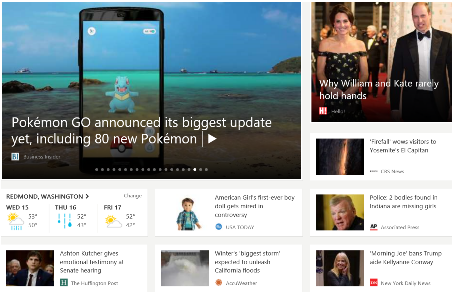
Corralling a Band of Bandit Algorithms

Alekh Agarwal[1], **Haipeng Luo**[1], Behnam Neyshabur[2], Rob Schapire[1]

[1]Microsoft Research, New York
[2]Toyota Technological Institute at Chicago

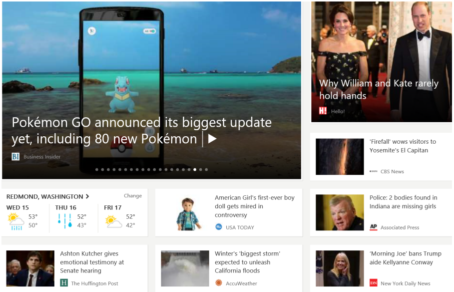
Contextual Bandits in Practice

Personalized news recommendation in MSN:



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

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: ✓✓✓✓✓✗✓✗✗✓✗✗✓✗✗✓✓✓✗✗✓✗✗✓✗✓✓✗✗✓✗✓✗

Alg_2: ✗✗✓✗✗✓✓✗✓✗✓✓✓✓✗✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓

when run separately

Difficulties

An example:

Alg_1: ✓✓✓✓✓✗✓✗✗✓✗✗✓✗✗✓✓✓✗✗✓✗✗✓✗✓✓✗✗✓✗✓✗

Alg_2: ✗✗✓✗✗✓✓✗✓✗✓✓✓✓✗✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓

when run separately

Alg_1: ✓✓✓✓✓✗✓✗✗✓✗✗✓✗✗✓✓✓✗✗✓✗✗✓✗✓✓✗✗✓✗✓✗

Alg_2: ✗✗✓✗✗✓✓✗✓✗✓✗✓✓✗✓✗✗✓✗✓✓✗✗✓✗✗✓✗✗✗✓✓

when run with a master

Difficulties

An example:

Alg_1: ✓✓✓✓✓✗✓✗✗✓✗✗✓✗✗✓✓✓✗✗✓✗✗✓✗✓✓✗✗✓✗✓✗

Alg_2: ✗✗✓✗✗✓✓✗✓✗✓✓✓✓✗✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓

when run separately

Alg_1: ✓✓✓✓✓✗✓✗✗✓✗✗✓✗✗✓✓✓✗✗✓✗✗✓✗✓✓✗✗✓✗✓✗

Alg_2: ✗✗✓✗✗✓✓✗✓✗✓✗✓✓✗✓✗✗✓✗✓✓✗✗✓✗✗✓✗✗✗✓✓

when run with a master

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

Difficulties

An example:

Alg_1: ✓✓✓✓✓✗✓✗✗✓✗✗✓✗✗✓✓✓✗✗✓✗✗✓✗✓✓✗✗✓✗✓✗

Alg_2: ✗✗✓✗✗✓✓✗✓✗✓✓✓✓✗✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓

when run separately

Alg_1: ✓✓✓✓✓✗✓✗✗✓✗✗✓✗✗✓✓✓✗✗✓✗✗✓✗✓✓✗✗✓✗✓✗

Alg_2: ✗✗✓✗✗✓✓✗✓✗✓✗✓✓✗✓✗✗✓✗✓✓✗✗✓✗✗✓✗✗✗✓✓

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 $\leftrightarrow$ less feedback

Difficulties

An example:

Alg_1: ✓✓✓✓✓✗✓✗✗✓✗✗✓✗✗✓✓✓✗✗✓✗✗✓✗✓✓✗✗✓✗✓✗

Alg_2: ✗✗✓✗✗✓✓✗✓✗✓✓✓✓✗✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓

when run separately

Alg_1: ✓✓✓✓✓✗✓✗✗✓✗✗✓✗✗✓✓✓✗✗✓✗✗✓✗✓✓✗✗✓✗✓✗

Alg_2: ✗✗✓✗✗✓✓✗✓✗✓✗✓✓✗✓✗✗✓✗✓✓✗✗✓✗✗✓✗✗✗✓✓

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 $\leftrightarrow$ 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

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

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

Base Algorithms

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

Base Algorithms

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

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

Base Algorithms

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

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

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 $\mathcal{B}_1, \ldots, \mathcal{B}_M$.

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

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

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

Goal

Suppose running $\mathcal{B}_i$ alone gives

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

When running master with all base algorithms, want

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

Goal

Suppose running $\mathcal{B}_i$ alone gives

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

When running master with all base algorithms, want

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

Example: for contextual bandits

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

Goal

Suppose running $\mathcal{B}_i$ alone gives

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

When running master with all base algorithms, want

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

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

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

When running master with all base algorithms, want

$$\mathrm{REG}_{\mathcal{M}} \leq \mathcal{O}(\text{poly}(M)\mathcal{R}_i(T))$$ **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 $i_t \sim \boldsymbol{p}_t$
- 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 $i_t \sim \boldsymbol{p}_t$
- 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

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

A Natural Assumption

Typical strategy:

- sample a base algorithm $i_t \sim \boldsymbol{p}_t$
- 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

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

A Natural Assumption

Typical strategy:

- sample a base algorithm $i_t \sim \boldsymbol{p}_t$
- 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

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

A Natural Assumption

Typical strategy:

- sample a base algorithm $i_t \sim \boldsymbol{p}_t$
- 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

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

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

A Natural Assumption

Typical strategy:

- sample a base algorithm $i_t \sim \boldsymbol{p}_t$
- 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

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

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

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

Outline

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

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 $\theta_t^1, \ldots, \theta_t^M$

Sample $i_t \sim \boldsymbol{p}_t$, 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 $\theta_t^1, \ldots, \theta_t^M$

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

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 $\theta_t^1, \ldots, \theta_t^M$

Sample $i_t \sim \boldsymbol{p}_t$, 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 $\boldsymbol{p}_{t+1} \leftarrow \text{Log-Barrier-OMD}(\boldsymbol{p}_t, \frac{f_t(\theta_t, x_t)}{p_{t,i_t}} \boldsymbol{e}_{i_t}, \boldsymbol{\eta})$

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:

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

then under the same environment CORRAL *ensures:*

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

Application

Contextual Bandits:

Algorithm	REG	environment
ILOVETOCONBANDITS (Agarwal et al., 2014)	$\sqrt{T}$	i.i.d.
BISTRO$+$ (Syrgkanis 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

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 ?

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

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 ?

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

- dependence on M: from polynomial to logarithmic?