗✗✗✗✗✗✗✗✗✗✗✗✓✓✓✓</td>
<td>X✗✗✗✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓</td>
</tr>
<tr>
<td>✓✓✓✓✗✗✗✗✗✗✗✗✗✗✗✗✗✗✓✓✓✓</td>
<td>X✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓</td>
</tr>
</tbody>
</table>

when run with a master

Right objective:
*perform almost as well as the best base algorithm if it was run on its own*

Difficulties:
- worse performance ↔ less feedback
Difficulties

An example:

| Alg1: ✔✔✔✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔✔ ✔✔ ✔ ✔ ✔✔ ✔✔ ✔✔ ✔✔ ✔ | Alg2: ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>when run separately</td>
<td></td>
</tr>
</tbody>
</table>

| Alg1: ✔✔✔✔ ✔ ✔ ✔ ✔ ✔✔ ✔✔ ✔✔ ✔✔ ✔✔ ✔✔ ✔✔ ✔✔ ✔✔ ✔✔ ✔✔ ✔ | Alg2: ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>when run with a master</td>
<td></td>
</tr>
</tbody>
</table>

Right objective: perform almost as well as the best base algorithm if it was run on its own

Difficulties:
- worse performance ↔ less feedback
- requires better tradeoff between exploration and exploitation
Related Work and Our Results

Maillard and Munos (2011) studied similar problems

- EXP3 with **higher amount of uniform exploration**

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
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
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

\begin{verbatim}
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 \( f_t : \Theta \times \mathcal{X} \mapsto [0, 1] \)
\end{verbatim}
A General Bandit Problem

for $t = 1$ to $T$ do

Environment reveals some side information $x_t \in X$

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

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

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

\begin{align*}
\text{for } t = 1 \text{ to } T \text{ do} \\
\quad \text{Environment reveals some side information } x_t \in \mathcal{X} \\
\quad \text{Player picks an action } \theta_t \in \Theta \\
\quad \text{Environment decides a loss function } f_t : \Theta \times \mathcal{X} \mapsto [0, 1] \\
\quad \text{Player suffers and observes (only) the loss } f_t(\theta_t, x_t)
\end{align*}

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 $B_1, \ldots, B_M$. 

Base Algorithms

Suppose given $M$ base algorithms $B_1, \ldots, B_M$.

At each round $t$, receive suggestions $\theta^1_t, \ldots, \theta^M_t$. 
Base Algorithms

Suppose given $M$ base algorithms $B_1, \ldots, B_M$.

At each round $t$, receive suggestions $\theta^1_t, \ldots, \theta^M_t$.

Hope: create a master s.t.

\[
\text{loss of master} \approx \text{loss of best base algorithm if run on its own}
\]
Base Algorithms

Suppose given $M$ base algorithms $B_1, \ldots, B_M$.

At each round $t$, receive suggestions $\theta^1_t, \ldots, \theta^M_t$.

Hope: create a master s.t.

\[
\text{loss of master} \approx \text{loss of best base algorithm if run on its own}
\]

How to formally measure?
Goal

Suppose running $B_i$ alone gives

$$REG_{B_i} \leq R_i(T)$$
Goal

Suppose running $B_i$ alone gives

$$\text{REG}_{B_i} \leq R_i(T)$$

When running master with all base algorithms, want

$$\text{REG}_M \leq O(\text{poly}(M)R_i(T))$$
Suppose running $B_i$ alone gives
\[ \text{REG}_{B_i} \leq R_i(T) \]

When running master with all base algorithms, want
\[ \text{REG}_M \leq O(\text{poly}(M)R_i(T)) \]

Example: for contextual bandits

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>REG</th>
<th>environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILOVETOCONBANDITS (Agarwal et al., 2014)</td>
<td>$\sqrt{T}$</td>
<td>i.i.d.</td>
</tr>
<tr>
<td>BISTRO+ (Syrgkanis et al., 2016)</td>
<td>$T^{2/3}$</td>
<td>hybrid</td>
</tr>
</tbody>
</table>
Goal

Suppose running $B_i$ alone gives

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

When running master with all base algorithms, want

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

Example: for contextual bandits

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>REG</th>
<th>environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master</td>
<td>$\sqrt{T}$</td>
<td>i.i.d.</td>
</tr>
<tr>
<td>Master</td>
<td>$T^{2/3}$</td>
<td>hybrid</td>
</tr>
</tbody>
</table>
Goal

Suppose running $B_i$ alone gives

$$\text{REG}_{B_i} \leq R_i(T)$$

When running master with all base algorithms, want

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

Example: for contextual bandits

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>REG</th>
<th>environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master</td>
<td>$\sqrt{T}$</td>
<td>i.i.d.</td>
</tr>
<tr>
<td>Master</td>
<td>$T^{2/3}$</td>
<td>hybrid</td>
</tr>
</tbody>
</table>
A Natural Assumption

Typical strategy:

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

Assume $B_i$ ensures

$$\text{REG}_{B_i} \leq R_i(T)?$$
A Natural Assumption

Typical strategy:

- Sample a base algorithm $i_t \sim p_t$
- Feed importance-weighted feedback to all $B_i$: $\frac{f_t(\theta_t, x_t)}{p_{t,i}} 1\{i = i_t\}$

Assume $B_i$ ensures

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

Typical strategy:

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

Assume $B_i$ ensures

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

Typical strategy:

- sample a base algorithm \( i_t \sim p_t \)
- feed importance-weighted feedback to all \( B_i: \frac{f_t(\theta_t, x_t)}{p_{t,i}} \mathbf{1}\{i = i_t\} \)

Assume \( B_i \) ensures

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

Want

\[
\text{REG}_M \leq \mathcal{O}(\text{poly}(M)\mathcal{R}_i(T))
\]
A Natural Assumption

Typical strategy:

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

Assume $B_i$ ensures

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

Want

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

Note: \textbf{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 $\frac{1}{p_{t,i}}$ to be small
A Special OMD

**Intuition 1:** want \( \frac{1}{p_{t,i}} \) to be small

<table>
<thead>
<tr>
<th>Mirror Map</th>
<th>( 1/p_{t,i} \approx )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Shannon Entropy</strong> (EXP3)</td>
<td>( \exp(\eta \cdot \text{loss}) )</td>
</tr>
</tbody>
</table>
A Special OMD

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

<table>
<thead>
<tr>
<th>Mirror Map</th>
<th>$1/p_{t,i} \approx$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shannon Entropy (EXP3)</td>
<td>$\exp(\eta \cdot \text{loss})$</td>
</tr>
<tr>
<td>Log Barrier: $-\frac{1}{\eta} \sum_i \ln p_i$ (Foster et al., 2016)</td>
<td>$\eta \cdot \text{loss}$</td>
</tr>
</tbody>
</table>
A Special OMD

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

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

In some sense, this provides the least extreme weighting
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** \( \eta_i \) 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: \textbf{CORRAL}

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

\textbf{for} $t = 1$ \textbf{to} $T$ \textbf{do}

- Observe $x_t$ and send $x_t$ to all base algorithms
- Receive suggested actions $\theta_1^t, \ldots, \theta_M^t$
- Sample $i_t \sim p_t$, predict $\theta_t = \theta_{i_t}^t$, observe loss $f_t(\theta_t, x_t)$
- Construct unbiased estimated loss $f_t^i(\theta, x)$ and send it to $B_i$
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 $\theta^1_t, \ldots, \theta^M_t$

Sample $i_t \sim p_t$, predict $\theta_t = \theta^i_t$, observe loss $f_t(\theta_t, x_t)$

Construct unbiased estimated loss $f^i_t(\theta, x)$ and send it to $B_i$

Update $p_{t+1} \leftarrow \text{LOG-BARRIER-OMD}(p_t, f_t(x_t, \theta_t, x_t) e_i, \eta)$
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 $\theta^1_t, \ldots, \theta^M_t$

Sample $i_t \sim p_t$, predict $\theta_t = \theta^i_t$, observe loss $f_t(\theta_t, x_t)$

Construct unbiased estimated loss $f^i_t(\theta, x)$ and send it to $B_i$

Update $p_{t+1} \leftarrow \text{LOG-BARRIER-OMD}(p_t, f_t(\theta_t, x_t) / p_{t,i_t} e_{i_t}, \eta)$

for $i = 1$ to $M$ do

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

If for some environment there exists a base algorithm $B_i$ such that:

$$\text{REG}_{B_i} \leq \mathbb{E} \left[ \rho_{T,i}^{\alpha_i} \right] R_i(T)$$

then under the same environment Corral ensures:

$$\text{REG}_M = \tilde{O} \left( \frac{M}{\eta} + T\eta - \frac{\mathbb{E}[\rho_{T,i}]}{\eta} + \mathbb{E}[\rho_{T,i}^{\alpha_i}]R_i(T) \right)$$
## Contextual Bandits:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>REG</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILOVETOCONBANDITS (Agarwal et al., 2014)</td>
<td>$\sqrt{T}$</td>
<td>i.i.d.</td>
</tr>
<tr>
<td>BISTRO+ (Syrgkanis et al., 2016)</td>
<td>$T^{2/3}$</td>
<td>hybrid</td>
</tr>
</tbody>
</table>
## Application

### Contextual Bandits:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>REG</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Corral</strong></td>
<td>$\sqrt{T}$</td>
<td>i.i.d.</td>
</tr>
<tr>
<td><strong>Corral</strong></td>
<td>$T^{3/4}$</td>
<td>hybrid</td>
</tr>
</tbody>
</table>
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?**

\[ \text{REG}_M \leq O(\text{poly}(M)\mathcal{R}_i(T)) \]
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?

\[ \text{REG}_M \leq \mathcal{O}(\text{poly}(M)R_i(T)) \]

- dependence on \( M \): from **polynomial** to **logarithmic**?