Online Convex Optimization

Tareq Si Salem

Postdoctoral Research Associate at Northeastern University

October, 2023

Outline I

Outline II

8. k-Experts Problem and Switching Costs

Motivating Problem: Prediction from Expert Advice

Motivating Problem: Prediction from Expert Advice

The player has to choose among the advice of n given experts. After making their choice, a loss between in $[0, 1]$ is incurred. A player constructs a "belief" (distribution) over the set of experts

$$\triangle_n \triangleq \left\{ \boldsymbol{x} \in [0,1]^n : \|\boldsymbol{x}\|_1 = 1 \right\}. \tag{1}$$

At each time step $t = 1$ to T,

- Player decides $\boldsymbol{x}_t \in \triangle_n$ and samples expert $i_t \sim \boldsymbol{x}_t$
- Adversary picks losses of each expert $\boldsymbol{l}_t = (l_{t,i})_{i \in [n]}$ and reveals them to the player.
- Player suffers an expected loss $f_t(\boldsymbol{x}_t) = \mathbb{E}_{i \sim \boldsymbol{x}_t} [l_{t,i}] = \boldsymbol{l}_t \cdot \boldsymbol{x}_t$.

In this setting, the optimization error is ill-defined. Instead, the goal of the player is to minimize *regret*:

$$\mathrm{regret}_T = \sum_{t=1}^{T} \boldsymbol{l}_t \cdot \boldsymbol{x}_t - \min_{i^\star \in [n]} \sum_{t=1}^{T} l_{t,i^\star}. \tag{2}$$

The goal of the player is to have sublinear regret ($\mathrm{regret}_T = o(T)$), i.e., on average the losses experienced by the player is **as good as** the best expert in hindsight.

Motivating Problem: Prediction from Expert Advice

A natural strategy. Consider that at timeslot t, the player greedily selects the best expert seen so far, also known as the follow the leader strategy (or fictitious play in game theory).

Theorem 1

FTL strategy incurs $\mathrm{regret}_T = \Omega(T)$.

Proof.

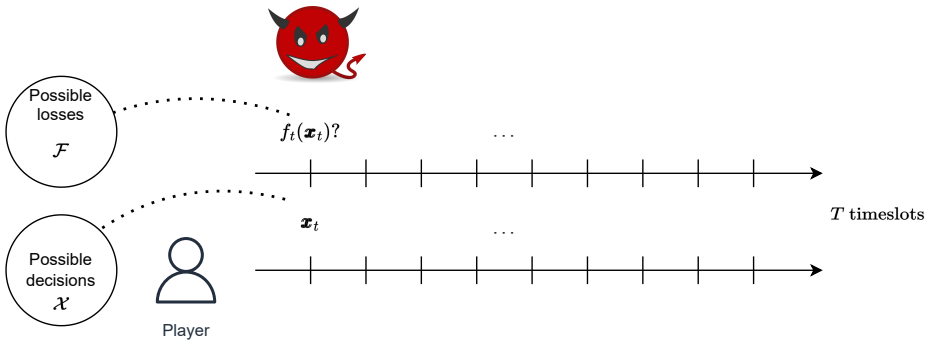
Consider two experts $n = 2$ and a sequence of losses $\boldsymbol{l}_1 = (1, 0), \boldsymbol{l}_2 = (0, 1), \ldots$. The best expert oscillates between 1 and 2. The player will incur a total loss of T (loss of 1 at every timeslot), whereas selecting a fixed expert incurs a total loss of $T/2$, i.e.,

$$\mathrm{regret}_T \geq T - T/2 = \Omega(T). \tag{3}$$

□

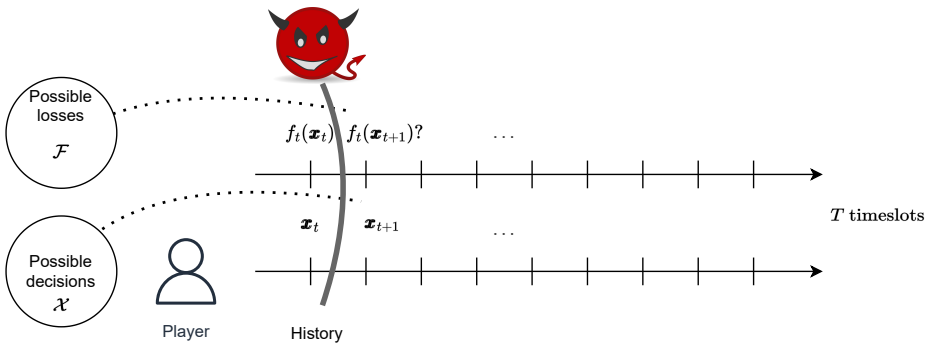
The Online Convex Optimization Setting

A player makes decisions iteratively. At the time of making the decision the outcome or outcomes associated with it is unknown to the player, and can even depend on the action taken by the decision maker.



The Online Convex Optimization Setting

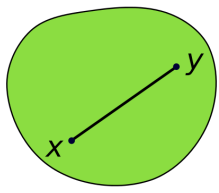
A player makes decisions iteratively. At the time of making the decision the outcome or outcomes associated with it is unknown to the player, and can even depend on the action taken by the decision maker.



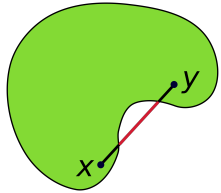
- The losses determined by an adversary should not be allowed to be unbounded.
- The decision set must be bounded and/or structured, and possibly infinite.

The Online Convex Optimization Setting

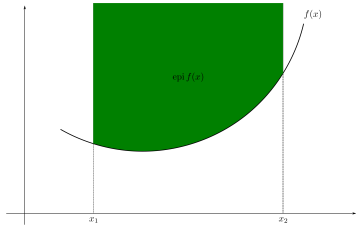
OCO models the decision set as a convex set in the Euclidean space $\mathcal{X} \subseteq \mathbb{R}^d$, and the costs as bounded convex functions.



(a) Convex set



(b) Non-convex set



(c) Convex functions

A More Formal Description of OCO

At iteration t, the player picks a new decision $\boldsymbol{x}_{t+1}$ after incurring the cost f_t, according to a mapping (algorithm) $\mathcal{A} : (\mathcal{X} \times \mathcal{F})^t \to \mathcal{X}$ given as

$$\boldsymbol{x}_{t+1} = \mathcal{A}\left(\left(\boldsymbol{x}_1, f_1\right), \left(\boldsymbol{x}_2, f_2\right), \ldots, \left(\boldsymbol{x}_t, f_t\right)\right) \in \mathcal{X}. \tag{4}$$

The regret of the player under policy $\mathcal{A}$ is then

$$\mathrm{regret}_T(\mathcal{A}) \triangleq \sup_{(f_t)_{t\in[T]} \in \mathcal{F}^T} \left\{ \sum_{t=1}^{T} f_t(\boldsymbol{x}_t) - \min_{\boldsymbol{x}^\star \in \mathcal{X}} \sum_{t=1}^{T} f_t(\boldsymbol{x}^\star) \right\}. \tag{5}$$

When the regret is sublinear ($\mathrm{regret}_T = o(T)$), i.e., on average the losses experienced by the player are **as good as** the best decision in hindsight.

Online Spam Filtering

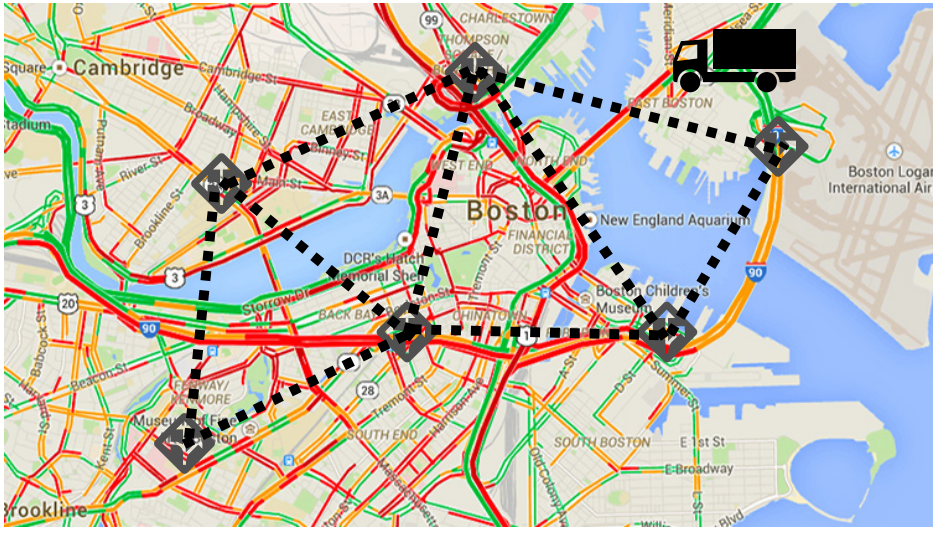
Online Spam Filtering

Consider an online spam-filtering system. Repeatedly, emails arrive in the system and are classified as spam or valid. For simplicity, consider a linear variant of the model:

- Each email is represented as a vector $\boldsymbol{a} \in \mathbb{R}^d$, where where d is the number of words in the dictionary. The email is the aggregation of one-hot encoding of words in the dictionary ("bag-of-words" representation).
- To predict whether an email is spam, we learn a filter, a vector $\boldsymbol{x} \in \mathbb{R}^d$.
- Given an email $\boldsymbol{a}_t$ and a label b_t, a loss $l_t(\boldsymbol{x}) = l_{\boldsymbol{a}_t, b_t}(\boldsymbol{x})$ is revealed to the player (e.g., the hinge loss $\text{hinge}(b_t \cdot \boldsymbol{x}^\top \boldsymbol{a}_t) = \max\left\{0, 1 - b_t \cdot \boldsymbol{x}^\top \boldsymbol{a}_t\right\}$).

Such a system has to cope with adversarially generated data and dynamically change with the varying input.

Online Shortest Paths/Routing



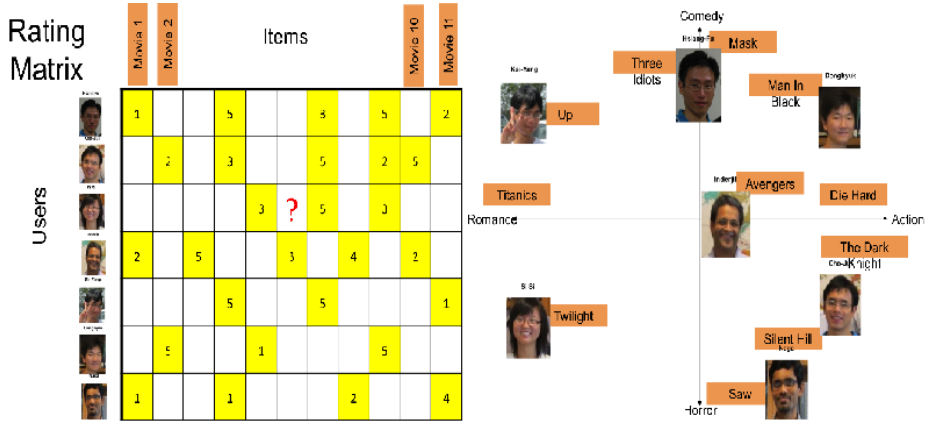
Online Shortest Paths/Routing

In the online shortest path problem, the decision maker is given a directed graph $G = (V, E)$ and a source-sink pair $u, v \in V$. The set $\mathcal{X}$ of all all distributions over paths (flows) in a graph is a convex set in $\mathbb{R}^E$, with $\mathcal{O}(m + |V|)$ constraints.

At each time step $t = 1$ to T

- The player chooses a flow $\boldsymbol{x}_t \in \mathcal{X}$ and samples a path $\boldsymbol{p}_t \sim \boldsymbol{x}_t$
- The adversary chooses weights on the edges of the graph $\boldsymbol{w}_t : E \to \mathbb{R}$. The player incurs the expected loss $f_t(\boldsymbol{x}_t) = \boldsymbol{x}_t \cdot \boldsymbol{w}_t$.

Matrix Completion and Recommendation Systems



Matrix Completion and Recommendation Systems

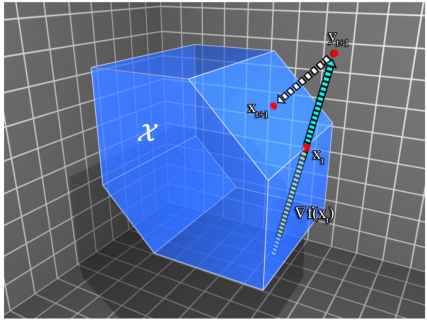
For example, for the case of binary recommendations for music, we have a matrix $\boldsymbol{X} \in \{0,1\}^{n \times m}$ where n is the number of persons considered, m is the number of songs in our library, and $0/1$ signifies dislike/like respectively.

At each time step $t = 1$ to T

- The player selects a preference matrix $\boldsymbol{X}_t \in \mathcal{X}$, where $\mathcal{X} \subseteq [0,1]^{n \times m}$ (with a rank constraint).
- An adversary then chooses a user/song pair (i_t, j_t) along with a "real" preference for this pair $y_t \in \{0,1\}$. The loss experienced by the decision maker can be described by some convex loss function, $f_t(\boldsymbol{X}) = (X_{i_t,j_t} - y_t)^2$.

The natural comparator in this scenario is a low-rank matrix, which corresponds to the intuitive assumption that preference is determined by few unknown factors

Online Gradient Descent



At time t, Online Gradient Descent (OGD)'s update rule is

$$\text{(gradient update)} \qquad \boldsymbol{y}_{t+1} = \boldsymbol{x}_t - \eta_t \nabla f_t(\boldsymbol{x}_t)$$

$$\text{(projection step)} \qquad \boldsymbol{x}_{t+1} = \Pi_{\mathcal{X}}(\boldsymbol{y}_{t+1})$$

The operator $\Pi_{\mathcal{X}} : \mathbb{R}^d \to \mathcal{X}$ is the Eulidean projection $\Pi_{\mathcal{X}}(\boldsymbol{y}) \triangleq \arg\min_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x} - \boldsymbol{y}\|_2$.

Regret of Online Gradient Descent

Theorem 2

Online gradient descent with fixed step sizes η guarantees the following for all $T \geq 1$:

$$\mathbf{regret}_T \leq \frac{D^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\nabla f_t(\boldsymbol{x}_t)\|_2^2 . \tag{6}$$

For a fixed learning learning rate $\eta = \frac{D}{G\sqrt{T}}$, the regret is upper bounded by $\mathrm{regret}_T \leq DG\sqrt{T}$.

Regret of Online Gradient Descent (Proof)

A fundamental property of projections into convex bodies is that for an arbitrary $\boldsymbol{x}' \in \mathbb{R}^d$, we have for all $\boldsymbol{x} \in \mathcal{X}$:

$$\left\|\Pi_{\mathcal{X}}(\boldsymbol{x}') - \boldsymbol{x}\right\|_2^2 \leq \left\|\boldsymbol{x}' - \boldsymbol{x}\right\|_2^2$$

Applying the above,

$$\begin{aligned}
\left\|\boldsymbol{x}_t - \boldsymbol{x}^\star\right\|^2 - \left\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\right\|_2^2 &= \left\|\boldsymbol{x}_t - \boldsymbol{x}^\star\right\|_2^2 - \left\|\Pi_{\mathcal{X}}(\boldsymbol{x}_t - \eta\nabla f_t(\boldsymbol{x}_t)) - \boldsymbol{x}^\star\right\|_2^2 && (7)\\
&\geq \left\|\boldsymbol{x}_t - \boldsymbol{x}^\star\right\|_2^2 - \left\|\boldsymbol{x}_t - \eta\nabla f_t(\boldsymbol{x}_t) - \boldsymbol{x}^\star\right\|_2^2 && (8)\\
&= 2\eta\nabla f_t(\boldsymbol{x}_t)\cdot\left(\boldsymbol{x}_t - \boldsymbol{x}^\star\right) - \eta^2\left\|\nabla f_t(\boldsymbol{x}_t)\right\|_2^2 && (9)
\end{aligned}$$

And so,

$$\nabla f_t(\boldsymbol{x}_t)\cdot\left(\boldsymbol{x}_t - \boldsymbol{x}^\star\right) \leq \frac{1}{2\eta}\left(\left\|\boldsymbol{x}_t - \boldsymbol{x}^\star\right\|_2^2 - \left\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\right\|_2^2\right) + \frac{\eta}{2}\left\|\nabla f_t(\boldsymbol{x}_t)\right\|_2^2 \tag{10}$$

Summing over t,

$$\begin{aligned}
\sum_{t=1}^{T}\nabla f_t(\boldsymbol{x}_t)\cdot\left(\boldsymbol{x}_t - \boldsymbol{x}^\star\right) &\leq \frac{1}{2\eta}\left(\left\|\boldsymbol{x}_1 - \boldsymbol{x}^\star\right\|_2^2 - \left\|\boldsymbol{x}_{T+1} - \boldsymbol{x}^\star\right\|_2^2\right) + \frac{\eta}{2}\sum_{t=1}^{T}\left\|\nabla f_t(\boldsymbol{x}_t)\right\|_2^2 && (11)\\
&\leq \frac{D^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\left\|\nabla f_t(\boldsymbol{x}_t)\right\|_2^2 && (12)
\end{aligned}$$

Expert Problem – Applying OGD

We can apply OGD to the expert problem. Recall the the decision set $\mathcal{X}$ is the probability simplex $\triangle_n$ and the loss of picking an expert is in $[0, 1]$. The diameter of the set is $\|\boldsymbol{x} - \boldsymbol{x}'\|_2 \leq D = 1$ for $\boldsymbol{x}, \boldsymbol{x}' \in \triangle_n$. The gradients are bounded by $\|\nabla f_t(\boldsymbol{x}_t)\|_2^2 \leq G = \sqrt{n}$. The regret is sublinear with the rate $\sqrt{nT}$. **The dependency on the number of experts can be improved further!**
We will describe a gradient-based scheme that can be tailored to the geometry of the problem.

Expert Problem – Adapting To Geometry via Online Mirror Descent

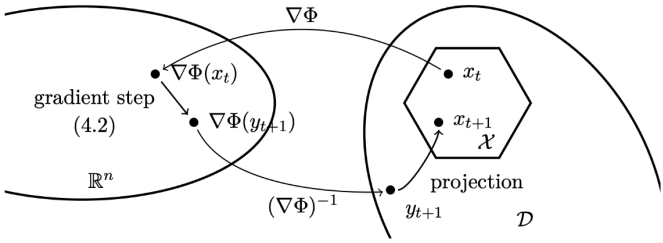


Figure: Illustration of online mirror descent

The policy is parameterized by (1) a learning rate η, and (2) a mirror map $\Phi : \mathcal{D} \subseteq \mathbb{R}^d \to \mathbb{R}$. The projection step generalizes Euclidean projection and it is given by

$$\Pi_{\mathcal{X}}^{\Phi}(\boldsymbol{y}) \triangleq \operatorname*{arg\,min}_{\boldsymbol{x}\in\mathcal{D}\cap\mathcal{X}} D_{\Phi}(\boldsymbol{x},\boldsymbol{y}) = \operatorname*{arg\,min}_{\boldsymbol{x}\in\mathcal{D}\cap\mathcal{X}} \Phi(\boldsymbol{x}) - \Phi(\boldsymbol{y}) - \nabla\Phi(\boldsymbol{y})\cdot(\boldsymbol{x}-\boldsymbol{y}) \qquad \text{(Bregman projection)}. \tag{13}$$

Regret of Online Mirror Descent

Theorem 3

Let Φ be a mirror map ρ-strongly convex on $\mathcal{X} \cap \mathcal{D}$ w.r.t. $\|\cdot\|$. Let $R^2 = \sup_{\boldsymbol{x} \in \mathcal{X} \cap \mathcal{D}} \Phi(\boldsymbol{x}) - \Phi(\boldsymbol{x}_1)$, and f_t be convex and G-Lipschitz w.r.t. $\|\cdot\|$. Then online mirror descent with $\eta = \frac{R}{G}\sqrt{\frac{2\rho}{T}}$ satisfies

$$\sum_{t=1}^{T} f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}) \leq \frac{D_\Phi(\boldsymbol{x}, \boldsymbol{x}_1)}{\eta} + \eta \frac{\sum_{t=1}^{T} \|\nabla f_t(\boldsymbol{x}_t)\|_\star^2}{2\rho} = RL\sqrt{\frac{2}{\rho}T}. \tag{14}$$

- A differentiable function f is α-strongly-convex with respect to a norm $\|\cdot\|$ if $\forall \boldsymbol{x} \in \mathcal{X}$

$$f(\boldsymbol{x}) - f(\boldsymbol{y}) \leq \nabla f(\boldsymbol{x}) \cdot (\boldsymbol{x} - \boldsymbol{y}) - \frac{\alpha}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2, \qquad \forall \boldsymbol{y} \in \mathcal{X}. \tag{15}$$

- Let $\|\cdot\|$ be a norm on $\mathbb{R}^d$. The dual norm $\|\cdot\|_\star$ is defined as $\|\boldsymbol{u}\|_\star = \sup_{\boldsymbol{x} \in \mathbb{R}^d, \|\boldsymbol{x}\| \leq 1} \boldsymbol{x} \cdot \boldsymbol{u}$.

Ball setup. When $\Phi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2$, the dual and primal spaces are identical ($\nabla\Phi(\boldsymbol{x}) = \boldsymbol{x}$ is the identity mapping) and the Bregman projection is simply the Euclidean projection $\Pi_{\mathcal{X}}$. OGD is an instance of OMD.

Simplex/Entropic Setup of OMD

Simplex/Entropic setup. A more interesting choice of a mirror map is given by the negative entropy

$$\Phi(\boldsymbol{x}) = \sum_{i=1}^{d} x_i \log(x_i). \tag{16}$$

The Bregman divergence of this mirror map is given by $D_\Phi(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{d} x_i \log(\frac{x_i}{y_i})$. The policy configured for simplex decision sets $\triangle_n$ amounts to

$$\text{(gradient update)} \qquad \boldsymbol{x}_{t+1,i} = x_{t,i} e^{-\eta l_{t,i}}, i \in [d] \tag{17}$$

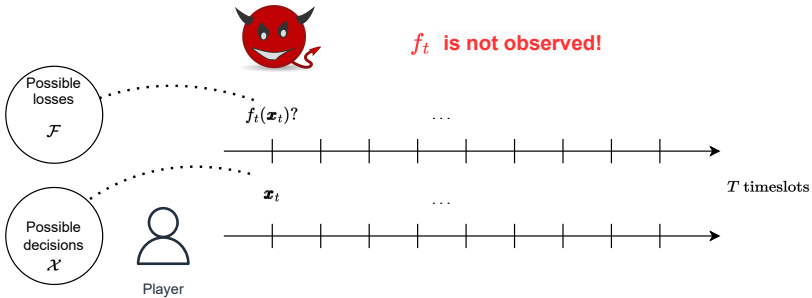
$$\text{(projection step)} \qquad \boldsymbol{x}_{t+1} = \frac{\boldsymbol{y}_{t+1}}{\|\boldsymbol{y}_{t+1}\|_1}. \tag{18}$$

For $\boldsymbol{x}_1 = (1/n, 1/n, \ldots, 1/n)$, one has $R^2 = \log(n)$ and $\|\nabla f_t(\boldsymbol{y})\|_\infty \leq G = 1$. (The norm $\|\cdot\|_\infty$ is the dual norm of $\|\cdot\|_1$.) The map Φ is 1-strongly convex w.r.t. $\|\cdot\|_1$ over the simplex $\triangle_n$ (Pinsker's inequality). The regret under this configuration has rate of $\sqrt{\log(n)T}$ instead of of $\sqrt{nT}$ for OGD. This setup corresponds to the well known **Hedge/Multiplicative Weights Update** algorithm.

Going Beyond Full-Information

Going Beyond Full-Information – Bandit Convex Optimization

As opposed to the OCO model, in which the decision maker has access to a gradient oracle for f_t over $\mathcal{X}$, in BCO the loss $f_t(\boldsymbol{x}_t)$ **is the only feedback available** to the online player at iteration t.



Multi-Armed Bandits (MAB)

A classical model for decision making under uncertainty is the multi-armed bandit (MAB) model. This setting is identical to the setting of prediction from expert advice, the only difference being the feedback available to the decision maker. **Only the loss of the selected expert is revealed!**
The MAB problem exhibits an exploration-exploitation tradeoff:

- An efficient (low regret) algorithm has to explore the value of the different actions in order to make the best decision.
- On the other hand, having gained sufficient information about the environment, a reasonable algorithm needs to exploit this action by picking the best action.

A Reduction from Limited Information to Full Information

1: Input: convex set $\mathcal{K} \subset \mathbb{R}^n$, first order online algorithm $\mathcal{A}$.
2: Let $\mathbf{x}_1 = \mathcal{A}(\emptyset)$.
3: **for** $t = 1$ to T **do**
4: Generate distribution $\mathcal{D}_t$, sample $\mathbf{y}_t \sim \mathcal{D}_t$ with $\mathbf{E}[\mathbf{y}_t] = \mathbf{x}_t$.
5: Play $\mathbf{y}_t$.
6: Observe $f_t(\mathbf{y}_t)$, generate $\mathbf{g}_t$ with $\mathbf{E}[\mathbf{g}_t] = \nabla f_t(\mathbf{x}_t)$.
7: Let $\mathbf{x}_{t+1} = \mathcal{A}(\mathbf{g}_1, ..., \mathbf{g}_t)$.
8: **end for**

Lemma 4

Let $(f'_t(\boldsymbol{x}) = \boldsymbol{g}_t \cdot \boldsymbol{x})_{t\in[T]}$ be a sequence of linear cost functions, and $(f_t)_{t\in[T]}$ be a sequence of differentiable function, and $\mathcal{X}$ be the convex decision set. Let $\mathcal{A}$ be an OCO policy with regret bound $\mathrm{regret}_T(\boldsymbol{\mathcal{A}}) \le B_{\boldsymbol{\mathcal{A}}}(\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_T)$ in the full-information setting against losses f'_t. Define the points $(\boldsymbol{x}_t)$ as: $\boldsymbol{x}_1 \in \mathcal{X}$, $\boldsymbol{x}_t \leftarrow \mathcal{A}_t((\boldsymbol{x}_s, f'_s,)_{s\in[t-1]})$ where each $\boldsymbol{g}_t$ is a vector valued random variable such that:

$$\mathbb{E}[\boldsymbol{g}_t | \boldsymbol{x}_1, f_1, \ldots, \boldsymbol{x}_t, f_t] = \nabla f_t(\boldsymbol{x}_t), \tag{19}$$

Then the following holds for all $\boldsymbol{x}^\star \in \mathcal{X}$:

$$\mathbb{E}\left[\sum_{t=1}^{T} f_t(\boldsymbol{x}_t)\right] - \sum_{t=1}^{T} f_t(\boldsymbol{x}^\star) \le \mathbb{E}\left[B_{\boldsymbol{\mathcal{A}}}(\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_T)\right]. \tag{20}$$

MAB: EXP3 Simultaneous Exploration and Exploitation

1: Input: parameter $\varepsilon > 0$. Set $\mathbf{x}_1 = (1/n)\mathbf{1}$.
2: **for** $t \in \{1, 2, ..., T\}$ **do**
3: Choose $i_t \sim \mathbf{x}_t$ and play i_t.
4: Let

$$\hat{\ell}_t(i) = \begin{cases} \frac{1}{\mathbf{x}_t(i_t)} \cdot \ell_t(i_t), & i = i_t \\ 0, & \text{otherwise} \end{cases}$$

5: Update $\mathbf{y}_{t+1}(i) = \mathbf{x}_t(i) e^{-\varepsilon \hat{\ell}_t(i)}$, $\mathbf{x}_{t+1} = \frac{\mathbf{y}_{t+1}}{\|\mathbf{y}_{t+1}\|_1}$
6: **end for**

Theorem 5

EXP3 with non-negative losses and $\epsilon = \sqrt{\frac{\log(n)}{Tn}}$ *guarantees the following regret bound:*

$$\mathbb{E}\left[\sum_{t=1}^{T} l_{t,i_t}\right] - \min_{i^\star \in [n]} \sum_{t=1}^{T} l_{t,i^\star} \leq 2\sqrt{Tn \log(n)}. \tag{21}$$

Flaxman/Kalai/McMahan (FKM) Algorithm

1: Input: decision set $\mathcal{K}$ containing $\mathbf{0}$, set $\mathbf{x}_1 = \mathbf{0}$, parameters δ, η.
2: **for** $t = 1$ to T **do**
3: Draw $\mathbf{u}_t \in \mathbb{S}_1$ uniformly at random, set $\mathbf{y}_t = \mathbf{x}_t + \delta \mathbf{u}_t$.
4: Play $\mathbf{y}_t$, observe and incur loss $f_t(\mathbf{y}_t)$. Let $\mathbf{g}_t = \frac{n}{\delta} f_t(\mathbf{y}_t)\, \mathbf{u}_t$.
5: Update $\mathbf{x}_{t+1} = \prod_{\mathcal{K}_\delta} [\mathbf{x}_t - \eta \mathbf{g}_t]$.
6: **end for**

Theorem 6

FKM algorithm with parameters $\frac{D}{nT^{3/4}}$ *and* $\delta = \frac{1}{T^{1/4}}$ *guarantees the following expected regret bound*

$$\mathbb{E}\left[\sum_{t=1}^{T} f_t(\boldsymbol{x}_t)\right] - \sum_{t=1}^{T} f_t(\boldsymbol{x}^\star) \leq 9nDGT^{3/4}. \tag{22}$$

Proof.

(Proof Sketch) The algorithm constructs an unbiased estimator of a convex function that is δ–far from the true cost function f_t. The estimator's variances scales with $\frac{1}{\delta^2}$. Selecting $\delta = \frac{1}{T^{1/4}}$ with a slightly less aggressive OGD, configured with stepsize $\eta = \frac{D}{nT^{3/4}}$; combined with the reduction in Lemma 4 ensures the above bound. □

PAC (Probably Approximately Correct) learning and OCO

A learning algorithm has access to samples from an unknown distribution

$$(\boldsymbol{x}, y) \sim \mathcal{D}, \boldsymbol{x} \in \mathcal{X}, y \in \mathcal{Y}. \tag{23}$$

The goal is to be able to predict y as a function of $\boldsymbol{x}$, i.e., to learn a mapping (hypothesis) $h : \mathcal{X} \to \mathcal{Y}$, that minimizes a prediction error according to a (usually convex) loss function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ given as

$$\text{error}(h) \triangleq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[l(h(\boldsymbol{x}), y) \right]. \tag{24}$$

Definition 7

(Agnostic PAC learning) The hypothesis class $\mathcal{H}$ is agnostically PAC learnable with respect to loss function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ if the following holds. There exists an algorithm $\mathcal{A}$ that accepts $S_T = \{(x_t, y_t), t \in [T]\}$ and returns hypothesis $h_{\mathcal{A}(S_T)} \in \mathcal{H}$ that satisfies: for any $\epsilon, \delta > 0$ there exists a sufficiently large natural number $T = T(\epsilon, \delta)$ such that for any distribution $\mathcal{D}$ over pairs $(\boldsymbol{x}, y)$ and $T(\epsilon, \delta)$ samples from this distribution, it holds that with probability at least $1 - \delta$

$$\text{error}(h_{\mathcal{A}(S_T)}) \leq \min_{h\in\mathcal{H}} \text{error}(h) + \epsilon \tag{25}$$

Reduction: Learning → OCO

1: Input: OCO algorithm $\mathcal{A}$, convex hypothesis class $\mathcal{H} \subseteq \mathbb{R}^d$, convex los
function ℓ.
2: Let $h_1 \leftarrow \mathcal{A}(\emptyset)$.
3: **for** $t = 1$ to T **do**
4: Draw labeled example $(\mathbf{x}_t, y_t) \sim \mathcal{D}$.
5: Let $f_t(h) = \ell(h(\mathbf{x}_t), y_t)$.
6: Update

$$h_{t+1} = \mathcal{A}(f_1, ..., f_t).$$

7: **end for**
8: Return $\bar{h} = \frac{1}{T}\sum_{t=1}^{T} h_t$.

Theorem 8

Let $\mathcal{A}$ be an OCO algorithm whose regret after T iterations is guaranteed to be bounded by $\mathrm{regret}_T(\mathcal{A})$. Then for any $\delta > 0$, with probability at least $1 - \delta$, it holds that

$$\mathrm{error}(\bar{h}) \leq \min_{h^\star \in \mathcal{H}} \mathrm{error}(h^\star) + \frac{\mathrm{regret}_T(\mathcal{A})}{T} + \sqrt{\frac{8\log(2/\delta)}{T}}. \tag{26}$$

For $T = \mathcal{O}\left(\frac{1}{\epsilon^2}\log(1/\delta) + T_\epsilon(\mathcal{A})\right)$, where $T_\epsilon(\mathcal{A})$ is the integer T such that $\mathrm{regret}_T(\mathcal{A})/T \leq \epsilon$, we have

$$\mathrm{error}(\bar{h}) \leq \min_{h^\star \in \mathcal{H}} \mathrm{error}(h^\star) + \epsilon. \tag{27}$$

k-Experts Problem

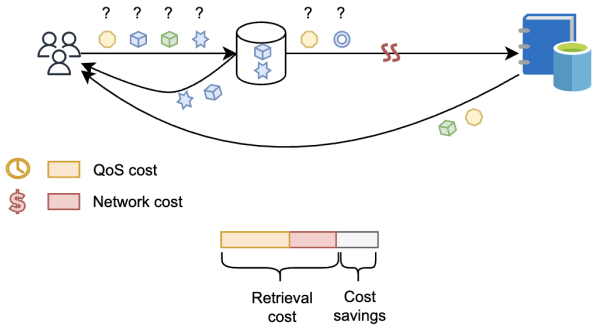
Let us consider a generalization of the expert problem.[1] The player has to choose k experts among n given experts. The number of configurations under this setting is $\binom{n}{k}$, so constructing a distribution $\triangle_{\binom{n}{k}}$ already prohibits having any efficient algorithms. Instead, we can keep track of the marginal probabilities instead, i.e., the decisions in

$$\triangle_{n,k} \triangleq \left\{ \boldsymbol{x} \in [0,1]^n : \|\boldsymbol{x}\|_1 = k \right\}. \tag{28}$$

[1] No-Regret Caching via Online Mirror Descent. ACM Transactions on Modeling and Performance Evaluation of Computing Systems. T. Si Salem, S. Ioannidis, G. Neglia.

k-Experts: Casting as a Caching Problem and Dimensionality Reduction

For $\boldsymbol{x} \in \triangle_{n,k}$, the component x_i corresponds to the probability of selecting file $i \in [n]$.

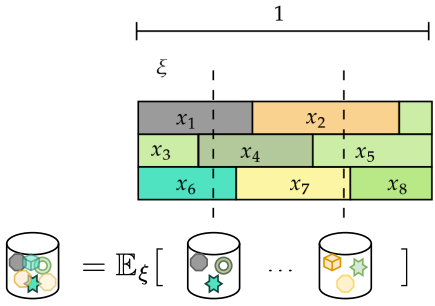


When a request batch $\boldsymbol{r}_t$ arrives, the cache incurs the following cost:

$$f_{\boldsymbol{r}_t}(\boldsymbol{x}_t) = \sum_{i=1}^{N} w_i r_{t,i}(1 - x_{t,i}). \tag{29}$$

k-Experts: Recovering the Original Setting

We can construct a sampling scheme Ξ that outputs $\boldsymbol{z}_t \in \triangle_{n,k} \cap \{0,1\}^n$ such that $\mathbb{E}[\boldsymbol{z}_t] = \boldsymbol{x}_t$.



This scheme is called Madow's sampling. Under Ξ the expected cost of $\boldsymbol{z}_t$ is identical to the cost of $\boldsymbol{x}_t$, i.e.,

$$\mathbb{E}_{\Xi}\left[f_{\boldsymbol{r}_t}(\boldsymbol{z}_t)\right] = f_{\boldsymbol{r}_t}(\boldsymbol{x}_t). \tag{30}$$

The expected regret $\mathrm{regret}_T(\mathcal{A}, \Xi)$ is the same as the regret of $\mathcal{A}$!

k-Experts: Switching Costs

In practice, there is a cost associated with switching the decision from $\boldsymbol{z}_t$ to $\boldsymbol{z}_{t+1}$, denoted by $\mathrm{UC}_{\boldsymbol{r}_t}(\boldsymbol{z}_t, \boldsymbol{z}_{t+1})$. In the fractional setup, gradient-based algorithms adapt their states proportionally to the learning rate, i.e., $\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\| = \mathcal{O}(\eta)$. The total update cost measured in some norm $\|\cdot\|$ is given by

$$\sum_{t=1}^{T} \|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\| = \mathcal{O}(\eta T) = \mathcal{O}\left(\sqrt{T}\right). \tag{31}$$

The last equality is obtained for $\eta = \Theta\left(\frac{1}{\sqrt{T}}\right)$.
Randomness prevents this property to be transferred to the integral actions $\boldsymbol{z}_t$.

k-Experts: Independent Sampling

When considering the extended regret ($\text{E-Regret}_T(\mathcal{A}, \Xi)$) we lose immediately the regret guarantee:

Theorem

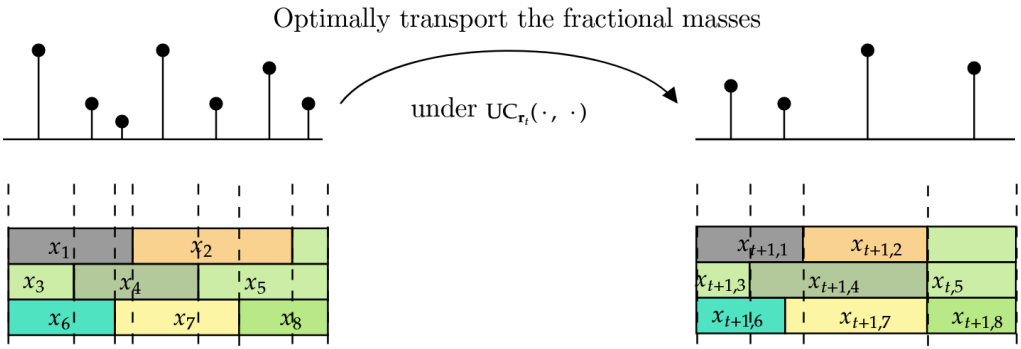
Any randomized caching policy constructed by an online policy $\mathcal{A}$ combined with online independent rounding as Ξ leads to $\Omega(T)$ $\text{E-Regret}(\mathcal{A}, \Xi)$.

Imposing dependence (coupling) between the two consecutive random states may significantly reduce the expected update cost.
Optimizing for all possible coupling is a **min-cost flow problem**, or when establishing flows between distributions it is an **optimal transport problem**.

$$\boldsymbol{f} = \underset{[f_{i,j}]_{(i,j)\in[|\boldsymbol{p}_t|]\times[|\boldsymbol{p}_{t+1}|]}}{\arg\min} \mathbb{E}\left[\text{UC}\left(\boldsymbol{z}_t, \boldsymbol{z}_{t+1}\right)\right] = \sum_{i=1}^{|\boldsymbol{p}_t|}\sum_{j=1}^{|\boldsymbol{p}_{t+1}|} \text{UC}_{\boldsymbol{r}_t}\left(\boldsymbol{\zeta}_t^i, \boldsymbol{\zeta}_{t+1}^j\right) f_{i,j}$$

$$\text{s.t.} \quad \sum_{j=1}^{|\boldsymbol{p}_{t+1}|} f_{i,j} = p_{t,i}, \quad \sum_{i=1}^{|\boldsymbol{p}_t|} f_{i,j} = p_{t+1,j}, \quad f_{i,j} \in [0,1], \forall (i,j) \in [|\boldsymbol{p}_t|] \times [|\boldsymbol{p}_{t+1}|].$$

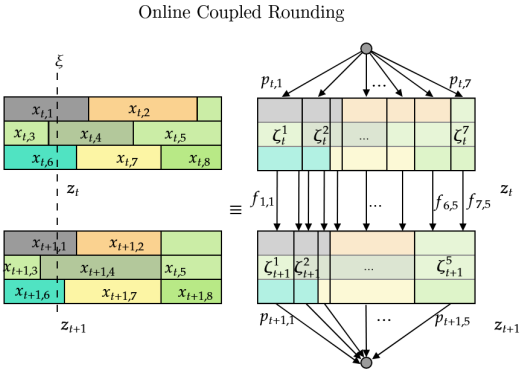
k-Experts: Optimal Transport



Remark

We prove that this scheme selected as Ξ, coupled with a no-regret policy $\mathcal{A}$, has sublinear extended regret guarantee. However, it has a time-complexity $\mathcal{O}\left(N^3\right)$.

k-Experts: Simpler Approach

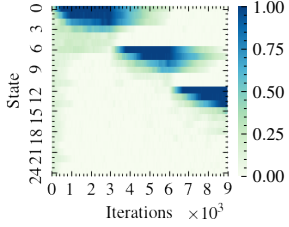


Theorem

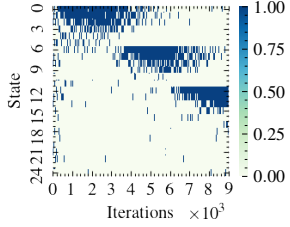
A no-regret policy $\mathcal{A}$ combined with online coupled rounding Ξ has $\mathcal{O}\left(\sqrt{T}\right)$ $\text{E-Regret}_T(\mathcal{A}, \Xi)$.

Online Coupled Rounding has linear time complexity.

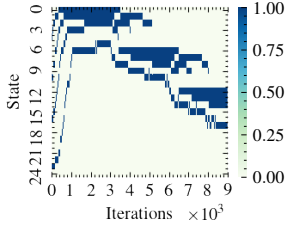
k-Experts: Coupling Schemes a Qualitative Evaluation



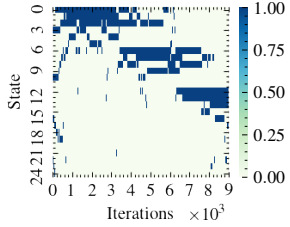
(a) Fractional cache states



(b) Online independent rounding



(c) Online coupled rounding



(d) Online optimally-coupled rounding