ITERATIVE METHODS AND REGULARIZATION IN THE DESIGN OF FAST ALGORITHMS

An unified framework for optimization and online learning beyond Multiplicative Weight Updates

Lorenzo Orecchia, MIT Math

Talk Outline: A Tale of Two Halves

PART 1: REGULARIZATION AND ITERATIVE TECHNIQUES FOR ONLINE LEARNING

- Online Linear Optimization
- Online Linear Optimization over Simplex and Multiplicative Weight Updates (MWUs)
- A Regularization Framework to generalize MWUs: Follow the Regularized Leader

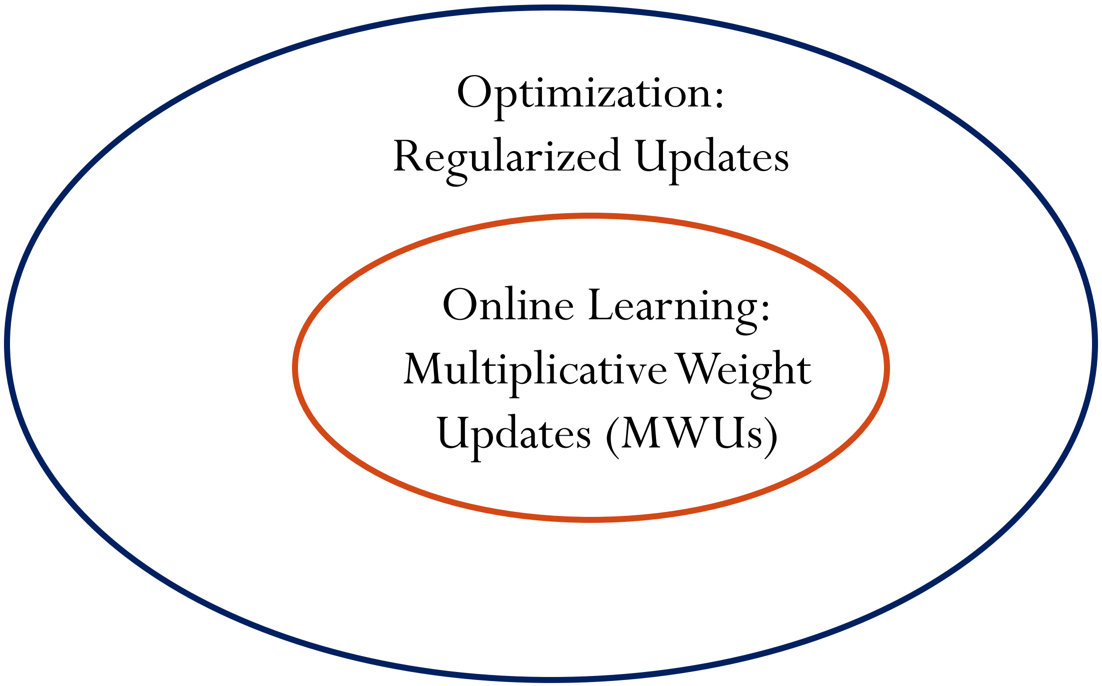
MESSAGE: REGULARIZATION IS A POWERFUL ALGORITHMIC TECHNIQUE

Talk Outline: A Tale of Two Halves

PART 1: REGULARIZATION AND ITERATIVE TECHNIQUES FOR ONLINE LEARNING

- Online Linear Optimization
- Online Linear Optimization over Simplex and Multiplicative Weight Updates (MWUs)
- A Regularization Framework to generalize MWUs: Follow the Regularized Leader

MESSAGE: REGULARIZATION IS A POWERFUL ALGORITHMIC TECHNIQUE



Talk Outline: A Tale of Two Halves

PART 1: REGULARIZATION AND ITERATIVE TECHNIQUES FOR ONLINE LEARNING

- Online Linear Optimization
- Online Linear Optimization over Simplex and Multiplicative Weight Updates (MWUs)
- A Regularization Framework to generalize MWUs: Follow the Regularized Leader

MESSAGE: REGULARIZATION IS A POWERFUL ALGORITHMIC TECHNIQUE

PART 2: NON-SMOOTH OPTIMIZATION AND FAST ALGORITHMS FOR MAXFLOW

- Non-smooth vs Smooth Convex Optimization
- Non-smooth Convex Optimization reduces to Online Linear Optimization
- Application: Understanding Undirected Maxflow algorithms based on MWUs

MESSAGE: FASTEST ALGORITHMS REQUIRE PRIMAL-DUAL APPROACH

TOC Applications of MWUs

- Fast Algorithms for solving specific LPs and SDPs:
  - Maximum Flow problems [PST], [GK], [F], [CKMST]
  - Covering-packing problems [PST]
  - Oblivious routing [R], [M]

- Fast Approximation Algorithms based on LP and SDP relaxations:
  - Maxcut [AK]
  - Graph Partitioning Problems [AK], [S], [OSV]

- Proof Technique
  - Hardcore Lemma [BHK]
  - QIP = PSPACE [W]
  - Derandomization [Y]

… and more

Machine Learning meets Optimization meets TCS

These techniques have been rediscovered multiple times in different fields:

Machine Learning, Convex Optimization, TCS

Three surveys emphasizing the different viewpoints and literatures:

1) ML: **Prediction, Learning and Games** by Gabor and Lugosi
2) Optimization: **Lectures in Modern Convex Optimization** by Ben Tal and Nemirowski
3) TCS: **The Multiplicative Weights Update Method: a Meta Algorithm and Applications** by Arora, Hazan and Kale

REGULARIZATION 101

What is Regularization?

Regularization is a fundamental technique in optimization

OPTIMIZATION PROBLEM → WELL-BEHAVED OPTIMIZATION PROBLEM

- Stable optimum
- Unique optimal solution
- Smoothness conditions

. . .

What is Regularization?

Regularization is a fundamental technique in optimization

OPTIMIZATION PROBLEM → WELL-BEHAVED OPTIMIZATION PROBLEM

$$\min_{x \in H} L(x)$$

$$\min_{x \in H} L(x) + \lambda \cdot F(x)$$

Parameter $\lambda > 0$

Regularizer F

Benefits of Regularization in Learning and Statistics:

- Prevents overfitting
- Increases stability
- Decreases sensitivity to random noise

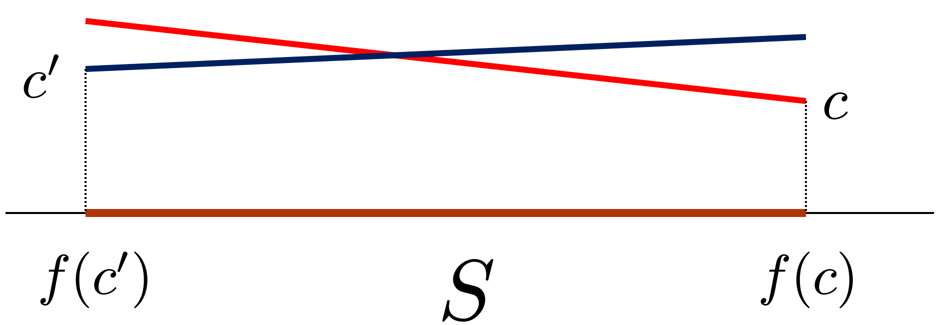
Example: Regularization Helps Stability

Consider a convex set $S \subset R^n$ and a linear optimization problem:

$$f(c) = \arg\min_{x \in S} c^T x$$

The optimal solution $f(c)$ may be very unstable under perturbation of c :

$$\|c' - c\| \cdot \delta \quad \text{and} \quad \|f(c') - f(c)\| >> \delta$$



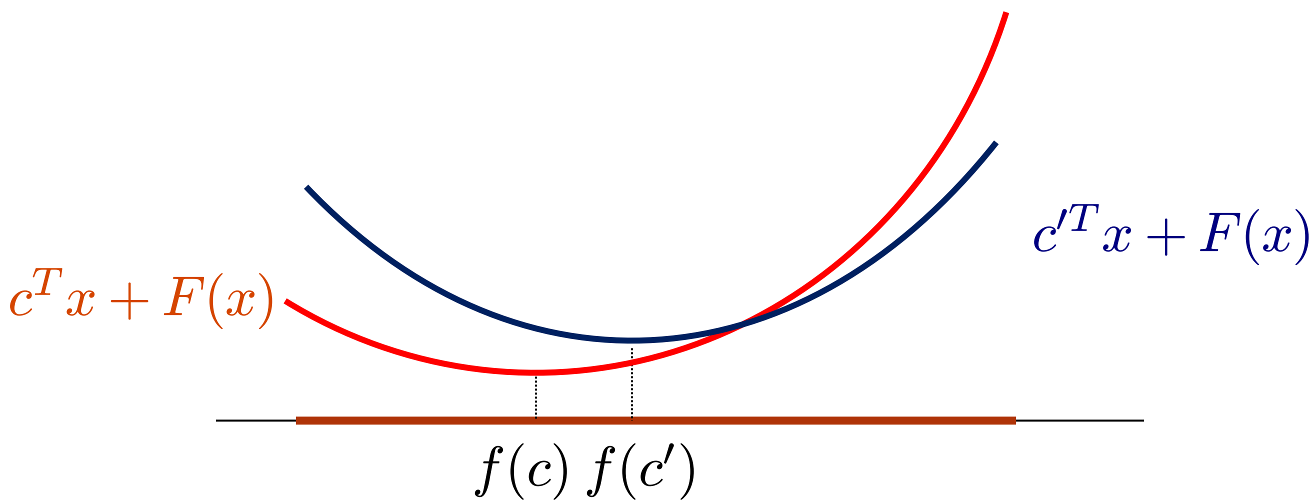
Example: Regularization Helps Stability

Consider a convex set $S \subset R^n$ and a **regularized** linear optimization problem

$$f(c) = \arg\min_{x \in S} c^T x + F(x)$$

where F is σ-strongly convex.

Then: $\|c' - c\| \cdot \delta$ implies $\|f(c') - f(c)\|\|\| \cdot \frac{\delta}{\sigma}$



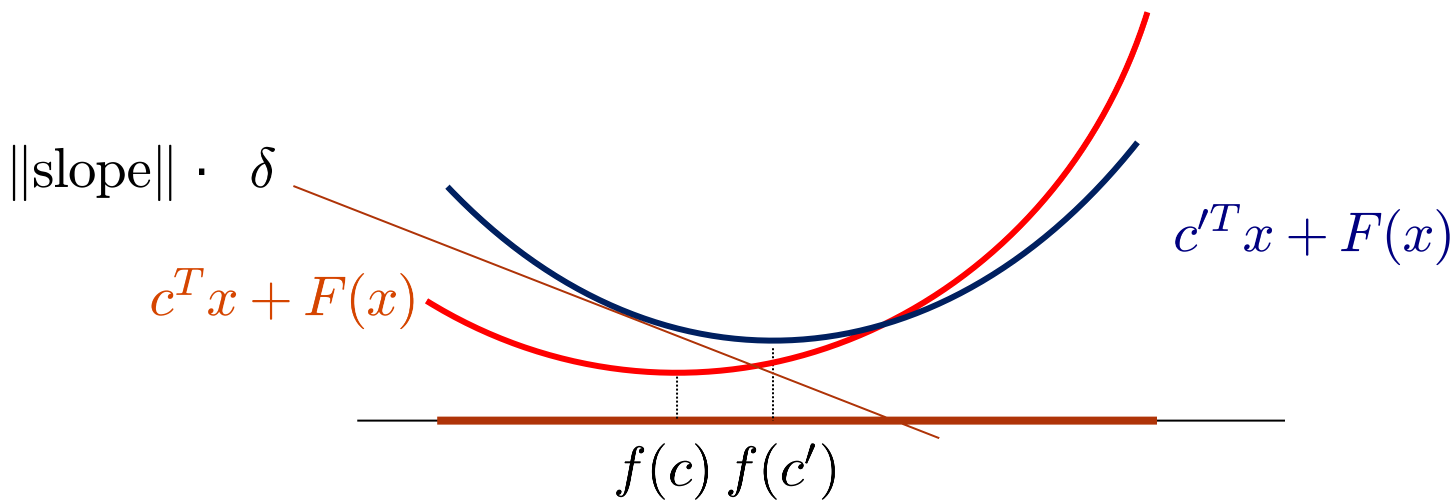
Example: Regularization Helps Stability

Consider a convex set $S \subset R^n$ and a **regularized** linear optimization problem

$$f(c) = \arg\min_{x \in S} c^T x + F(x)$$

where F is σ-strongly convex.

Then: $\|c' - c\| \cdot \delta$ implies $\|f(c') - f(c)\|\|\| \cdot \frac{\delta}{\sigma}$



ONLINE LINEAR OPTIMIZATION
AND
MULTIPLICATIVE WEIGHT UPDATES

Online Linear Minimization

SETUP: Convex set $X \subseteq \mathbb{R}^n$, generic norm, repeated game over T rounds.

At round t,

ALGORITHM

$$x^{(t)} \in X$$

Current solution

ADVERSARY

Online Linear Minimization

SETUP: Convex set $X \subseteq \mathrm{R}^n$, generic norm, repeated game over T rounds.

At round t,

ALGORITHM

$x^{(t)} \in X$

Current solution

ADVERSARY

$\ell^{(t)} \in \mathrm{R}^n, \|\nabla \ell^{(t)}\|_* \cdot \; \rho$

Current linear objective

Loss vector

Online Linear Minimization

SETUP: Convex set $X \subseteq \mathrm{R}^n$, generic norm, repeated game over T rounds.

At round t,

ALGORITHM **ADVERSARY**

$$x^{(t)} \in X \longrightarrow \ell^{(t)} \in \mathrm{R}^n, \|\nabla \ell^{(t)}\|_* \cdot \ \rho$$

Current solution

Current linear objective

Loss vector

$$\ell^{(t)^T} x^{(t)}$$

Algorithm's loss

Online Linear Minimization

SETUP: Convex set $X \subseteq \mathrm{R}^n$, generic norm, repeated game over T rounds.

At round t,

ALGORITHM

$x^{(t)} \in X$

ADVERSARY

$\ell^{(t)} \in \mathrm{R}^n, \|\nabla \ell^{(t)}\|_* \cdot \quad \rho$

$x^{(t+1)} \in X$

Updated solution

Online Linear Minimization

SETUP: Convex set $X \subseteq \mathrm{R}^n$, generic norm, repeated game over T rounds.

At round t,

ALGORITHM

$$x^{(t)} \in X$$

ADVERSARY

$$\ell^{(t)} \in \mathrm{R}^n, \|\nabla \ell^{(t)}\|_* \cdot \ \rho$$

$$x^{(t+1)} \in X$$

Updated solution

$$\ell^{(t+1)} \in \mathrm{R}^n, \|\nabla \ell^{(t)}\|_* \cdot \ \rho$$

New Loss Vector

Online Linear Minimization

SETUP: Convex set $X \subseteq \mathrm{R}^n$, generic norm, repeated game over T rounds.

At round t,

ALGORITHM | **ADVERSARY**

$x^{(t)} \in X \longrightarrow \ell^{(t)} \in \mathrm{R}^n, \|\nabla \ell^{(t)}\|_* \cdot \;\; \rho$

$x^{(t+1)} \in X \longrightarrow \ell^{(t+1)} \in \mathrm{R}^n, \|\nabla \ell^{(t)}\|_* \cdot \;\; \rho$

GOAL: update $x^{(t)}$ to **minimize regret**

$$\frac{1}{T} \cdot \sum_{t=1}^{T} {\ell^{(t)}}^T x^T - \min_{x \in X} \frac{1}{T} \cdot \sum_{t=1}^{T} {\ell_i^{(t)}}^T x$$

Average Algorithm's Loss $\hat{L}$ | A Posteriori Optimum L^*

Simplex Case: Learning with Experts

SETUP: Simplex $X \subseteq \mathrm{R}^n$ under ℓ_1 norm. At round t,

ALGORITHM

$p^{(t)}$

distribution over experts

ADVERSARY

Simplex Case: Learning with Experts

SETUP: Simplex $X \subseteq \mathbb{R}^n$ under ℓ_1 norm. At round t,

ALGORITHM

$p^{(t)}$ →

distribution over dimensions
i.e. experts

ADVERSARY

$\|\ell^{(t)}\|_\infty \cdot \rho$

Experts' losses

Simplex Case: Learning with Experts

SETUP: Simplex $X \subseteq \mathbb{R}^n$ under ℓ_1 norm. At round t,

ALGORITHM

$p^{(t)}$

distribution over experts

ADVERSARY

$\|\ell^{(t)}\|_\infty \cdot \rho$

Experts' losses

$$E_{i \leftarrow p^{(t)}}\left[\ell_i^{(t)}\right] = {p^{(t)}}^T \ell^{(t)}$$

Algorithm's loss

Simplex Case: Learning with Experts

SETUP: Simplex $X \subseteq \mathrm{R}^n$ under ℓ_1 norm. At round t,

ALGORITHM

$$p^{(t)}$$

distribution over experts

ADVERSARY

$$\|\ell^{(t)}\|_\infty \cdot \rho$$

Experts' losses

$$p^{(t+1)}$$

Update distribution

Simplex Case: Multiplicative Weight Updates

ALGORITHM $p^{(t)}$ → **ADVERSARY** $\ell^{(t)}$

Weights: $w_i^{(t+1)} = (1-\epsilon)^{\ell_i^{(t)}} w_i^{(t)}$, $w_1 = \vec{1}$

Simplex Case: Multiplicative Weight Updates

ALGORITHM → **ADVERSARY**

$p^{(t)}$ → $\ell^{(t)}$

Weights: $$w_i^{(t+1)} = (1-\epsilon)^{\ell_i^{(t)}} w_i^{(t)} \quad , \quad w_1 = \vec{1}$$

Distribution: $$p_i^{(t+1)} = \frac{w_i^{(t)}}{\sum_{j=1}^{n} w_j^{(t)}}$$

Simplex Case: Multiplicative Weight Updates

ALGORITHM **ADVERSARY**

$$p^{(t)} \longrightarrow \ell^{(t)}$$

Weights: $$w_i^{(t+1)} = \boxed{(1-\epsilon)^{\ell_i^{(t)}} w_i^{(t)}}\ , \quad w_1 = \vec{1}$$

Distribution: $$p_i^{(t+1)} = \frac{w_i^{(t)}}{\sum_{j=1}^{n} w_j^{(t)}}$$

MULTIPLICATIVE WEIGHT UPDATE

Simplex Case: Multiplicative Weight Updates

ALGORITHM $\quad\longrightarrow\quad$ **ADVERSARY**

$p^{(t)} \longrightarrow \ell^{(t)}$

Weights: $w_i^{(t+1)} = (1-\epsilon)^{\ell_i^{(t)}} w_i^{(t)}, \quad w_1 = \vec{1}$

Distribution: $p_i^{(t+1)} = \frac{w_i^{(t)}}{\sum_{j=1}^n w_j^{(t)}}$

CONSERVATIVE ... **AGGRESSIVE**

0 $\longrightarrow$ 1

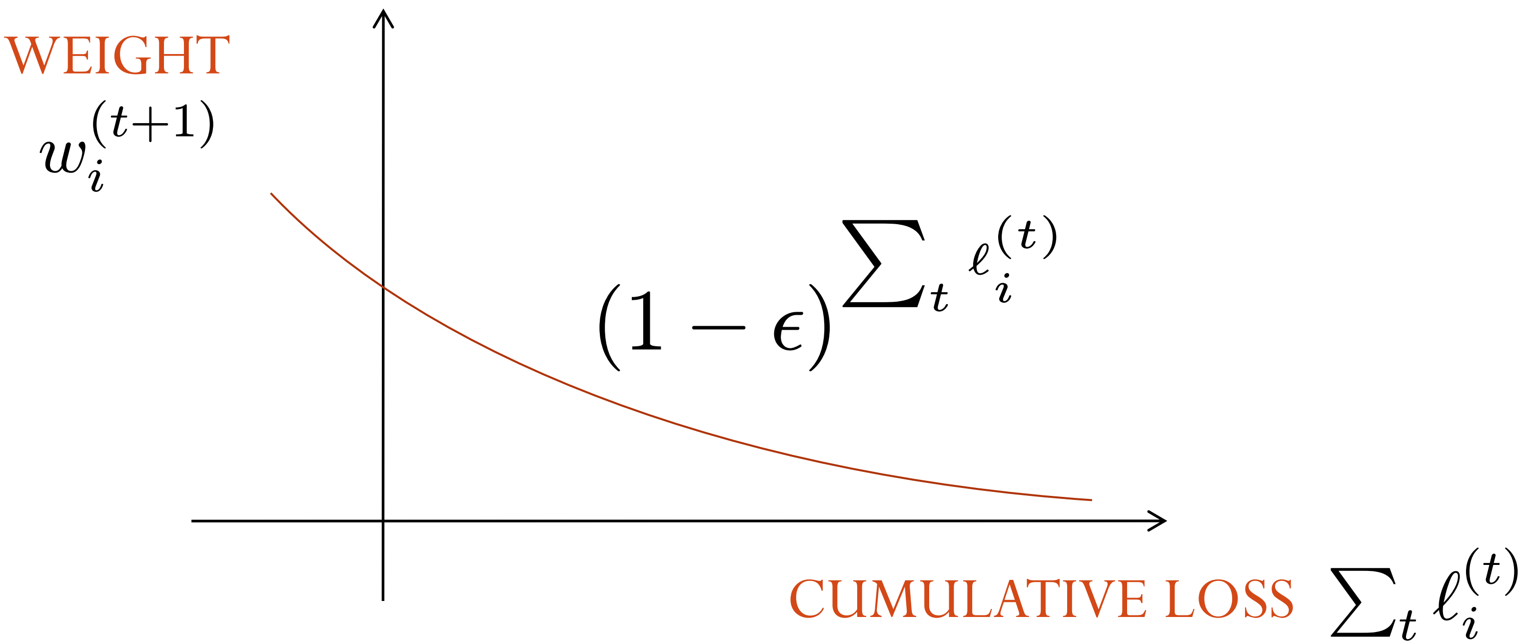
$$\epsilon \in (0,1)$$

MWUs: Unraveling the Update

ALGORITHM $\quad\quad\quad$ **ADVERSARY**

$p^{(t)} \longrightarrow \ell^{(t)}$

Update: $\quad p_i^{(t+1)} \propto w_i^{(t+1)} = (1-\epsilon)^{\ell_i^{(t)}} \cdot w_i^{(t)}$



MWUs: Regret Bound

ALGORITHM $\quad\quad\quad\quad\quad\quad\quad\quad$ **ADVERSARY**

$$p^{(t)} \longrightarrow \ell^{(t)}$$

Update: $p_i^{(t+1)} \propto w_i^{(t+1)} = (1-\epsilon)^{\ell_i^{(t)}} \cdot w_i^{(t)}$

For $\epsilon < \frac{1}{2}$ and $\|\ell^{(t)}\|_\infty \cdot \rho$

$$\boxed{\hat{L} - L^\star \cdot \frac{\rho \log n}{\epsilon T} + \rho\epsilon}$$

MWUs: Regret Bound

ALGORITHM **ADVERSARY**

$p^{(t)} \longrightarrow \ell^{(t)}$

Update: $p_i^{(t+1)} \propto w_i^{(t+1)} = (1-\epsilon)^{\ell_i^{(t)}} \cdot w_i^{(t)}$

For $\epsilon < \frac{1}{2}$ and $\|\ell^{(t)}\|_\infty \cdot \ \rho$

$$\boxed{\hat{L} - L^\star \cdot \ \frac{\rho \log n}{\epsilon T} + \rho\epsilon}$$

Algorithm's Regret

Start-up Penalty

Penalty for being greedy

ONLINE LINEAR OPTIMIZATION BEYOND MWUs

A REGULARIZATION FRAMEWORK

MWUs: Proof Sketch of Regret Bound

Update: $p_i^{(t+1)} \propto w_i^{(t+1)} = (1-\epsilon)^{\sum_{s=1}^{t} \ell_i^{(s)}}$

- Proof is potential function argument

$$\Phi^{(t+1)} = \log_{1-\epsilon} \sum_{i=1}^{n} w_i^{(t+1)}$$

MWUs: Proof Sketch of Regret Bound

Update: $p_i^{(t+1)} \propto w_i^{(t+1)} = (1-\epsilon)^{\sum_{s=1}^{t} \ell_i^{(s)}}$

- Proof is potential function argument

$$\Phi^{(t+1)} = \log_{1-\epsilon} \sum_{i=1}^{n} w_i^{(t+1)}$$

- Potential function bounds loss of best expert

$$\Phi^{(t+1)} \cdot \log_{1-\epsilon} \min_{i=1}^{n} w_i^{(t+1)} = \min_{i=1}^{n} \left(\sum_{s=1}^{t} \ell_i^{(s)} \right)$$

MWUs: Proof Sketch of Regret Bound

Update: $$p_i^{(t+1)} \propto w_i^{(t+1)} = (1-\epsilon)^{\sum_{s=1}^{t} \ell_i^{(s)}}$$

- Proof is <u>potential function argument</u>

$$\Phi^{(t+1)} = \log_{1-\epsilon} \sum_{i=1}^{n} w_i^{(t+1)}$$

- Potential function <u>bounds loss of best expert</u>

$$\Phi^{(t+1)} \cdot \log_{1-\epsilon} \min_{i=1}^{n} w_i^{(t+1)} = \min_{i=1}^{n} \left(\sum_{s=1}^{t} \ell_i^{(s)} \right)$$

- Potential function is <u>related to algorithm's performance</u>

$$\Phi^{(t+1)} - \Phi^{(t)} \geq \left({\ell^{(t)}}^T p^{(t)} \right) - \epsilon$$

MWUs: Proof Sketch of Regret Bound

Update: $p_i^{(t+1)} \propto w_i^{(t+1)} = (1-\epsilon)^{\sum_{s=1}^{t} \ell_i^{(s)}}$

- Proof is potential function argument

$$\Phi^{(t+1)} = \log_{1-\epsilon} \sum_{i=1}^{n} w_i^{(t+1)}$$

- Potential function bounds loss of best expert

$$\Phi^{(t+1)} \cdot \log_{1-\epsilon} \min_{i=1}^{n} w_i^{(t+1)} = \min_{i=1}^{n} \left(\sum_{s=1}^{t} \ell_i^{(s)} \right)$$

- Potential function is related to algorithm's performance

$$\Phi^{(t+1)} - \Phi^{(t)} \geq \left({\ell^{(t)}}^T p^{(t)} \right) - \epsilon$$

DOES THIS PROOF TECHNIQUE GENERALIZE TO BEYOND SIMPLEX CASE?

Designing a Regularized Update

GOAL: Design an update and its potential function analysis

QUESTION: Choice of potential function?

DESIDERATA: 1) lower bounds best expert's loss

2) tracks algorithm's performance

Designing a Regularized Update

QUESTION: Choice of potential function?

DESIDERATA: 1) lower bounds best expert's loss

2) tracks algorithm's performance

Attempt 1 – FOLLOW THE LEADER: Cumulative loss $L^{(t)} = \sum_{s=1}^{t} \ell^{(s)}$

$$x^{(t+1)} = \arg\min_{x \in X} x^T L^{(t)}$$

Pick best current solution

$$\Phi^{(t+1)} = \min_{x \in X} x^T L^{(t)}$$

Potential is current best loss

Designing a Regularized Update

QUESTION: Choice of potential function?

DESIDERATA: 1) lower bounds best expert's loss ✓

2) tracks algorithm's performance

Attempt 1 – FOLLOW THE LEADER: Cumulative loss $L^{(t)} = \sum_{s=1}^{t} \ell^{(s)}$

$$x^{(t+1)} = \arg\min_{x \in X} x^T L^{(t)}$$

Pick best current solution

$$\Phi^{(t+1)} = \min_{x \in X} x^T L^{(t)}$$

Potential is current best loss

Designing a Regularized Update

QUESTION: Choice of potential function?

DESIDERATA: 1) lower bounds best expert's loss ✓

2) tracks algorithm's performance ✗

Attempt 1 – FOLLOW THE LEADER: Cumulative loss $L^{(t)} = \sum_{s=1}^{t} \ell^{(s)}$

$$x^{(t+1)} = \arg\min_{x \in X} x^T L^{(t)}$$

Pick best current solution

$$\Phi^{(t+1)} = \min_{x \in X} x^T L^{(t)}$$

Potential is current best loss

✗ Fails if best expert changes moves drastically

Designing a Regularized Update

QUESTION: Choice of potential function?

DESIDERATA: 1) lower bounds best expert's loss ✓

2) tracks algorithm's performance ✗

Attempt 1 – FOLLOW THE LEADER: Cumulative loss $L^{(t)} = \sum_{s=1}^{t} \ell^{(s)}$

$$x^{(t+1)} = \arg\min_{x \in X} x^T L^{(t)}$$

How to make update more stable?

$$\Phi^{(t+1)} = \min_{x \in X} x^T L^{(t)}$$

Regularized Update: Definition

QUESTION: Choice of potential function?

DESIDERATA: 1) lower bounds best expert's loss

2) tracks algorithm's performance

Attempt 2 – FOLLOW THE REGULARIZED LEADER:

$$x^{(t+1)} = \arg\min_{x \in X} x^T L^{(t)} + \eta \cdot \mathbf{F}(\mathbf{x})$$

$$\Phi^{(t+1)} = \min_{x \in X} x^T L^{(t)} + \eta \cdot \mathbf{F}(\mathbf{x})$$

Properties of **Regularizer F(x):**

1. Convex, differentiable
2. σ-strong convex w.r.t. norm

Parameter $\eta \geq 0$, TBD

Regularized Update: Definition

QUESTION: Choice of potential function?

DESIDERATA: 1) lower bounds best expert's loss

2) tracks algorithm's performance

Attempt 2 – FOLLOW THE REGULARIZED LEADER:

$$x^{(t+1)} = \arg\min_{x \in X} x^T L^{(t)} + \eta \cdot \mathbf{F}(\mathbf{x})$$

$$\Phi^{(t+1)} = \min_{x \in X} x^T L^{(t)} + \eta \cdot \mathbf{F}(\mathbf{x})$$

Properties of **Regularizer F(x):**

1. Convex, differentiable
2. σ-strong convex w.r.t. norm

Parameter $\eta \geq 0$, TBD

These properties are actually sufficient to get a regret bound

Regularized Update: Analysis

<u>QUESTION:</u> Choice of <u>potential function</u>?

<u>DESIDERATA</u>: 1) lower bounds best expert's loss ✓

2) tracks algorithm's performance

<u>Attempt 2 – FOLLOW THE REGULARIZED LEADER:</u>

$$x^{(t+1)} = \arg\min_{x \in X} x^T L^{(t)} + \eta \cdot \mathbf{F}(\mathbf{x})$$

$$\Phi^{(t+1)} = \min_{x \in X} x^T L^{(t)} + \eta \cdot \mathbf{F}(\mathbf{x})$$

Properties of **Regularizer F(x):**

1. Convex, differentiable
2. σ-strong convex w.r.t. norm

Parameter $\eta \geq 0$, TBD

✓ $$\Phi^{(t+1)} \cdot \quad \min_{x \in X} {L^{(t)}}^T x + \eta \cdot \max_{x \in X} F(x)$$

Regularized Update: Analysis

QUESTION: Choice of potential function?

DESIDERATA: 1) lower bounds best expert's loss ✓

2) tracks algorithm's performance

Attempt 2 – FOLLOW THE REGULARIZED LEADER:

$$x^{(t+1)} = \arg\min_{x \in X} x^T L^{(t)} + \eta \cdot \mathbf{F}(\mathbf{x})$$

$$\Phi^{(t+1)} = \min_{x \in X} x^T L^{(t)} + \eta \cdot \mathbf{F}(\mathbf{x})$$

Properties of **Regularizer F(x):**

1. Convex, differentiable
2. σ-strong convex w.r.t. norm

Parameter $\eta \geq 0$, TBD

✓ $$\Phi^{(t+1)} \cdot \min_{x \in X} {L^{(t)}}^T x + \boxed{\eta \cdot \max_{x \in X} F(x)}$$

Regularization error

Regularized Update: Analysis

QUESTION: Choice of potential function?

DESIDERATA: 1) lower bounds best expert's loss ✓

2) tracks algorithm's performance ?

Attempt 2 – FOLLOW THE REGULARIZED LEADER:

$$x^{(t+1)} = \arg\min_{x \in X} x^T L^{(t)} + \eta \cdot \mathbf{F}(\mathbf{x})$$

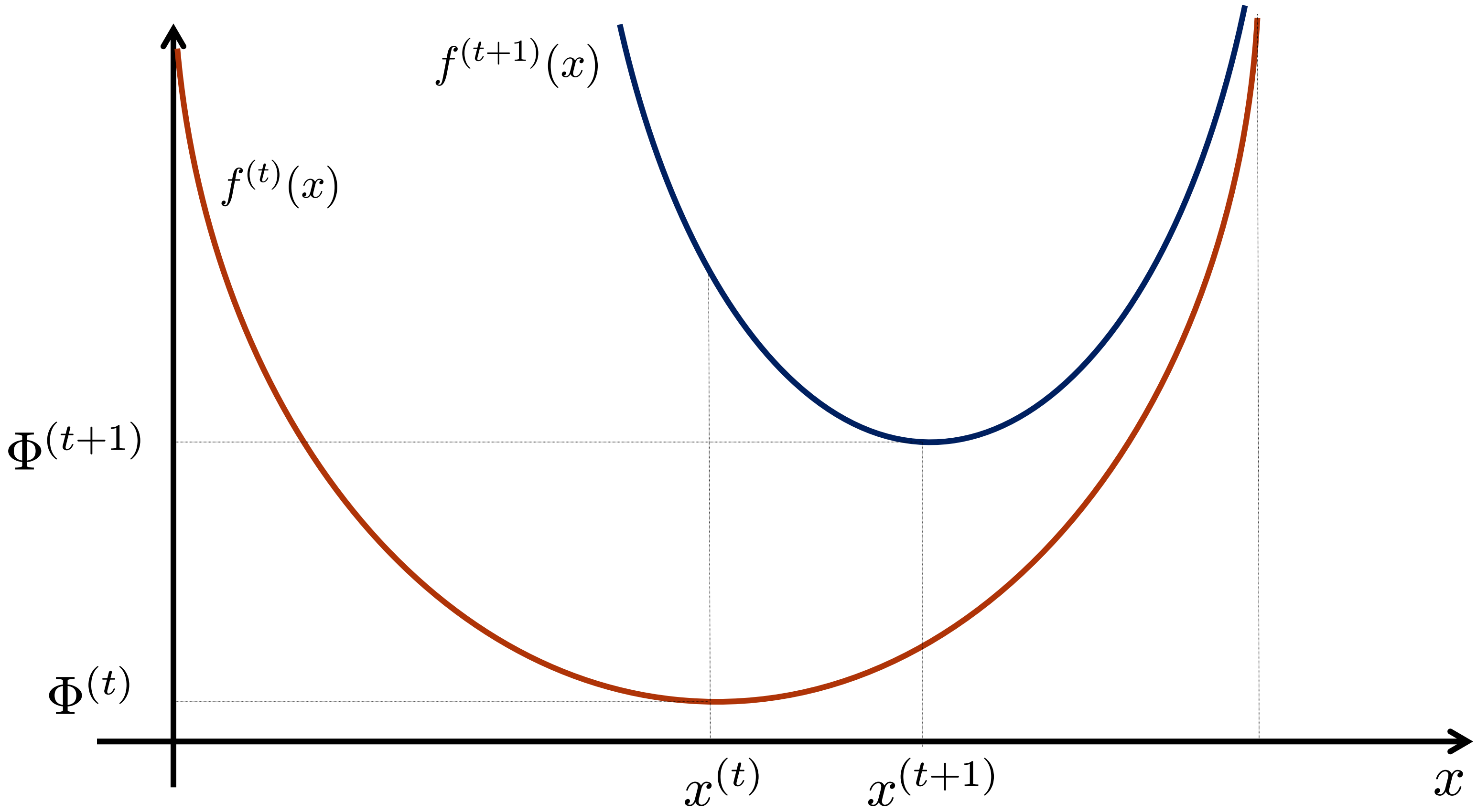
$$\Phi^{(t+1)} = \underbrace{\min_{x \in X} x^T L^{(t)} + \eta \cdot \mathbf{F}(\mathbf{x})}_{f^{(t+1)}(x)}$$

Properties of **Regularizer F(x):**

1. Convex, differentiable
2. σ-strong convex w.r.t. norm

Parameter $\eta \geq 0$, TBD

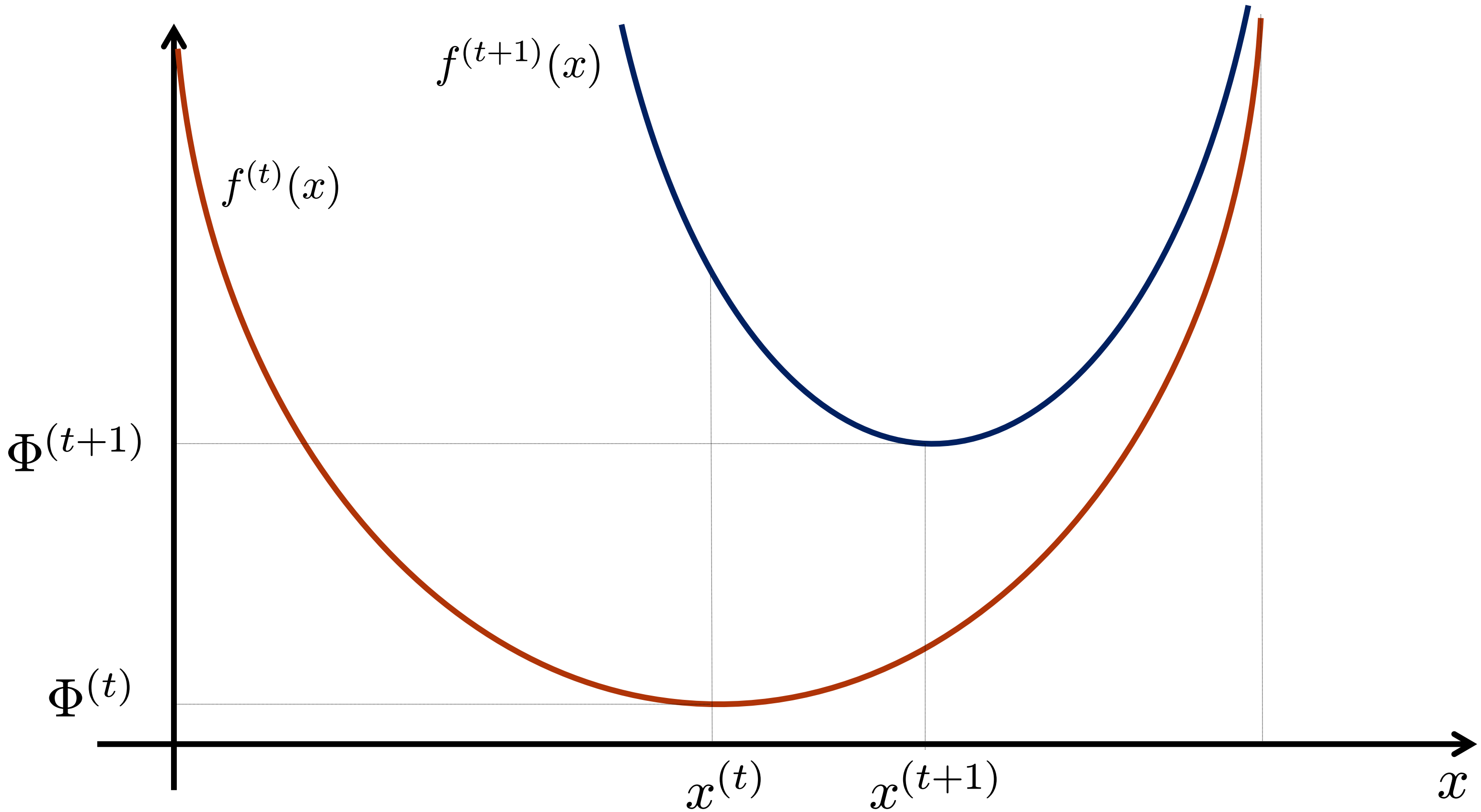
Tracking the Algorithm: Proof by Picture



Define:

$$f^{(t+1)}(x) = x^T L^{(t)} + \eta \cdot F(x)$$

Tracking the Algorithm: Proof by Picture



Define:

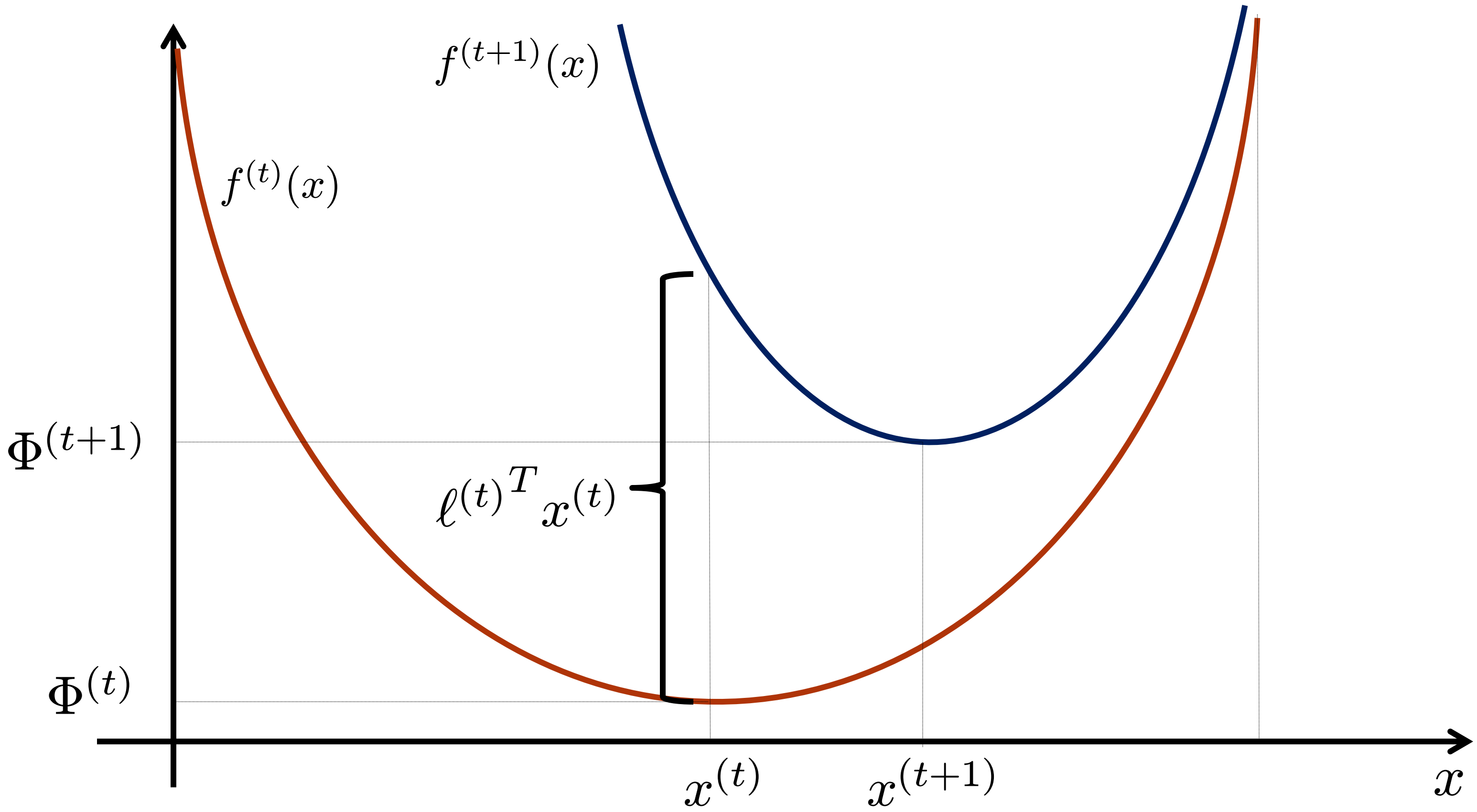
$$f^{(t+1)}(x) = x^T L^{(t)} + \eta \cdot F(x)$$

Notice:

$$f^{(t+1)}(x) - f^{(t)}(x) = {\ell^{(t)}}^T x$$

Latest loss vector

Tracking the Algorithm: Proof by Picture



Define:

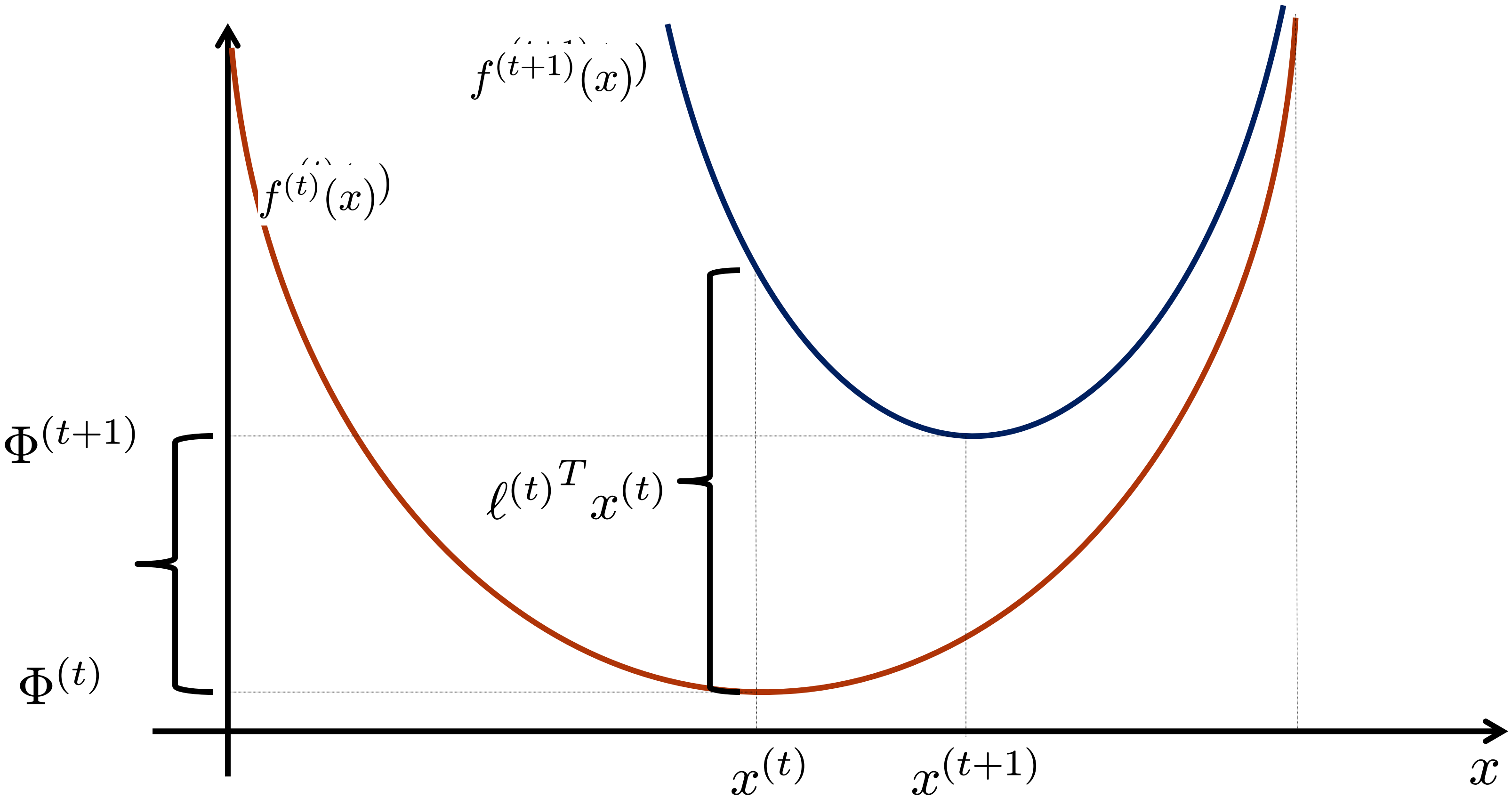
$$f^{(t+1)}(x) = {L^{(t)}}^T x + \eta \cdot F(x)$$

Notice:

$$f^{(t+1)}(x) - f^{(t)}(x) = {\ell^{(t)}}^T x$$

Latest loss vector

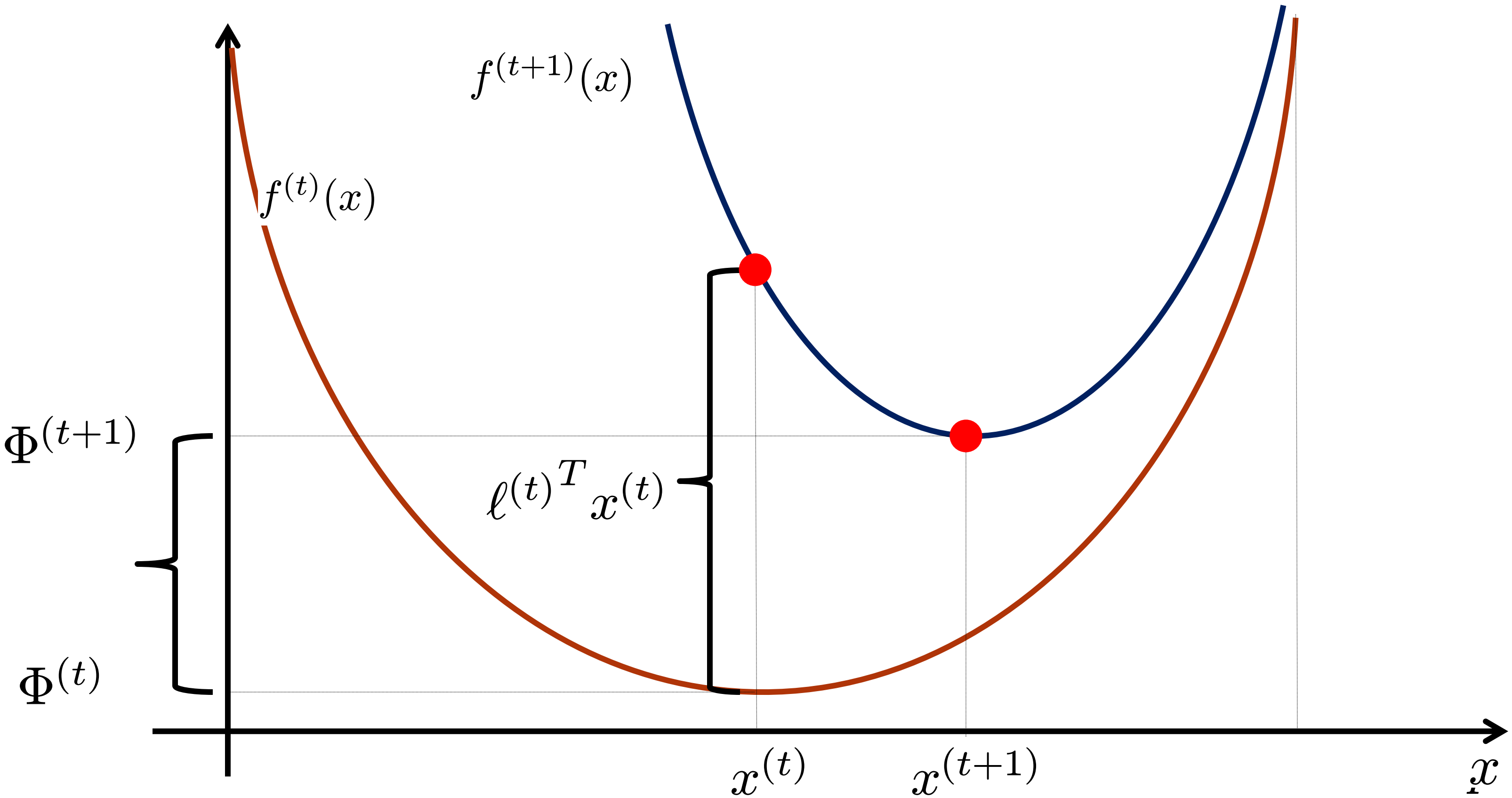
Tracking the Algorithm: Proof by Picture



Compare:

$${\ell^{(t)}}^T x^{(t)} \quad \text{and} \quad \Phi^{(t+1)} - \Phi^{(t)}$$

Tracking the Algorithm: Proof by Picture

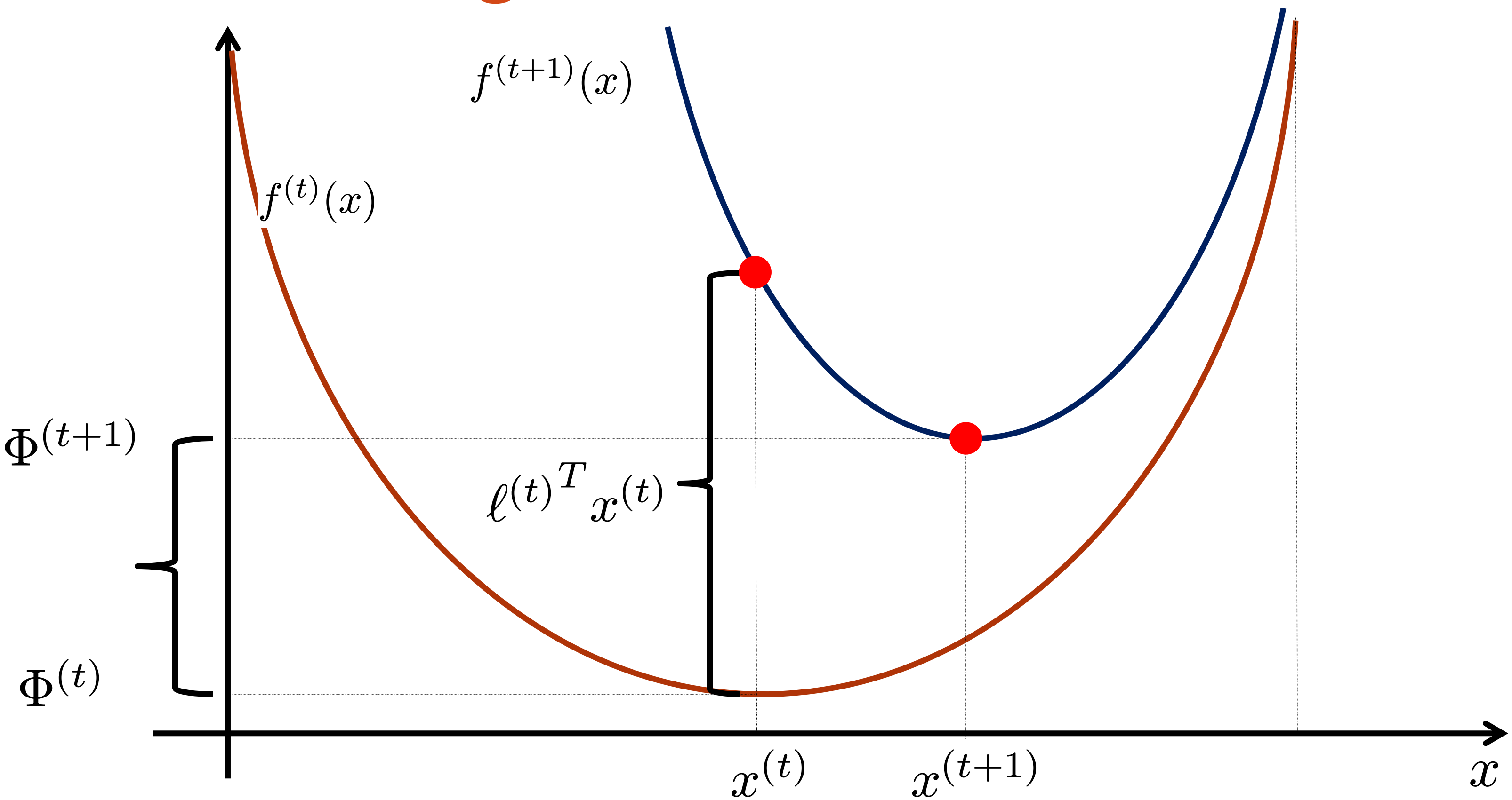


$$\Phi^{(t+1)} - \Phi^{(t)} = f^{(t+1)}(x^{(t+1)}) - f^{(t+1)}(x^{(t)}) + {\ell^{(t)}}^T x^{(t)}$$

Want:

$$f^{(t+1)}(x^{(t)}) \approx f^{(t+1)}(x^{(t+1)})$$

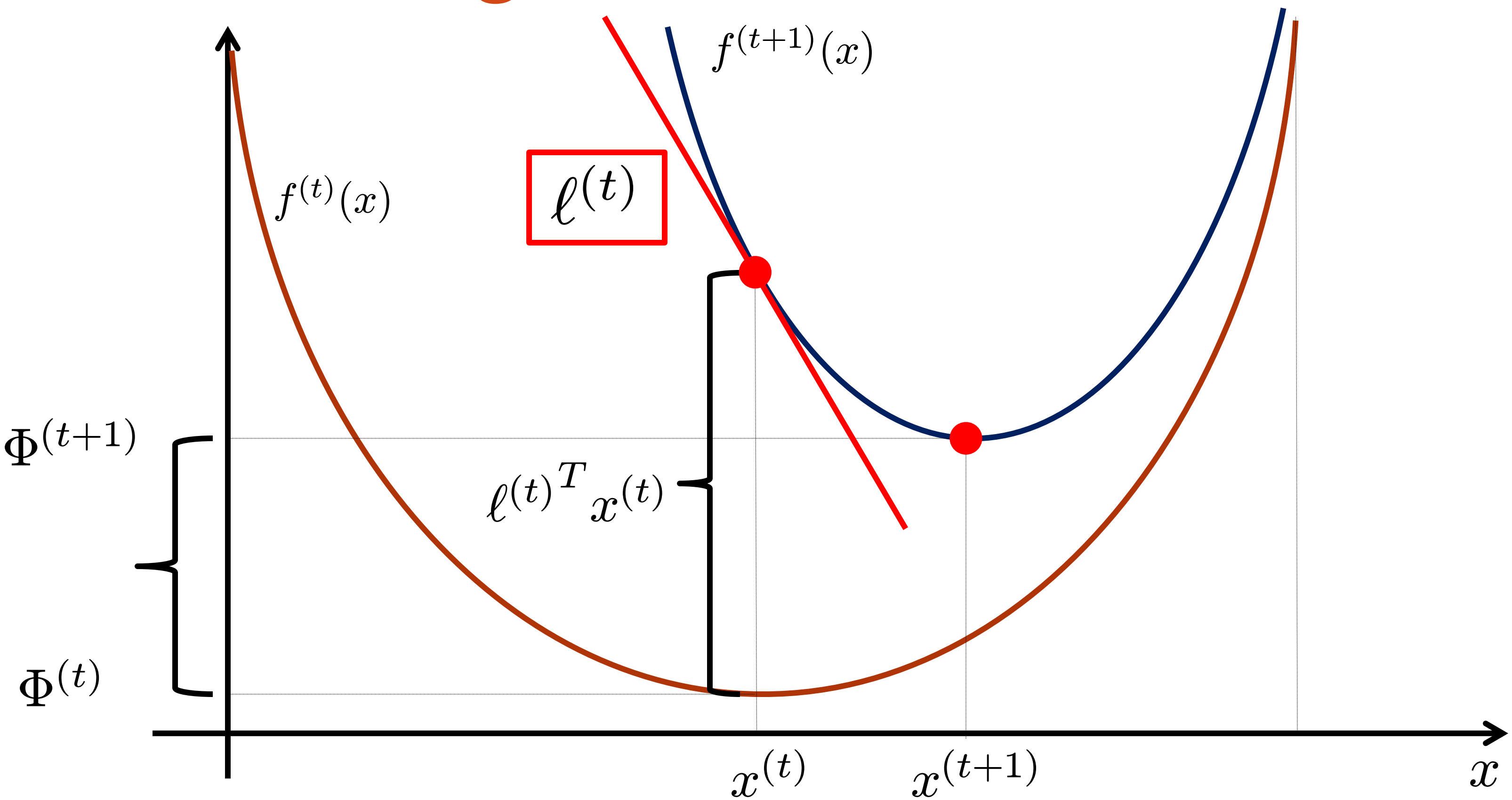
Regularization in Action



$$f^{(t+1)}(x) = {L^{(t)}}^T x + \eta \cdot F(x)$$

REGULARIZATION → $f^{(t)}$ is $(\eta \cdot \sigma)$-strongly-convex

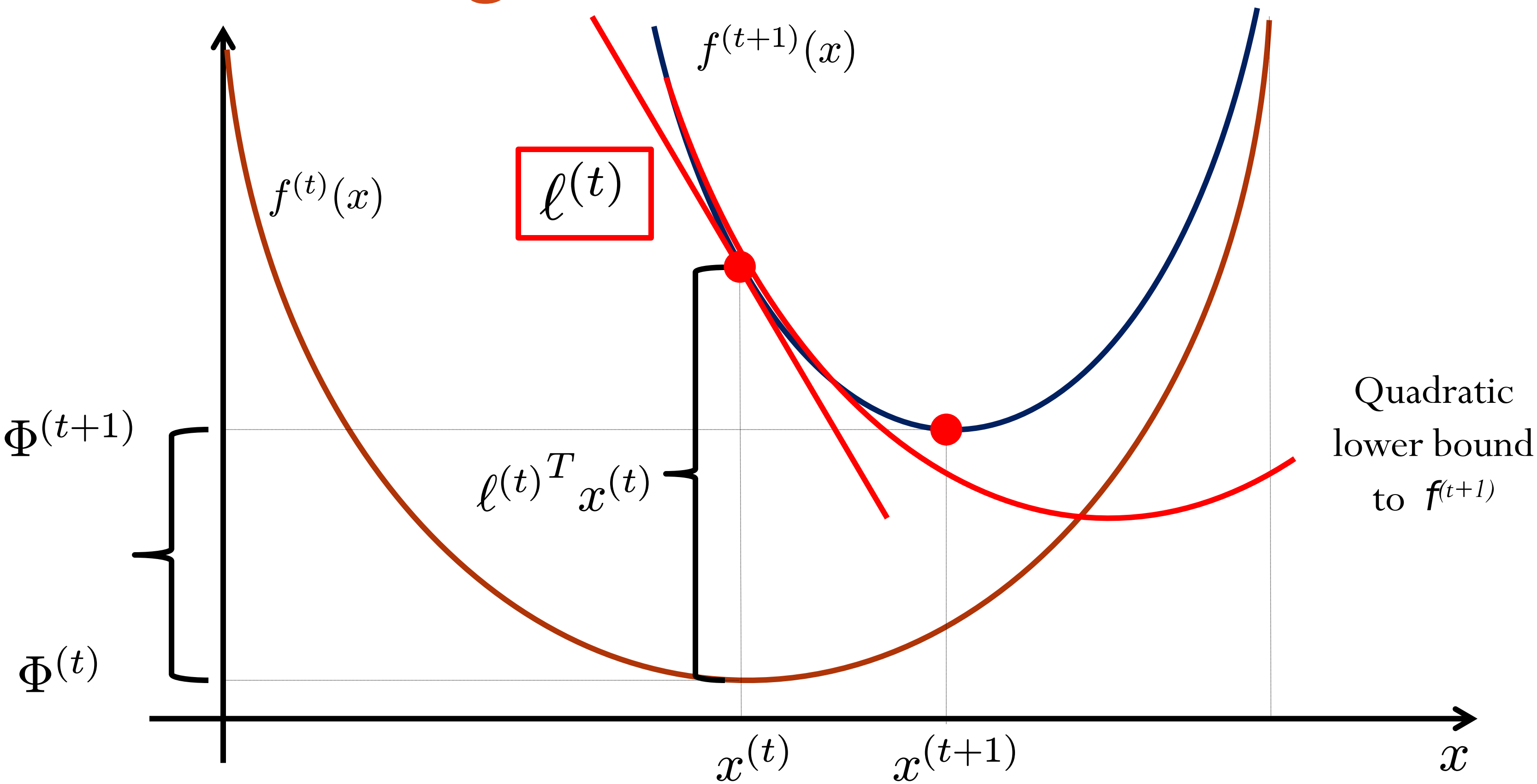
Regularization in Action



$$f^{(t+1)}(x) = {L^{(t)}}^T x + \eta \cdot F(x) \xrightarrow{\text{REGULARIZATION}} f^{(t)} \text{ is } (\eta \cdot \sigma\ )\text{-strongly-convex}$$

$$\|f^{(t+1)} - f^{(t)}\|_* = \|\ell^{(t)}\|_* \xrightarrow{\text{STABILITY}} ||x^{(t+1)} - x^{(t)}|| \cdot \frac{||\ell^{(t)}||_*}{\eta \cdot \sigma}$$

Regularization in Action



$$f^{(t+1)}(x) = {L^{(t)}}^T x + \eta \cdot F(x) \xrightarrow{\text{REGULARIZATION}} f^{(t)} \text{ is } (\eta \cdot \sigma\ )\text{-strongly-convex}$$

$$\|f^{(t+1)} - f^{(t)}\| = \|\ell^{(t)}\| \xrightarrow{\text{STABILITY}} ||x^{(t+1)} - x^{(t)}||_* \cdot \frac{||\ell^{(t)}||}{\eta \cdot \sigma}$$

Analysis: Progress in One Iteration

$$\Phi^{(t+1)} - \Phi^{(t)} = f^{(t+1)}(x^{(t+1)}) - f^{(t+1)}(x^{(t)}) + {\ell^{(t)}}^T x^{(t)}$$

$$\nabla f^{(t+1)}(x^{(t)}) = \ell^{(t)}$$

$$||x^{(t)} - x^{(t)}|| \cdot \quad \frac{||\ell^{(t)}||_*}{\eta \cdot \sigma}$$

$f^{(t+1)}$ is $(\eta \cdot \sigma)$-strongly-convex

$$f^{(t+1)}(x^{(t+1)}) - f^{(t+1)}(x^{(t)}) \geq {\ell^{(t)}}^T (x^{(t+1)} - x^{(t)}) + \frac{||\ell^{(t)}||_*^2}{2\eta \cdot \sigma}$$

Analysis: Progress in One Iteration

$$\Phi^{(t+1)} - \Phi^{(t)} = f^{(t+1)}(x^{(t+1)}) - f^{(t+1)}(x^{(t)}) + {\ell^{(t)}}^T x^{(t)}$$

$$\nabla f^{(t+1)}(x^{(t)}) = \ell^{(t)}$$

$$||x^{(t)} - x^{(t)}|| \cdot \quad \frac{||\ell^{(t)}||_*}{\eta \cdot \sigma}$$

$f^{(t+1)}$ is $(\eta \cdot \sigma)$-strongly-convex

$$f^{(t+1)}(x^{(t+1)}) - f^{(t+1)}(x^{(t)}) \geq {\ell^{(t)}}^T (x^{(t+1)} - x^{(t)}) + \frac{||\ell^{(t)}||_*^2}{2\eta \cdot \sigma}$$

$$\geq -\|\ell^{(t)}\|_* \|x^{(t+1)} - x^{(t)}\| + \frac{||\ell^{(t)}||_*}{2\eta \cdot \sigma} \geq -\frac{||\ell^{(t)}||_*^2}{2\eta \cdot \sigma}$$

Completing the Analysis

Progress in one iteration:

$$\Phi^{(t+1)} - \Phi^{(t)} \geq {\ell^{(t)}}^T x^{(t)} - \frac{\|\ell^{(t)}\|_*}{2\sigma\eta}$$

Regret at iteration t

Completing the Analysis

Progress in one iteration:

$$\Phi^{(t+1)} - \Phi^{(t)} \geq {\ell^{(t)}}^T x^{(t)} - \frac{\|\ell^{(t)}\|_*}{2\sigma\eta}$$

Telescopic sum:

$$\Phi^{(T+1)} \geq \sum_{t=1}^{T} {\ell^{(t)}}^T p^{(t)} + \Phi^{(1)} - T \cdot \frac{||\ell^{(t)}||}{2\eta \cdot \sigma}$$

Completing the Analysis

Progress in one iteration:

$$\Phi^{(t+1)} - \Phi^{(t)} \geq {\ell^{(t)}}^T x^{(t)} - \frac{\|\ell^{(t)}\|_*}{2\sigma\eta}$$

Telescopic sum:

$$\Phi^{(T+1)} \geq \sum_{t=1}^{T} {\ell^{(t)}}^T p^{(t)} + \Phi^{(1)} - T \cdot \frac{||\ell^{(t)}||}{2\eta \cdot \sigma}$$

Final regret bound:

$$\frac{1}{T}\left(\sum_{t=1}^{T} {\ell^{(t)}}^T x^{(t)} - \min_{x \in X} \sum_{t=1}^{T} {\ell^{(t)}}^T x\right) \cdot \quad \frac{\eta}{T} \cdot (\max_{x \in X} F(x) - \min_{x \in X} F(x)) + \frac{\rho^2}{2\sigma\eta}$$

Completing the Analysis

Regret bound: with regularizer F and $||\ell^{(t)}||_* \cdot \rho$

$$\frac{1}{T}\left(\sum_{t=1}^{T} {\ell^{(t)}}^T x^{(t)} - \min_{x\in X}\sum_{t=1}^{T} {\ell^{(t)}}^T x\right) \cdot \frac{\eta}{T}\cdot(\max_{x\in X} F(x) - \min_{x\in X} F(x)) + \frac{\rho^2}{2\sigma\eta}$$

Start-up Penalty

Penalty for being greedy

SAME TYPE OF BOUND AS FOR MWUs

Reinterpreting MWUs

Potential function: $$\Phi^{(t+1)} = \min_{\substack{p \geq 0, \\ \sum p_i = 1}} p^T L^{(t)} + \eta \cdot \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{p_i} \log \mathbf{p_i}$$

Regularizer: $F(p) = \sum_{i=1}^{n} p_i \log p_i$ is negative entropy

Reinterpreting MWUs

Potential function: $$\Phi^{(t+1)} = \min_{\substack{p \geq 0, \\ \sum p_i = 1}} p^T L^{(t)} + \eta \cdot \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{p_i} \log \mathbf{p_i}$$

Regularizer: $F(p) = \sum_{i=1}^{n} p_i \log p_i$ is negative entropy

SOFT-MAX

$F(p)$ is 1-strongly-convex w.r.t. $\| \cdot \|_\infty$

Update: $$p^{(t+1)} = \arg \min_{\substack{p \geq 0, \\ \sum p_i = 1}} p^T L^{(t)} + \eta \cdot \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{p_i} \log \mathbf{p_i}$$

$$p_i^{(t+1)} = \frac{e^{-\frac{1}{\eta} L_i^{(t)}}}{\sum_{i=1}^{n} e^{-\frac{1}{\eta} L_i^{(t)}}} = \frac{(1-\epsilon)^{L_i^{(t)}}}{\sum_{i=1}^{n} (1-\epsilon)^{L_i^{(t)}}}.$$

Reinterpreting MWUs

Potential function: $$\Phi^{(t+1)} = \min_{\substack{p \geq 0, \\ \sum p_i = 1}} p^T L^{(t)} + \eta \cdot \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{p_i} \log \mathbf{p_i}$$

Regularizer: $$F(p) = \sum_{i=1}^{n} p_i \log p_i \text{ is negative entropy}$$

$F(p)$ is 1-strongly-convex w.r.t. $\|\cdot\|_\infty$

Update: $$p^{(t+1)} = \arg \min_{\substack{p \geq 0, \\ \sum p_i = 1}} p^T L^{(t)} + \eta \cdot \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{p_i} \log \mathbf{p_i}$$

$$p_i^{(t+1)} = \frac{e^{-\frac{1}{\eta} L_i^{(t)}}}{\sum_{i=1}^{n} e^{-\frac{1}{\eta} L_i^{(t)}}} = \frac{(1-\epsilon)^{L_i^{(t)}}}{\sum_{i=1}^{n} (1-\epsilon)^{L_i^{(t)}}}.$$

Beyond MWUs: which regularizer?

Regret bound: optimizing over η

$$\frac{1}{T}\left(\sum_{t=1}^{T} {\ell^{(t)}}^T x^{(t)} - \min_{x\in X}\sum_{t=1}^{T} {\ell^{(t)}}^T x\right) \cdot \quad \frac{\rho\sqrt{(2\cdot(\max_{x\in X} F(x) - \min_{x\in X} F(x))}}{\sqrt{\sigma T}}$$

Best choice of regularizer and norm minimizes

$$\frac{\max_t ||\ell^{(t)}||_*^2 \cdot (\max_{x\in X} F(x) - \min_{x\in X} F(x))}{\sigma}$$

Beyond MWUs: which regularizer?

Regret bound: optimizing over η

$$\frac{1}{T}\left(\sum_{t=1}^{T} {\ell^{(t)}}^T x^{(t)} - \min_{x\in X}\sum_{t=1}^{T} {\ell^{(t)}}^T x\right) \cdot \quad \frac{\rho\sqrt{(2\cdot(\max_{x\in X} F(x) - \min_{x\in X} F(x))}}{\sqrt{\sigma T}}$$

Best choice of regularizer and norm minimizes

$$\frac{\max_t ||\ell^{(t)}||_*^2 \cdot (\max_{x\in X} F(x) - \min_{x\in X} F(x))}{\sigma}$$

Negative entropy with ℓ_1-norm is approximately optimal for simplex

QUESTION: are other regularizers ever useful?

Different Regularizers in Algorithm Design

QUESTION 1:

Are other regularizers, besides entropy, **ever useful?**

YES! Applications:

- **Graph Partitioning and Random Walks**
  - Spectral algorithms for balanced separator running in time $\tilde{O}(m)$

    Uses random-walk framework and SDP MWUs

    Different walks correspond to **different regularizers** for eigenvector problem

SDP MWU	$F(X) = \mathrm{Tr}(X \log X)$	$\longrightarrow$	Heat Kernel Random Walk
p-norm, $1 \leq p \leq \infty$	$F(X) = \mathrm{Tr}(X^p)$	$\longrightarrow$	Lazy Random Walk
NEW REGULARIZER	$F(X) = \mathrm{Tr}(X^{1/2})$	$\longrightarrow$	Personalized PageRank

[Mahoney, Orecchia, Vishnoi 2011], [Orecchia, Sachdeva, Vishnoi 2012]

Different Regularizers in Algorithm Design

QUESTION 1:

Are other regularizers, besides entropy, **ever useful?**

YES! Applications:

- **Graph Partitioning and Random Walks**
- **Sparsification**
  - ϵ-spectral-sparsifiers with $O(\frac{n \log n}{\epsilon^2})$ edges

    Uses Matrix concentration bound equivalent to SDP MWUs

    [Spielman, Srivastava 2008]
  - ϵ-spectral-sparsifiers with $O(\frac{n}{\epsilon^2})$ edges

    Can be interpreted as different regularizer: $F(X) = \mathrm{Tr}(X^{1/2})$

    [Batson, Spielman, Srivastava 2009]

Different Regularizers in Algorithm Design

QUESTION 1:

Are other regularizers, besides entropy, **ever useful?**

YES! Applications:

- **Graph Partitioning and Random Walks**
- **Sparsification**

Many more in Online Learning

- **Bandit Online Learning [AHR], …**

NON-SMOOTH CONVEX OPTIMIZATION REDUCES TO ONLINE LINEAR OPTIMIZATION

Convex Optimization Setup

$$\min_{x \in X} f(x)$$

f convex, differentiable

$X \subseteq \mathrm{R}^n$ closed, convex set

NON-SMOOTH

$$\forall x \in X,$$
$$\|\nabla f(x)\|_* \cdot \quad \rho$$

ρ-Lipschitz continuous

SMOOTH

$$\forall x, y \in X,$$
$$\|\nabla f(y) - \nabla f(x)\|_* \cdot \quad L\|y - x\|$$

ρ-Lipschitz continuous gradient

Convex Optimization Setup

$$\min_{x \in X} f(x)$$

f convex, differentiable

$X \subseteq \mathrm{R}^n$ closed, convex set

NON-SMOOTH

$$\forall x \in X,$$

$$\|\nabla f(x)\|_* \cdot \quad \rho$$

ρ-Lipschitz continuous

SMOOTH

$$\forall x, y \in X,$$

$$\|\nabla f(y) - \nabla f(x)\|_* \cdot \quad L\|y - x\|$$

ρ-Lipschitz continuous gradient

Gradient step is guaranteed to decrease function value

$$f(x^{(t+1)}) \cdot \quad f(x^{(t)}) - \frac{\|\nabla f(x^{(t)})\|_*^2}{2L}$$

Convex Optimization Setup

$$\min_{x \in X} f(x)$$

f convex, differentiable

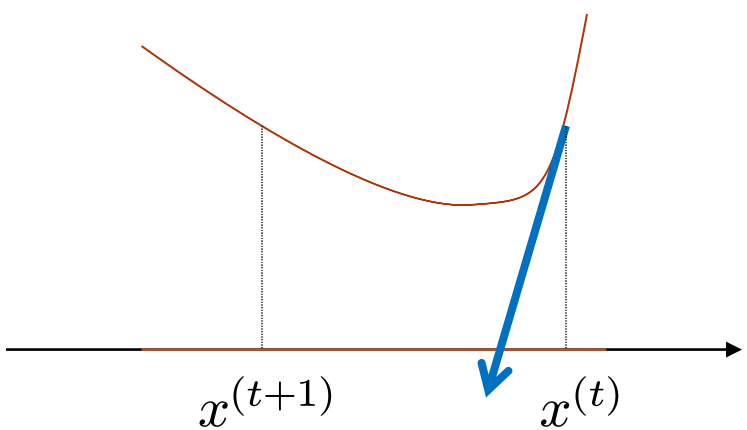
$X \subseteq \mathrm{R}^n$ closed, convex set

NON-SMOOTH

$$\forall x \in X,$$
$$\|\nabla f(x)\|_* \cdot \ \rho$$

ρ-Lipschitz continuous

NO GRADIENT STEP GUARANTEE



SMOOTH

$$\forall x, y \in X,$$
$$\|\nabla f(y) - \nabla f(x)\|_* \cdot \ L\|y - x\|$$

ρ-Lipschitz continuous gradient

Gradient step is guaranteed to decrease function value

$$f(x^{(t+1)}) \cdot \ f(x^{(t)}) - \frac{\|\nabla f(x^{(t)})\|_*^2}{2L}$$

Convex Optimization Setup

$$\min_{x \in X} f(x)$$

f convex, differentiable

$X \subseteq \mathrm{R}^n$ closed, convex set

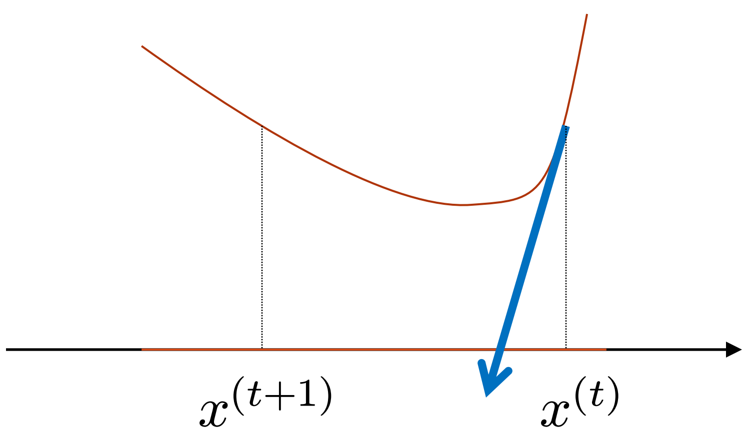
NON-SMOOTH

$$\forall x \in X,$$

$$\|\nabla f(x)\|_* \cdot \ \rho$$

ρ-Lipschitz continuous

NO GRADIENT STEP GUARANTEE



ONLY DUAL GUARANTEE

SMOOTH

$$\forall x, y \in X,$$

$$\|\nabla f(y) - \nabla f(x)\|_* \cdot \ L\|y - x\|$$

ρ-Lipschitz continuous gradient

Gradient step is guaranteed to decrease function value

$$f(x^{(t+1)}) \cdot \ f(x^{(t)}) - \frac{\|\nabla f(x^{(t)})\|_*^2}{2L}$$

Non-Smooth Setup: Dual Approach

$$\min_{x \in X} f(x)$$

f convex, differentiable

$X \subseteq \mathrm{R}^n$ closed, convex set

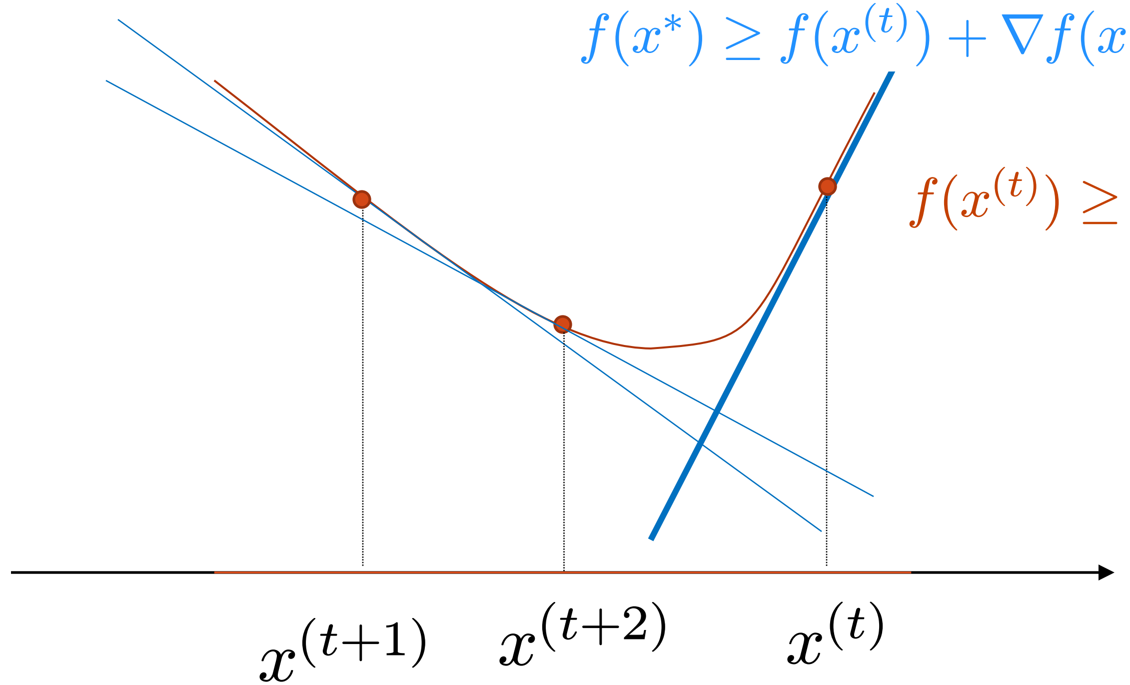
$$\forall x \in X,\ \|\nabla f(x)\|_* \cdot \ \rho$$

ρ-Lipschitz continuous

APPROACH: Each iterate solution provides a **lower bound** and an **upper bound**

$$f(x^*) \geq f(x^{(t)}) + \nabla f(x^{(t)})^T (x^* - x^{(t)})$$

$$f(x^{(t)}) \geq f(x^*)$$



Non-Smooth Setup: Dual Approach

$$\min_{x \in X} f(x)$$

f convex, differentiable

$X \subseteq \mathrm{R}^n$ closed, convex set

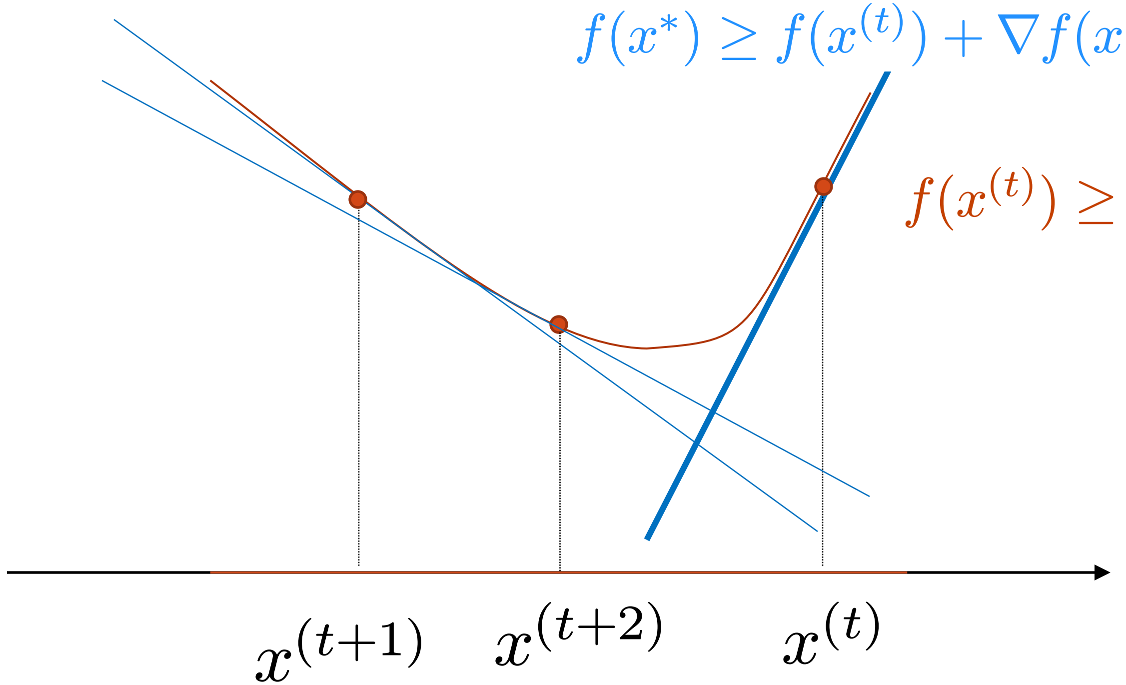
$$\forall x \in X,\ \|\nabla f(x)\|_* \cdot \ \rho$$

ρ-Lipschitz continuous

APPROACH: Each iterate solution provides a **lower bound** and an **upper bound**

$$f(x^*) \geq f(x^{(t)}) + \nabla f(x^{(t)})^T (x^* - x^{(t)})$$

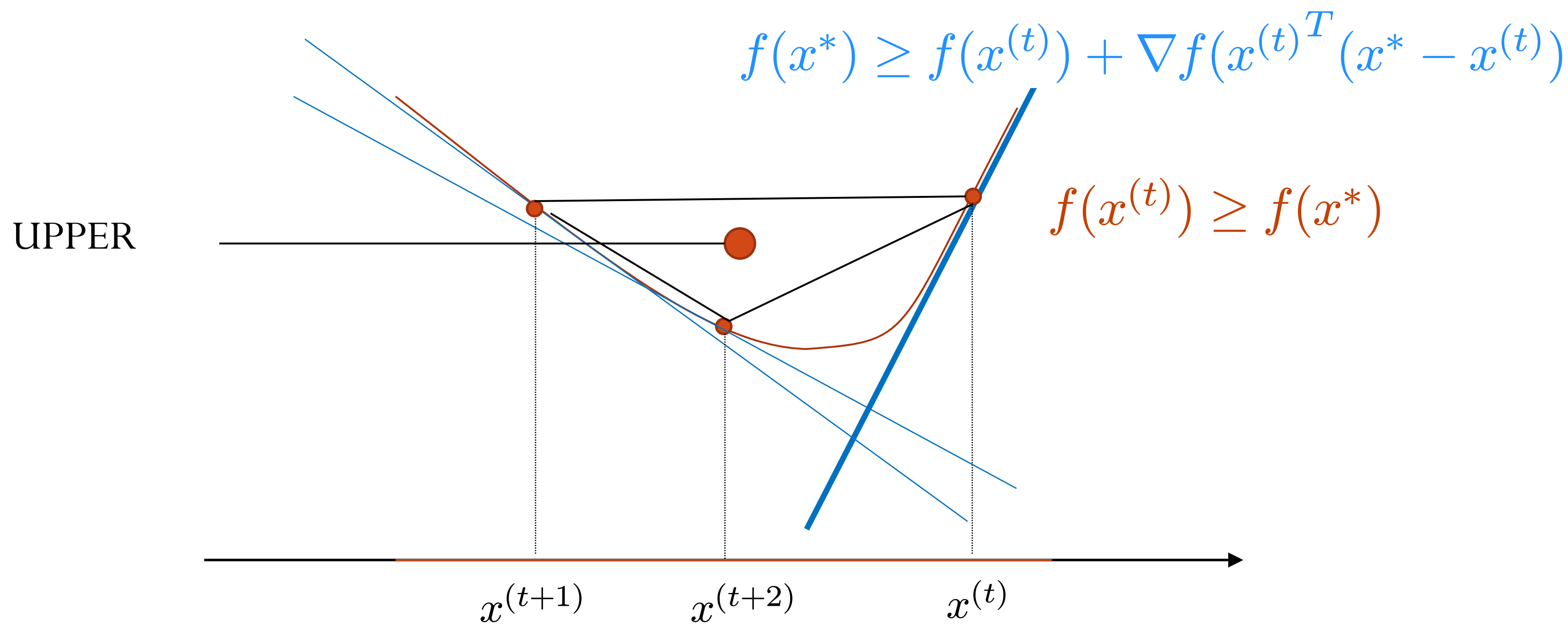
$$f(x^{(t)}) \geq f(x^*)$$



CAN WEAKEN DIFFERENTIABILITY ASSUMPTION: SUBGRADIENTS SUFFICE

Non-Smooth Setup: Dual Approach

APPROACH: Each iterate solution provides a **lower bound** and an **upper bound**



Take convex combination of both upper bounds and lower bounds with weights γ_t

UPPER BOUND: $$\frac{1}{\sum_{t=1}^T \gamma_t} \left(\sum_{t=1}^T \gamma_t f(x^{(t)}) \right) \geq f(x^*)$$

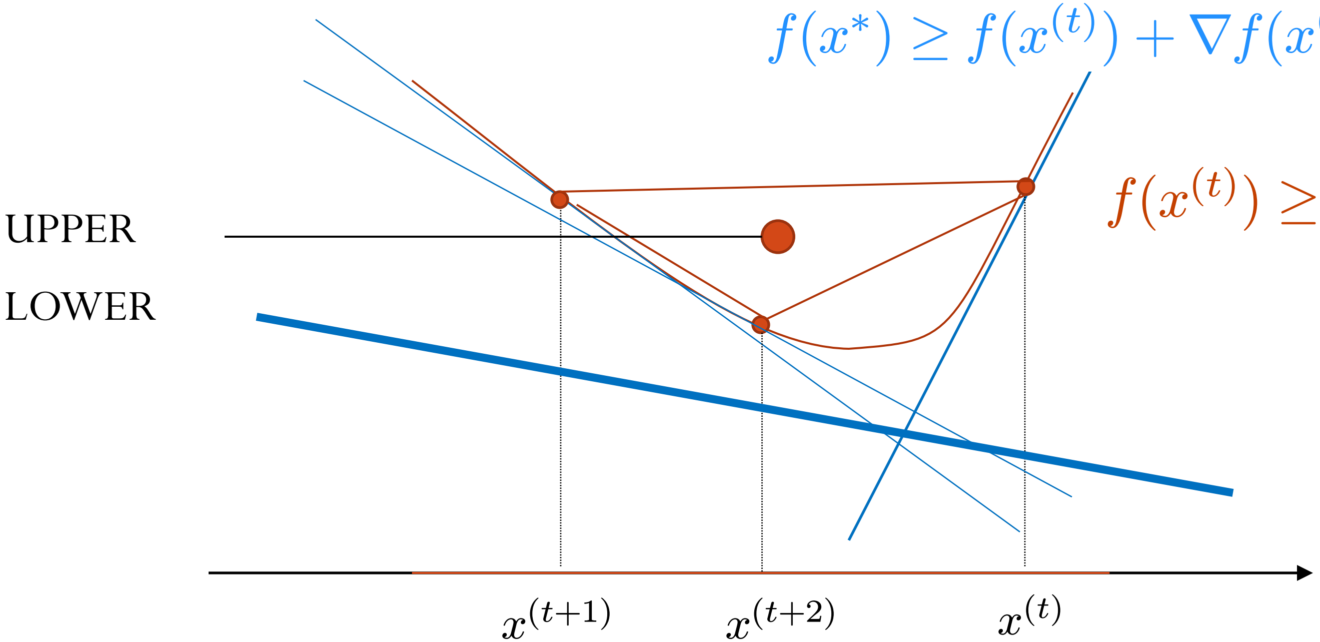
LOWER BOUND:

Non-Smooth Setup: Dual Approach

APPROACH: Each iterate solution provides a **lower bound** and an **upper bound**

$$f(x^*) \geq f(x^{(t)}) + \nabla f(x^{(t)})^T (x^* - x^{(t)})$$

$$f(x^{(t)}) \geq f(x^*)$$



Take convex combination of both upper bounds and lower bounds with weights γ_t

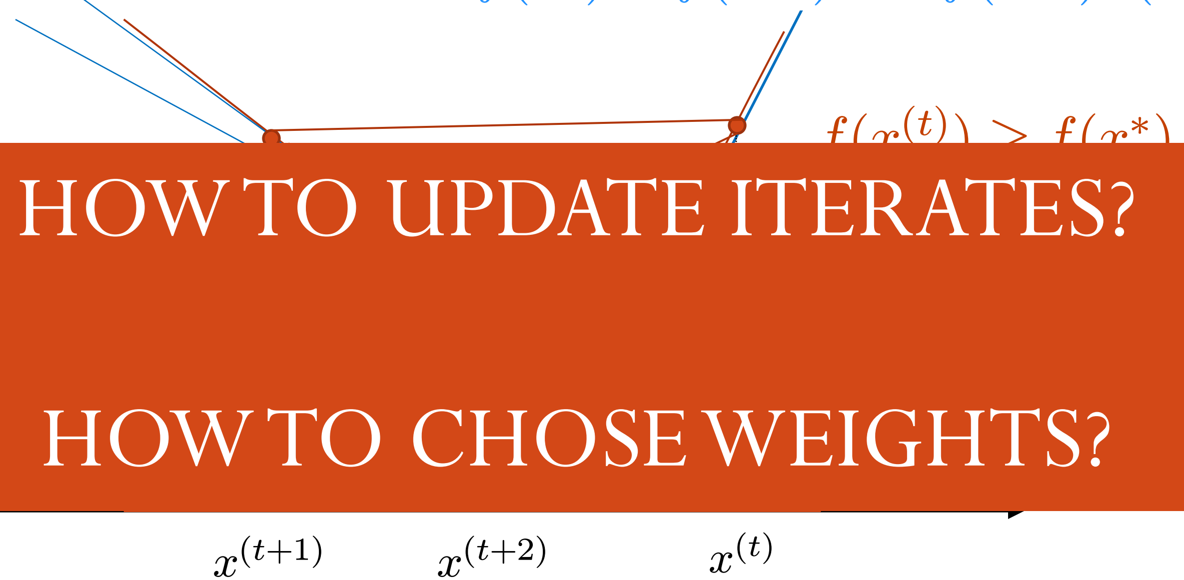
UPPER: $$\frac{1}{\sum_{t=1}^{T} \gamma_t} \left(\sum_{t=1}^{T} \gamma_t f(x^{(t)}) \right) \geq f(x^*)$$

LOWER : $$f(x^*) \geq \frac{1}{\sum_{t=1}^{T} \gamma_t} \left[\sum_{t=1}^{T} \gamma_t \left(f(x^{(t)}) + \nabla f(x^{(t)})^T (x^* - x^{(t)}) \right) \right]$$

Non-Smooth Setup: Dual Approach

APPROACH: Each iterate solution provides a **lower bound** and an **upper bound**

$$f(x^*) \geq f(x^{(t)}) + \nabla f(x^{(t)})^T (x^* - x^{(t)})$$



UPP

LOW

HOW TO UPDATE ITERATES?

HOW TO CHOSE WEIGHTS?

Take convex combination of both upper bounds and lower bounds with weights γ_t

UPPER: $$\frac{1}{\sum_{t=1}^T \gamma_t}\left(\sum_{t=1}^T \gamma_t f(x^{(t)})\right) \geq f(x^*)$$

LOWER : $$f(x^*) \geq \frac{1}{\sum_{t=1}^T \gamma_t}\left[\sum_{t=1}^T \gamma_t \left(f(x^{(t)}) + \nabla f(x^{(t)})^T (x^* - x^{(t)})\right)\right]$$

Reduction to Online Linear Minimization

Fix weights γ_t to be uniform for simplicity:

UPPER: $$\frac{1}{\sum_{t=1}^{T}\gamma_t}\left(\sum_{t=1}^{T}\gamma_t f(x^{(t)})\right) \geq f(x^*)$$

LOWER : $$f(x^*) \geq \frac{1}{\sum_{t=1}^{T}\gamma_t}\left[\sum_{t=1}^{T}\gamma_t (f(x^{(t)}) + \nabla f(x^{(t)})^T (x^* - x^{(t)}))\right]$$

DUALITY GAP:

$$\left[\sum_{t=1}^{T}\frac{\gamma_t}{\sum_{t=1}^{T}\gamma_t} f(x^{(t)})\right] - f(x^*) \cdot \quad \sum_{t=1}^{T} -\nabla f(x^{(t)})^T (x^* - x^{(t)})$$

LINEAR FUNCTION

Reduction to Online Linear Minimization

Fix weights γ_t to be uniform for simplicity:

DUALITY GAP:

$$\left[\sum_{t=1}^{T} \frac{\gamma_t}{\sum_{t=1}^{T} \gamma_t} f(x^{(t)})\right] - f(x^*) \cdot \sum_{t=1}^{T} -\nabla f(x^{(t)})^T (x^* - x^{(t)})$$

ONLINE SETUP

ALGORITHM

$$x^{(t)} \in X$$

ADVERSARY

$$-\nabla f(x^{(t)})$$

Reduction to Online Linear Minimization

Fix weights γ_t to be uniform for simplicity:

DUALITY GAP:

$$\left[\sum_{t=1}^{T} \frac{\gamma_t}{\sum_{t=1}^{T} \gamma_t} f(x^{(t)})\right] - f(x^*) \cdot \sum_{t=1}^{T} -\nabla f(x^{(t)})^T (x^* - x^{(t)})$$

ONLINE SETUP

ALGORITHM

$$x^{(t)} \in X \longrightarrow$$

ADVERSARY

$$\ell^{(t)} = -\nabla f(x^{(t)})$$

Loss vector is gradient

Recall that by assumption: $\|\ell^{(t)}\|_* = \|\nabla f(x^{(t)})\|_* \cdot \rho$

Reduction to Online Linear Minimization

Fix weights γ_t to be uniform for simplicity:

DUALITY GAP:

$$\left[\sum_{t=1}^{T} \frac{1}{T} f(x^{(t)})\right] - f(x^*) \cdot \quad \frac{1}{T} \cdot \sum_{t=1}^{T} -\nabla f(x^{(t)})^T (x^* - x^{(t)})$$

ONLINE SETUP

ALGORITHM → **ADVERSARY**

$$x^{(t)} \in X \longrightarrow \ell^{(t)} = -\nabla f(x^{(t)})$$

Loss vector is gradient

Recall that by assumption: $\|\ell^{(t)}\|_* = \|\nabla f(x^{(t)})\|_* \cdot \quad \rho$

$$\frac{1}{T} \cdot \sum_{t=1}^{T} -\nabla f(x^{(t)})^T (x^* - x^{(t)}) = \text{REGRET}$$

Final Bound

ONLINE SETUP

ALGORITHM

$$x^{(t)} \in X$$

→

ADVERSARY

$$\ell^{(t)} = -\nabla f(x^{(t)})$$

Loss vector is gradient

Recall that by assumption: $\|\ell^{(t)}\|_* = \|\nabla f(x^{(t)})\|_* \cdot \ \rho$

$$\sum_{t=1}^{T} -\nabla f(x^{(t)})^T (x^* - x^{(t)}) = \text{REGRET}$$

RESULTING ALGORITHM: MIRROR DESCENT

Error bound with σ-strongly-convex regularizer F

$$\epsilon_{\text{MD}} \cdot \ \frac{\rho\sqrt{2 \cdot (\max_{x \in X} F(x) - \min_{x \in X} F(x))}}{\sigma\sqrt{T}}$$

Final Bound

ONLINE SETUP

ALGORITHM

$x^{(t)} \in X$

ADVERSARY

$\ell^{(t)} = -\nabla f(x^{(t)})$

Loss vector is gradient

Recall that by assumption: $\|\ell^{(t)}\|_* = \|\nabla f(x^{(t)})\|_* \cdot \ \rho$

$$\sum_{t=1}^{T} -\nabla f(x^{(t)})^T (x^* - x^{(t)}) = \text{REGRET}$$

RESULTING ALGORITHM: MIRROR DESCENT

Error bound with σ-strongly-convex regularizer F

$$\epsilon_{\text{MD}} \cdot \ \frac{\rho\sqrt{2 \cdot (\max_{x \in X} F(x) - \min_{x \in X} F(x))}}{\sigma\sqrt{T}}$$

ASYMPTOTICALLY OPTIMAL BY INFORMATION COMPLEXITY LOWER BOUND

Non-Smooth Optimization over Simplex

RESULTING ALGORITHM:

MIRROR DESCENT OVER SIMPLEX = MWU

Regularizer F is negative entropy, with $\|\nabla f(x^{(t)})\|_\infty \cdot \rho$

$$\epsilon_{\mathrm{MD}} \cdot \frac{\rho\sqrt{2 \cdot \log n}}{\sqrt{T}}$$

APPLICATIONS IN ALGORITHM DESIGN

Warm-up Example: Linear Programming

$$A \in \mathrm{R}^{m \times n},$$
$$?\exists x \in X \ : Ax - b \geq 0$$

LP Feasibility problem

Easy constraints
Maintain feasible

Hard constraints
Require fixing

Warm-up Example: Linear Programming

$$A \in \mathrm{R}^{m \times n},$$
$$?\exists x \in X \;\; : Ax - b \geq 0$$

LP Feasibility problem

Convert into non-smooth optimization problem over simplex:

$$\min_{p \in \Delta_m} \max_{x \in X} p^T (b - Ax)$$

Non-differentiable objective:

$$f(p) = \max_{x \in X} p^T (b - Ax)$$

Warm-up Example: Linear Programming

$$A \in \mathrm{R}^{m \times n},$$
$$?\exists x \in X \ : Ax - b \geq 0$$

LP Feasibility problem

Convert into non-smooth optimization problem over simplex:

$$\min_{p \in \Delta_m} \max_{x \in X} p^T (b - Ax)$$

Non-differentiable objective:

$$f(p) = \max_{x \in X} p^T (b - Ax)$$

Best response to dual solution p

Warm-up Example: Linear Programming

$$A \in \mathrm{R}^{m \times n},$$
$$?\exists x \in X \ : b - Ax \geq 0$$

LP Feasibility problem

Convert into non-smooth optimization problem over simplex:

$$\min_{p \in \Delta_m} \max_{x \in X} p^T (b - Ax)$$

Non-differentiable objective

$$f(p) = \max_{x \in X} p^T (b - Ax)$$

Admits subgradients, for all p:

$$x_p : p^T (b - Ax_p) \geq 0;$$
$$(b - Ax_p) \in \partial f(p)$$

Subgradient is slack
in constraints

Warm-up Example: Linear Programming

$$A \in \mathrm{R}^{m \times n},$$
$$?\exists x \in X \ : b - Ax \geq 0$$

LP Feasibility problem

Convert into non-smooth optimization problem over simplex:

$$\min_{p \in \Delta_m} \max_{x \in X} p^T(b - Ax)$$

Non-differentiable objective

$$f(p) = \max_{x \in X} p^T(b - Ax)$$

Admits subgradients, for all p:

$$x_p : p^T(b - Ax_p) \geq 0;$$
$$(b - Ax_p) \in \partial f(p)$$

If we can pick x_p such that $\|b - Ax_p\|_\infty \cdot \ \rho$, then

$$\epsilon_{\text{MD}} \cdot \frac{\rho\sqrt{2 \cdot \log n}}{\sqrt{T}} \longrightarrow T \cdot \frac{2 \cdot \rho^2 \cdot \log n}{\epsilon^2}$$

MWU and s-t Maxflow

Minaximum flow feasibility for value F over undirected graph G with incidence matrix B:

$$\forall e \in E, F \cdot \frac{|f_e|}{c_e} \cdot \; 1$$

$$B^T f = e_s - e_t$$ ⟶ Will enforce this

Turn into non-smooth minimization problem over simplex:

$$f(p) = \min_{B^T f = e_s - e_t} \sum_{e \in E} p_e \cdot \frac{F \cdot |f_e|}{c_e} - 1$$

Best response f_p is shortest s-t path with lengths $p_e \,/\, c_e$.

For any p, if f_p has length $>\; 1$, there is no subgradient, i.e. problem is infeasible.

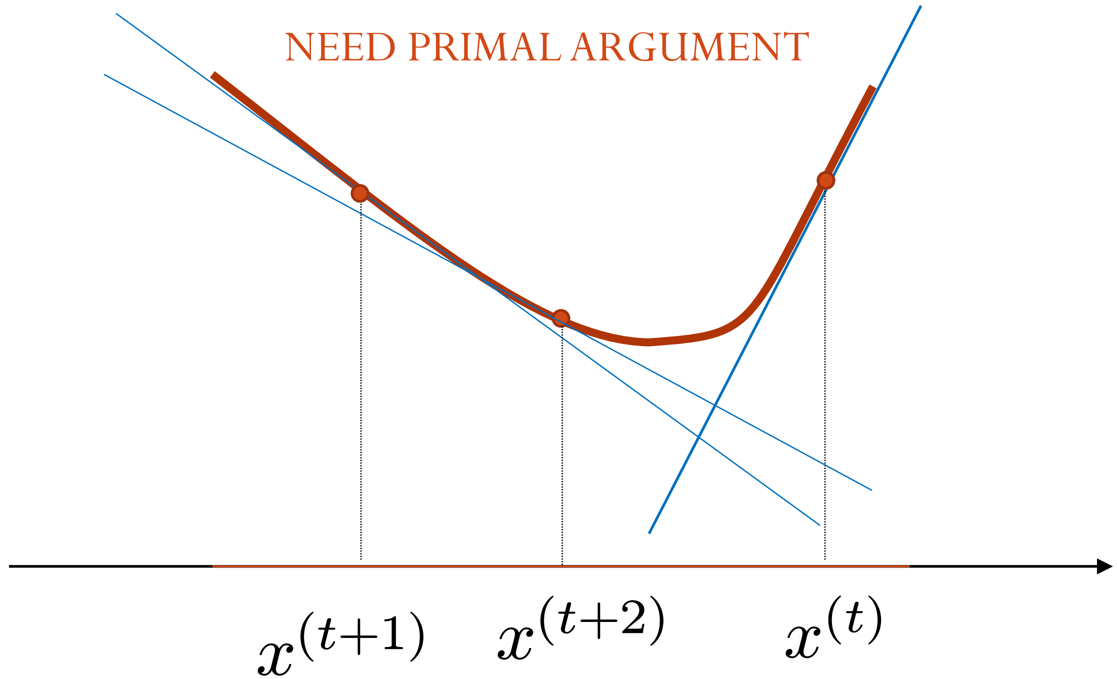
Otherwise, the following is a subgradient

$$\partial f(p)_e = \frac{F \cdot |(f_p)_e|}{c_e} - 1$$

Unfortunately, width can be large

$$\|\partial f(p)_e\|_\infty \cdot \; \frac{F}{c_{\min}} \xrightarrow{\text{[PST 91]}} T = O\left(\frac{F \log n}{\epsilon^2 c_{\min}}\right)$$

Width Reduction: make function nicer



PROBLEM: Optimal for this specific formulation $\|\partial f(p)_e\|_\infty \cdot \frac{F}{c_{\min}}$

SOLUTION: Regularize primal

$$f(p) = \min_{B^T f = e_s - e_t} F \cdot \sum_{e \in E} \frac{f_e}{c_e}\left(p_e + \frac{\epsilon}{m}\right) - 1$$

Width Reduction: make primal nicer

PROBLEM: Optimal for this specific formulation $\|\partial f(p)_e\|_\infty \cdot \dfrac{F}{c_{\min}}$

SOLUTION: Regularize primal

$$f(p) = \min_{B^T f = e_s - e_t} F \cdot \sum_{e \in E} \frac{f_e}{c_e} \left(p_e + \frac{\epsilon}{m} \right) - 1$$

REGULARIZATION ERROR: ϵF

NEW WIDTH: $\|\partial f(p)_e\|_\infty \cdot \dfrac{m}{\epsilon}$

ITERATION BOUND:

$$T = O\left(\frac{m \log n}{\epsilon^2} \right)$$

[GK 98]

Electrical Flow Approach [CKMST]

Different formulation yields basis for CKMST algorithm:

$$\forall e \in E, F \cdot \frac{f_e^2}{c_e^2} \cdot \; 1$$

$$B^T f = e_s - e_t \qquad \longrightarrow$$ Will enforce this

Non-smooth optimization problem:

$$f(p) = \min_{B^T f = e_s - e_t} \sum_{e \in E} p_e \cdot \frac{F \cdot f_e^2}{c_e^2} - 1$$

Electrical Flow Approach [CKMST]

Different formulation yields basis for CKMST algorithm:

$$\forall e \in E, F \cdot \frac{f_e^2}{c_e^2} \cdot \; 1$$

$$B^T f = e_s - e_t$$ ⟶ Will enforce this

Non-smooth optimization problem:

$$f(p) = \min_{B^T f = e_s - e_t} \sum_{e \in E} p_e \cdot \frac{F \cdot f_e^2}{c_e^2} - 1$$

Best response is electrical flow f_p

Original width: $\|\partial f(p)_e\|_\infty \cdot \; m$

Electrical Flow Approach [CKMST]

Different formulation yields basis for CKMST algorithm:

$$\forall e \in E, F \cdot \frac{f_e^2}{c_e^2} \cdot \ 1$$

$$B^T f = e_s - e_t \qquad \longrightarrow \qquad \text{Will enforce this}$$

Non-smooth optimization problem:

$$f(p) = \min_{B^T f = e_s - e_t} \sum_{e \in E} p_e \cdot \frac{F \cdot f_e^2}{c_e^2} - 1$$

Regularize primal:

$$f(p) = \min_{B^T f = e_s - e_t} F \cdot \sum_{e \in E} \frac{f_e^2}{c_e^2} \left(p_e + \frac{\epsilon}{m} \right) - 1$$

$$\|\partial f(p)_e\|_\infty \cdot \ \sqrt{\frac{m}{\epsilon}}$$

Conclusion: Take-away messages

- **Regularization** is a powerful tool for the design of fast algorithms.

- Most iterative algorithms can be understood as regularized updates:
  MWUs, Width Reduction, Interior Point, Gradient descent, ..

- Perform well in practice. Regularization also helps eliminate noise.

- **ULTIMATE GOAL:**

Development of a library of iterative methods for fast graph algorithms.

Regularization plays a fundamental role in this effort

THE END – THANK YOU