Simons Tutorial: Online Learning and Bandits Part I

Wouter Koolen and Alan Malek

August 31st, 2020

Positioning this Tutorial

- Building up tools in support of RL
- Exploring surrounding viewpoints, problems and methods
- Soaking up “Culture”

Working Definitions

Context: interactive decision making in unknown environment

Aim: Design systems to amass reward in many environments.

Working Definitions

Context: interactive decision making in unknown environment

Aim: Design systems to amass reward in many environments.

Main distinction: model of environment

- **Reinforcement Learning** action affects future state
- **Bandits** action affects observation
- **Full Inf. Online Learning** action affects reward

On the Menu

Two parts:

(1) Full Information Online Learning

(2) Bandits (w. Alan Malek)

Full Information Online Learning

1. Two Basic Problems
   - Online Convex Optimisation; Online Gradient Descent
   - The Experts Problem; Exponential Weights
2. Two Peeks Beyond the Basics
   - Follow the Regularised Leader and Mirror Descent
   - Online Quadratic Optimisation; Online Newton Step
3. Applications
   - Classical Optimisation
   - Stochastic Optimisation
   - Saddle Points in Two-player Zero-Sum Games
4. Conclusion and Extensions

Two Basic Problems

Setup

- Focus on losses (negative rewards)
- Model Environment as Adversary
- Online Convex Optimisation (OCO) abstraction.

OCO Problem

Protocol: Online Convex Optimisation

Given: game length T, convex action space $\mathcal{U} \subseteq \mathbb{R}^d$

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

- The learner picks action $\boldsymbol{w}_t \in \mathcal{U}$
- The adversary picks convex loss $f_t : \mathcal{U} \to \mathbb{R}$
- The learner observes f_t $\triangleleft$ full information
- The learner incurs loss $f_t(\boldsymbol{w}_t)$

OCO Problem

Protocol: Online Convex Optimisation

Given: game length T, convex action space $\mathcal{U} \subseteq \mathbb{R}^d$

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

- The learner picks action $\boldsymbol{w}_t \in \mathcal{U}$
- The adversary picks convex loss $f_t : \mathcal{U} \to \mathbb{R}$
- The learner observes f_t $\triangleleft$ full information
- The learner incurs loss $f_t(\boldsymbol{w}_t)$

The goal: control the regret (w.r.t. the best point after T rounds)

$$\mathcal{R}_T = \sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \min_{\boldsymbol{u} \in \mathcal{U}} \sum_{t=1}^{T} f_t(\boldsymbol{u})$$

using a computationally efficient algorithm for learner.

Design Principle

Learner needs to "chase" the best point $\arg\min_{\boldsymbol{u}\in\mathcal{U}} \sum_{t=1}^{T} f_t(\boldsymbol{w}_t)$.
But doing so naively overfits.

Idea: add regularisation. Two manifestations:

- Penalise excentricity "FTRL style"
- Update iterates, but only slowly "MD style"

Will see examples of both. For our purposes, these are roughly equivalent

Online Gradient Descent (OGD) Algorithm

Let $\mathcal{U}$ be a convex set containing $\mathbf{0}$. Fix a learning rate $\eta > 0$.

Algorithm: Online Gradient Descent (OGD)

OGD with learning rate $\eta > 0$ plays

$$\boldsymbol{w}_1 = \mathbf{0} \qquad \text{and} \qquad \boldsymbol{w}_{t+1} \;=\; \Pi_{\mathcal{U}}\left(\boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t)\right)$$

where $\Pi_{\mathcal{U}}(\boldsymbol{w}) = \arg\min_{\boldsymbol{u} \in \mathcal{U}} \|\boldsymbol{u} - \boldsymbol{w}\|$ is the projection onto $\mathcal{U}$.

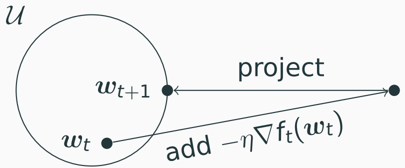


Figure 1: OGD update

Online Gradient Descent Result

Algorithm: OGD

$$\boldsymbol{w}_1 = \mathbf{0} \qquad \text{and} \qquad \boldsymbol{w}_{t+1} \; = \; \Pi_{\mathcal{U}}\left(\boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t)\right)$$

Assumption: Boundedness

Bounded domain $\max_{\boldsymbol{u} \in \mathcal{U}} \|\boldsymbol{u}\| \leq D$ and gradients $\|\nabla f_t(\boldsymbol{w}_t)\| \leq G$.

Online Gradient Descent Result

Algorithm: OGD

$$\boldsymbol{w}_1 = \boldsymbol{0} \qquad \text{and} \qquad \boldsymbol{w}_{t+1} \;=\; \Pi_{\mathcal{U}}\left(\boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t)\right)$$

Assumption: Boundedness

Bounded domain $\max_{\boldsymbol{u} \in \mathcal{U}} \|\boldsymbol{u}\| \leq D$ and gradients $\|\nabla f_t(\boldsymbol{w}_t)\| \leq G$.

Theorem (OGD regret bd, Zinkevich 2003)

$$\mathcal{R}_T \;=\; \sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \min_{\boldsymbol{u} \in \mathcal{U}} \sum_{t=1}^{T} f_t(\boldsymbol{u}) \;\leq\; \frac{1}{2\eta} D^2 + \frac{\eta}{2} T G^2$$

Online Gradient Descent Result

Algorithm: OGD

$$\boldsymbol{w}_1 = \boldsymbol{0} \qquad \text{and} \qquad \boldsymbol{w}_{t+1} \;=\; \Pi_{\mathcal{U}}\left(\boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t)\right)$$

Assumption: Boundedness

Bounded domain $\max_{\boldsymbol{u} \in \mathcal{U}} \|\boldsymbol{u}\| \leq D$ and gradients $\|\nabla f_t(\boldsymbol{w}_t)\| \leq G$.

Theorem (OGD regret bd, Zinkevich 2003)

$$\mathcal{R}_T \;=\; \sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \min_{\boldsymbol{u} \in \mathcal{U}} \sum_{t=1}^{T} f_t(\boldsymbol{u}) \;\leq\; \frac{1}{2\eta} D^2 + \frac{\eta}{2} T G^2$$

Corollary

Tuning $\eta = \frac{D}{G\sqrt{T}}$ *results in* $\mathcal{R}_T \leq DG\sqrt{T}$.

Online Gradient Descent Result

Algorithm: OGD

$$\boldsymbol{w}_1 = \boldsymbol{0} \qquad \text{and} \qquad \boldsymbol{w}_{t+1} \;=\; \Pi_{\mathcal{U}}\left(\boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t)\right)$$

Assumption: Boundedness

Bounded domain $\max_{\boldsymbol{u} \in \mathcal{U}} \|\boldsymbol{u}\| \leq D$ and gradients $\|\nabla f_t(\boldsymbol{w}_t)\| \leq G$.

Theorem (OGD regret bd, Zinkevich 2003)

$$\mathcal{R}_T \;=\; \sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \min_{\boldsymbol{u} \in \mathcal{U}} \sum_{t=1}^{T} f_t(\boldsymbol{u}) \;\leq\; \frac{1}{2\eta} D^2 + \frac{\eta}{2} T G^2$$

Corollary

Tuning $\eta = \frac{D}{G\sqrt{T}}$ *results in* $\mathcal{R}_T \leq DG\sqrt{T}$.

Sublinear regret: learning overhead per round $\to 0$.

Proof of OGD regret bound

Using convexity, we may analyse the tangent upper bound

$$f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u}) \;\leq\; \langle \boldsymbol{w}_t - \boldsymbol{u}, \nabla f_t(w_t) \rangle$$

Moreover,

$$\begin{aligned}
\|\boldsymbol{w}_{t+1} - \boldsymbol{u}\|^2 \;&=\; \|\Pi_{\mathcal{U}}\left(\boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t)\right) - \boldsymbol{u}\|^2 \\
&\leq\; \|\boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t) - \boldsymbol{u}\|^2 \\
&=\; \|\boldsymbol{w}_t - \boldsymbol{u}\|^2 - 2\eta \langle \boldsymbol{w}_t - \boldsymbol{u}, \nabla f_t(\boldsymbol{w}_t) \rangle + \eta^2 \|\nabla f_t(\boldsymbol{w}_t)\|^2
\end{aligned}$$

Hence

$$\langle \boldsymbol{w}_t - \boldsymbol{u}, \nabla f_t(\boldsymbol{w}_t) \rangle \;\leq\; \frac{\|\boldsymbol{w}_t - \boldsymbol{u}\|^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{u}\|^2}{2\eta} + \frac{\eta}{2} \|\nabla f_t(\boldsymbol{w}_t)\|^2$$

Proof of OGD regret bound (ctd)

Summing over T rounds, we find

$$
\begin{aligned}
\mathcal{R}_T^{\boldsymbol{u}} &\leq \sum_{t=1}^{T} \langle \boldsymbol{w}_t - \boldsymbol{u}, \nabla f_t(\boldsymbol{w}_t) \rangle \\
&\leq \underbrace{\sum_{t=1}^{T} \frac{\|\boldsymbol{w}_t - \boldsymbol{u}\|^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{u}\|^2}{2\eta}}_{\text{telescopes}} + \frac{\eta}{2} \sum_{t=1}^{T} \|\nabla f_t(\boldsymbol{w}_t)\|^2 \\
&\leq \frac{\|\boldsymbol{u}\|^2 - \cancel{\|\boldsymbol{w}_{T+1} - \boldsymbol{u}\|^2}}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\nabla f_t(\boldsymbol{w}_t)\|^2 \\
&\leq \frac{D^2}{2\eta} + \frac{\eta}{2} T G^2
\end{aligned}
$$

OCO Lower Bound

Is OGD regret bound of $\mathcal{R}_T \leq GD\sqrt{T}$ any good?

OCO Lower Bound

Is OGD regret bound of $\mathcal{R}_T \leq GD\sqrt{T}$ any good?

Scaling with G and D is natural. What about $\sqrt{T}$?

OCO Lower Bound

Is OGD regret bound of $\mathcal{R}_T \leq GD\sqrt{T}$ any good?

Scaling with G and D is natural. What about $\sqrt{T}$?

Theorem

Any OCO algorithm can be made to incur $\mathcal{R}_T = \Omega(\sqrt{T})$.

OCO Lower Bound

Is OGD regret bound of $\mathcal{R}_T \leq GD\sqrt{T}$ any good?

Scaling with G and D is natural. What about $\sqrt{T}$?

Theorem

Any OCO algorithm can be made to incur $\mathcal{R}_T = \Omega(\sqrt{T})$.

Proof (by probabilistic argument).

Consider interval $\mathcal{U} = [-1, 1]$ and linear losses $f_t(u) = x_t \cdot u$ with i.i.d. Rademacher coefficients $x_t \in \{\pm 1\}$. *Any* algorithm has expected loss zero. The expected loss of the best action (± 1) is $-\mathbb{E}[|\sum_{t=1}^T x_t|] = -\Omega(\sqrt{T})$. Then as the expected regret is $\mathbb{E}[\mathcal{R}_T] = \Omega(\sqrt{T})$, there is a deterministic witness. $\square$

Here, the regret arises from *overfitting* of the best point.

OGD Discussion

- Adversarial result, super strong!
- Proof reveals it is really about linear losses.
- Matching lower bounds

Successful in practise:

- Practically all deep learning uses versions of online gradient descent (e.g. TensorFlow has AdaGrad [Duchi et al., 2011]) even though objective not convex.

From Learning Parameters to Picking Actions

We now turn to the second elementary online learning task.

- Decision Theoretic Online Learning
- Experts setting (also: Hedge setting)
- Prediction with Expert Advice

From Learning Parameters to Picking Actions

We now turn to the second elementary online learning task.

- Decision Theoretic Online Learning
- Experts setting (also: Hedge setting)
- Prediction with Expert Advice

Protocol: Prediction With Expert Advice

Given: game length T, number K of experts

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

- Learner chooses a distribution $\boldsymbol{w}_t \in \triangle_K$ on K "experts".
- Adversary reveals loss vector $\boldsymbol{\ell}_t \in [0,1]^K$.
- Learner's loss is the **dot loss** $\boldsymbol{w}_t^\intercal \boldsymbol{\ell}_t = \sum_{k=1}^K w_t^k \ell_t^k$

From Learning Parameters to Picking Actions

We now turn to the second elementary online learning task.

- Decision Theoretic Online Learning
- Experts setting (also: Hedge setting)
- Prediction with Expert Advice

Protocol: Prediction With Expert Advice

Given: game length T, number K of experts

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

- Learner chooses a distribution $\boldsymbol{w}_t \in \triangle_K$ on K “experts”.
- Adversary reveals loss vector $\boldsymbol{\ell}_t \in [0, 1]^K$.
- Learner’s loss is the **dot loss** $\boldsymbol{w}_t^\intercal \boldsymbol{\ell}_t = \sum_{k=1}^K w_t^k \ell_t^k$

The goal: control the regret (w.r.t. the best expert after T rounds)

$$\mathcal{R}_T = \sum_{t=1}^T \boldsymbol{w}_t^\intercal \boldsymbol{\ell}_t - \min_{k \in [K]} \sum_{t=1}^T \ell_t^k$$

using a computationally efficient algorithm for learner.

Let's apply what we know

Observations:

- Dot loss $\boldsymbol{u} \mapsto \boldsymbol{u}^\intercal \boldsymbol{\ell}_t$ is *linear* (hence convex).
- Gradient $\boldsymbol{\ell}_t \in [0,1]^K$ bounded by $\|\boldsymbol{\ell}_t\| \le \sqrt{K}$.
- Probability simplex $\triangle_K$ is contained in unit ball.

So: Instance of Online Convex Optimisation.

OGD with $D = 1$ and $G = \sqrt{K}$ gives $\mathcal{R}_T \le \sqrt{KT}$.

Let's apply what we know

Observations:

- Dot loss $\boldsymbol{u} \mapsto \boldsymbol{u}^\intercal \boldsymbol{\ell}_t$ is *linear* (hence convex).
- Gradient $\boldsymbol{\ell}_t \in [0,1]^K$ bounded by $\|\boldsymbol{\ell}_t\| \le \sqrt{K}$.
- Probability simplex $\triangle_K$ is contained in unit ball.

So: Instance of Online Convex Optimisation.

OGD with $D = 1$ and $G = \sqrt{K}$ gives $\mathcal{R}_T \le \sqrt{KT}$.

Q: Optimal?

Let's apply what we know

Observations:

- Dot loss $\boldsymbol{u} \mapsto \boldsymbol{u}^\intercal \boldsymbol{\ell}_t$ is *linear* (hence convex).
- Gradient $\boldsymbol{\ell}_t \in [0, 1]^K$ bounded by $\|\boldsymbol{\ell}_t\| \leq \sqrt{K}$.
- Probability simplex $\triangle_K$ is contained in unit ball.

So: Instance of Online Convex Optimisation.

OGD with $D = 1$ and $G = \sqrt{K}$ gives $\mathcal{R}_T \leq \sqrt{KT}$.

Q: Optimal?

Maybe not. There are no points with loss difference $\sqrt{K}$ in the simplex . . .

Exponential Weigths / Hedge Algorithm

Algorithm: Exponential Weights (EW)

EW with *learning rate* $\eta > 0$ plays weights in round t:

$$w_t^k \;=\; \frac{e^{-\eta \sum_{s=1}^{t-1} \ell_s^k}}{\sum_{j=1}^K e^{-\eta \sum_{s=1}^{t-1} \ell_s^j}}. \tag{EW}$$

Exponential Weigths / Hedge Algorithm

Algorithm: Exponential Weights (EW)

EW with *learning rate* $\eta > 0$ plays weights in round t:

$$w_t^k \;=\; \frac{e^{-\eta \sum_{s=1}^{t-1} \ell_s^k}}{\sum_{j=1}^K e^{-\eta \sum_{s=1}^{t-1} \ell_s^j}}. \tag{EW}$$

or, equivalently, $w_1^k = \frac{1}{K}$ and

$$w_{t+1}^k \;=\; \frac{w_t^k e^{-\eta \ell_t^k}}{\sum_{j=1}^K w_t^j e^{-\eta \ell_t^j}} \tag{EW, incremental}$$

Exponential Weigths / Hedge Algorithm

Algorithm: Exponential Weights (EW)

EW with *learning rate* $\eta > 0$ plays weights in round t:

$$w_t^k \;=\; \frac{e^{-\eta \sum_{s=1}^{t-1} \ell_s^k}}{\sum_{j=1}^K e^{-\eta \sum_{s=1}^{t-1} \ell_s^j}}. \tag{EW}$$

or, equivalently, $w_1^k = \frac{1}{K}$ and

$$w_{t+1}^k \;=\; \frac{w_t^k e^{-\eta \ell_t^k}}{\sum_{j=1}^K w_t^j e^{-\eta \ell_t^j}} \tag{EW, incremental}$$

Theorem (EW regret bd, Freund and Schapire 1997)

The regret of EW is bounded by $\mathcal{R}_T \leq \frac{\ln K}{\eta} + T\frac{\eta}{8}$.

Corollary

Tuning $\eta = \sqrt{\frac{8 \ln K}{T}}$ *yields* $\mathcal{R}_T \leq \sqrt{T/2 \ln K}$.

EW Analysis

Applying *Hoeffding's Lemma* to the loss of each round gives

$$\sum_{t=1}^{T} \boldsymbol{w}_t^\intercal \boldsymbol{\ell}_t \ \le\ \sum_{t=1}^{T} \Bigg(\underbrace{\frac{-1}{\eta} \ln \left(\sum_{k=1}^{K} w_t^k e^{-\eta \ell_t^k} \right)}_{\text{“mix loss”}} + \underbrace{\eta/8}_{\text{overhead}} \Bigg)$$

Crucial observation is that cumulative mix loss *telescopes*

$$\begin{aligned}
\sum_{t=1}^{T} \frac{-1}{\eta} \ln \left(\sum_{k=1}^{K} w_t^k e^{-\eta \ell_t^k} \right) &= \sum_{t=1}^{T} \frac{-1}{\eta} \ln \left(\sum_{k=1}^{K} \frac{e^{-\eta \sum_{s=1}^{t-1} \ell_s^k}}{\sum_{j=1}^{K} e^{-\eta \sum_{s=1}^{t-1} \ell_s^j}} e^{-\eta \ell_t^k} \right) \\
&= \sum_{t=1}^{T} \frac{-1}{\eta} \ln \left(\frac{\sum_{k=1}^{K} e^{-\eta \sum_{s=1}^{t} \ell_s^k}}{\sum_{j=1}^{K} e^{-\eta \sum_{s=1}^{t-1} \ell_s^j}} \right) \\
&\overset{\text{telescopes}}{=} \frac{-1}{\eta} \ln \left(\sum_{k=1}^{K} e^{-\eta \sum_{t=1}^{T} \ell_t^k} \right) + \frac{\ln K}{\eta} \\
&\le \min_{k \in [K]} \sum_{t=1}^{T} \ell_t^k + \frac{\ln K}{\eta}.
\end{aligned}$$

Summary so far

Balancing act: "model complexity" vs "overfitting"

Theorem (OGD)

$$\mathcal{R}_T \;\leq\; \frac{D^2}{2\eta} + \frac{\eta}{2} G^2 T$$

Theorem (EW)

$$\mathcal{R}_T \;\leq\; \frac{\ln K}{\eta} + \frac{\eta}{8} T$$

Summary so far

Balancing act: "model complexity" vs "overfitting"

Theorem (OGD)

$$\mathcal{R}_T \;\le\; \frac{D^2}{2\eta} + \frac{\eta}{2} G^2 T$$

Theorem (EW)

$$\mathcal{R}_T \;\le\; \frac{\ln K}{\eta} + \frac{\eta}{8} T$$

Generates many follow-up questions:

- What if horizon T is not fixed? Anytime guarantees?
- What if gradient bound G is not known a priori?
- Can we have the actual gradient norms?
- What if model complexity (D) is not known? Not uniformly bounded? See Orabona and Cutkosky ICML'20 tutorial.

Summary so far

Balancing act: “model complexity” vs “overfitting”

Theorem (OGD)

$$\mathcal{R}_T \;\leq\; \frac{D^2}{2\eta} + \frac{\eta}{2} G^2 T$$

Theorem (EW)

$$\mathcal{R}_T \;\leq\; \frac{\ln K}{\eta} + \frac{\eta}{8} T$$

Generates many follow-up questions:

- What if horizon T is not fixed? Anytime guarantees?
- What if gradient bound G is not known a priori?
- Can we have the actual gradient norms?
- What if model complexity (D) is not known? Not uniformly bounded? See Orabona and Cutkosky ICML’20 tutorial.

Need refined analyses $\Rightarrow$ Restarts (doubling trick), decreasing η_t (AdaGrad/AdaHedge), learning the learning rate η (MetaGrad), ...

Summary so far

Balancing act: "model complexity" vs "overfitting"

Theorem (OGD)

$$\mathcal{R}_T \leq \frac{D^2}{2\eta} + \frac{\eta}{2} G^2 T$$

Theorem (EW)

$$\mathcal{R}_T \leq \frac{\ln K}{\eta} + \frac{\eta}{8} T$$

Generates many follow-up questions:

- What if horizon T is not fixed? Anytime guarantees?
- What if gradient bound G is not known a priori?
- Can we have the actual gradient norms?
- What if model complexity (D) is not known? Not uniformly bounded? See Orabona and Cutkosky ICML'20 tutorial.

Need refined analyses $\Rightarrow$ Restarts (doubling trick), decreasing η_t (AdaGrad/AdaHedge), learning the learning rate η (MetaGrad), ...

Active research area!

Two Peeks Beyond the Basics

FTRL/MD “sneak peek”

Q: What if my domain does not look like either ball or simplex?

FTRL/MD "sneak peek"

Q: What if my domain does not look like either ball or simplex?

Algorithm: Follow the Regularised Leader (FTRL)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathcal{U}} \sum_{s=1}^{t} \langle \boldsymbol{u}, \nabla f_s(w_s) \rangle + \frac{1}{\eta} R(\boldsymbol{u})$$

Algorithm: Mirror Descent (MD)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathcal{U}} \langle \boldsymbol{u}, \nabla f_t(w_t) \rangle + \frac{1}{\eta} B(\boldsymbol{u} \| \boldsymbol{w}_t)$$

FTRL/MD "sneak peek"

Q: What if my domain does not look like either ball or simplex?

Algorithm: Follow the Regularised Leader (FTRL)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathcal{U}} \sum_{s=1}^{t} \langle \boldsymbol{u}, \nabla f_s(w_s) \rangle + \frac{1}{\eta} R(\boldsymbol{u})$$

Algorithm: Mirror Descent (MD)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathcal{U}} \langle \boldsymbol{u}, \nabla f_t(w_t) \rangle + \frac{1}{\eta} B(\boldsymbol{u} \| \boldsymbol{w}_t)$$

Examples:

	Regularizer R	**Bregman Divergence** B
OGD	sq. Euclidean norm	sq. Euclidean distance
EW	Shannon entropy	Kullback-Leibler divergence

FTRL/MD "sneak peek"

Q: What if my domain does not look like either ball or simplex?

Algorithm: Follow the Regularised Leader (FTRL)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathcal{U}} \sum_{s=1}^{t} \langle \boldsymbol{u}, \nabla f_s(w_s) \rangle + \frac{1}{\eta} R(\boldsymbol{u})$$

Algorithm: Mirror Descent (MD)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathcal{U}} \langle \boldsymbol{u}, \nabla f_t(w_t) \rangle + \frac{1}{\eta} B(\boldsymbol{u} \| \boldsymbol{w}_t)$$

Examples:

	Regularizer R	**Bregman Divergence** B
OGD	sq. Euclidean norm	sq. Euclidean distance
EW	Shannon entropy	Kullback-Leibler divergence

Other entropies: Burg, Tsallis, Von Neumann, ... Connections to continuous exponential weights [van der Hoeven et al., 2018].

FTRL/MD “sneak peak” performance

Algorithm: Follow the Regularised Leader (FTRL)

$$\boldsymbol{w}_{t+1} \;=\; \operatorname*{arg\,min}_{\boldsymbol{u}\in\mathcal{U}} \; \sum_{s=1}^{t} \langle \boldsymbol{u}, \nabla f_s(w_s) \rangle + \frac{1}{\eta} R(\boldsymbol{u})$$

Algorithm: Mirror Descent

$$\boldsymbol{w}_{t+1} \;=\; \operatorname*{arg\,min}_{\boldsymbol{u}\in\mathcal{U}} \; \langle \boldsymbol{u}, \nabla f_t(w_t) \rangle + \frac{1}{\eta} B(\boldsymbol{u} \| \boldsymbol{w}_t)$$

FTRL/MD “sneak peak” performance

Algorithm: Follow the Regularised Leader (FTRL)

$$w_{t+1} = \underset{u \in \mathcal{U}}{\arg\min} \sum_{s=1}^{t} \langle u, \nabla f_s(w_s) \rangle + \frac{1}{\eta} R(u)$$

Algorithm: Mirror Descent

$$w_{t+1} = \underset{u \in \mathcal{U}}{\arg\min} \langle u, \nabla f_t(w_t) \rangle + \frac{1}{\eta} B(u \| w_t)$$

Theorem (AdaFTRL, Orabona and Pál 2015)

Fix a norm $\|\cdot\|$ with associated dual norm $\|\cdot\|_\star$. Let $R : \mathcal{U} \to [0, D^2]$ be strongly convex w.r.t. $\|\cdot\|$. AdaFTRL ensures

$$\mathcal{R}_T \leq 2D \sqrt{\sum_{t=1}^{T} \|\nabla f_t(w_t)\|_\star^2} + 2 \cdot \textit{loss range}.$$

Quadratic Losses

So far we used convexity to "linearise"

$$f_t(\boldsymbol{u}) \geq f_t(\boldsymbol{w}_t) + \langle \boldsymbol{u} - \boldsymbol{w}_t, \nabla f_t(\boldsymbol{w}_t) \rangle,$$

and our methods essentially operated on linear losses. But what if we know there is curvature?

- How to represent/quantify curvature?
- How to efficiently manipulate curvature?
- How much can we reduce the regret?

Curvature assumptions

Assumption: Quadratic loss lower bound

There is a matrix $M_t \succeq \mathbf{0}$ such that

$$f_t(\boldsymbol{u}) \;\geq\; \underbrace{f_t(\boldsymbol{w}_t) + \langle \boldsymbol{u} - \boldsymbol{w}_t, \nabla f_t(\boldsymbol{w}_t)\rangle + \frac{1}{2}(\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \boldsymbol{M}_t(\boldsymbol{u} - \boldsymbol{w}_t)}_{=: q_t(\boldsymbol{u})}$$

for each $\boldsymbol{u} \in \mathcal{U}$.

Curvature assumptions

Assumption: Quadratic loss lower bound

There is a matrix $M_t \succeq \mathbf{0}$ such that

$$f_t(\boldsymbol{u}) \;\geq\; \underbrace{f_t(\boldsymbol{w}_t) + \langle \boldsymbol{u} - \boldsymbol{w}_t, \nabla f_t(\boldsymbol{w}_t) \rangle + \frac{1}{2}(\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \boldsymbol{M}_t (\boldsymbol{u} - \boldsymbol{w}_t)}_{=: q_t(\boldsymbol{u})}$$

for each $\boldsymbol{u} \in \mathcal{U}$.

Two main classes of instances

- squared Euclidean distance: $f_t(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{u} - x_t\|^2$ satisfies the assumption with $\boldsymbol{M}_t = \boldsymbol{I}$. More generally, strongly convex functions have $\boldsymbol{M}_t \propto \boldsymbol{I}$.
- linear regression: $f_t(\boldsymbol{u}) = (y_t - \langle \boldsymbol{u}, \boldsymbol{x}_t \rangle)^2$ satisfies the assumption with $\boldsymbol{M}_t = \boldsymbol{x}_t \boldsymbol{x}_t^\intercal$. More generally, exp-concave functions have $\boldsymbol{M}_t \propto \nabla_t f_t(\boldsymbol{w}_t) \nabla_t f_t(\boldsymbol{w}_t)^\intercal$.

ONS Algorithm

Algorithm: Online Newton Step (FTRL variant)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathcal{U}} \sum_{s=1}^{t} q_s(\boldsymbol{u}) + \frac{1}{2} \|\boldsymbol{u}\|^2$$

Computing the iterate $\boldsymbol{w}_{t+1}$ amounts to minimising a convex quadratic. Often (depending on $\mathcal{U}$) closed-form solution or 1d line search.

- For $\boldsymbol{M}_t \propto \boldsymbol{I}$, takes $O(d)$ per round.
- For rank-one $\boldsymbol{M}_t$, can do update in $O(d^2)$ per round.
- In both cases, need to take care of projection onto $\mathcal{U}$.

ONS Performance

Algorithm: Online Newton Step (FTRL version)

$$\boldsymbol{w}_{t+1} \ = \ \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathcal{U}} \ \sum_{s=1}^{t} q_s(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u}\|^2$$

Theorem (ONS strcvx bd, Hazan et al. 2006)

For the strongly convex case $\boldsymbol{M}_t \propto \boldsymbol{I}$, ONS guarantees

$$\mathcal{R}_T \ = \ O(\ln T)$$

Algorithm reduces to OGD with specific decreasing step-size η_t

Theorem (ONS expccv bd, Hazan et al. 2006)

For the exp-concave case $\boldsymbol{M}_t \propto \boldsymbol{g}_t \boldsymbol{g}_t^\intercal$, ONS guarantees

$$\mathcal{R}_T \ = \ O(d \ln T)$$

ONS Discussion

- Convex quadratics closed under taking sums. Run-time independent of T.
- Curvature gives huge reduction in regret: $\sqrt{T}$ to $\ln T$.
- Matrix sketching techniques allow trading off run-time $O(d^2)$ vs $O(d)$ with regret $O(\ln T)$ vs $O(\sqrt{T})$ [Luo et al., 2016].

Applications

Application 1: Offline Optimisation

Problem

Given gradient access to a convex f, find a near-optimal point.

Application 1: Offline Optimisation

Problem

Given gradient access to a convex f, find a near-optimal point.

Idea: run OGD on $f_t = f$ for T rounds. Regret bound gives

$$\sum_{t=1}^{T} f(\boldsymbol{w}_t) - T \min_{\boldsymbol{u} \in \mathcal{U}} f(\boldsymbol{u}) \;\leq\; GD\sqrt{T}$$

We may divide by T and apply convexity to find

$$f\left(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{w}_t\right) - \min_{\boldsymbol{u} \in \mathcal{U}} f(\boldsymbol{u}) \;\leq\; \frac{GD}{\sqrt{T}}$$

Find ϵ-suboptimal point (iterate average) after $T = \frac{G^2 D^2}{\epsilon^2}$ rounds.

Application 1: Offline Optimisation

Problem

Given gradient access to a convex f, find a near-optimal point.

Idea: run OGD on $f_t = f$ for T rounds. Regret bound gives

$$\sum_{t=1}^{T} f(\boldsymbol{w}_t) - T \min_{\boldsymbol{u} \in \mathcal{U}} f(\boldsymbol{u}) \ \leq \ GD\sqrt{T}$$

We may divide by T and apply convexity to find

$$f\left(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{w}_t\right) - \min_{\boldsymbol{u} \in \mathcal{U}} f(\boldsymbol{u}) \ \leq \ \frac{GD}{\sqrt{T}}$$

Find ϵ-suboptimal point (iterate average) after $T = \frac{G^2 D^2}{\epsilon^2}$ rounds.

Why would we optimise this way? For example, what if $f_t \to f$.

Application 2: Online to Batch

Assumption: stochastic setting

Suppose training set $f_1, \ldots, f_T$ and test point f drawn i.i.d. from unknown $\mathbb{P}$.

Problem

Learn a point $\hat{\boldsymbol{w}}_T$ from the training set that generalises to $\mathbb{P}$, i.e. behaves like $\boldsymbol{u}^* = \arg\min_{\boldsymbol{u} \in \mathcal{U}} \mathbb{E}_f[f(\boldsymbol{u})]$

Application 2: Online to Batch

Assumption: stochastic setting

Suppose training set $f_1, \ldots, f_T$ and test point f drawn i.i.d. from unknown $\mathbb{P}$.

Problem

Learn a point $\hat{\boldsymbol{w}}_T$ from the training set that generalises to $\mathbb{P}$, i.e. behaves like $\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}\in\mathcal{U}} \mathbb{E}_f[f(\boldsymbol{u})]$

Idea: use online learning algorithm on training set $f_1, \ldots, f_T$, to get iterates $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_T$. Output the *average iterate estimator*

$$\hat{\boldsymbol{w}}_T = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{w}_t.$$

Theorem

An online regret bound $R_T \leq B(T)$ implies

$$\mathbb{E}_{iid\ f_1,\ldots,f_T,f}\left[f\left(\hat{\boldsymbol{w}}_T\right) - f(\boldsymbol{u}^*)\right] \leq \frac{B(T)}{T}$$

Application 3: Computing Saddle Points

Assumption: convex-concave

Fix an objective function

$$g(x, y)$$

convex in x, concave in y.

Application 3: Computing Saddle Points

Assumption: convex-concave

Fix an objective function

$$g(x, y)$$

convex in x, concave in y.

The game *value* is

$$V^* \;=\; \min_x \max_y g(x, y) \;=\; \max_y \min_x g(x, y).$$

An ϵ-*saddle point* $(\bar{x}, \bar{y})$ satisfies

$$V^* - \epsilon \leq \min_x g(x, \bar{y}) \leq V^* \leq \max_y g(\bar{x}, y) \leq V^* + \epsilon.$$

Application 3: Computing Saddle Points

Assumption: convex-concave

Fix an objective function

$$g(x, y)$$

convex in x, concave in y.

The game *value* is

$$V^* \;=\; \min_x \max_y g(x, y) \;=\; \max_y \min_x g(x, y).$$

An ϵ*-saddle point* $(\bar{x}, \bar{y})$ satisfies

$$V^* - \epsilon \le \min_x g(x, \bar{y}) \le V^* \le \max_y g(\bar{x}, y) \le V^* + \epsilon.$$

Problem

Find an ϵ*-saddle point*

Application 3: Computing Saddle Points

Assumption: convex-concave

Fix an objective function

$$g(x, y)$$

convex in x, concave in y.

The game *value* is

$$V^* \;=\; \min_x \max_y g(x, y) \;=\; \max_y \min_x g(x, y).$$

An ϵ*-saddle point* $(\bar{x}, \bar{y})$ satisfies

$$V^* - \epsilon \leq \min_x g(x, \bar{y}) \leq V^* \leq \max_y g(\bar{x}, y) \leq V^* + \epsilon.$$

Problem

Find an ϵ*-saddle point*

Idea: play regret minimisation algorithms for x and y.

Application 3: Saddle Point Algorithm

Algorithm: approximate saddle point solver

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

- Players play x_t and y_t.
- Players see loss functions $x \mapsto +g(x, y_t)$ and $y \mapsto -g(x_t, y)$.

Output average iterate pair $\bar{x}_T = \frac{1}{T}\sum_{t=1}^{T} x_t$ and $\bar{y}_T = \frac{1}{T}\sum_{t=1}^{T} y_t$

Assume the players have regret (bounds) $\mathcal{R}_T^x$ and $\mathcal{R}_T^y$, i.e.

$$\sum_{t=1}^{T} +g(x_t, y_t) - \min_x \sum_{t=1}^{T} +g(x, y_t) \;\leq\; \mathcal{R}_T^x$$

$$\sum_{t=1}^{T} -g(x_t, y_t) - \min_y \sum_{t=1}^{T} -g(x_t, y) \;\leq\; \mathcal{R}_T^y$$

Theorem (self-play, Freund and Schapire 1999)

$\bar{x}_T$ and $\bar{y}_T$ form an $\frac{\mathcal{R}_T^x + \mathcal{R}_T^y}{T}$-saddle point.

Application 3: Saddle Point Analysis

$$
\begin{aligned}
V^* &= \min_x \max_y g(x, y) \\
&\leq \max_y g(\bar{x}_T, y) \\
&\leq \max_y \frac{1}{T} \sum_{t=1}^{T} g(x_t, y) \\
&\leq \frac{1}{T} \sum_{t=1}^{T} g(x_t, y_t) + \frac{\mathcal{R}_T^y}{T} \\
&\leq \min_x \frac{1}{T} \sum_{t=1}^{T} g(x, y_t) + \frac{\mathcal{R}_T^x + \mathcal{R}_T^y}{T} \\
&\leq \min_x g(x, \bar{y}_T) + \frac{\mathcal{R}_T^x + \mathcal{R}_T^y}{T} \\
&\leq \min_x \max_y g(x, y) + \frac{\mathcal{R}_T^x + \mathcal{R}_T^y}{T} \\
&= V^* + \frac{\mathcal{R}_T^x + \mathcal{R}_T^y}{T}
\end{aligned}
$$

Conclusion and Extensions

Conclusion

- Online Learning a powerful and versatile tool
- Environment-as-black-box. Adversarial.
- Foundation for optimisation, statistical learning, games, ...

Conclusion

- Online Learning a powerful and versatile tool
- Environment-as-black-box. Adversarial.
- Foundation for optimisation, statistical learning, games, ...

Some (of many) cool things we left out:

- First-order (small loss) and second-order (small variance) bounds
- Adaptivity to friendly stochastic environments (best of both worlds, interpolation)
- Optimism (predicting the upcoming gradient)
- Non-stationarity (tracking, adaptive/dynamic regret, path length)
- Beyond convexity (star-convex, geometrically convex, ...)
- Supervised Learning and (stochastic) complexities (VC, Littlestone, Rademacher, ...)

Thanks!

References

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, August 1997.

Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

Elad Hazan, Adam Kalai, Satyen Kale, and Amit Agarwal. Logarithmic regret algorithms for online convex optimization. In *Learning Theory*, pages 499–513, 2006.

Haipeng Luo, Alekh Agarwal, Nicolò Cesa-Bianchi, and John Langford. Efficient second order online learning by sketching. In *Advances in Neural Information Processing Systems 29*, pages 902–910. 2016.

Francesco Orabona and Dávid Pál. Scale-free algorithms for online linear optimization. In *Algorithmic Learning Theory*, pages 287–301, 2015.

Dirk van der Hoeven, Tim van Erven, and Wojciech Kotłowski. The many faces of exponential weights in online learning. volume 75 of *Proceedings of Machine Learning Research*, pages 2067–2092, 06–09 Jul 2018.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, page 928–935, 2003.

Simons Tutorial: Online Learning and Bandits Part II

Wouter Koolen and **Alan Malek**

August 31st, 2020

What is a Bandit?

The Basic Bandit Game

Protocol: Finite-Arm Bandits

Given: game length T, number of arms K

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

- The learner picks action $I_t \in \{1, \ldots, K\}$
- The adversary simultaneously picks rewards $r_t \in \{1, \ldots, K\} \to [0, 1]$
- The learner observes and receives $r_t(I_t)$
- The learner does not observe $r_t(i)$ for $i \neq I_t$

The goal: control the regret (a random variable)

$$\mathcal{R}_T = \underbrace{\max_i \sum_{t=1}^{T} r_t(i)}_{\text{Best action in hindsight}} - \sum_{t=1}^{T} r_t(I_t)$$

Bandits are Super Simple MDP

- $S = \{\text{the\_state}\}$, $P(\text{the\_state}|\text{the\_state}, a) = 1$
- Why should we care about this in RL?
  - Creates a tension between
    - Exploration (learning about the loss of actions)
    - Exploitation (playing actions that will have low regret)
  - Exploration/Exploitation is absent in full-information but very present in reinforcement learning
  - Model is simple enough to allow for comprehensive theory
  - Easily incorporates adversarial data
  - Useful algorithm design principles

The Regret

$$\mathcal{R}_T = \underbrace{\max_i \sum_{t=1}^{T} r_t(i)}_{\text{Best action in hindsight}} - \sum_{t=1}^{T} r_t(I_t)$$

- $\mathcal{R}_T$ is a random variable we do not observe
- Different objectives, from easiest to hardest
  - Pseudo-regret $\overline{\mathcal{R}_T} = \max_i \mathbb{E}\left[\sum_{t=1}^{T} r_t(i)\right] - \mathbb{E}\left[\sum_{t=1}^{T} r_t(I_t)\right]$
  - Expected regret $\mathbb{E}[\mathcal{R}_T] = \mathbb{E}\left[\underbrace{\max_i \sum_{t=1}^{T} r_t(i)}_{\text{can depend on } I_t} - \sum_{t=1}^{T} r_t(I_t)\right]$
  - High probability bounds on the realized regret

The Regret

$$\mathcal{R}_T = \underbrace{\max_i \sum_{t=1}^{T} r_t(i)}_{\text{Best action in hindsight}} - \sum_{t=1}^{T} r_t(I_t)$$

- $\mathcal{R}_T$ is a random variable we do not observe
- Different objectives, from easiest to hardest
  - Pseudo-regret $\overline{\mathcal{R}_T} = \max_i \mathbb{E}\left[\sum_{t=1}^{T} r_t(i)\right] - \mathbb{E}\left[\sum_{t=1}^{T} r_t(I_t)\right]$
  - Expected regret $\mathbb{E}[\mathcal{R}_T] = \mathbb{E}\left[\underbrace{\max_i \sum_{t=1}^{T} r_t(i)}_{\text{can depend on } I_t} - \sum_{t=1}^{T} r_t(I_t)\right]$
  - High probability bounds on the realized regret
- We always have $\overline{\mathcal{R}_T} \leq \mathbb{E}[\mathcal{R}_T]$
- If the adversary is *reactive*, then the distribution of r_t can be a function of $I_1, \ldots, I_{t-1}$
- Otherwise, the adversary is *oblivious* and $\overline{\mathcal{R}_T} = \mathbb{E}[\mathcal{R}_T]$

Our Focus

- Introduce most popular bandit problems
  - Adversarial Bandits
  - Stochastic Bandits
  - Pure Exploration Bandits
  - Contextual Bandits (time permitting)
- Concentrate on useful algorithm design principles
  - Exponential weights (still useful)
  - Optimism in the face of Uncertainty
  - Probability matching (i.e. Thompson sampling)
  - Action-Elimination

Other Settings that Have Been Considered

- Data models for r_t
  - chosen by an adversary
  - sampled i.i.d.
  - stochastic with adversarial perturbations...
- Action spaces
  - Finite number of arms
  - A vector space (r_t are functions)
  - Combinatorial (e.g. subsets, paths on a graph)
- Objectives
  - Pseudo-regret (the expectation over the learner's randomness)
  - Realized regret (with high probability)
  - Best-arm identification a.k.a. pure exploration
- Side information
  - Linear rewards
  - Competing with a policy class
- ...

Adversarial Bandits

Adversarial Protocol

Protocol: Finite-Arm Adversarial Bandits

Given: game length T, number of arms K

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

- The learner picks action $I_t \in \{1, \dots, K\}$
- The adversary simultaneously picks losses $\ell_t \in [0, 1]^K$
- The learner observes and receives $\ell_t(I_t)$

- The results are easier to state using losses instead of rewards
- Randomization of I_t is *essential*
- We are familiar with adversarial data from the first half
- The simple idea of estimating ℓ_t from $\ell_t(I_t)$ and then applying a full-information algorithm works very well

Algorithm Design Principle: Exponential Weights

Algorithm: Exp3 [Auer et al., 2002b]

Given: number of arms K, learning rate $\eta > 0$, length T

Initialize $p_1(i) = 1/K$, $\hat{L}_0(i) = 0$ for all $i \in [K]$

For $t = 1, 2, \ldots, T$:

- Sample $I_t \sim p_t$ and observe $\ell_t(I_t)$
- Estimate $\hat{\ell}_t(i) = \frac{\ell_t(I_t)}{p_t(I_t)} \mathbb{1}_{\{I_t = i\}}$ and $\hat{L}_t = \hat{\ell}_t + \hat{L}_{t-1}$
- Calculate $W_t = \sum_j e^{-\eta \hat{L}_t(j)}$ and $p_{t+1}(i) = \frac{1}{W_t} e^{-\eta \hat{L}_t(i)}$

Algorithm Design Principle: Exponential Weights

Algorithm: Exp3 [Auer et al., 2002b]

Given: number of arms K, learning rate $\eta > 0$, length T

Initialize $p_1(i) = 1/K$, $\hat{L}_0(i) = 0$ for all $i \in [K]$

For $t = 1, 2, \ldots, T$:

- Sample $I_t \sim p_t$ and observe $\ell_t(I_t)$
- Estimate $\hat{\ell}_t(i) = \frac{\ell_t(I_t)}{p_t(I_t)} \mathbb{1}_{\{I_t = i\}}$ and $\hat{L}_t = \hat{\ell}_t + \hat{L}_{t-1}$
- Calculate $W_t = \sum_j e^{-\eta \hat{L}_t(j)}$ and $p_{t+1}(i) = \frac{1}{W_t} e^{-\eta \hat{L}_t(i)}$

- Exp3 = Exponential Weights for Exploration and Exploitation
- $\hat{\ell}_t$ is the importance-weighted estimator of ℓ_t
- $\hat{\ell}_t$ is unbiased:

$$\mathbb{E}_{I_t \sim p_t}[\hat{\ell}_t(i)] = \mathbb{E}\left[\frac{\ell_t(I_t)}{p_t(I_t)} \mathbb{1}_{\{I_t = i\}}\right] = \sum_j p_t(j) \frac{\ell_t(j)}{p_t(j)} \mathbb{1}_{\{j = i\}} = \ell_t(i).$$

- Exp3 runs exponential weights on $\hat{\ell}_t$

Exp3: Abridged Analysis

- Using the same $\frac{W_t}{W_{t-1}}$ telescoping procedure as in the full information case with i^* arbitrary but fixed,

$$\sum_{t=1}^{T}\sum_{j} p_t(j)\mathbb{E}[\hat{\ell}_t(j) - \hat{L}_T(i^*)] \leq \frac{\log(K)}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\mathbb{E}\left[\sum_{j} p_t(j)\hat{\ell}_t(j)^2\right].$$

Exp3: Abridged Analysis

- Using the same $\frac{W_t}{W_{t-1}}$ telescoping procedure as in the full information case with i^* arbitrary but fixed,

$$\sum_{t=1}^{T} \sum_{j} p_t(j) \mathbb{E}[\hat{\ell}_t(j) - \hat{L}_T(i^*)] \leq \frac{\log(K)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \mathbb{E}\left[\sum_{j} p_t(j) \hat{\ell}_t(j)^2\right].$$

- Because $\hat{\ell}_t$ is unbiased,

$$\sum_{t=1}^{T} \sum_{j} p_t(j) \mathbb{E}[\hat{\ell}_t(j) - \hat{L}_T(i^*)] = \sum_{t=1}^{T} \sum_{j} p_t(j) \ell_t(j) - L_T(i^*) \geq \overline{\mathcal{R}}_T.$$

Exp3: Abridged Analysis

- Using the same $\frac{W_t}{W_{t-1}}$ telescoping procedure as in the full information case with i^* arbitrary but fixed,

$$\sum_{t=1}^{T}\sum_{j} p_t(j)\mathbb{E}[\hat{\ell}_t(j) - \hat{L}_T(i^*)] \leq \frac{\log(K)}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\mathbb{E}\left[\sum_{j} p_t(j)\hat{\ell}_t(j)^2\right].$$

- Because $\hat{\ell}_t$ is unbiased,

$$\sum_{t=1}^{T}\sum_{j} p_t(j)\mathbb{E}[\hat{\ell}_t(j) - \hat{L}_T(i^*)] = \sum_{t=1}^{T}\sum_{j} p_t(j)\ell_t(j) - L_T(i^*) \geq \overline{\mathcal{R}}_T.$$

- Bounding the variance term turns out to be easy:

$$\mathbb{E}\left[\sum_{j} p_t(j)\frac{\ell_t(I_t)^2}{p_t(I_t)^2}\mathbb{1}_{\{I_t=i\}}\right] \leq \mathbb{E}\left[\sum_{j} p_t(j)\frac{\mathbb{1}_{\{I_t=i\}}}{p_t(I_t)^2}\right] = \mathbb{E}\left[\frac{1}{p_t(I_t)}\right] = K.$$

Exp3: Analysis

So, plugging this in, we find

$$\sum_{t=1}^{T} \mathbb{E}[\ell(I_t)] - L_T(i^*) \leq \frac{\log(K)}{\eta} + \frac{\eta}{2} TK.$$

Theorem (Exp3 upper bound [Auer et al., 2002b])

With $\eta = \sqrt{\frac{2\log(T)}{TK}}$*, Exp3 has* $\overline{\mathcal{R}}_T \leq \sqrt{2TK\log(K)}$.

Only get pseudo-Regret bounds because the i^* in the proof was fixed, not a function of $I_1, \ldots, I_T$

Lower Bounds

Theorem (Adversarial Bandits lower bound [Auer et al., 2002b])

Any adversarial bandit algorithm must have

$$\overline{\mathcal{R}}_T = \Omega(\sqrt{TK})$$

- Exp3 upper bound: $\overline{\mathcal{R}}_T \leq \sqrt{2TK \log(K)}$
- First matching upper bound achieved by INF [Audibert and Bubeck, 2009] (which is Mirror Descent)

Upgrades

- High Probability bounds: requires a lower-variance estimate of $\hat{\ell}_t$ or an algorithm that keeps $p_t(i)$ away from zero
  - Exp3.P [Auer et al., 2002b] uses $\hat{\ell}_t(i) = \frac{\mathbb{1}_{\{I_t=i\}}\ell_t(I_t)-\beta}{p_t(I_t)}$
  - Exp3-IX [Neu, 2015] uses $\hat{\ell}_t(i) = \frac{\mathbb{1}_{\{I_t=i\}}\ell_t(I_t)}{p_t(I_t)+\gamma}$
- Experts with bandits; each arm is an expert that recommends actions: you compete with the best expert (Exp4 algorithm) [Auer et al., 2002b]
- Competing with strategies that can switch [Auer, 2002]
- Feedback determined by a graph [Mannor and Shamir, 2011]
- Partial Monitoring [Bartók et al., 2014]
- Combinatorial action spaces...

Stochastic Bandits

Protocol

Protocol: Stochastic Bandits

Given: game length T, number of arms K, reward distributions $\nu_1, \ldots, \nu_K$

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

- The learner picks action $I_t \in \{1, \ldots, K\}$
- The learner observes and receives reward $X_t \sim \nu_{I_t}$

Protocol

Protocol: Stochastic Bandits

Given: game length T, number of arms K, reward distributions $\nu_1, \ldots, \nu_K$

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

- The learner picks action $I_t \in \{1, \ldots, K\}$
- The learner observes and receives reward $X_t \sim \nu_{I_t}$

- Stochastic bandits is an old problem [Thompson, 1933]
- We will use the following notation
  - Reward of arm i is sampled from ν_i with $\mu_i := \mathbb{E}_{X \sim \nu_i}[X]$
  - $i^* = \arg\max_i \mu_i$ is the best arm
  - Gaps $\Delta_i := \mu_{i^*} - \mu_i \geq 0$,
  - Number of pulls $N_{i,t} := \sum_{s=1}^{t} \mathbb{1}_{\{I_s = i\}}$
  - Empirical mean $\hat{\mu}_{i,t} := \frac{\sum_{s=1}^{t} X_s \mathbb{1}_{\{I_s = i\}}}{N_{i,t}}$

Protocol

Protocol: Stochastic Bandits

Given: game length T, number of arms K, reward distributions $\nu_1, \ldots, \nu_K$

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

- The learner picks action $I_t \in \{1, \ldots, K\}$
- The learner observes and receives reward $X_t \sim \nu_{I_t}$

- We still want to minimize the expected regret, which has the useful decomposition
$$\mathbb{E}[\mathcal{R}_T] = T\mu_{i^*} - \sum_{t=1}^T \mathbb{E}[X_t] = \sum_i \Delta_i \mathbb{E}[N_{i,T}]$$

Protocol

Protocol: Stochastic Bandits

Given: game length T, number of arms K, reward distributions $\nu_1, \ldots, \nu_K$

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

- The learner picks action $I_t \in \{1, \ldots, K\}$
- The learner observes and receives reward $X_t \sim \nu_{I_t}$

- We still want to minimize the expected regret, which has the useful decomposition
$$\mathbb{E}[\mathcal{R}_T] = T\mu_{i^*} - \textstyle\sum_{t=1}^{T} \mathbb{E}[X_t] = \sum_i \Delta_i \mathbb{E}[N_{i,T}]$$

Assumption: 1-sub-Gaussian reward distributions

For all stochastic bandit problems, we will assume that all arms are 1-sub-Gaussian, i.e. $\mathbb{E}_{X\sim\mu}[e^{\lambda(X-\mu)^2-\lambda^2/2}] \leq 1$. For $X_1, \ldots, X_t$, This implies the Hoeffding bound

$$P\left(\frac{1}{t}\sum_{s=1}^{t} X_s - \mu_i \geq \epsilon\right) \leq e^{-\frac{\epsilon^2 t}{2}}.$$

Warm-up: Explore-Then-Commit

Algorithm: Explore-Then-Commit

Given: Game length T, exploration parameter M

For $t = 1, 2, \ldots, MK$:

- Choose $i_t = (t \mod K)$, see $X_t \sim \nu_{i_t}$

Compute empirical means $\hat{\mu}_{i,mK}$

For $t = MK + 1, MK + 2, \ldots, T$:

- Pull arm $i = \arg\max_i \hat{\mu}_{i,mK}$

- The first strategy you might try
- A proof idea that we will return to: bound regret by first bounding $\mathbb{E}[N_{i,T}]$.
- In this simple algorithm,

$$\mathbb{E}[N_{i,T}] = M + (T - MK)P\left(i = \arg\max_j \hat{\mu}_{j,MK}\right)$$

Explore-Then-Commit Upper Bound

Using the sub-Gaussian concentration bound,

$$
\begin{aligned}
P\left(i = \arg\max_j \hat{\mu}_{j,MK}\right) &\leq P\left(\hat{\mu}_{i,MK} \geq \hat{\mu}_{i^*,MK}\right) \\
&= P\left((\hat{\mu}_{i,MK} - \mu_i) \geq (\hat{\mu}_{i^*,MK} - \mu_{i^*}) + \Delta_i\right) \\
&\leq e^{-\frac{M\Delta_i^2}{4}} \text{ (the difference is } \sqrt{2/M}\text{-sub-Gaussian)}
\end{aligned}
$$

Explore-Then-Commit Upper Bound

Using the sub-Gaussian concentration bound,

$$P\left(i = \arg\max_j \hat{\mu}_{j,MK}\right) \leq P\left(\hat{\mu}_{i,MK} \geq \hat{\mu}_{i^*,MK}\right)$$

$$= P\left((\hat{\mu}_{i,MK} - \mu_i) \geq (\hat{\mu}_{i^*,MK} - \mu_{i^*}) + \Delta_i\right)$$

$$\leq e^{-\frac{M\Delta_i^2}{4}} \text{ (the difference is } \sqrt{2/M}\text{-sub-Gaussian)}$$

Theorem (Explore-Then-Commit upper bound)

$$\mathbb{E}[\mathcal{R}_T] = \sum_i \Delta_i \mathbb{E}[N_{i,T}] \leq \sum_{i=1}^K \Delta_i \left(M + (T - MK)e^{-\frac{M\Delta_i^2}{4}}\right)$$

- For the two arm case, if we know Δ, then $m = \frac{4}{\Delta_1^2} \log \frac{T\Delta_1^2}{4}$, results in $\mathbb{E}[\mathcal{R}_T] \leq \sum_{i=1}^K \frac{4}{\Delta_1} \log \frac{T\Delta_1^2}{4} + T\frac{4}{T\Delta_1^2} = O\left(\frac{K\log(T)}{\Delta_1}\right)$
- But we don't know Δ...can we be adaptive?

Algorithm Design Principle: OFU

- OFU: Optimism in the Face of Uncertainty
- We establish some confidence set for the problem instance (e.g. means) to within some confidence set
- We then assume the most favorable instance in the confidence set and act greedily

Algorithm Design Principle: OFU

- OFU: Optimism in the Face of Uncertainty
- We establish some confidence set for the problem instance (e.g. means) to within some confidence set
- We then assume the most favorable instance in the confidence set and act greedily

Algorithm: UCB1 [Auer et al., 2002a]

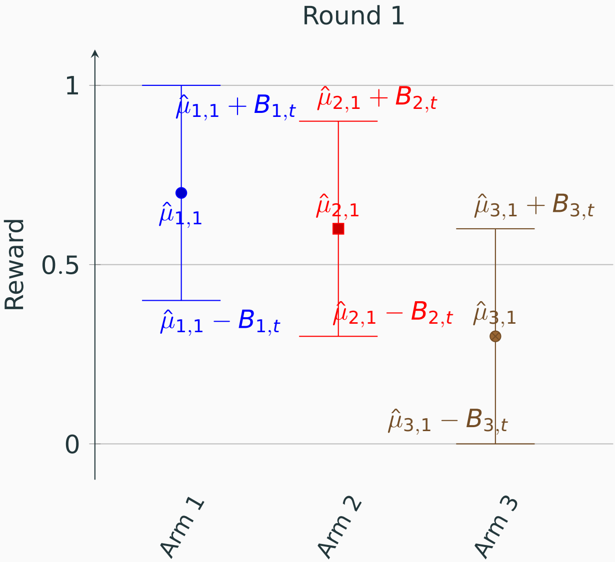
Given: Game length T

Initialize: play every arm once

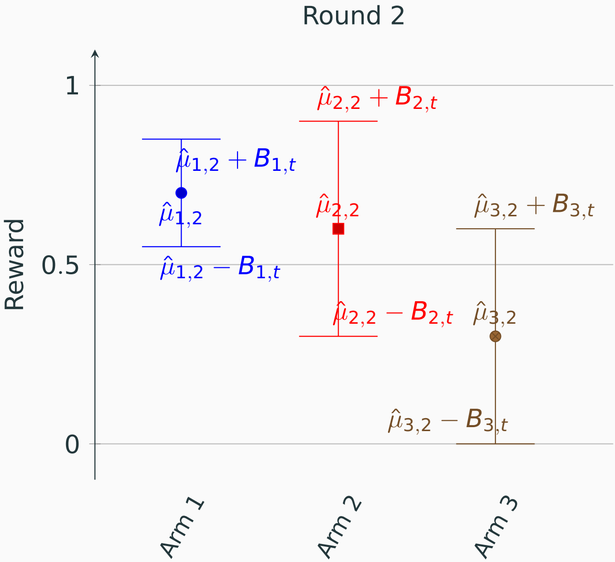
For $t = K+1, 2, \dots, T$:

- Compute upper confidence bounds $B_{i,t-1} = \sqrt{\frac{6\log(t)}{N_{i,t-1}}}$
- Choose $I_t = \arg\max_i \hat{\mu}_{i,t-1} + B_{i,t-1}$, observe $X_t \sim \nu_{I_t}$
- Update $N_{i,t} = N_{i,t-1} + \mathbb{1}_{\{I_t=i\}}$ and $\hat{\mu}_{i,t} = \frac{\sum_{s=1}^{t} \mathbb{1}_{\{I_s=i\}} X_s}{N_{i,t}}$

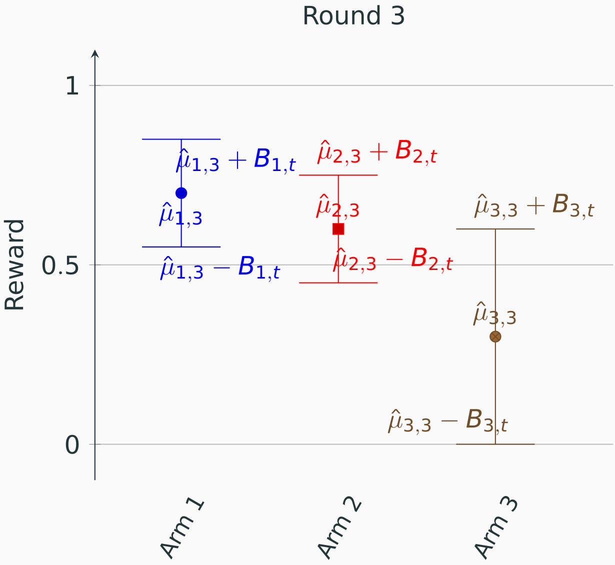
UCB Illustration



UCB Illustration



UCB Illustration



UCB: Intuition

- Naturally balances exploration and exploitation: an arm has a high UCB if
  - It has a high $\hat{\mu}_{i,t}$, or
  - $B_{i,t}$ is large because $N_{i,t-1}$ is small
- Optimistic because we pretend the rewards are the plausibly best and then do the greedy thing

UCB: Analysis

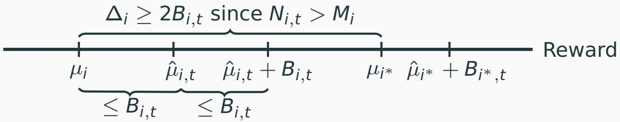
- Define $M_i = \left\lceil \frac{12\log(T)}{\Delta_i^2} \right\rceil$, the number of pulls of arm i such that $B_{i,t} = \sqrt{\frac{6\log(t)}{N_{i,t}}} \leq \sqrt{\frac{6\log(T)}{N_{i,t}}} \leq \frac{\Delta_i}{2}$
- The intuition of the proof is
  1. Since $\overline{\mathcal{R}_T} = \sum_i \Delta_i \mathbb{E}[N_{i,T}]$, we bound $\mathbb{E}[N_{i,t}]$ first.
  2. With high probability, we will never pull arm i more than M_i times, so

$$\mathbb{E}[N_{i,T}] = \mathbb{E}\sum_{t=1}^{T} \mathbb{1}_{\{I_t=i\}} \leq M_i + \sum_{t=M_i}^{T} \underbrace{\mathbb{E}\mathbb{1}_{\{I_t=i, N_{i,t}>M_i\}}}_{\text{we will bound this}}$$

  3. If $\{I_t = i, N_{i,t} > M_i\}$ occurs, then the UCB for i^* or for i must be wrong (next slide)

UCB: Analysis

Claim: if $\{I_t = i, N_{i,t} > M_i\}$ occurs, then either $\hat{\mu}_{i,t}$ must be too high or $\hat{\mu}_{i^*,t}$ must be too low. In a picture:



In an equation: suppose that $N_{i,t} > M_i$, $\hat{\mu}_{i,t} - B_{i,t} \geq \mu_i$, and $\hat{\mu}_{i^*,t} + B_{i^*,t} \geq \mu_{i^*}$. Then

$$\hat{\mu}_{i^*,t} + B_{i^*,t} \geq \mu_{i^*} = \mu_i + \Delta_i \geq \mu_i + \underbrace{2B_{i,t}}_{\text{by choice of } B_{i,t}} \geq \hat{\mu}_{i,t} + B_{i,t},$$

so the algorithm will not choose $I_t = i$.

If $I_t = i$, at least one of the bounds must be wrong, implying

$$P(I_t = i, N_{i,t} > M_i) \leq P\left(\hat{\mu}_{i,t} \leq \mu_i + B_{i,t}\right) + P\left(\hat{\mu}_{i^*,t} + B_{i^*,t} \leq \mu_{i^*}\right).$$

UCB: Analysis

Using the Hoeffding bound,

$$\begin{aligned}
P\left(\hat{\mu}_{i,t} - \mu_i \le B_{i,t}\right) &\le P\Bigg(\underbrace{\exists s \le t}_{\text{we don't know } N_{i,t-1}} : \hat{\mu}_{i,s} - \mu_i \le \sqrt{\frac{6\log(t)}{s}}\Bigg) \\
&\le \sum_{s=1}^{t} P\left(\hat{\mu}_{i,s} - \mu_i \le \sqrt{\frac{6\log(t)}{s}}\right) \\
&\le \sum_{s=1}^{t} \exp\left\{-\frac{3\log(t)}{s}\right\} \le \sum_{s=1}^{t} t^{-3} = t^{-2}.
\end{aligned}$$

UCB: Analysis

Using the Hoeffding bound,

$$
\begin{aligned}
P\left(\hat{\mu}_{i,t} - \mu_i \leq B_{i,t}\right) &\leq P\Bigg(\underbrace{\exists s \leq t}_{\text{we don't know } N_{i,t-1}} : \hat{\mu}_{i,s} - \mu_i \leq \sqrt{\frac{6\log(t)}{s}}\Bigg) \\
&\leq \sum_{s=1}^{t} P\left(\hat{\mu}_{i,s} - \mu_i \leq \sqrt{\frac{6\log(t)}{s}}\right) \\
&\leq \sum_{s=1}^{t} \exp\left\{-\frac{3\log(t)}{s}\right\} \leq \sum_{s=1}^{t} t^{-3} = t^{-2}.
\end{aligned}
$$

The same inequality holds for i^*, so

$$
\overline{\mathcal{R}_T} = \sum_i \Delta_i \mathbb{E}[N_{i,T}] \leq \sum_i \Delta_i \left(\frac{12\log(T)}{\Delta_i^2} + 2\sum_{t=M_i+1}^{T} t^{-2}\right).
$$

UCB: Analysis

Theorem (UCB upper bound [Auer, 2002])

The UCB1 algorithm on 1-sub-Gaussian data has

$$\overline{\mathcal{R}_T} \leq \sum_i \frac{12\log(T)}{\Delta_j} + o(1).$$

UCB: Analysis

Theorem (UCB upper bound [Auer, 2002])

The UCB1 algorithm on 1-sub-Gaussian data has

$$\overline{\mathcal{R}_T} \leq \sum_i \frac{12 \log(T)}{\Delta_j} + o(1).$$

Theorem (Lower Bound [Lai and Robbins, 1985])

Suppose we have a parametric family P_θ and $\theta_1, \ldots, \theta_k$. For any "admissible" algorithm,

$$\liminf_{T \to \infty} \frac{\overline{R_T}}{\log(T)} \geq \sum_{i \neq i*} \frac{\Delta_i}{KL(P_{\theta_i}, P_{\theta_{i*}})} \approx O\left(\sum_{i \neq i*} \frac{1}{\Delta_i}\right)$$

E.g. if P_θ is Bernoulli, then $\frac{(\theta_i - \theta_{i*})^2}{\theta_{i*}(1-\theta_{i*})} \geq KL(P_{\theta_i}, P_{\theta_{i*}}) \geq 2(\theta_i - \theta_{i^*})^2$.

Algorithm Design Principle: Probability Matching

- We put a prior π over means μ_i and a likelihood $\nu_i = P(\cdot|\mu_i)$ over rewards
- Choose $P(I_t = i) = P(\mu_i = \mu_{i^*}|history)$ (the matching)
- We usually pick conjugate models (e.g. $\mu_i \sim N(0, 1)$, $X_t \sim N(\mu_i, 1)$)

Algorithm Design Principle: Probability Matching

- We put a prior π over means μ_i and a likelihood $\nu_i = P(\cdot|\mu_i)$ over rewards
- Choose $P(I_t = i) = P(\mu_i = \mu_{i^*}|history)$ (the matching)
- We usually pick conjugate models (e.g. $\mu_i \sim N(0, 1)$, $X_t \sim N(\mu_i, 1)$)

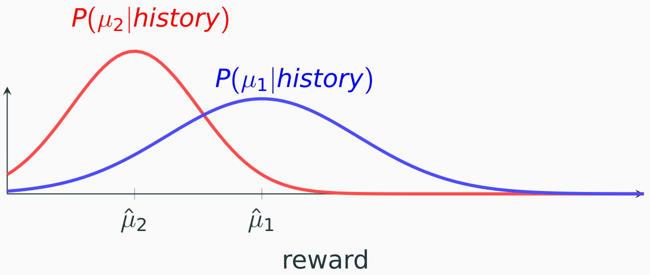
Algorithm: Thompson Sampling

Given: game length T, prior $\pi(\mu)$, likelihoods $p(\cdot|\mu)$
Initialize posteriors $p_{i,0}(\mu) = \pi(\mu)$

For $t = 1, 2, \ldots, T$:

- Draw $\theta_{i,t} \sim p_{i,t-1}$ for all i
- Choose $I_t = \arg\max_i \theta_{i,t}$ (implements the matching)
- Receive and observe $X_t \sim \nu_{I_t}$
- Update the posterior $p_{I_T,t}(\mu) = p(X_t|\mu)p_{I_t,t-1}(\mu)$

Thompson Sampling: Overview



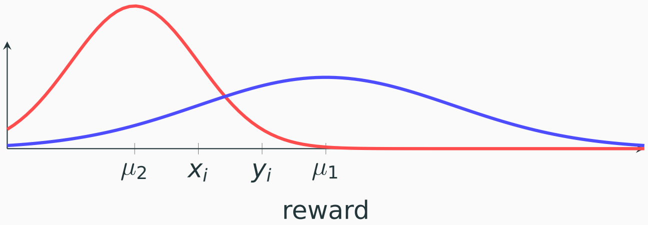
- *Not* Bayesian: a Bayesian method would maximize the Bayes regret (the expectation under the probability model)
- The regret bound is frequentist
- Arms with small $N_{i,t}$ implies a wide posterior, hence a good probability of being selected
- Generally performs empirically better that UCB (it is much more aggressive)
- Analysis is difficult

Thompson Sampling: Upper Bound

Theorem (Agrawal and Goyal [2013])

For binary rewards, Gamma-Beta Thompson sampling has $\mathbb{E}[R_T] \leq (1+\epsilon)\sum_{i \neq i^*} \Delta_i \frac{\log(T)}{KL(\mu_i, \mu_{i^*})} + O\left(\frac{N}{\epsilon^2}\right)$.

- The proof is much more technical that UCB's
- We cannot rely on the upper bounds being correct w.h.p.



For some to-be-tuned $\mu_i \leq x_i \leq y_i \leq \mu_{i^*}$, we have

$$
\begin{aligned}
\mathbb{E}[N_{i,T}] &\leq \sum_{t=1}^{T} P(I_t = i) \\
&\leq \sum_{t=1}^{T} P(I_t = i, \hat{\mu}_{i,t-1} \leq x_i, \theta_{i,t} \geq y_i) \qquad \left(O\left(\frac{\log(T)}{kl(x_j, y_j)}\right)\right) \\
&\quad + \sum_{t=1}^{T} P(I_t = i, \hat{\mu}_{i,t-1} \leq x_i, \theta_{i,t} \leq y_i) \quad \text{(the tricky case)} \\
&\quad + \sum_{t=1}^{T} P(I_t = i, \hat{\mu}_{i,t-1} \geq x_i) \qquad \text{(Small by concentration)}
\end{aligned}
$$

Thompson Sampling: Proof Outline

- The tricky case is $\sum_{t=1}^{T} P(I_t = i, \hat{\mu}_{i,t-1} \leq x_i, \theta_{i,t} \leq y_i)$
- This happens when we have enough samples of i but not many of i^*
- A key lemma argues that, on $\hat{\mu}_{i,t-1} \leq x_i, \theta_{i,t} \leq y_i$, the probability of picking i is a constant less than of picking i^*:

$$\sum_{t=1}^{T} P(I_t = i, \hat{\mu}_{i,t-1} \leq x_i, \theta_{i,t} \leq y_i)$$

$$\leq \sum_{t=1}^{T} \underbrace{\frac{P(\theta_{i^*,t} \leq y_i)}{P(\theta_{i^*,t} > y_j)}}_{\text{exponentially small}} P(I_t = i^*, \hat{\mu}_{i,t-1} \leq x_i, \theta_{i,t} \leq y_i) = O(1)$$

- Hence, we will quickly get enough samples of i^*

Best of Both Worlds

- The stochastic and adversarial algorithms are quite different
- A natural question: is there an algorithm that
  - gets $\mathcal{R}_T = O(\sqrt{TK})$ regret for adversarial
  - gets $\mathcal{R}_t = O(\sum_i \log(T)/\Delta_i)$ regret for stochastic
  - without knowing the setting?

Best of Both Worlds

- The stochastic and adversarial algorithms are quite different
- A natural question: is there an algorithm that
  - gets $\mathcal{R}_T = O(\sqrt{TK})$ regret for adversarial
  - gets $\mathcal{R}_t = O(\sum_i \log(T)/\Delta_i)$ regret for stochastic
  - without knowing the setting?
- Bubeck and Slivkins [2012] proposed an algorithm that assumes stochastic but falls back to UCB once adversarial data is detected
- Zimmert and Seldin [2019] showed that (for pseudo-regret), it is possible
  - Their algorithm: online mirror descent with $\frac{1}{2}$-Tsallis entropy
  - $\Psi(w) = -\sum_i 4(\sqrt{w_i} - \frac{1}{2}w_i)$

Pure Exploration

A New Problem

- What if we only wanted to identify the best arm i^* without caring about loss along the way?
- Intuitively, we would explore more; we are happy to accrue less reward if we get more useful samples.
- More similar to hypothesis testing; useful for selecting treatments
- Known as “Best Arm Identification” or “Pure Exploration”

Two Settings

Protocol: Best-arm Identification

Given:number of arms K, arm distributions $\nu_1, \ldots, \nu_K$

For $t = 1, 2, \ldots,$

- The learner picks arm $I_t \in \{1, \ldots, K\}$
- The learner observes $X_t \sim \nu_{I_t}$
- The learner decides whether to stop

The learner returns arm A

Two settings:

	fixed-confidence	*fixed-budget*
Input	$\delta > 0$,	T
Goal	$P(A = i^*) \geq 1 - \delta$	maximize $P(A = i^*)$
Stopping	once learner is confident	after T rounds

- Standard stochastic bandit algorithms under explore (they fail to meet lower bounds on this problem)
- Many can be adapted
  - LUCB [Kalyanakrishnan et al., 2012]
  - Top-Two Thompson Samping [Russo, 2016]
- Instead, we will describe a new algorithm design principle

Algorithm Design Principle: Action Elimination

Algorithm: Successive Elimination

Given: confidence $\delta > 0$

Initialize plausibly-best set $S = \{1, \ldots, K\}$

For $t = 1, 2, \ldots$:

- Pull all arms in S and update $\hat{\mu}_{i,t}$
- Calculate $B_t = \sqrt{2t^{-1}\log(4Kt^2/\delta)}$
- Remove i from S if $\underbrace{\max_{j \in S} \hat{\mu}_{j,t} - B_t}_{\text{Lowest } \mu_i^* \text{ could be}} \geq \underbrace{\hat{\mu}_{i,t} + B_t}_{\text{highest } \mu_i \text{ could be}}$
- If $|S| = 1$, stop and return $A = S$.

- S is a list of plausibly-best arms
- Each epoch, all arms that cannot be the best (if the bounds hold) are removed

Successive Elimination Analysis

- Define the "bad event" $\mathcal{E} = \bigcup_{i,t}\{|\hat{\mu}_{i,t} - \mu_i| \geq B_t(\delta)\}$: we have

$$P(\mathcal{E}) \leq \sum_{i,t} P\left(|\hat{\mu}_{i,t} - \mu_i| \geq \sqrt{2t^{-1}\log(4Kt^2/\delta)}\right) \leq \sum_{i,t} 2e^{-\log\left(\frac{4Kt^2}{\delta}\right)}$$

$$\leq \sum_{i,t} \frac{2\delta}{4Kt^2} = \frac{2\pi^2}{24}\delta \leq \delta$$

Successive Elimination Analysis

- Define the "bad event" $\mathcal{E} = \bigcup_{i,t}\{|\hat{\mu}_{i,t} - \mu_i| \geq B_t(\delta)\}$: we have

$$P(\mathcal{E}) \leq \sum_{i,t} P\left(|\hat{\mu}_{i,t} - \mu_i| \geq \sqrt{2t^{-1}\log(4Kt^2/\delta)}\right) \leq \sum_{i,t} 2e^{-\log\left(\frac{4Kt^2}{\delta}\right)}$$

$$\leq \sum_{i,t} \frac{2\delta}{4Kt^2} = \frac{2\pi^2}{24}\delta \leq \delta$$

- (Correctness) If $\mathcal{E}$ does not happen,
  - $|\hat{\mu}_{i^*} - \mu_{i^*}| \leq B_t$ and $|\mu_j - \hat{\mu}_j| \leq B_t$ for all j. Thus, for all j
    $\hat{\mu}_j - \hat{\mu}_{i^*} \leq (\mu_{i^*} - \hat{\mu}_{i^*}) + (\mu_j - \mu_{i^*}) + (\hat{\mu}_j - \mu_j) \leq 2B_t$
  - i is removed if $\max_{j\in S} \hat{\mu}_{j,t} - \hat{\mu}_{i,t} \geq 2B_t \Rightarrow i^*$ is never removed
  - $\lim_{t\to\infty} B_t(\delta) \to 0$: every arm will eventually be removed
  - Successive Elimination is correct with probability $1 - \delta$

Successive Elimination Analysis

- Define the "bad event" $\mathcal{E} = \bigcup_{i,t}\{|\hat{\mu}_{i,t} - \mu_i| \geq B_t(\delta)\}$: we have

$$P(\mathcal{E}) \leq \sum_{i,t} P\left(|\hat{\mu}_{i,t} - \mu_i| \geq \sqrt{2t^{-1}\log(4Kt^2/\delta)}\right) \leq \sum_{i,t} 2e^{-\log\left(\frac{4Kt^2}{\delta}\right)}$$

$$\leq \sum_{i,t} \frac{2\delta}{4Kt^2} = \frac{2\pi^2}{24}\delta \leq \delta$$

- (Correctness) If $\mathcal{E}$ does not happen,
  - $|\hat{\mu}_{i^*} - \mu_{i^*}| \leq B_t$ and $|\mu_j - \hat{\mu}_j| \leq B_t$ for all j. Thus, for all j $\hat{\mu}_j - \hat{\mu}_{i^*} \leq (\mu_{i^*} - \hat{\mu}_{i^*}) + (\mu_j - \mu_{i^*}) + (\hat{\mu}_j - \mu_j) \leq 2B_t$
  - i is removed if $\max_{j \in S} \hat{\mu}_{j,t} - \hat{\mu}_{i,t} \geq 2B_t \Rightarrow i^*$ is never removed
  - $\lim_{t\to\infty} B_t(\delta) \to 0$: every arm will eventually be removed
  - Successive Elimination is correct with probability $1 - \delta$
- (Sample Complexity): arm i will be eliminated once $\Delta_i \leq 2B_t$
  - We can verify that $N_i = O\left(\Delta_i^{-2}\log(K/\delta\Delta_i)\right)$ is sufficient
  - Total sample complexity of $\sum_i \Delta_i^{-2}\log(K/\delta\Delta_i)$

Theorem

Successive Elimination is $(0, \delta)$-PAC with sample complexity

$$O\left(\sum_i \Delta_i^{-2} \log(K/\delta\Delta_i)\right)$$

Theorem

For any best-arm identification algorithm, there is a problem instance that requires

$$\Omega\left(\sum_i \Delta_i^{-2} \log\log\left(\frac{1}{\delta\Delta_i^2}\right)\right)$$

samples.

Linear Stochastic Bandits

Bonus: Linear Contextual Bandits

Protocol: Contextual Linear Bandit

Given: game length T, number of arms K

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

- The learner sees one context per arm $c_{1,t}, \ldots, c_{K,t}$
- The learner picks action $I_t \in \{1, \ldots, K\}$
- The learner observes and receives reward $X_t = \langle c_{I_t,t}, \theta^* \rangle + \xi_t$

Regret is defined w.r.t. an agent that knows the true θ:

$$\overline{\mathcal{R}}_T = \sum_{t=1}^{T} \max_i x_{i,t}^\intercal \theta^* - \sum_{t=1}^{T} x_{I_t,t}^\intercal \theta^*$$

Algorithm Design Principle: Optimism

Algorithm: OFUL [Abbasi-Yadkori et al., 2011]

Initialize $\hat{\theta}_0 = 0$, $B_0 = \mathbb{R}^d$

For $t = 1, 2, \ldots, T$:

- Receive contexts $c_{1,t}, \ldots, c_{K,t}$
- Choose $(I_t, \tilde{\theta}_t) = \arg\max_{i \in \{1,\ldots,K\}, \theta \in B_{t-1}} \theta^\intercal c_{i,t}$ (optimism)
- Observe $X_t = c_{I_t,t}^\intercal \theta^* + \xi_t$
- Calculate $V_t = \sum_{s=1}^t c_s c_s^\intercal + \lambda I$ and $r_t = \sqrt{\log \frac{\det(V_t)}{\delta^2 \lambda^d}} + \sqrt{\lambda} \|\theta^*\|$
- Calculate $\hat{\theta}_t = V_t^{-1} \left(\sum_{s=1}^t c_s X_s \right)$ (ridge)
- Update $B_t = \{\theta : (\theta - \hat{\theta}_t)^\intercal V_t (\theta - \hat{\theta}_t) \leq r_t)$

- If ξ_t is 1-sub-Gaussian, B_t is a confidence sequence with $P(\forall t > 0 : \theta^* \in B_t) \geq 1 - \delta$ (more examples in [de la Peña et al., 2009, Howard et al., 2020])

Analysis

- Regret decomposes over rounds:
- Recall that $(I_t, \tilde{\theta}_t) = \arg\max_{i \in \{1,\dots,K\}, \theta \in B_{t-1}} \theta^\intercal c_{i,t}$

$$
\begin{aligned}
\mathcal{R}_t - \mathcal{R}_{t-1} &= c_{i_t^*}^\intercal \theta^* - c_{I_t}^\intercal \theta^* \\
&\le c_{I_t}^\intercal \tilde{\theta}_t - c_{I_t}^\intercal \theta^* && \text{(by optimism)} \\
&\le c_{I_t}^\intercal \left(\tilde{\theta}_t - \hat{\theta}_{t-1}\right) + c_{I_t}^\intercal \left(\hat{\theta}_{t-1} - \theta^*\right) \\
&\le \|c_{I_t}\|_{V_t} \underbrace{\left\|\tilde{\theta}_t - \hat{\theta}_{t-1}\right\|_{V_t}}_{\le r_t} + \|c_{I_t}\|_{V_t} \underbrace{\left\|\hat{\theta}_{t-1} - \theta^*\right\|_{V_t}}_{\le r_t}
\end{aligned}
$$

- After some algebra, we can show, with probability $\ge 1 - \delta$, that

$$\mathcal{R}_T = O\left(\frac{d \log(1/\delta)}{\Delta}\right)$$

- The shared structure lets us learn a lot!

Review

- Setting: adversarial bandits
  - Exp3 (exponential weights)
- Setting: stochastic bandits
  - UCB (optimism)
  - Thompson Sampling (probablity matching)
- Setting: pure exploration
  - Successive Elimination (action-elimination)
- Setting: linear contextual bandits
  - OFUL (optimism)

Thanks!

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

Shipra Agrawal and Navin Goyal. Further optimal regret bounds for thompson sampling. In *Artificial intelligence and statistics*, pages 99–107, 2013.

Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. 2009.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov): 397–422, 2002.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002a.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, (1):48–77, 2002b.

Gábor Bartók, Dean P Foster, Dávid Pál, Alexander Rakhlin, and Csaba Szepesvári. Partial monitoring – classification, regret bounds, and algorithms. *Mathematics of Operations Research*, (4):967–997, 2014.

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

Victor H. de la Peña, Michael J. Klass, and Tze Leung Lai. Theory and applications of multivariate self-normalized processes. *Stochastic Processes and their Applications*, 119(12): 4210–4227, December 2009. ISSN 0304-4149.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform chernoff bounds via nonnegative supermartingales. *Probab. Surveys*, 17:257–317, 2020. doi: 10.1214/18-PS321. URL https://doi.org/10.1214/18-PS321.

Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. Pac subset selection in stochastic multi-armed bandits. In *ICML*, volume 12, pages 655–662, 2012.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, (1):4–22, 1985.

Shie Mannor and Ohad Shamir. From bandits to experts: On the value of side-observations. In *Advances in Neural Information Processing Systems*, pages 684–692, 2011.

Gergely Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3168–3176, 2015.

Daniel Russo. Simple bayesian algorithms for best arm identification. In *Conference on Learning Theory*, pages 1417–1418, 2016.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Julian Zimmert and Yevgeny Seldin. An optimal algorithm for stochastic and adversarial bandits. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 467–475, 2019.

Extras

Aside: Lower Bound Reasoning

- Fix a strategy and consider two problem instances:
  1. $\nu_1, \nu_2, \ldots, \nu_K$; with P as the joint distribution over $(I_t, r_{i,t})$
  2. $\nu_1, \nu_2', \ldots, \nu_K$; with P' as the joint distribution over $(I_t, r_{i,t})$
  3. The optimal arm is different: $\mu_2' \geq \mu_1 \geq \mu_2 \geq \mu_3 \geq \ldots$
  4. The data from P and P' will look very similar
- An algorithm that does well on P must not pull arm 2 too many times; hence, it will not do well on P'
- "Similar" is quantified by a change-of-measure identity; e.g. $P'(A) = e^{-\widehat{kl}_{N_{2,T}}} P(A)$, where $\widehat{kl}_t = \sum_{s=1}^{t} \log \frac{d\nu_2}{d\nu_2'}(X_{2,s})$
- Hence, an algorithm cannot tell if it is P or P' and must get high regret under P', mistakenly believing it is playing in P

Exp3: Analysis Full Detail

Exp3: Analysis

- Following the EW analysis, W_t is a potential function
- For any i^*, $e^{-\eta \hat{L}_T(i^*)} \leq \sum_j e^{-\eta \hat{L}_T(j)} = W_T = W_0 \prod_{t=1}^T \frac{W_t}{W_{t-1}}$.

$$
\begin{aligned}
\frac{W_t}{W_{t-1}} &= \frac{\sum_j e^{-\eta \hat{L}_{t-1}(j)} e^{-\eta \hat{\ell}_t(j)}}{\sum_j e^{-\eta \hat{L}_{t-1}(j)}} = \sum_j p_{t-1}(j) e^{-\eta \hat{\ell}_t(j)} \\
&\leq \underbrace{\sum_j p_{t-1}(j) \left(1 - \eta \hat{\ell}_t(j) + \frac{\eta^2}{2} \hat{\ell}_t(j)^2\right)}_{\text{since } e^x \leq 1 + x + \frac{1}{2}x^2 \text{ for } x \leq 0} \\
&= 1 - \eta \sum_j p_t(j) \hat{\ell}_t(j) + \frac{\eta^2}{2} \sum_j p_t(j) \hat{\ell}_t(I)^2 \\
&\leq \underbrace{e^{-\eta \sum_j p_t(j) \hat{\ell}_t(j) + \frac{\eta^2}{2} \sum_j p_t(j) \hat{\ell}_t(j)^2}}_{\text{since } 1 + x \leq e^x}
\end{aligned}
$$

Exp3: Analysis

$$e^{-\eta \hat{L}_T(i^*)} \leq W_0 \prod_{t=1}^{T} \frac{W_t}{W_{t-1}} \leq K \prod_{t=1}^{T} e^{-\eta \sum_j p_t(j)\hat{\ell}_t(j) + \frac{\eta^2}{2} \sum_j p_t(j)\hat{\ell}_t(j)^2}$$

$$\Leftrightarrow -\eta \hat{L}_T(i^*) \leq \log(K) - \eta \sum_j p_t(j)\hat{\ell}_t(j) + \frac{\eta^2}{2} \sum_j p_t(j)\hat{\ell}_t(j)^2$$

$$\Leftrightarrow \sum_{t=1}^{T} \sum_j p_t(j)\hat{\ell}_t(j) - \hat{L}_T(i^*) \leq \frac{\log(K)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \mathbb{E}[\sum_j p_t(j)\hat{\ell}_t(j)^2$$

$$\Rightarrow \sum_{t=1}^{T} \sum_j p_t(j)\mathbb{E}[\hat{\ell}_t(j) - \hat{L}_T(i^*)] \leq \frac{\log(K)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \mathbb{E}\left[\sum_j p_t(j)\hat{\ell}_t(j)^2\right]$$

$$\Leftrightarrow \sum_{t=1}^{T} \mathbb{E}[\ell(I_t)] - L_T(i^*) \leq \frac{\log(K)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \underbrace{\mathbb{E}\left[\sum_j p_t(j) \frac{\ell_t(I_t)^2}{p_t(I_t)^2} \mathbb{1}_{\{I_t = i\}}\right]}_{\text{variance term}}$$

Exp3: Analysis

Bounding the variance term turns out to be easy:

$$\mathbb{E}\left[\sum_j p_t(j)\frac{\ell_t(I_t)^2}{p_t(I_t)^2}\mathbb{1}_{\{I_t=i\}}\right] \leq \mathbb{E}\left[\sum_j p_t(j)\frac{\mathbb{1}_{\{I_t=i\}}}{p_t(I_t)^2}\right]$$
$$= \mathbb{E}\left[\frac{1}{p_t(I_t)}\right] = K$$

So, plugging this in, $\sum_{t=1}^{T} \mathbb{E}[\ell(I_t)] - L_T(i^*) \leq \frac{\log(K)}{\eta} + \frac{\eta}{2}TK$

Theorem (Exp3 upper bound [Auer et al., 2002b])

With $\eta = \sqrt{\frac{2\log(T)}{TK}}$, UCB has $\overline{\mathcal{R}}_T \leq \sqrt{2TK\log(K)}$.

Only get pseudo-Regret bounds because the i^* in the proof was fixed, not a function of $I_1, \ldots, I_T$