MS&E 233
Game Theory, Data Science and AI
Lecture 2

Vasilis Syrgkanis
Assistant Professor
Management Science and Engineering
(by courtesy) Computer Science and Electrical Engineering
Institute for Computational and Mathematical Engineering

Class Music Auction!

We will be experimenting with putting music for the first three minutes of the class as people arrive!

You have the chance to choose the song of the day!

Each of you has a total budget of 100 fake dollars for the whole class! You can choose to spend them however you want on each lecture.

For each lecture you can choose to bid anywhere from 0 to 20 dollars.

We will then choose uniformly at random among the highest bidders. The winner of the auction will get to choose the song of the day and they have to pay their bid, i.e. the amount they bid will be subtracted from their 100$ budget.

If you submit an illegal bid (i.e. a bid that goes beyond your total budget, your bid will be disqualified and ignored).

Please be appropriate in your choice of songs; I might need to censor and ask you to choose something else. I'll be emailing the winner on the morning of the lecture to email me the spotify link for the song.

Submit your bid by 11:59pm the day before the lecture. You should submit your bid using the corresponding canvas quiz that will be setup for each lecture

Class Music Auction: Game Theory, Data Science and AI (stanford.edu)

Go to canvas and check the quizzes section.

If there is no participation in the auction, I'll just choose the music myself. But that's not much fun...

Spotify playlist that will be populated with the songs we play each day: https://open.spotify.com/playlist/03yGb6URnCzG4pVV6RhK4C?si=wpINDMSGRJOho_6daaSLsA&pt=ff706933952e0f64d8f8b797368a83ed

Computational Game Theory for Complex Games

1
- Basics of game theory and zero-sum games (T)
- **Basics of online learning theory (T)**
- Solving zero-sum games via online learning (T)
- *HW1: implement simple algorithms to solve zero-sum games*
- Applications to ML and AI (T+A)
- *HW2: implement boosting as solving a zero-sum game*

2
- Basics and applications of extensive-form games (T+A)
- Solving extensive-form games via online learning (T)
- *HW3: implement agents to solve very simple variants of poker*

3
- General games and equilibria (T)
- Online learning in general games, multi-agent RL (T+A)
- *HW4: implement no-regret algorithms that converge to correlated equilibria in general games*

Data Science for Auctions and Mechanisms

4
- Basics and applications of auction theory (T+A)
- Learning to bid in auctions via online learning (T)
- *HW5: implement bandit algorithms to bid in ad auctions*

5
- Optimal auctions and mechanisms (T)
- Simple vs optimal mechanisms (T)
- *HW6: calculate equilibria in simple auctions, implement simple and optimal auctions, analyze revenue empirically*

6
- Optimizing mechanisms from samples (T)
- Online optimization of auctions and mechanisms (T)
- *HW7: implement procedures to learn approximately optimal auctions from historical samples and in an online manner*

Further Topics

7
- Econometrics in games and auctions (T+A)
- A/B testing in markets (T+A)
- *HW8: implement procedure to estimate values from bids in an auction, empirically analyze inaccuracy of A/B tests in markets*

Guest Lectures

- Mechanism Design for LLMs, Renato Paes Leme, Google Research
- Auto-bidding in Sponsored Search Auctions, Kshipra Bhawalkar, Google Research

Introduction to Online Learning

Example

- You have a daily commute: Stanford→Berkeley
- Every day you contemplate between two options: (1) Bay Bridge, (2) Dumbarton Bridge
- Don’t know which route will have more traffic jams (due to un-predictable events, e.g., accidents)
- You take one of the two options
- After the fact, you observe the traffic jams that occurred on both routes

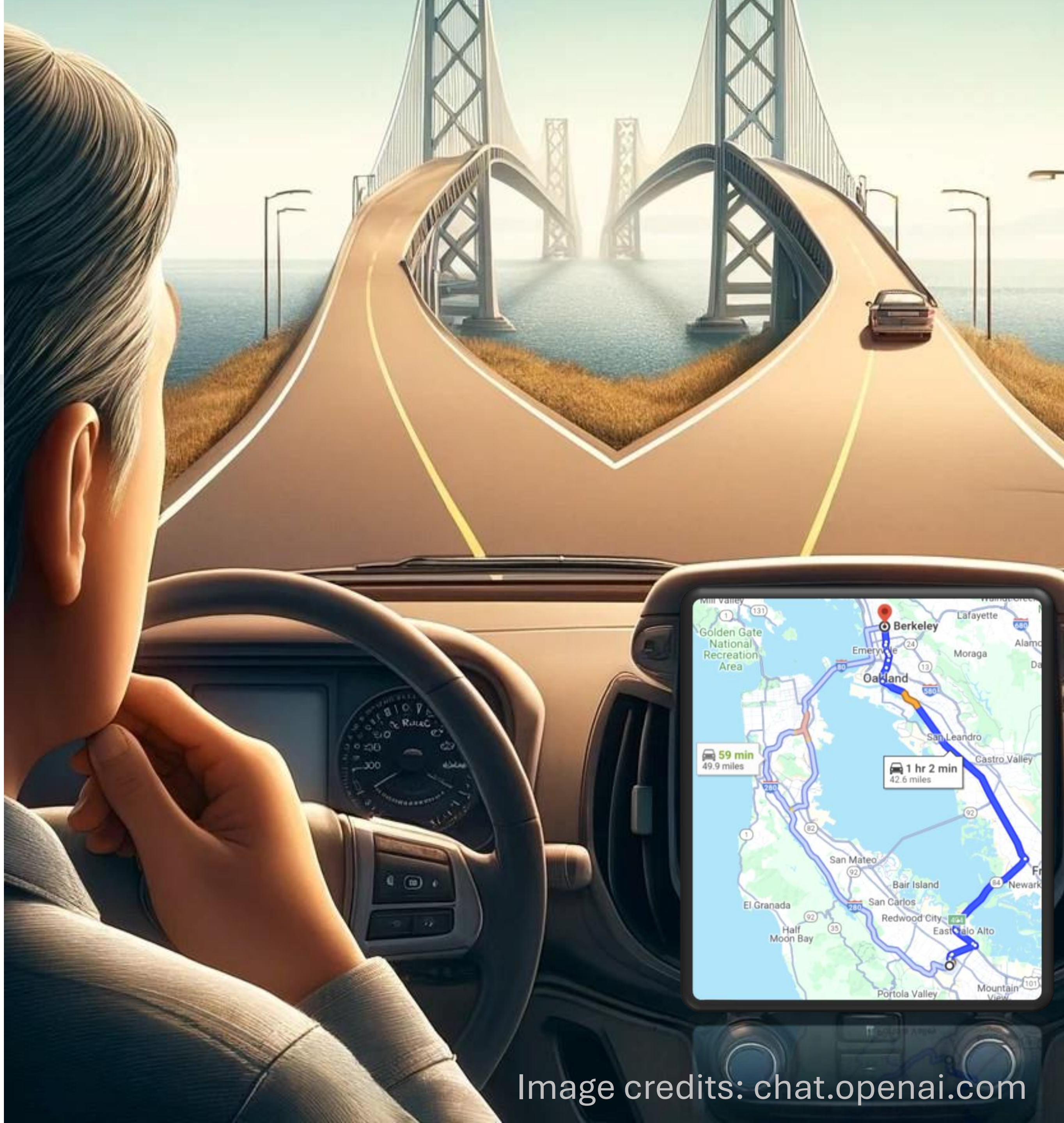


Image credits: chat.openai.com

Example in Math

- Every day $t \in \{1, \dots, T\}$ you have two options: (1) Bay, (2) Dumbarton

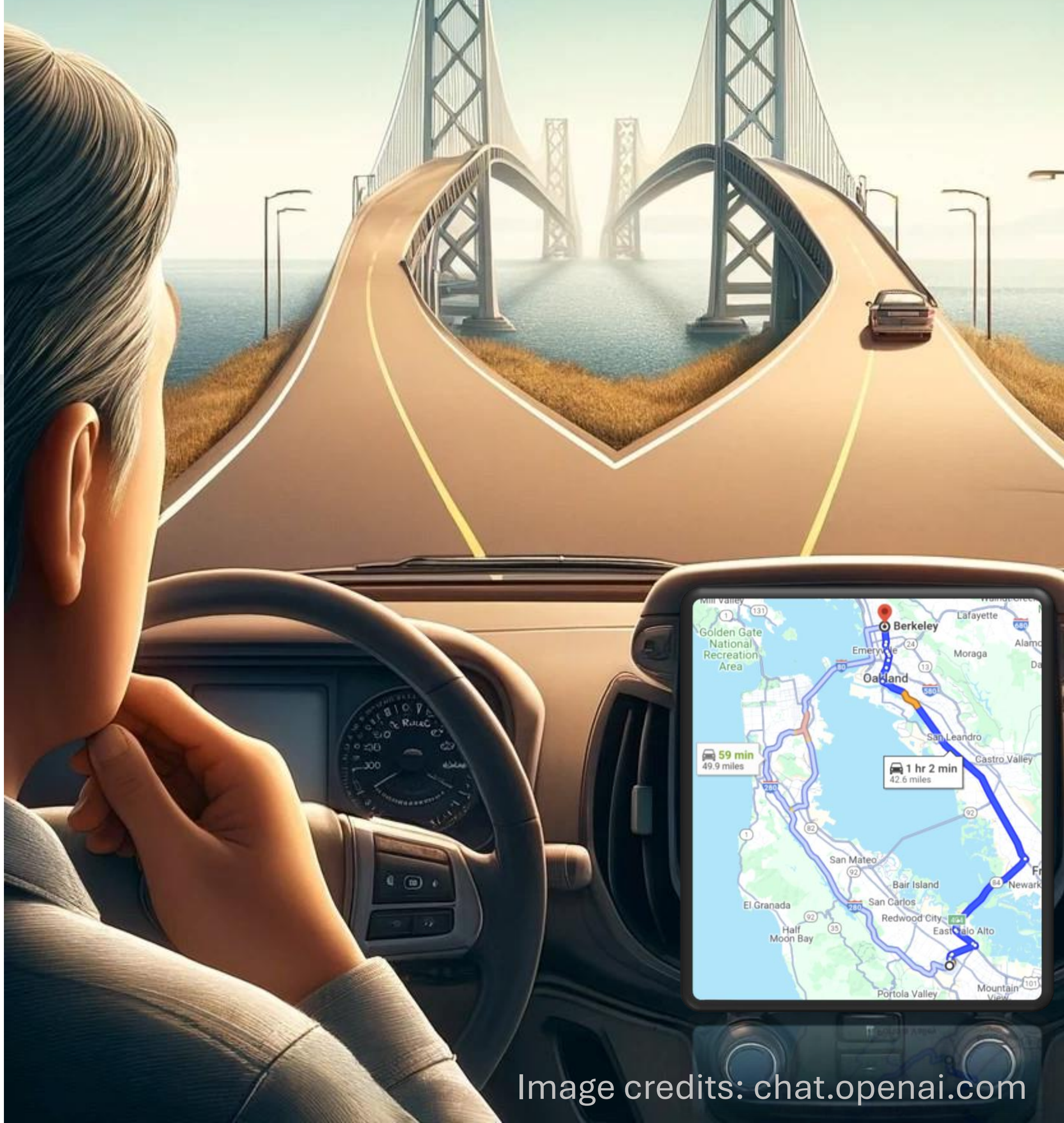


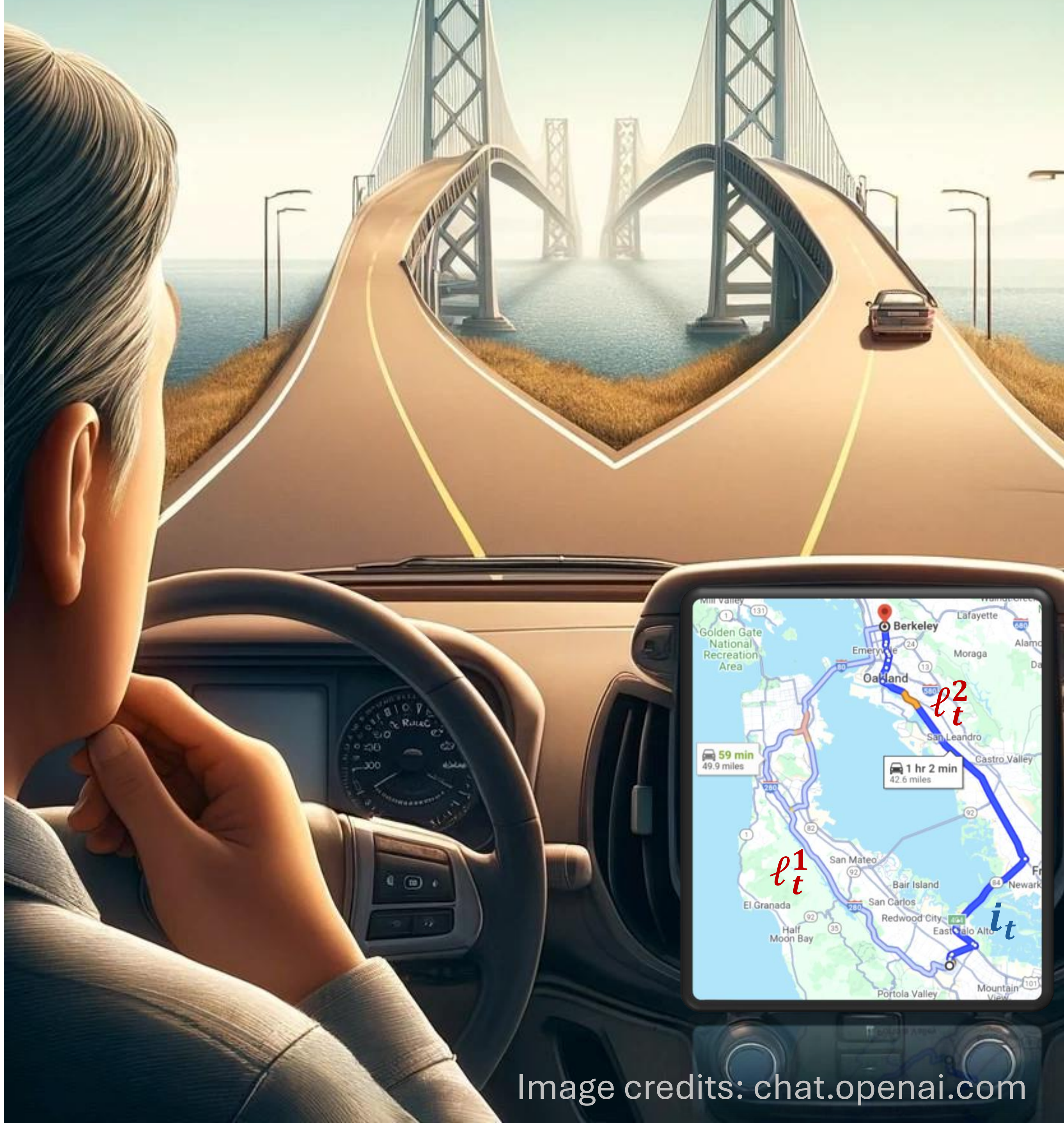
Image credits: chat.openai.com

Example in Math

- Every day $t \in \{1, \dots, T\}$ you have two options: (1) Bay, (2) Dumbarton
- Don't know which route will have more traffic jams: $\ell_t = \left(\boxed{\ell_t^1}, \ell_t^2\right)$

  # of jams on route (1)

- You choose some option $i_t \in \{1,2\}$
- You observe the traffic jams on both routes, i.e., you observe ℓ_t



Example in Math

- Device a choice picking algorithm $\boldsymbol{i_t}$

- **Goal.** At end of the year, looking back, *not regret much* either "always taking Bay" or "always taking Dumbarton"

$$\mathrm{Regret}(\ell_{1:T}) = \boxed{\frac{1}{T}\sum_{t=1}^{T} \ell_t^{i_t}} - \min_{i\in\{1,2\}} \boxed{\frac{1}{T}\sum_{t=1}^{T} \ell_t^{i}}$$

Short-hand notation for sequence of loss vectors $(\ell_1, \dots, \ell_T)$

Average # of jams you encountered

Average # of jams you would have encountered had you always chosen bridge i

Image credits: chat.openai.com

A choice picking algorithm is called a *no-regret learning algorithm* if the *worst-case regret* over any sequence of losses

$$R(T) = \sup_{\ell_{1:T}} \mathrm{Regret}(\ell_{1:T})$$

vanishes to zero with the number of periods

$$R(T) \to 0$$

Elements of a No-Regret Algorithm

Natural Algorithm

- Every day, choose option with the best historical performance

$$i_t = \underset{i \in \{1,2\}}{\operatorname{argmin}} \boxed{\sum_{\tau=1}^{t-1} \ell_i^\tau}$$

Total # of jams you on bridge i in the past

- Many times, referred to as "Follow-the-Leader" (FTL) as we are following the action that has the leading historical performance

Failure of the Natural Algorithm

- Suppose traffic jams alternate every day between the two bridges
- Suppose that ties are broken in favorite of Bay bridge

option \ day	1	2	3	4	5	6	…
Bay	1	0	1	0	1	1	…
Du.	0	1	0	1	0	0	…
Choice							

Failure of the Natural Algorithm

- Suppose traffic jams alternate every day between the two bridges
- Suppose that ties are broken in favorite of Bay bridge

day / option	1	2	3	4	5	6	…
Bay	1	0	1	0	1	0	…
Du.	0	1	0	1	0	1	…
Choice	**Bay**	**Du.**	**Bay**	**Du.**	**Bay**	**Du.**	**…**
Historical Losses	Bay: 0 Du.: 0	Bay: 1 Du.: 0	Bay: 1 Du.: 1	Bay: 2 Du.: 1	Bay: 2 Du.: 2	Bay: 3 Du.: 2	

Failure of the Natural Algorithm

- Suppose traffic jams alternate every day between the two bridges
- Suppose that ties are broken in favorite of Bay bridge

option \ day	1	2	3	4	5	6	...
Bay	**1**	0	**1**	0	**1**	0	...
Du.	0	**1**	0	**1**	0	**1**	...
Choice	**Bay**	**Du.**	**Bay**	**Du.**	**Bay**	**Du.**	**...**

Historical Losses						
	Bay: 0 Du.: 0	Bay: 1 Du.: 0	Bay: 1 Du.: 1	Bay: 2 Du.: 1	Bay: 2 Du.: 2	Bay: 3 Du.: 2

- Total loss of algorithm is $T \Rightarrow$ Average loss is 1
- Loss of any fixed action is $T/2 \Rightarrow$ Average loss $1/2$

$$\text{Regret} = \frac{1}{2}$$

Problematic Traits of FTL

- The choice of an action each day is deterministic

- The chosen action is very unstable and can change even daily

Problematic Traits of FTL

- The choice of an action each day is deterministic
- We need to introduce randomization in our choices

- The chosen action is very unstable and can change even daily
- We need to make sure that our choice distribution does not change too much at each step

Why is randomization necessary?

Theorem. Any deterministic algorithm has worst-case regret $\geq 1/2$

Why is randomization necessary?

Theorem. Any deterministic algorithm has worst-case regret $\geq 1/2$

Proof.

- Consider the sequence of losses that assign loss 1 to the choice of the algorithm and 0 to the other choice
- Total loss of the algorithm is $T \Rightarrow$ average loss is 1
- The sum of losses of the two options is T
- Hence, one of two options must have total loss of at most $T/2$
- Average loss of that option is $1/2$

Randomized Algorithms

- At each period, choose action 1 with probability p_t and action 2 with probability $1 - p_t$
- Our expected loss is

Overloaded short-hand notation for expected loss

$$\boxed{\ell_t(p_t)} = p_t \ell_t^1 + (1 - p_t)\ell_t^2$$

- Our expected regret is

$$\text{Regret}(\ell_{1:T}) = \boxed{\frac{1}{T}\sum_{t=1}^{T} \ell_t(p_t)} - \min_{i \in \{1,2\}} \boxed{\frac{1}{T}\sum_{t=1}^{T} \ell_t^i}$$

Expected average # of jams you encountered

Average # of jams you would have encountered had you always chosen bridge i

A randomized choice picking algorithm is called a *no-regret learning algorithm* if the *worst-case expected regret* over any sequence of losses

$$R(T) = \sup_{\ell_{1:T}} \mathrm{Regret}(\ell_{1:T})$$

vanishes to zero with the number of periods

$$R(T) \to 0$$

Why is stability useful?

- For the FTL algorithm, regret for a loss sequence $\ell_{1:T}$ is upper bounded by stability of algorithm's choice, under that sequence

$$\text{Regret}(\ell_{1:T}) \leq \frac{1}{T}\sum_{t=1}^{T} 1\{i_t \neq i_{t-1}\} = \text{average \# of changes}$$

- **Intuition.** We behave *as if* we think that the historically best option will be the best option for the next period; if the historically best option change after we observe the next period loss, then our *assumption* is roughly accurate

Why is stability useful?

Suppose algorithm makes *relatively stable and historically well-performing* choices

- **Adversary** chooses $\ell_{1:T}$ trying to hurt us a lot, while keeping loss of one of the options small
- Assume adversary uses $\ell_{1:T}$ such that option 1 will be the best performing option at the end

$$\text{Regret}(\ell_{1:T}) = \frac{1}{T}\sum_{t=1}^{T} E\left[\ell_{i_t}^t - \ell_1^t\right] = \frac{1}{T}\sum_{t=1}^{T} E\left[\left(\ell_2^t - \ell_1^t\right)1\{i_t \neq 1\}\right] = \sum_{t=1}^{T}\left(\ell_2^t - \ell_1^t\right)\Pr(i_t \neq 1)$$

Why is stability useful?

Suppose algorithm makes *relatively stable and historically well-performing* choices

- **Adversary** chooses $\ell_{1:T}$ trying to hurt us a lot, while keeping loss of one of the options small
- Assume adversary uses $\ell_{1:T}$ such that option 1 will be the best performing option at the end

$$\mathrm{Regret}(\ell_{1:T}) = \frac{1}{T}\sum_{t=1}^{T} E\left[\ell_{i_t}^t - \ell_1^t\right] = \frac{1}{T}\sum_{t=1}^{T} E\left[\left(\ell_2^t - \ell_1^t\right)1\{i_t \neq 1\}\right] = \sum_{t=1}^{T} \boxed{\left(\ell_2^t - \ell_1^t\right)} \, \boxed{\Pr(i_t \neq 1)}$$

Adversary goal: Make us choose option 1 with small probability, while keeping difference $\ell_2^t - \ell_1^t$ large on average

Why is stability useful?

Suppose algorithm makes *relatively stable and historically well-performing* choices

- **Adversary** chooses $\ell_{1:T}$ trying to hurt us a lot, while keeping loss of one of the options small
- Assume adversary uses $\ell_{1:T}$ such that option 1 will be the best performing option at the end

$$\mathrm{Regret}(\ell_{1:T}) = \frac{1}{T}\sum_{t=1}^{T} E\left[\ell_{i_t}^t - \ell_1^t\right] = \frac{1}{T}\sum_{t=1}^{T} E\left[\left(\ell_2^t - \ell_1^t\right)1\{i_t \neq 1\}\right] = \sum_{t=1}^{T} \boxed{\left(\ell_2^t - \ell_1^t\right)}\ \boxed{\Pr(i_t \neq 1)}$$

Adversary goal: Make us choose option 1 with small probability, while keeping difference $\ell_2^t - \ell_1^t$ large on average

- **If we are not stable**, they can convince us to move to option 2, by introducing a "*single bad apple period*" for option 1

- **If we are stable**, they need to introduce "*many bad apple periods*" for option 1, to make us move, which will decrease the average difference $\ell_2^2 - \ell_1^t$ by a lot

Constructing a No-Regret Algorithm Formally

Stability and Regret, Formally

For convenience, let's rewrite FTL in terms of probabilistic choices

$$p_t = \operatorname*{argmin}_p L_{t-1}(p) := \boxed{\sum_{\tau=1}^{t-1} \ell_\tau(p)}$$

Historical performance of always choosing one with probability p

Short-hand notation for past performance of probability p

Stability and Regret, Formally

For convenience, let's rewrite FTL in terms of probabilistic choices

$$p_t = \underset{p}{\operatorname{argmin}} L_{t-1}(p) := \boxed{\sum_{\tau=1}^{t-1} \ell_\tau(p)}$$

Historical performance of always choosing one with probability p

Theorem. For any loss sequence, with $\ell_t^i \in [0,1]$:

$$\text{Regret}(\ell_{1:T}) \leq \boxed{2\frac{1}{T}\sum_{t=1}^{T}|p_{t+1} - p_t|}$$

Average stability of algorithm's choice distribution

Proof of Regret via Stability

- Thought experiment: suppose we could look one-step ahead!
- We then modify our FTL algorithm to include that next step loss

$$\tilde{p}_t = \underset{p}{\operatorname{argmin}} L_t(p) \coloneqq \boxed{\sum_{\tau=1}^{t} \ell_\tau(p)}$$

Historical performance of always choosing one with probability p, **including next period loss**

- We will call this Be-The-Leader (BTL)

Proof of Regret via Stability

- Thought experiment: suppose we could look one-step ahead!
- We then modify our FTL algorithm to include that next step loss

$$\tilde{p}_t = \underset{p}{\operatorname{argmin}}\, L_t(p) := \boxed{\sum_{\tau=1}^{t} \ell_\tau(p)}$$

Historical performance of always choosing one with probability p, **including next period loss**

- We will call this Be-The-Leader (BTL)

Lemma 1. The difference in average loss between FTL and BTL is upper bounded by the average stability (*Proof.* $p_{t+1} = \tilde{p}_t$)

Lemma 2 (Be-The-Leader Lemma). BTL has zero regret

Be-The-Leader Lemma (Proof by Induction)

- Suppose that up until period $t - 1$ we have zero regret

$$\sum_{\tau=1}^{t-1} \ell_\tau(\tilde{p}_\tau) \leq \min_p \sum_{\tau=1}^{t-1} \ell_\tau(p)$$

Be-The-Leader Lemma (Proof by Induction)

- Suppose that up until period $t-1$ we have zero regret

$$\sum_{\tau=1}^{t-1} \ell_\tau(\tilde{p}_\tau) \leq \min_p \sum_{\tau=1}^{t-1} \ell_\tau(p)$$

- Hence, up until period $t-1$ we have no regret against always choosing the next period probability $\tilde{p}_t$

$$\boxed{\sum_{\tau=1}^{t-1} \ell_\tau(\tilde{p}_\tau)} \leq \boxed{\sum_{\tau=1}^{t-1} \ell_\tau(\tilde{p}_t)}$$

Historical performance (until period $t-1$) of BTL algorithm

Historical performance (until period $t-1$) of always choosing the next period probability $\tilde{p}_t$ of BTL

Be-The-Leader Lemma (Proof by Induction)

- Suppose that up until period $t-1$ we have zero regret

$$\sum_{\tau=1}^{t-1} \ell_\tau(\tilde{p}_\tau) \leq \min_p \sum_{\tau=1}^{t-1} \ell_\tau(p)$$

- Hence, up until period $t-1$ we have no regret against always choosing the next period probability $\tilde{p}_t$

$$\ell_t(\tilde{p}_t) + \boxed{\sum_{\tau=1}^{t-1} \ell_\tau(\tilde{p}_\tau)} \leq \boxed{\sum_{\tau=1}^{t-1} \ell_\tau(\tilde{p}_t)} + \ell_t(\tilde{p}_t)$$

Add performance of BTL
choice on next period
loss on both sides

Be-The-Leader Lemma (Proof by Induction)

- Suppose that up until period $t-1$ we have zero regret

$$\sum_{\tau=1}^{t-1} \ell_\tau(\tilde{p}_\tau) \leq \min_p \sum_{\tau=1}^{t-1} \ell_\tau(p)$$

- Hence, up until period $t-1$ we have no regret against always choosing the next period probability $\tilde{p}_t$

$$\boxed{\ell_t(\tilde{p}_t) + \sum_{\tau=1}^{t-1} \ell_\tau(\tilde{p}_\tau)} \leq \boxed{\sum_{\tau=1}^{t-1} \ell_\tau(\tilde{p}_t) + \ell_t(\tilde{p}_t)}$$

Historical performance **(until period t)** of BTL algorithm

Historical performance **(until period t)** of always choosing probability $\tilde{p}_t$

Be-The-Leader Lemma (Proof by Induction)

- Suppose that up until period $t-1$ we have zero regret

$$\sum_{\tau=1}^{t-1} \ell_\tau(\tilde{p}_\tau) \le \min_p \sum_{\tau=1}^{t-1} \ell_\tau(p)$$

- Hence, up until period $t-1$ we have no regret against always choosing the next period probability $\tilde{p}_t$

$$\boxed{\sum_{\tau=1}^{t} \ell_\tau(\tilde{p}_\tau)} \le \boxed{\sum_{\tau=1}^{t} \ell_\tau(\tilde{p}_t)}$$

Historical performance **(until period t)** of BTL algorithm

Historical performance **(until period t)** of always choosing probability $\tilde{p}_t$

Be-The-Leader Lemma (Proof by Induction)

- Suppose that up until period $t-1$ we have zero regret

$$\sum_{\tau=1}^{t-1} \ell_\tau(\tilde{p}_\tau) \leq \min_p \sum_{\tau=1}^{t-1} \ell_\tau(p)$$

- Hence, up until period $t-1$ we have no regret against always choosing the next period probability $\tilde{p}_t$

$$\boxed{\sum_{\tau=1}^{t} \ell_\tau(\tilde{p}_\tau)} \leq \boxed{\sum_{\tau=1}^{t} \ell_\tau(\tilde{p}_t)} \boxed{\leq \min_p \sum_{\tau=1}^{t} \ell_\tau(p)}$$

Historical performance **(until period t)** of BTL algorithm

Historical performance **(until period t)** of always choosing probability $\tilde{p}_t$

By the definition of $\tilde{p}_t$ as the probability that minimizes this quantity

Recap: Stability and Regret

For convenience, let's rewrite FTL in terms of probabilistic choices

$$p_t = \operatorname*{argmin}_p L_{t-1}(p) := \boxed{\sum_{\tau=1}^{t-1} \ell_\tau(p)}$$

Historical performance of always choosing one with probability p

Theorem. For any loss sequence, with $\ell_t^i \in [0,1]$:

$$\text{Regret}(\ell_{1:T}) \leq 2 \boxed{\frac{1}{T} \sum_{t=1}^{T} |p_t - p_{t-1}|}$$

Average stability of algorithm's choice distribution

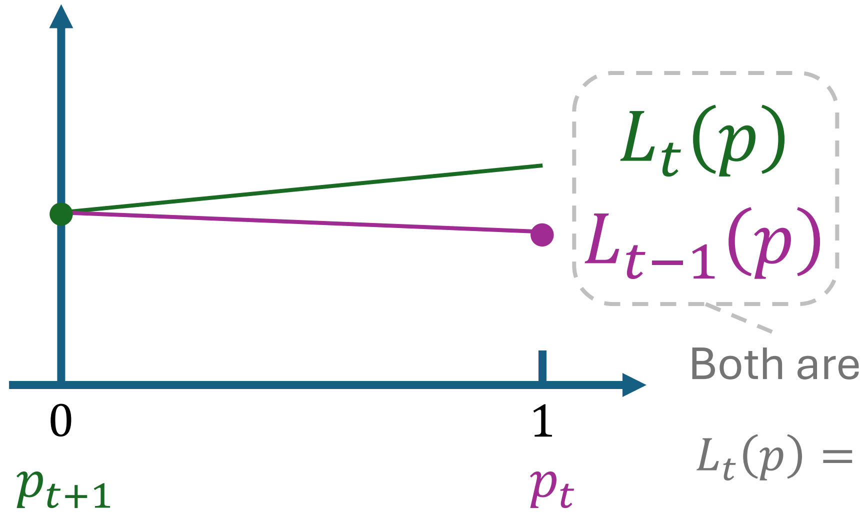
How do we stabilize FTL, such that it is stable irrespective of the loss sequence?

Closeness of optima of nearby functions

- The probabilities p_t and p_{t+1} are optima of very similar functions
$$p_t = \operatorname*{argmin}_p L_{t-1}(p)\,, \qquad p_{t+1} = \operatorname*{argmin}_p L_t(p)$$
- Note that: $L_t(p) - L_{t-1}(p) = \ell_t(p) \in [0, 1]$
- Given that these two functions only differ in the final loss, can we claim that their optima are close to each other?

Closeness of optima of nearby functions

- The probabilities p_t and p_{t+1} are optima of very similar functions
$$p_t = \operatorname*{argmin}_p L_{t-1}(p), \qquad p_{t+1} = \operatorname*{argmin}_p L_t(p)$$
- Note that: $L_t(p) - L_{t-1}(p) = \ell_t(p) \in [0,1]$
- Given that these two functions only differ in the final loss, can we claim that their optima are close to each other?

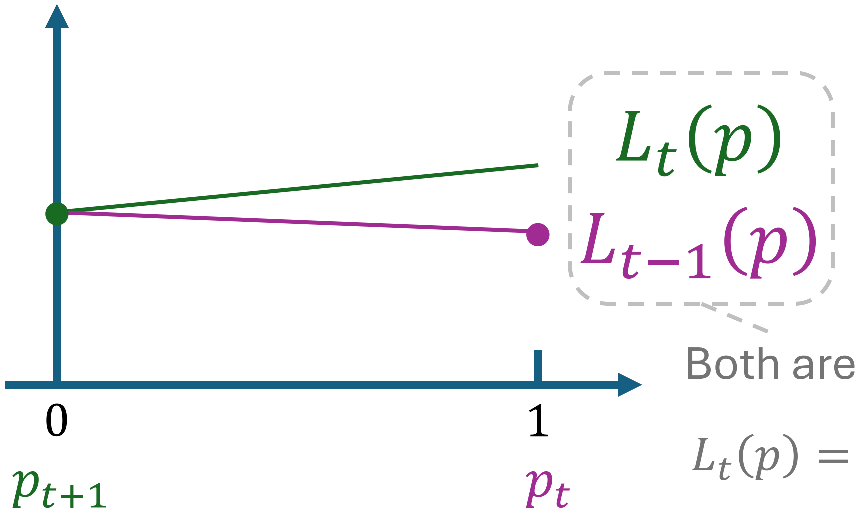


Both are linear functions of p

$$L_t(p) = p \sum_{\tau<t} \ell_\tau^1 + (1-p) \sum_{\tau<t} \ell_\tau^2$$

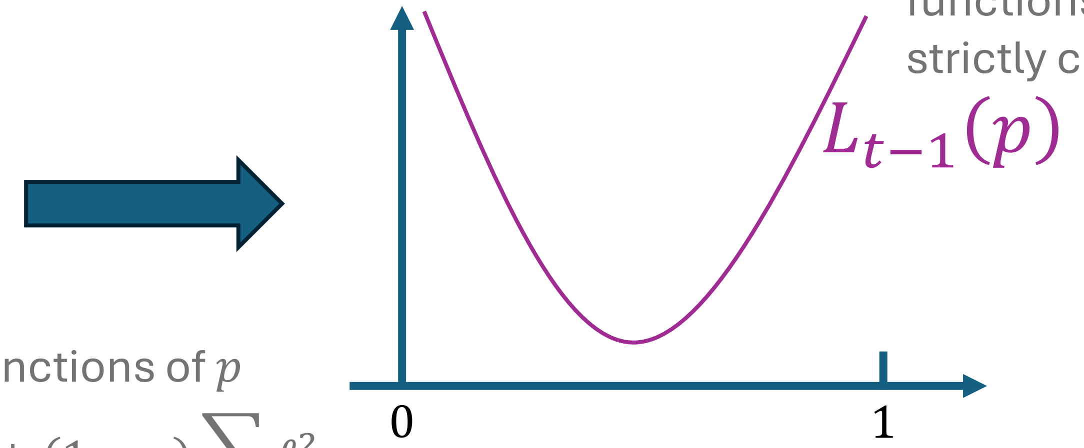
Closeness of optima of nearby functions

- The probabilities p_t and p_{t+1} are optima of very similar functions
$$p_t = \operatorname*{argmin}_p L_{t-1}(p), \qquad p_{t+1} = \operatorname*{argmin}_p L_t(p)$$
- Note that: $L_t(p) - L_{t-1}(p) = \ell_t(p) \in [0,1]$
- Given that these two functions only differ in the final loss, can we claim that their optima are close to each other?



Both are linear functions of p

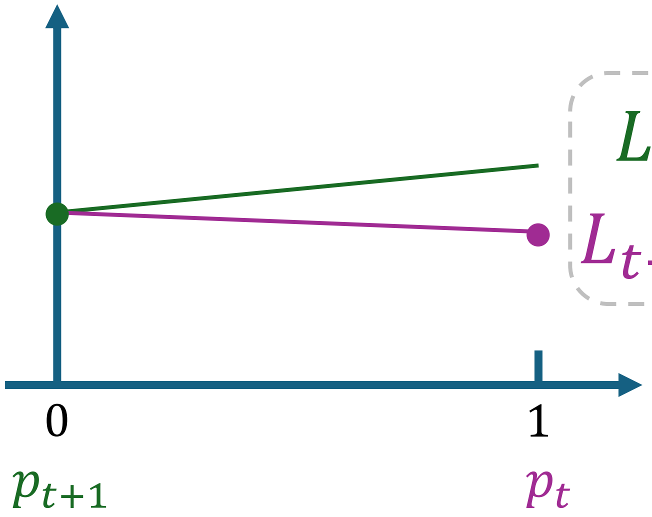
$$L_t(p) = p \sum_{\tau<t} \ell_\tau^1 + (1-p) \sum_{\tau<t} \ell_\tau^2$$



What if loss functions were strictly convex?

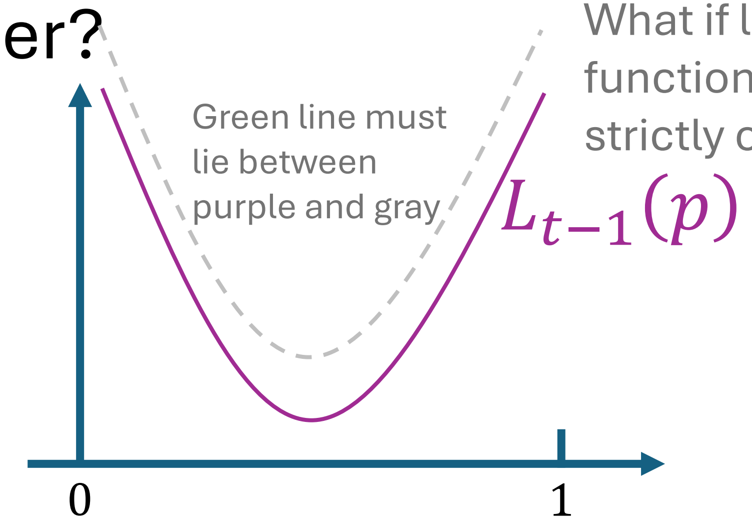
Closeness of optima of nearby functions

- The probabilities p_t and p_{t+1} are optima of very similar functions
$$p_t = \operatorname*{argmin}_p L_{t-1}(p), \qquad p_{t+1} = \operatorname*{argmin}_p L_t(p)$$
- Note that: $L_t(p) - L_{t-1}(p) = \ell_t(p) \in [0,1]$
- Given that these two functions only differ in the final loss, can we claim that their optima are close to each other?



Both are linear functions of p

$$L_t(p) = p\sum_{\tau<t}\ell_\tau^1 + (1-p)\sum_{\tau<t}\ell_\tau^2$$



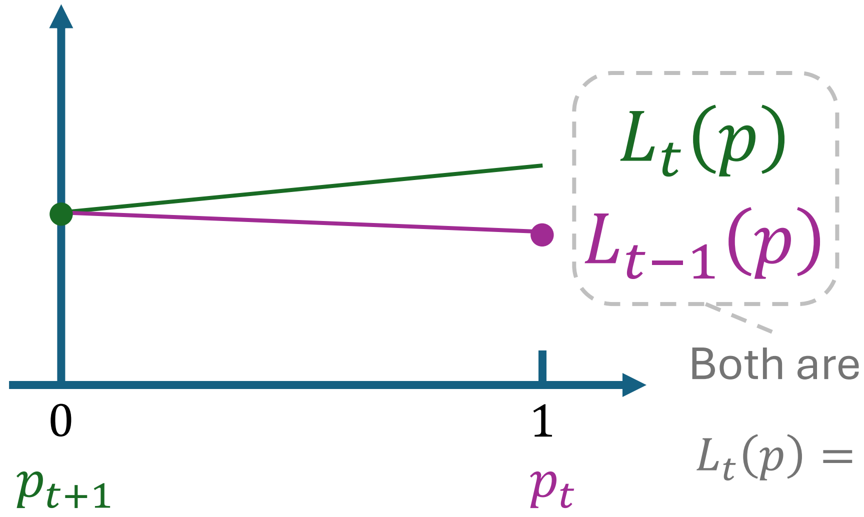
What if loss functions were strictly convex?

Closeness of optima of nearby functions

- The probabilities p_t and p_{t+1} are optima of very similar functions

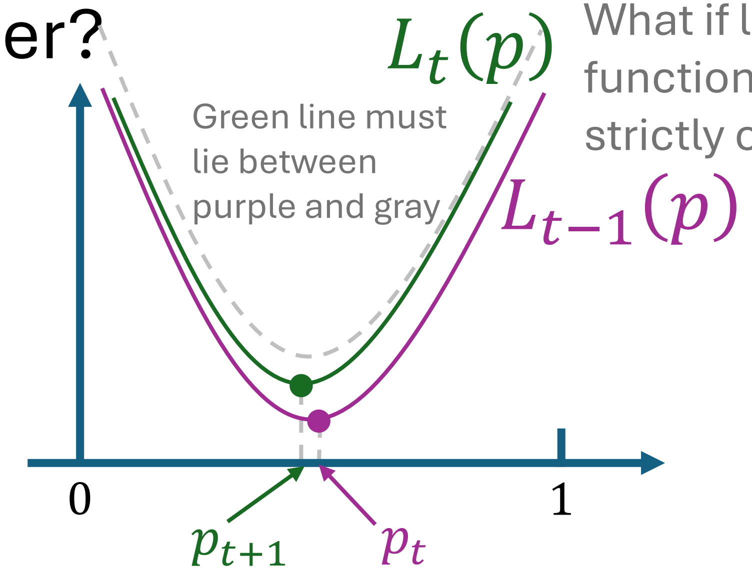
$$p_t = \underset{p}{\operatorname{argmin}}\, L_{t-1}(p)\,, \qquad p_{t+1} = \underset{p}{\operatorname{argmin}}\, L_t(p)$$

- Note that: $L_t(p) - L_{t-1}(p) = \ell_t(p) \in [0, 1]$
- Given that these two functions only differ in the final loss, can we claim that their optima are close to each other?



Both are linear functions of p

$$L_t(p) = p \sum_{\tau<t} \ell_\tau^1 + (1-p) \sum_{\tau<t} \ell_\tau^2$$



What if loss functions were strictly convex?

Stability via Convexity Theorem

Suppose two functions $f, g\colon [0,1] \to R$ are $1/\eta$-strictly convex

$$f''(p), g''(p) \geq \frac{1}{\eta}$$

and their difference $h(p) = g(p) - f(p)$ is L-Lipschitz

$$|h(p) - h(p')| \leq L \cdot |p - p'|$$

Let p_f, p_g be their corresponding minima. Then

$$\left|p_f - p_g\right| \leq \eta \cdot L$$

Proof. For any strictly convex function the value grows at least quadratically as we move away from the optimum (*Taylor expansion*)

$$f(p) - f(p_f) = \underbrace{\boxed{f'(p_f) \cdot (p - p_f)}}_{\geq 0} + \underbrace{\boxed{\frac{f''(\bar{p})}{2}}}_{\geq 1/2\eta}(p - p_f)^2 \geq \frac{1}{2\eta}(p - p_f)^2$$

≥ 0 by first-order optimality of p_f

$\geq 1/2\eta$ by strict convexity

Proof. For any strictly convex function the value grows at least quadratically as we move away from the optimum (*Taylor expansion*)

$$\boxed{f(p) - f(p_f)} = \underbrace{\boxed{f'(p_f) \cdot (p - p_f)}}_{\geq 0 \text{ by first-order optimality of } p_f} + \underbrace{\boxed{\frac{f''(\bar{p})}{2}}}_{\geq 1/2\eta \text{ by strict convexity}} (p - p_f)^2 \geq \boxed{\frac{1}{2\eta}(p - p_f)^2}$$

sub-optimality of any point p

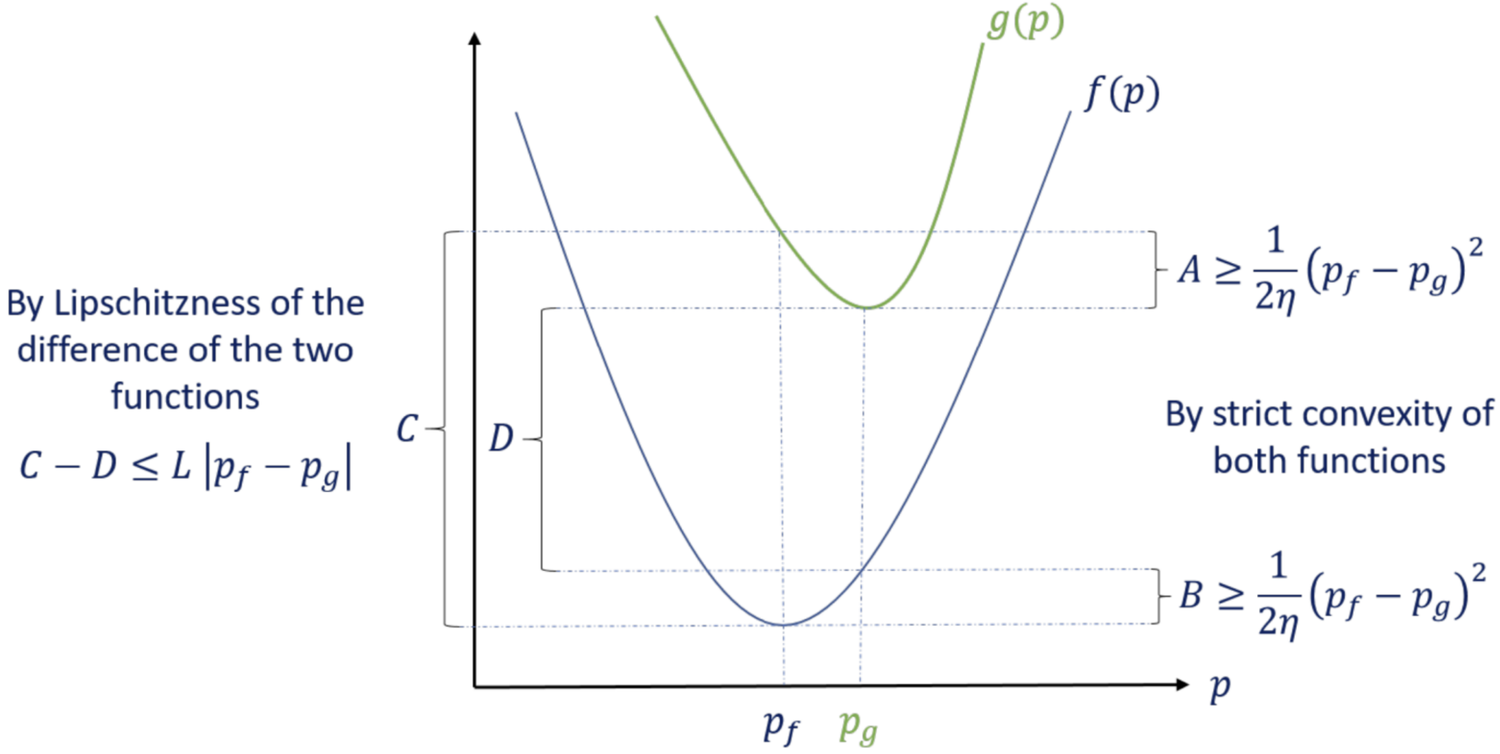
≥ 0
by first-order optimality of p_f

$\geq 1/2\eta$
by strict convexity

Grows quadratically with distance from optimum

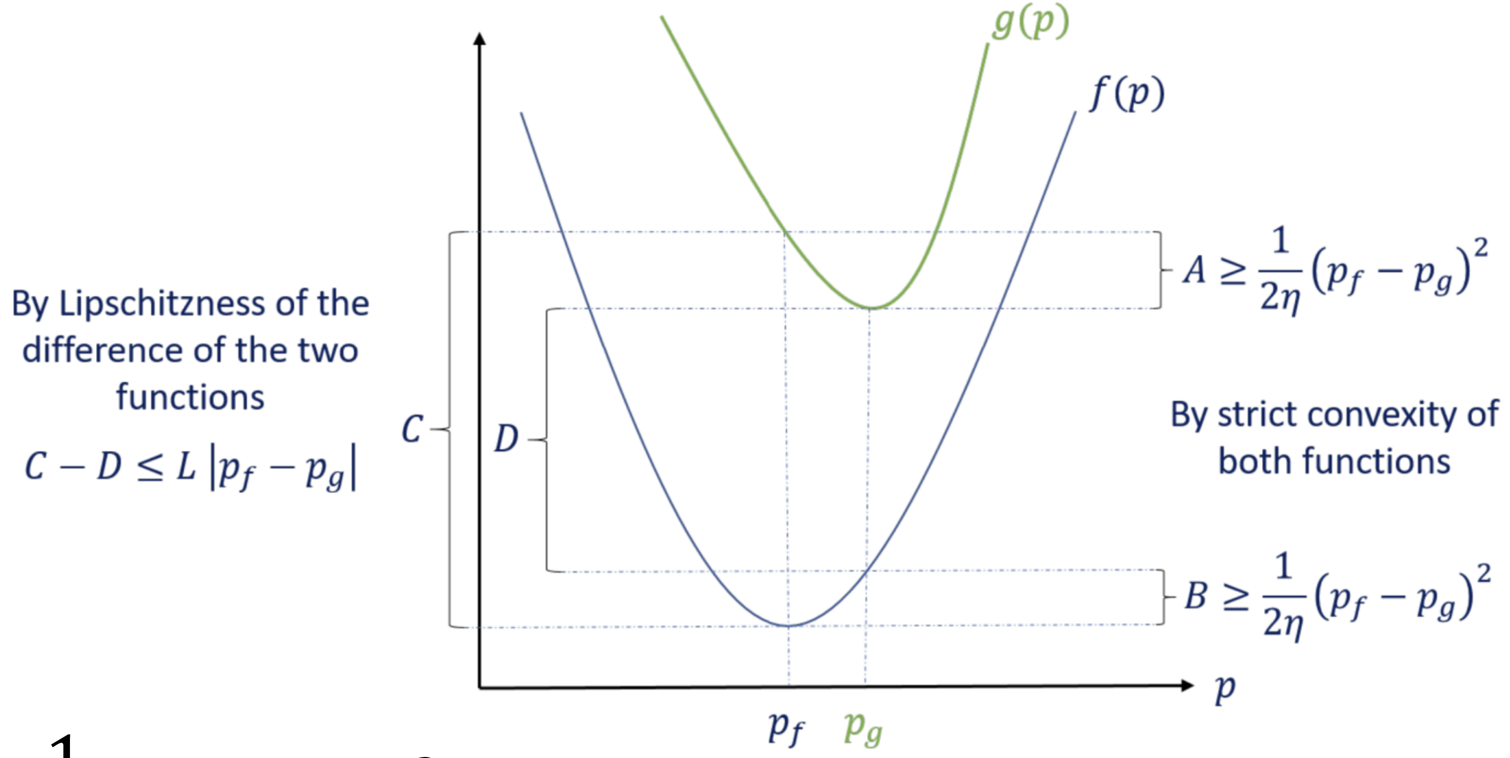
Proof. For any strictly convex function the value grows at least quadratically as we move away from the optimum

$$f(p) - f(p_f) = f'(p_f) \cdot (p - p_f) + \frac{f''(\bar{p})}{2}(p - p_f)^2 \geq \frac{1}{2\eta}(p - p_f)^2$$



Proof. For any strictly convex function the value grows at least quadratically as we move away from the optimum

$$f(p) - f(p_f) = f'(p_f) \cdot (p - p_f) + \frac{f''(\bar{p})}{2}(p - p_f)^2 \geq \frac{1}{2\eta}(p - p_f)^2$$



$$\frac{1}{\eta}(p_f - p_g)^2 = A + B = C - D \leq L \cdot |p_f - p_g|$$

How do we use the stability property of strictly convex functions to stabilize FTL?

Follow-the-Regularized-Leader (FTRL)

Add a strictly convex "regularizer" to the FTL objective

$$p_t = \operatorname*{argmin}_p \boxed{L_{t-1}(p)} + \frac{1}{\eta}\boxed{\mathcal{R}(p)}$$

1-strictly convex function of p that stabilizes the minimizer

Historical performance of always choosing one with probability p

Regret of FTRL

Add a strictly convex "regularizer" to the FTL objective

$$p_t = \operatorname*{argmin}_p \boxed{L_{t-1}(p)} + \frac{1}{\eta}\boxed{\mathcal{R}(p)}$$

1-strictly convex function of p that stabilizes the minimizer

Historical performance of always choosing one with probability p

Theorem. For any loss sequence, with $\ell_t^i \in [0,1]$:

$$\text{Regret}(\ell_{1:T}) \leq \boxed{2\frac{1}{T}\sum_{t=1}^{T}|p_{t+1} - p_t|} + \boxed{\frac{1}{\eta T}\left(\max_p \mathcal{R}(p) - \min_p \mathcal{R}(p)\right)}$$

Average stability of algorithm's choice distribution

Average loss distortion caused by regularizer

Proof of Regret of FTRL

Suppose we could foresee the next period loss and played

$$\tilde{p}_t = \underset{p}{\operatorname{argmin}} \boxed{L_t(p)} + \frac{1}{\eta} \boxed{\mathcal{R}(p)}$$

1-strictly convex function of p that stabilizes the minimizer

Historical performance **(including next period t)** of always choosing one with probability p

We will call this algorithm Be-The-*Regularized*-Leader (BTRL)

Proof of Regret of FTRL

Suppose we could foresee the next period loss and played

$$\tilde{p}_t = \underset{p}{\operatorname{argmin}} \boxed{L_t(p)} + \frac{1}{\eta} \boxed{\mathcal{R}(p)}$$

1-strictly convex function of p that stabilizes the minimizer

Historical performance **(including next period t)** of always choosing one with probability p

We will call this algorithm Be-The-*Regularized*-Leader (BTRL)

Lemma 1. The difference in average loss between FTRL and BTRL is upper bounded by the average stability (*Proof.* $p_{t+1} = \tilde{p}_t$)

Lemma 2 (Be-The-Regularized-Leader). BTRL regret $\leq$ **distortion**

Be-the-Regularized-Leader Lemma

BTRL Lemma. BTRL has regret $\leq \frac{1}{\eta T}\left(\max_{p} \mathcal{R}(p) - \min_{p} \mathcal{R}(p)\right)$

Be-the-Regularized-Leader Lemma

BTRL Lemma. BTRL has regret $\leq \frac{1}{\eta T}\left(\max_p \mathcal{R}(p) - \min_p \mathcal{R}(p)\right)$

Proof

- We can think of the regularizer as a "*loss at time* 0"
- Then BTRL is BTL on this augmented loss sequence
- Invoking the BTL lemma we get by induction (with $p_0 = \min_p \mathcal{R}(p)$)

$$\frac{1}{\eta}\mathcal{R}(p_0) + \sum_{t=1}^{T} \ell_t(\tilde{p}_t) \leq \min_p \frac{1}{\eta}\mathcal{R}(p) + \sum_{t=1}^{T} \ell_t(p) \leq \min_p \sum_{t=1}^{T} \ell_t(p) + \max_p \frac{1}{\eta}\mathcal{R}(p)$$

Recap: Regret of FTRL

Add a strictly convex "regularizer" to the FTL objective

$$p_t = \operatorname*{argmin}_p \boxed{L_{t-1}(p)} + \frac{1}{\eta}\boxed{\mathcal{R}(p)}$$

1-strictly convex function of p that stabilizes the minimizer

Historical performance of always choosing one with probability p

Theorem. For any loss sequence, with $\ell_t^i \in [0,1]$:

$$\text{Regret}(\ell_{1:T}) \leq \boxed{2\frac{1}{T}\sum_{t=1}^{T}|p_{t+1} - p_t|} + \boxed{\frac{1}{\eta T}\left(\max_p \mathcal{R}(p) - \min_p \mathcal{R}(p)\right)}$$

Average stability of algorithm's choice distribution

Average loss distortion caused by regularizer

Stability of FTRL

Theorem. For the FTRL algorithm: $|p_t - p_{t+1}| \leq 2 \cdot \eta$

Stability of FTRL

Theorem. For the FTRL algorithm: $|p_t - p_{t+1}| \leq 2 \cdot \eta$

Proof. Invoke stability of strictly convex functions theorem with

$$f(p) = L_{t-1}(p) + \frac{1}{\eta}\mathcal{R}(p), \qquad g(p) = L_t(p) + \frac{1}{\eta}\mathcal{R}(p)$$

$$h(p) = g(p) - f(p) = \ell_t(p) = p\left(\ell_t^1 - \ell_t^2\right) + \ell_t^2 \Rightarrow 2 - \text{Lipschitz}$$

(linear) + ($1/\eta$-strictly convex) function is $1/\eta$-strictly convex

Punchline

(FTRL) $$p_t = \operatorname*{argmin}_p \boxed{L_{t-1}(p)} + \frac{1}{\eta}\boxed{\mathcal{R}(p)}$$

1-strictly convex function of p that stabilizes the minimizer

Historical performance of always choosing one with probability p

Corollary. The regret of FTRL $\leq \boxed{2\eta} + \boxed{\frac{1}{\eta T}\left(\max_p \mathcal{R}(p) - \min_p \mathcal{R}(p)\right)}$

Average stability induced by regularizer

Average loss distortion caused by regularizer

What is a good regularizer?

Regularizing Probabilities via Negative Entropy

- A natural regularizer on distributions is the ***negative entropy***
$$R(p) = p\log(p) + (1-p)\log(1-p)$$
- **Intuition:** FTRL with negative entropy picks distribution over choices that tries to *minimize historical loss, while having large entropy* (i.e. not very deterministic)
- Negative entropy is 1-strictly convex and takes values in $[-\log(2)\,, 0]$

$$\mathcal{R}'(p) = \log(p) + 1 - \log(1-p) - 1 = \log\left(\frac{p}{1-p}\right)$$
$$\mathcal{R}''(p) = \frac{1}{p} + \frac{1}{1-p} = \frac{1}{p\,(1-p)} \geq 1$$

FTRL with Negative Entropy

Corollary. Regret of FTRL with negative entropy regularizer is

$$R(T) \leq 2\eta + \frac{\log(2)}{\eta\, T}$$

Choosing $\eta = \sqrt{\frac{\log(2)}{2T}}$, to make the two terms equal

$$R(T) \leq \sqrt{\frac{2\log(2)}{T}} \to 0$$

Closed Form of FTRL with Negative Entropy

We are optimizing

$$p_t = \min_p L_{t-1}(p) + \frac{1}{\eta}\mathcal{R}(p)$$

Note that:

$$L_{t-1}(p) = p \sum_{\tau<t} \ell_t^1 + (1-p)\sum_{\tau<t} \ell_t^2 = p\left(\overbrace{\boxed{\sum_{\tau<t} \ell_t^1}}^{\mathcal{L}_t^1} - \overbrace{\boxed{\sum_{\tau<t} \ell_t^2}}^{\mathcal{L}_t^2}\right) + \sum_{\tau<t} \ell_t^2$$

Hence our minimization is of the form

$$p_t = \min_p p\left(\mathcal{L}_t^1 - \mathcal{L}_t^2\right) + \frac{1}{\eta}\mathcal{R}(p)$$

Closed Form of FTRL with Negative Entropy

$$p_t = \min_p p\left(\mathcal{L}_t^1 - \mathcal{L}_t^2\right) + \frac{1}{\eta}\mathcal{R}(p)$$

First order conditions

$$\mathcal{L}_t^1 - \mathcal{L}_t^2 + \frac{1}{\eta}\mathcal{R}'(p) = 0 \Rightarrow \mathcal{L}_t^1 - \mathcal{L}_t^2 + \frac{1}{\eta}\log\left(\frac{p}{1-p}\right) = 0$$

which implies that

$$\frac{p}{1-p} = \exp\left(-\eta\left(\mathcal{L}_t^1 - \mathcal{L}_t^2\right)\right) \Rightarrow p = \frac{\exp\left(-\eta\left(\mathcal{L}_t^1 - \mathcal{L}_t^2\right)\right)}{1 + \exp\left(-\eta\left(\mathcal{L}_t^1 - \mathcal{L}_t^2\right)\right)}$$

$$= \frac{\exp\left(-\eta\mathcal{L}_t^1\right)}{\exp\left(-\eta\mathcal{L}_t^2\right) + \exp\left(-\eta\mathcal{L}_t^1\right)}$$

Punchline

At each period t play each action $i \in \{1, 2\}$ with probability

$$p_t^i \propto \boxed{\exp(-\eta \mathcal{L}_t^i)}$$

Play each option with probability proportional to the exponential of its historical performance

Choosing $\eta = \sqrt{\frac{\log(2)}{2T}}$ we get $R(T) \le \sqrt{\frac{2\log(2)}{T}} \to 0$

Simpler update implementation

$$p_t^i = \frac{\exp(-\eta \mathcal{L}_t^i)}{\sum_j \exp\left(-\eta \mathcal{L}_t^j\right)} = \frac{\exp(-\eta \mathcal{L}_{t-1}^i)\exp(-\eta \ell_{t-1}^i)}{\sum_j \exp\left(-\eta \mathcal{L}_{t-1}^j\right)\exp\left(-\eta \ell_{t-1}^j\right)} = \boxed{\frac{p_{t-1}^i \exp(-\eta \ell_{t-1}^i)}{\sum_j p_{t-1}^j \exp\left(-\eta \ell_{t-1}^j\right)}}$$

Exponential weight updates algorithm!
(aka Hedge, Multiplicative Weight Updates, EXP, ...)

What if we have many options?

The n action case

Short-hand for n-dimensional simplex
$\Delta(n) \coloneqq \left\{x \in R^n : x_i \geq 0, \sum_{i=1}^{n} x_i = 1\right\}$

At each period choose a distribution $p_t \in \Delta(n)$ over n actions

Observe a loss vector $\ell_t \in [0,1]^n$ and incur loss $\langle p_t, \ell_t \rangle$

Short-hand for inner product between two vectors

$$\mathrm{Regret}(\ell_{1:T}) = \frac{1}{T}\sum_{t=1}^{T}\langle p_t, \ell_t\rangle - \min_{i=1}^{n} \frac{1}{T}\sum_{t=1}^{T} \ell_t^i$$

The n action case

Short-hand for n-dimensional simplex
$\Delta(n) \coloneqq \left\{x \in R^n : x_i \geq 0, \sum_{i=1}^n x_i = 1\right\}$

At each period choose a distribution $p_t \in \Delta(n)$ over n actions

Observe a loss vector $\ell_t \in [0,1]^n$ and incur loss $\langle p_t, \ell_t \rangle$

Short-hand for inner product between two vectors

(FTRL) $$p_t = \min_p L_{t-1}(p) + \frac{1}{\eta}\mathcal{R}(p) = \min_p \sum_{\tau<t} \langle p, \ell_t \rangle + \frac{1}{\eta}\mathcal{R}(p)$$

The n action case

Short-hand for n-dimensional simplex
$\Delta(n) := \{x \in R^n : x_i \geq 0, \sum_{i=1}^n x_i = 1\}$

At each period choose a distribution $p_t \in \Delta(n)$ over n actions

Observe a loss vector $\ell_t \in [0,1]^n$ and incur loss $\langle p_t, \ell_t \rangle$

Short-hand for inner product between two vectors

(FTRL) $$p_t = \min_p L_{t-1}(p) + \frac{1}{\eta}\mathcal{R}(p) = \min_p \sum_{\tau<t} \langle p, \ell_t \rangle + \frac{1}{\eta}\mathcal{R}(p)$$

For the *negative entropy* regularizer, leads to the simple EXP algorithm

$$p_t^i \propto \boxed{p_{t-1}^i \exp\left(-\eta \ell_{t-1}^i\right)}$$

Play each action with probability proportional to the exponential of its historical performance

The negative entropy is *1-strongly convex* and now takes values in $[-\log(\mathrm{n}), 0]$

$$R(T) \leq 2\eta + \frac{\log(n)}{\eta T} \boxed{\leq \sqrt{\frac{2\log(n)}{T}} \to 0}$$

For $\eta = \sqrt{\frac{\log(n)}{2T}}$

Strong-convexity in n-dimensions

Gradient of a function: $\nabla f(p) = \left(\frac{\partial}{\partial p_1} f(p), \ldots, \frac{\partial}{\partial p_n} f(p)\right)$

A function $f: R^n \to R$ is σ-strongly convex if

$$f(p) - f(p') \geq \langle \nabla f(p'), p - p' \rangle + \frac{\sigma}{2} \|p - p'\|^2$$

Some norm in the n-dimensional vector space:

e.g. $\|p\|_2 = \sqrt{\sum_{i=1}^{n} p_i^2}$ or $\|p\|_1 = \sum_{i=1}^{n} |p_i|$

Strong-convexity in n-dimensions

Gradient of a function: $\nabla f(p) = \left(\frac{\partial}{\partial p_1} f(p), \dots, \frac{\partial}{\partial p_n} f(p)\right)$

A function $f: R^n \to R$ is σ-strongly convex if

$$f(p) - f(p') \geq \langle \nabla f(p'), p - p' \rangle + \frac{\sigma}{2} \|p - p'\|^2$$

Some norm in the n-dimensional vector space:

e.g. $\|p\|_2 = \sqrt{\sum_{i=1}^{n} p_i^2}$ or $\|p\|_1 = \sum_{i=1}^{n} |p_i|$

For a twice-differentiable function f, implied by

$$\forall \bar{p}:\ (p - p')^\top \nabla^2 f(\bar{p}) (p - p') \geq \sigma \|p - p'\|^2$$

Hessian of a function: $\nabla^2 f(p) = \begin{pmatrix} \frac{\partial^2}{\partial p_1^2} f(p) & \cdots & \frac{\partial^2}{\partial p_1 \partial p_n} f(p) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial p_n \partial p_1} f(p) & \cdots & \frac{\partial^2}{\partial p_n^2} f(p) \end{pmatrix}$

Strong-convexity in n-dimensions

Gradient of a function: $\nabla f(p) = \left(\frac{\partial}{\partial p_1} f(p), \ldots, \frac{\partial}{\partial p_n} f(p)\right)$

A function $f: R^n \to R$ is σ-strongly convex if

$$f(p) - f(p') \geq \langle \boxed{\nabla f(p')}, p - p' \rangle + \frac{\sigma}{2} \boxed{\|p - p'\|^2}$$

Some norm in the n-dimensional vector space:

e.g. $\|p\|_2 = \sqrt{\sum_{i=1}^n p_i^2}$ or $\|p\|_1 = \sum_{i=1}^n |p_i|$

Theorem. Suppose two functions $f, g: [0,1] \to R$ are $1/\eta$-strongly convex and their difference $h(p) = g(p) - f(p)$ is L-Lipschitz

$$|h(p) - h(p')| \leq L \cdot \|p - p'\|$$

Let p_f, p_g be their corresponding minima. Then

$$\|p_f - p_g\| \leq \eta \cdot L$$

Punchline

(FTRL) $$p_t = \operatorname*{argmin}_p \boxed{L_{t-1}(p)} + \frac{1}{\eta}\boxed{\mathcal{R}(p)}$$

1-strongly convex function of p that stabilizes the minimizer

Historical performance of always choosing one with probability p

Theorem. Assuming the loss function at each period
$$\ell_t(p) = \langle p, \ell_t \rangle$$
is L-Lipschitz with respect to some norm $||\cdot||$ and the regularizer is 1-strongly convex with respect to the same norm then

$$\text{Regret} - \text{FTRL}(T) \leq \boxed{\eta L} + \boxed{\frac{1}{\eta T}\left(\max_p \mathcal{R}(p) - \min_p \mathcal{R}(p)\right)}$$

Average stability induced by regularizer

Average loss distortion caused by regularizer

Punchline

(EXP) $$p_t^i \propto p_{t-1}^i \exp\left(-\eta \ell_{t-1}^i\right)$$

Corollary. If all actions have losses $\ell_t^i \in [0,1]$, then loss function is 1-Lipschitz with respect to the norm $\|\cdot\|_1$. The negative entropy is 1-strongly convex with respect to the norm $\|\cdot\|_1$ *(bonus exercise)*.

$$\text{Regret} - \text{EXP}(T) \leq 2\eta + \frac{\log(n)}{\eta T} \leq \boxed{\sqrt{\frac{2\log(n)}{T}} \to 0}$$

For $\eta = \sqrt{\frac{\log(n)}{2T}}$

What if loss function is not linear in chosen vector?

The general convex case

Convex

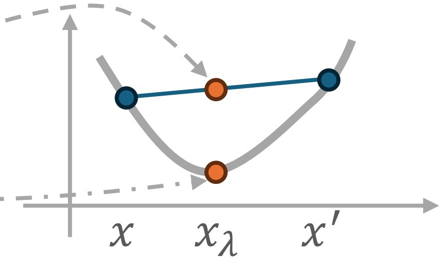
Not Convex

Convex set: $x, x' \in S \Rightarrow \lambda x + (1-\lambda)x' \in S$

At each period choose a vector $p_t \in S \subseteq R^n$, where S is a convex set

Observe a convex loss function $\ell_t : S \to R$ and incur loss $\ell_t(p_t)$

Convex loss: $\ell(\lambda x + (1-\lambda)x') \leq \lambda \ell(x) + (1-\lambda)\ell(x')$



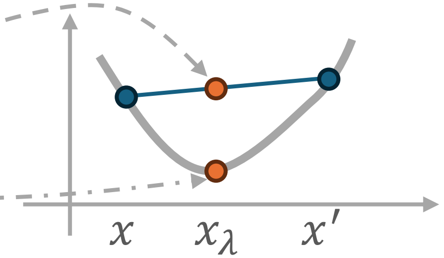
The general convex case

Convex set: $x, x' \in S \Rightarrow \lambda x + (1-\lambda)x' \in S$

At each period choose a vector $p_t \in S \subseteq R^n$, where S is a convex set

Observe a convex loss function $\ell_t: S \to R$ and incur loss $\ell_t(p_t)$

Convex loss: $\ell(\lambda x + (1-\lambda)x') \leq \lambda \ell(x) + (1-\lambda)\ell(x')$



$$\begin{pmatrix}\text{Linearized}\\ \text{FTRL}\end{pmatrix} p_t = \underset{p}{\operatorname{argmin}}\, \bar{L}_{t-1}(p) + \frac{1}{\eta}\mathcal{R}(p) = \underset{p}{\operatorname{argmin}} \sum_{\tau<t} \boxed{\langle p, \nabla \ell_t(p_\tau)\rangle} + \frac{1}{\eta}\mathcal{R}(p)$$

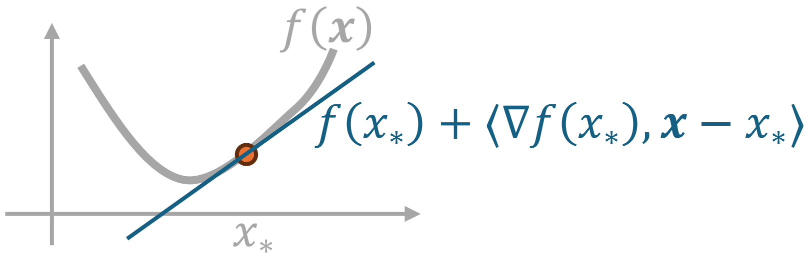
Linear approximation of loss around chosen point $\bar{\ell}_t(p) = \langle p, \nabla \ell_t(p_t)\rangle$

Linearization Lemma. $\mathrm{Regret}(\ell_{1:T}) \leq \mathrm{Regret}(\bar{\ell}_{1:T})$

Linearization Lemma: $\mathrm{Regret}(\ell_{1:T}) \leq \mathrm{Regret}(\bar{\ell}_{1:T})$

Convexity of function implies it lies above the linear approximation

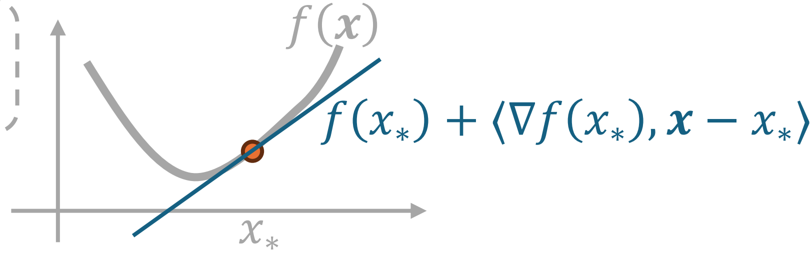
$$f(x) \geq f(x_*) + \langle \nabla f(x_*), x - x_* \rangle$$



Linearization Lemma: $\mathrm{Regret}(\ell_{1:T}) \leq \mathrm{Regret}\left(\bar{\ell}_{1:T}\right)$

Convexity of function implies it lies above the linear approximation

$$f(x) \geq f(x_*) + \langle \nabla f(x_*), x - x_* \rangle$$



By convexity of losses

$$\mathrm{Regret}(\ell_{1:T}) = \min_p \sum_{t=1}^{T} \ell_t(p_t) - \ell_t(p)$$

$$\leq \min_p \sum_{t=1}^{T} \langle \nabla \ell_t(p_t), p_t - p \rangle$$

$$= \min_p \sum_{t=1}^{T} \bar{\ell}_t(p_t) - \bar{\ell}_t(p) = \mathrm{Regret}\left(\bar{\ell}_{1:T}\right)$$

Punchline

(Linearized FTRL) $$p_t = \operatorname*{argmin}_p \boxed{\bar{L}_{t-1}(p)} + \frac{1}{\eta}\boxed{\mathcal{R}(p)}$$

1-strongly convex function of p that stabilizes the minimizer

Linearized historical performance of always choosing vector p

Theorem. Assuming the linearized loss function at each period

$$\bar{\ell}_t(p) = \langle p, \nabla \ell_t(p_t) \rangle$$

is L-Lipschitz with respect to some norm $||\cdot||$ and the regularizer is 1-strongly convex with respect to the same norm then

$$\text{Regret} - \text{FTRL}(T) \leq \boxed{\eta L} + \boxed{\frac{1}{\eta T}\left(\max_p \mathcal{R}(p) - \min_p \mathcal{R}(p)\right)}$$

Average stability induced by regularizer

Average loss distortion caused by regularizer

Another "*decent*" regularizer

Squared ℓ_2 norm is 1-strongly convex regularizer with respect to ℓ_2

$$\mathcal{R}(p) = \frac{1}{2}\|p\|^2 = \frac{1}{2}\sum_{i=1}^{n} p_i^2, \qquad \nabla^2\mathcal{R}(p) = \mathrm{I}$$

At each period we solve the minimization problem

$$\min_p \left\langle p, \sum_{\tau<t} \nabla\ell_\tau(p_\tau) \right\rangle + \frac{1}{2\eta}\|p\|^2$$

Another "*decent*" regularizer

Squared ℓ_2 norm is 1-strongly convex regularizer with respect to ℓ_2

$$\mathcal{R}(p) = \frac{1}{2}\|p\|^2 = \frac{1}{2}\sum_{i=1}^{n} p_i^2, \qquad \nabla^2\mathcal{R}(p) = \mathrm{I}$$

At each period we solve the minimization problem

$$\min_p \left\langle p, \sum_{\tau<t} \nabla\ell_\tau(p_\tau) \right\rangle + \frac{1}{2\eta}\|p\|^2$$

First order condition: $\sum_{\tau<t} \nabla\ell_\tau(p_\tau) + \frac{1}{\eta}p = 0 \Rightarrow p_t = -\eta \sum_{\tau<t} \nabla\ell_\tau(p_\tau)$

Update rule: $p_t = \boxed{p_{t-1} - \eta\nabla\ell_{t-1}(p_{t-1})}$

Online/Stochastic Gradient *Descent* Algorithm
(aka SGD)

Punchline: The Master Algorithms of our Times

(Linearized FTRL) $$p_t = \operatorname*{argmin}_p \bar{L}_{t-1}(p) + \frac{1}{\eta}\mathcal{R}(p)$$

$$\mathcal{R}(p) = \sum_{i=1}^{n} p_i \log(p_i)$$

$$p_t \propto p_{t-1} \exp(\eta\, \ell_{t-1})$$

$$\mathcal{R}(p) = \frac{1}{2}\|p\|^2$$

$$p_t = p_{t-1} - \eta \nabla \ell_{t-1}(p_{t-1})$$

Exponential weight updates algorithm!
(aka Hedge, Multiplicative Weight Updates, EXP, ...)

Online/Stochastic Gradient *Descent* Algorithm
(aka SGD)