535520: Optimization Algorithms
Lecture 9 — Frank-Wolfe Method and Mirror Descent

Ping-Chun Hsieh (謝秉均)

November 11, 2024

This Lecture

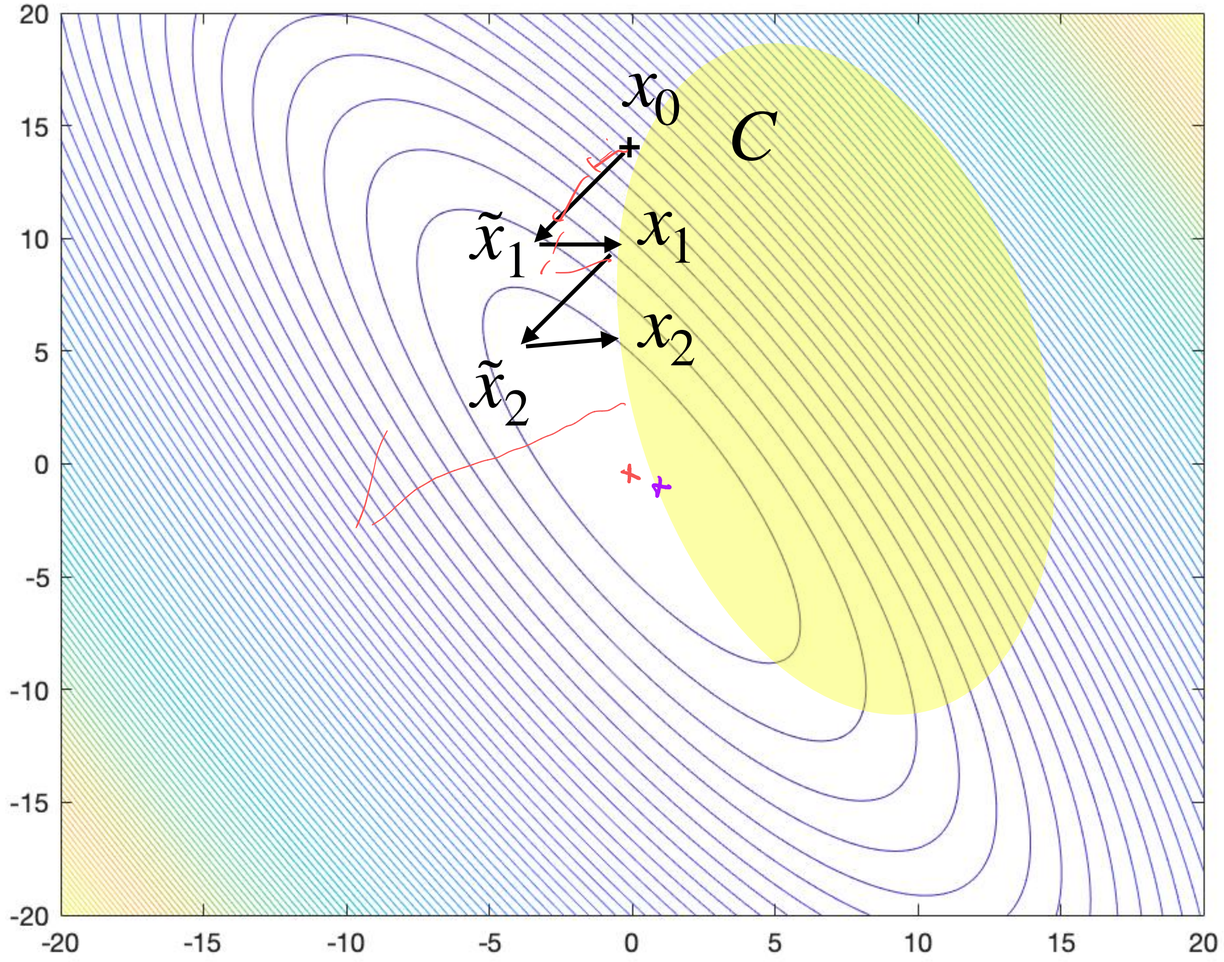
1. Projected Gradient Descent and Frank-Wolfe

2. Mirror Descent

- Reading Material:
  - M. Jaggi, “Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization,” ICML 2013
  - Amir Beck and Marc Teboulle, “Mirror descent and nonlinear projected subgradient methods for convex optimization,” Operations Research Letters, 2003.
  - Lecture notes of Prof. Anupam Gupta at CMU (http://www.cs.cmu.edu/~15850/notes/lec19.pdf)
  - Chapter 9 of Amir Beck’s textbook “First-Order Methods in Optimization”
  - Part of the material is adapted from Prof. Yuxin Chen’s lecture notes

Part 1. Gradient Methods for Constrained Problems

Review: Projected Gradient Descent (PGD)



- Constrained optimization

$$\min \; f(x)$$
$$\textbf{subject to } x \in C$$

- Under PGD, the iterates are updated as

$$x_{t+1} = \Pi_C(\underbrace{x_t - \eta_t \nabla f(x_t)}_{=:\tilde{x}_{t+1}})$$

where the projection is defined as

$$\Pi_C(x) := \arg\min_{c \in C} \|x - z\|$$

Review: Projection Theorem

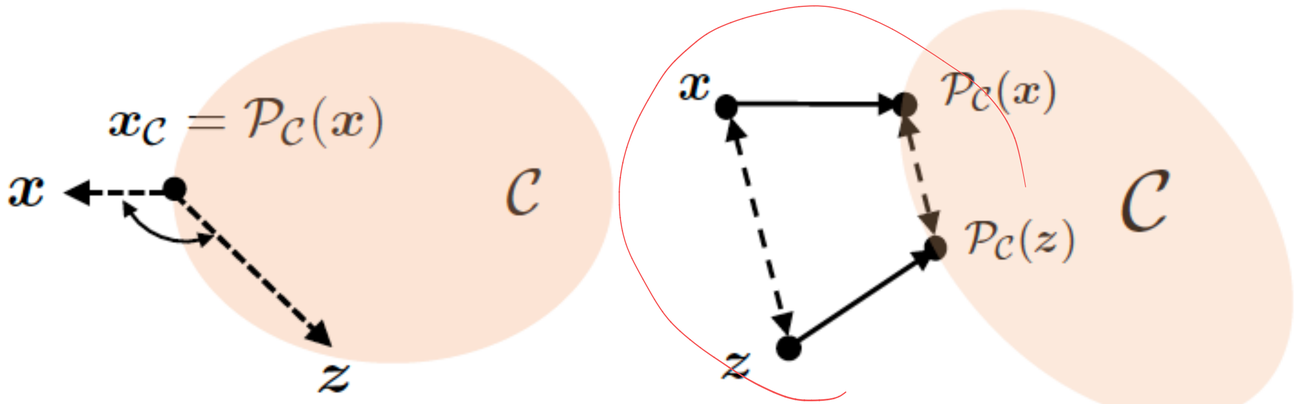
Theorem (Projection): Let C be a convex set.

(1) Given some vector $x \in \mathbb{R}^d$, a vector $x_C \in C$ is equal to the projection $\Pi_C(x)$ if and only if

$$(x - x_C)^\top (z - x_C) \leq 0, \quad \forall z \in C$$

(2) The mapping $h : \mathbb{R}^d \to C$ defined by $\Pi_C(x)$ is continuous and non-expansive, i.e.,

$$\|x_C - z_C\| \leq \|x - z\|, \quad \forall x, z \in \mathbb{R}^d$$



(Figure Credit: Yuxin Chen)

Review: PGD Stops at Local Minimizers

- FONC-C (Necessary condition for local minimizers of constrained problems)

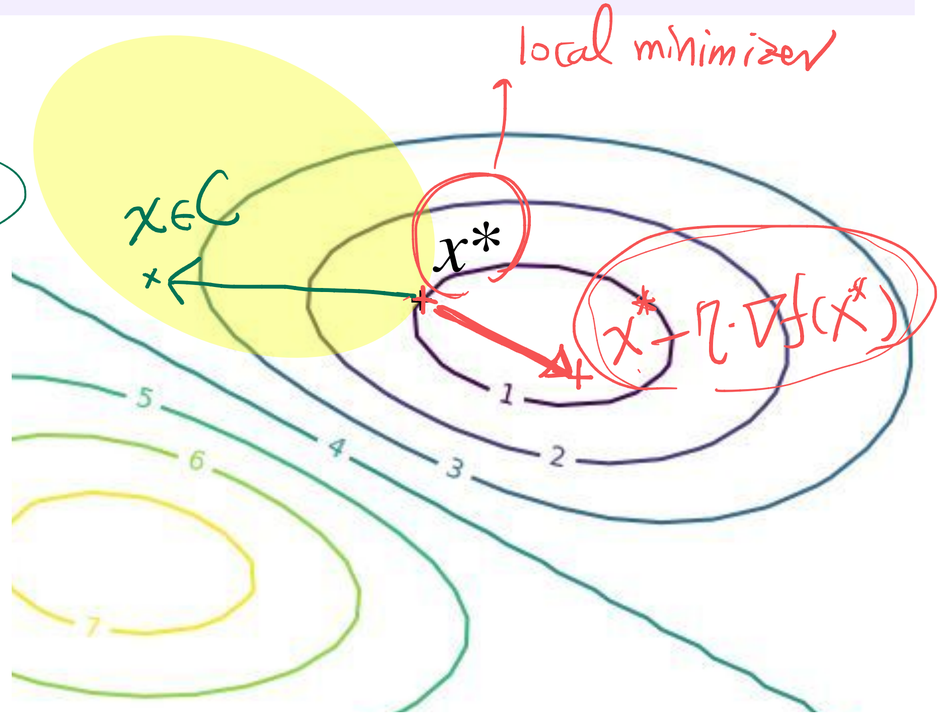
If x^* is a local minimizer of f over a convex feasible set C, then

$$\nabla f(x^*)^\top (x - x^*) \geq 0, \forall x \in C$$

- The above condition is equivalent to

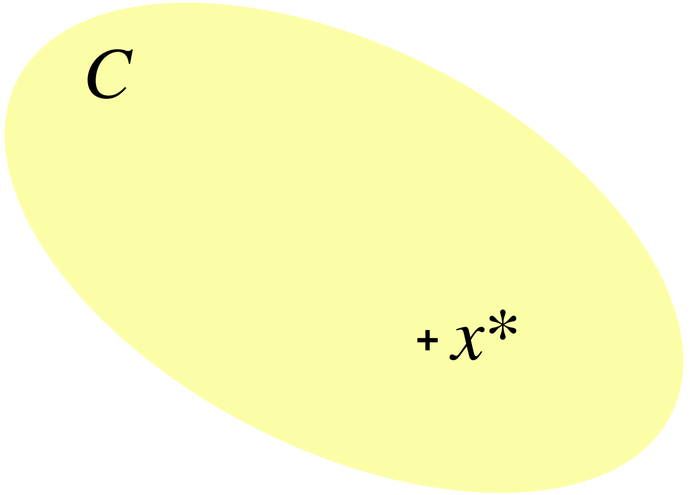
$$\Big((x^* - \eta \nabla f(x^*)) - x^*\Big)^\top (x - x^*) \leq 0, \forall x \in C$$

By Projection Theorem, this condition holds if and only if x^* is the projection of $x^* - \eta \nabla f(x^*)$ onto C

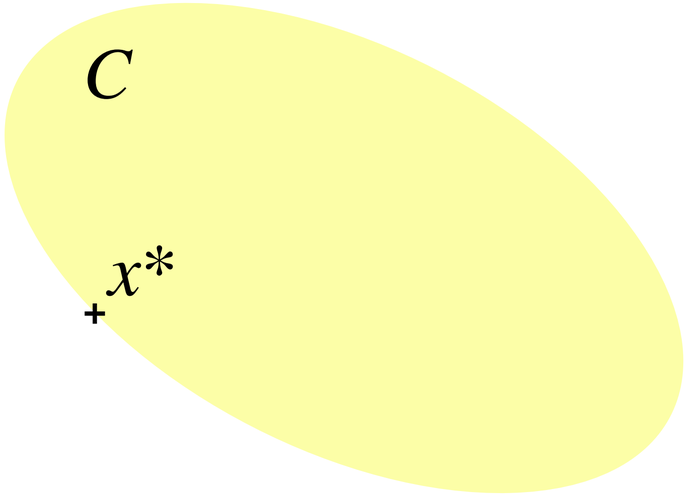


Review: Convergence of PGD: Two Possible Scenarios

- What are the possible scenarios of x^* in constrained problems?
  - Scenario 1: x^* is in the interior of C
    - Is this case challenging for PGD?



  - Scenario 2: x^* is on the boundary of C
    - Why is this case challenging for PGD?



Convergence of PGD: Strongly-Convex and Smooth Problems

Theorem (Convergence of PGD):

Let f be μ-strongly convex and L-smooth. Under PGD with constant step sizes $\eta = 1/L$, we have

$$\|x_t - x^*\|^2 \leqslant \left(1 - \frac{\mu}{L}\right)^t \cdot \|x_0 - x^*\|^2$$

- Question: Comparison with the convergence rate of Scenario #1?

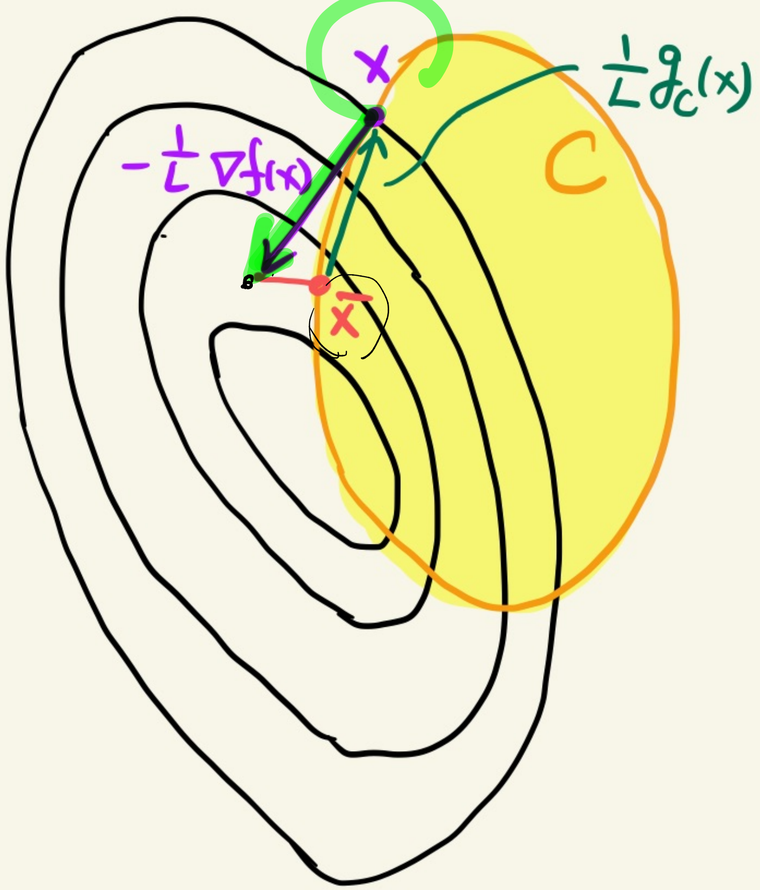
Proof Idea: Rewrite PGD in a form similar to GD.

- Define $\bar{x} := \Pi_C\left(x - \frac{1}{L}\nabla f(x)\right)$

$$g_C(x) := \frac{1}{\eta}(x - \bar{x}) = L(x - \bar{x})$$

- $g_C(x)$ can be viewed as "the generalized $\nabla f(x)$"

  as we know $g_C(x^*) = 0$



Claim: $$g_C(x)^T(x - x^*) \geq \frac{\mu}{2}\|x - x^*\|^2 + \frac{1}{2L}\|g_C(x)\|^2$$

(Intuition?)

$g_C(x) = 0$

$g_C(x) \approx 0$

(Given this claim, we can reuse the analysis of GD for PGD)

Want: $g_C(x)^T(x-x^*) \geq \frac{\mu}{2}\|x-x^*\|^2 + \frac{1}{2L}\|g_C(x)\|^2$

Proof of Claim:

Step1: $0 \leq f(\bar{x}) - f(x^*)$

$= \left(f(\bar{x}) - f(x)\right) + \left(f(x) - f(x^*)\right)$ (by L-Smoothness; by μ-Strong convexity.)

why? $\leq \left(\nabla f(x)^T(\bar{x}-x) + \frac{L}{2}\|\bar{x}-x\|^2\right) + \left(\nabla f(x)^T(x-x^*) - \frac{\mu}{2}\|x-x^*\|^2\right)$

$= \nabla f(x)^T(\bar{x}-x^*) + \frac{1}{2L}\|g_C(x)\|^2 - \frac{\mu}{2}\|x-x^*\|^2$

Step2: We also have $\nabla f(x)^T(\bar{x}-x^*) \leq g_C(x)^T(\bar{x}-x^*)$ since $\left(\bar{x} - \left(x - \frac{1}{L}\nabla f(x)\right)\right)^T(\bar{x}-x^*) \leq 0$ (by Projection theorem)

By Combining Step 1 and Step 2, we can verify the claim □

PGD for Convex and Smooth Problems

Convergence of PGD for Convex and Smooth Problems

Theorem (Convergence of PGD):

Let f be convex and L-smooth. Under PGD with constant step sizes $\eta = 1/L$, we have

$$f(x_t) - f(x^*) \leqslant \frac{3L\|x_0 - x^*\|^2 + \big(f(x_0) - f(x^*)\big)}{t+1}, \quad \forall t \in \mathbb{N}$$

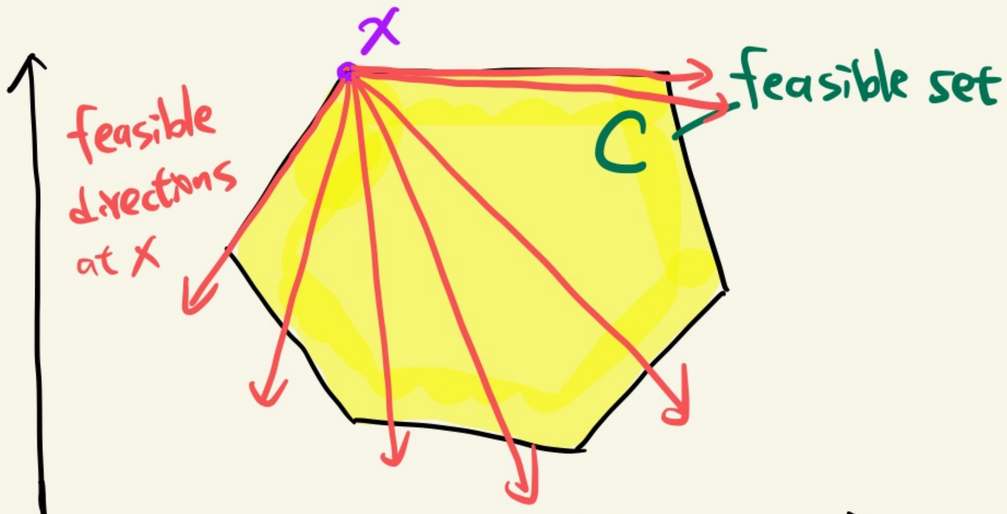
- Remark: PGD achieves $O(1/t)$ rate as GD for unconstrained problems

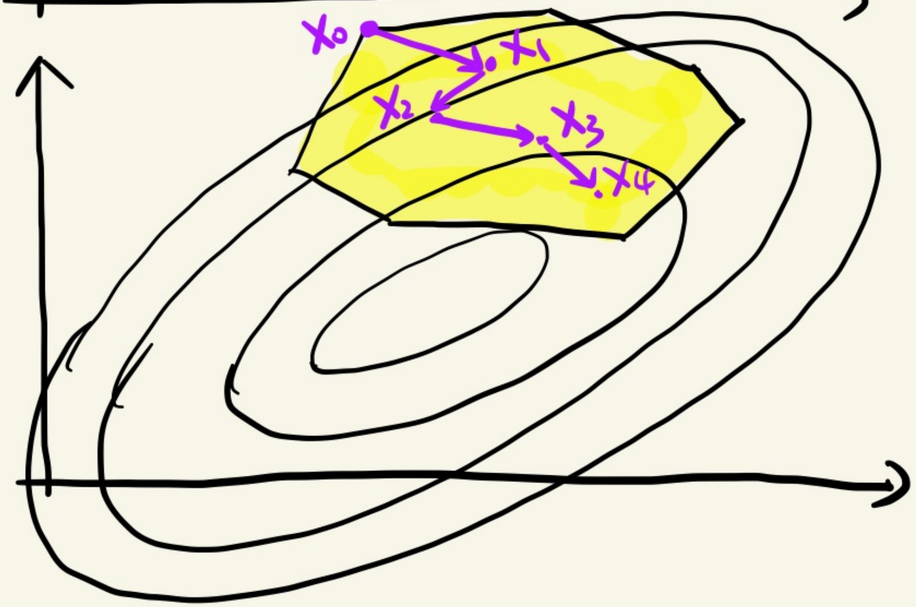
(Proof: HW2 problem for step-by-step analysis)

Can we design a gradient method *without* projection?

Frank-Wolfe Method (or Conditional Gradient)!

Feasible Direction Method





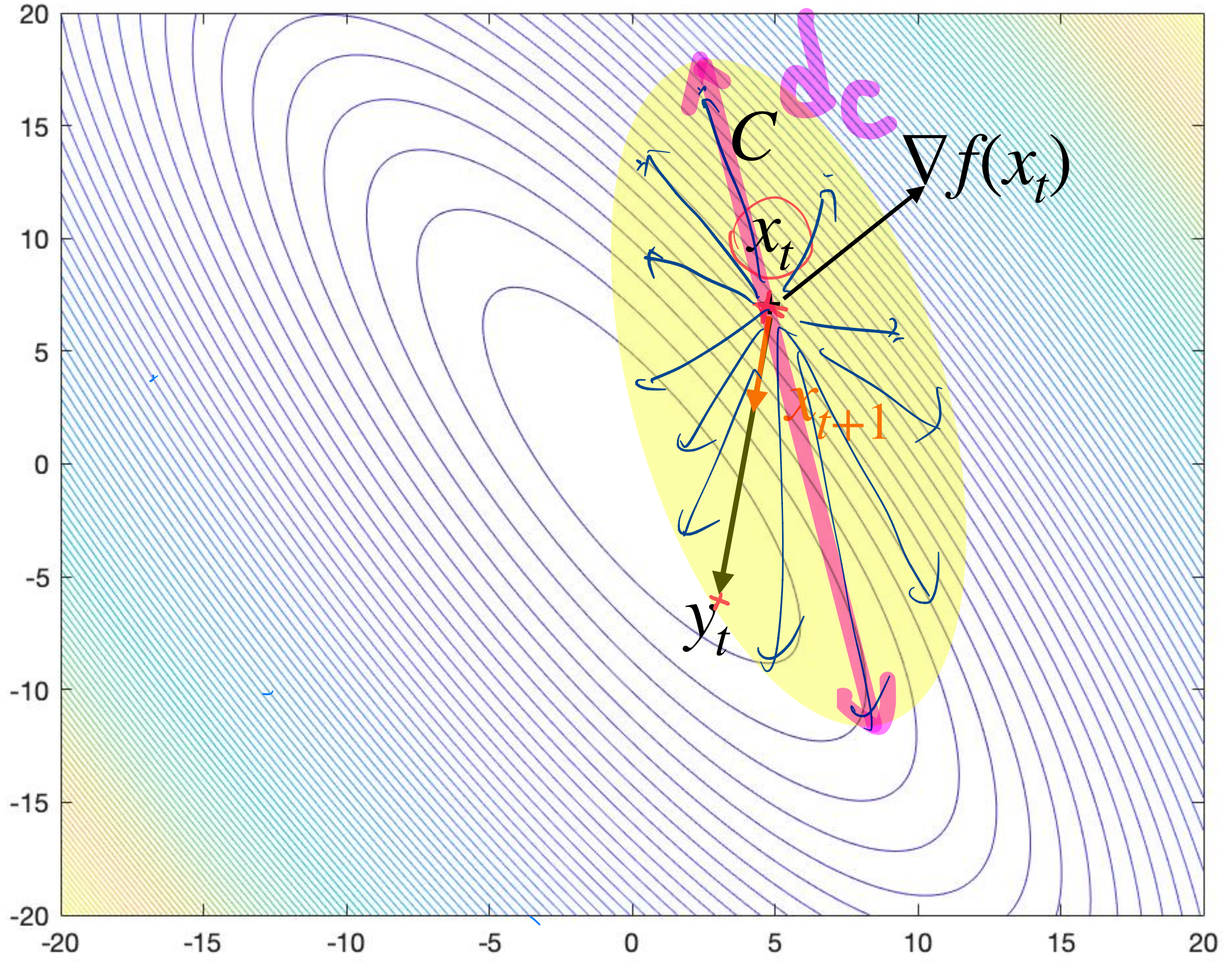
Definition: Given a vector X, we say d is a "feasible direction" at X if $(X+\alpha d)$ is also feasible for sufficiently small α.

Feasible Direction Method:

$$X_{t+1} = X_t + \alpha_t \cdot d_t,$$

where d_t is a feasible direction with $\nabla f(X_t)^T \cdot d_t < 0$

Frank-Wolfe (FW) Method



- Under FW, the iterates are updated as

$$y_t = \arg\min_{x \in C} \nabla f(x_t)^\top x$$

$$x_{t+1} = (1 - \eta_t) x_t + \eta_t y_t$$

$= x_t + \eta_t (y_t - x_t)$

- Question: Is $(y_t - x_t)$ a feasible direction? Yes!
- Question: Is FW a feasible direction method? Yes!
- Question: Is FW projection-free? Yes!

Convergence of Frank-Wolfe for Convex and Smooth Problems

Theorem (Convergence of FW):

Let f be convex and L-smooth. Under FW with step sizes $\eta_t = 2/(t+2)$, we have

$$f(x_t) - f(x^*) \leq \frac{2Ld_C^2}{t+2}$$

where $d_C := \sup_{x,y \in C} \|x - y\|$ is the diameter of C

- Remark: FW achieves $O(1/t)$ rate as PGD for convex smooth problems

M. Jaggi, "Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization," ICML 2013

Proof of Theorem

Step 1:

$$f(x_{t+1}) - f(x_t)$$

$$\leq \nabla f(x_t)^T (x_{t+1} - x_t) + \frac{L}{2} \| x_{t+1} - x_t \|^2 \quad \cdots \text{(by } L\text{-smoothness)}$$

$$\leq \eta_t \cdot \nabla f(x_t)^T (y_t - x_t) + \frac{L}{2} \eta_t^2 d_C^2 \quad \cdots \left(\begin{array}{l} x_{t+1} - x_t = \eta_t (y_t - x_t) \\ d_C = \sup \| x - y \| \geq \| y_t - x_t \| \end{array} \right)$$

$$\leq \eta_t \cdot \nabla f(x_t)^T (x^* - x_t) + \frac{L}{2} \eta_t^2 d_C^2 \quad \cdots \left(\nabla f(x_t)^T y_t \leq \nabla f(x_t)^T x, \text{ for any } x \in C \right)$$

$$\leq \eta_t \cdot \left(f(x^*) - f(x_t) \right) + \frac{L}{2} \eta_t^2 d_C^2 \quad \cdots \text{(by convexity)}$$

Step 2: By letting $\Delta_t := f(x_t) - f(x^*)$, we get a recursion

$$\Delta_{t+1} \leq (1 - \eta_t) \Delta_t + \frac{L}{2} \eta_t^2 d_C^2$$

$$\Delta_{t+1} \leq (1-\eta_t)\Delta_t + \frac{1}{t^2}$$

Suppose $\eta_t = \frac{1}{t}$, $\Delta_t \approx \text{const.} \cdot \frac{1}{t}$

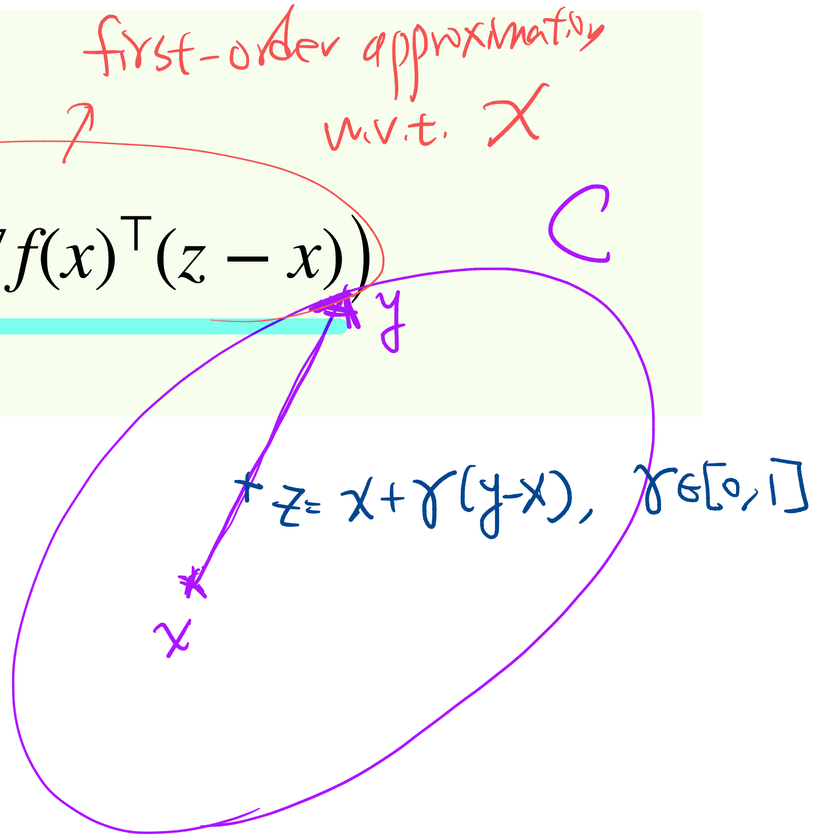
$$(1-\eta_t)\cdot\Delta_t = \frac{t-1}{t} \cdot \frac{1}{t} = \frac{t-1}{t^2}$$

Remark on Diameter and Curvature Constant

- FW has a convergence rate that depends on $L \cdot d_C^2$
- This can be slightly improved by considering the curvature constant (Jaggi, 2013)

Definition: The curvature constant C_f is defined as

$$C_f := \sup_{x,y \in C, \gamma \in [0,1], z = x + \gamma(y-x)} \frac{2}{\gamma^2} \left(f(z) - f(x) - \nabla f(x)^\top (z - x) \right)$$

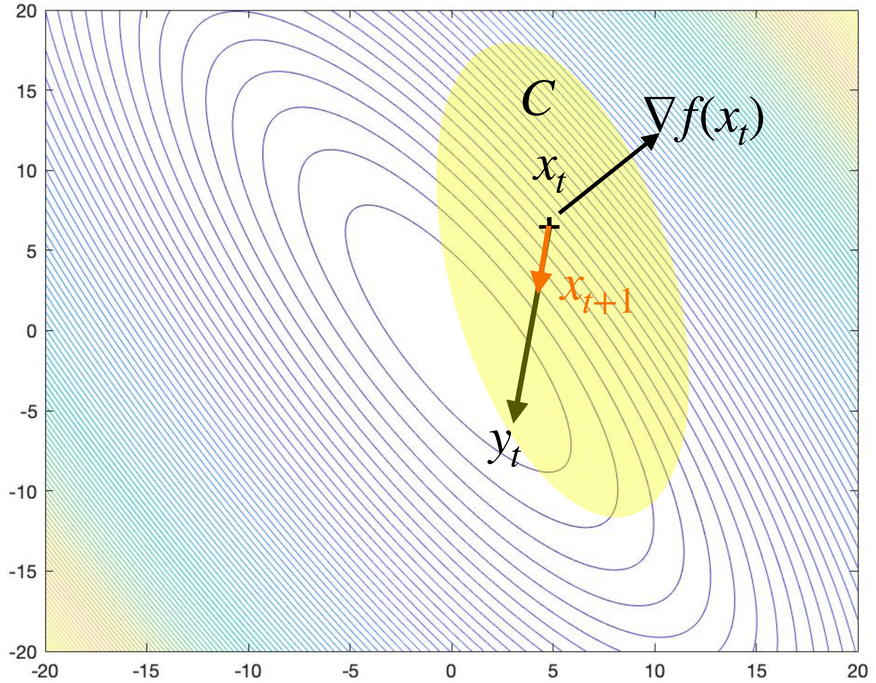


- Property: If f is L-smooth, then $C_f \leq L \cdot d_C^2$ (why?)

Remarks on the Step Size of Frank-Wolfe

- Step size $\eta_t = \frac{2}{t+2}$ does not guarantee "descent" (despite that it achieves $O(\frac{1}{t})$ convergence rate)
- To ensure "descent" in each iteration, FW can be combined with exact line search, i.e.,

$$\eta_t \in \arg\min_{\eta \in [0,1]} f(x_t + \eta(y_t - x_t))$$



How to Characterize the Stopping Criterion of Frank-Wolfe?

- Recall: Necessary condition for local minimizers of constrained problems

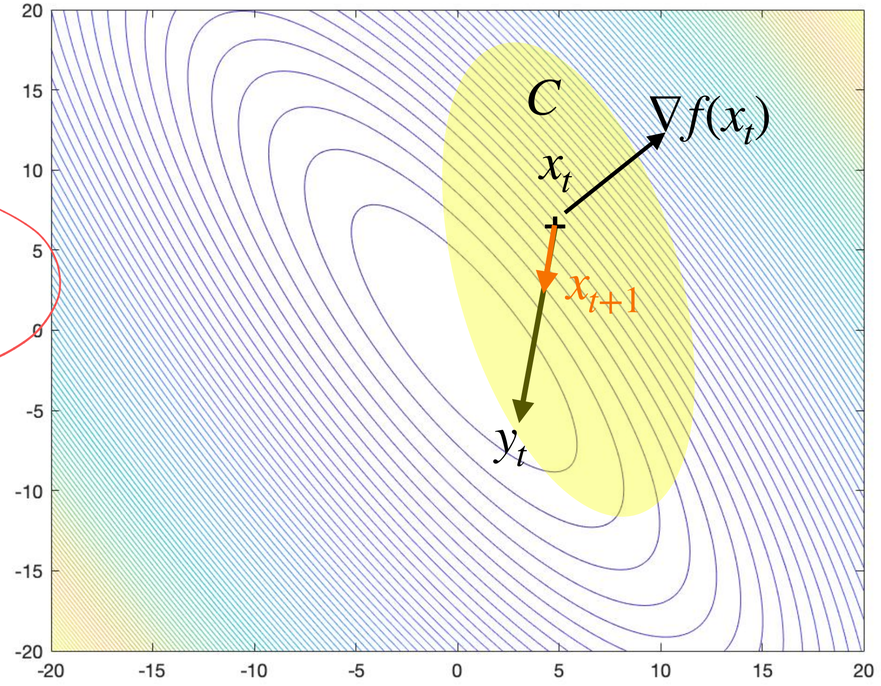
> *If x^* is a local minimizer of f over a convex feasible set C, then*
>
> $$\nabla f(x^*)^\top (x - x^*) \geq 0, \forall x \in C$$

- Idea: Convert the above into Frank-Wolfe gap

$$G_{FW}(x_t) := \max_{x \in C} \nabla f(x_t)^\top (x_t - x) = \nabla f(x_t)^\top (x_t - y_t)$$

$\nabla f(x)^\top (z - x) \geq 0$

  - $G_{FW}(x) \geq 0$, for all $x \in C$
  - $G_{FW}(x) = 0$ if and only if x is a local minimizer
  - If f is convex, then $f(x_t) - f(x^*) \leq G_{FW}(x_t)$



Proof of Stopping Criterion of Frank-Wolfe

surrogate

$$G_{FW}(x_t) := \max_{x \in C} \nabla f(x_t)^\top (x_t - x) = \nabla f(x_t)^\top (x_t - y_t)$$

- $G_{FW}(x) \geq 0$, for all $x \in C$

$G_{FW}(x) := \max_{z \in C} \nabla f(x)^\top (x - z) \geq \nabla f(x)^\top (x - x) = 0$

- $G_{FW}(x) = 0$ if and only if x is a local minimizer

✓ $G_{FW}(x)=0 \Rightarrow x$ is a local minimizer

$\nabla f(x)^\top (x - z) \leq 0$ for all $z \in C$

Essentially, x satisfies FONC-C

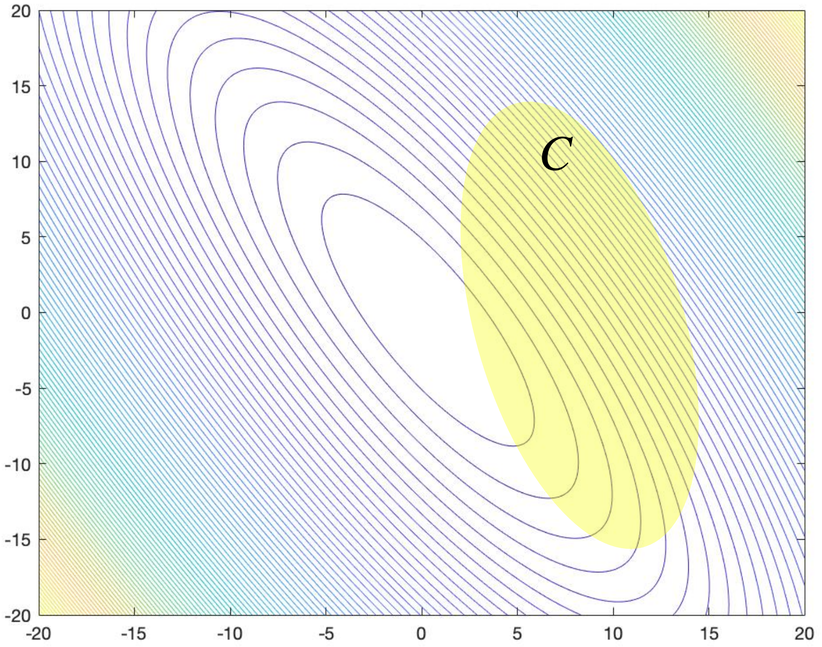
Hence, x is a local minimizer.

x is a local minimizer $\Rightarrow G_{FW}(x)=0$.

Since x is a local minimizer, x satisfies FONC-C, i.e.,

$\nabla f(x)^\top (z - x) \geq 0$ for all $z \in C$

$\Leftrightarrow \nabla f(x)^\top (x - z) \leq 0$, for all $z \in C \Rightarrow G_{FW}(x)=0$.



Proof of Stopping Criterion of Frank-Wolfe (Cont.)

$$G_{FW}(x_t) := \max_{x \in C} \nabla f(x_t)^\top (x_t - x) = \nabla f(x_t)^\top (x_t - y_t)$$

- If f is convex, then $f(x_t) - f(x^*) \leq G_{FW}(x_t)$

Question: Can FW attain a better convergence rate under *strong convexity* (compared to only under convexity)?

Nope in general!

A Counterexample on Impossibility of Faster Convergence Than $O(1/t)$

$$\min_{x} \quad \frac{1}{2} x^\top Q x + b^\top x$$

subject to $x \in \textbf{Conv}\{a_1, \cdots, a_k\} =: C$

- Suppose x^* is on the boundary of C
- Suppose the interior of C is non-empty
- Q is positive definite

- A Classic Result by (Canon & Cullum, 1968): Under FW, there exists an initial point x_0 in the interior of C such that for any $\epsilon > 0$, we have

$$f(x_t) - f(x^*) \geq \frac{1}{t^{1+\epsilon}}, \quad \text{infinitely often}$$

M. Canon and C. Cullum, “A tight upper bound on the rate of convergence of Frank-Wolfe algorithm,” SIAM Journal on Control, 1968.

Another Counterexample on Impossibility of Faster Convergence Than $O(1/t)$

$$\begin{aligned} \min_{x} \quad & \|x\|^2 \\ \textbf{subject to} \quad & x^{(i)} \geq 0, \sum_{i=1}^{d} x^{(i)} = 1 \end{aligned}$$

- A Well-Known Result by (Jaggi, 2013): Under FW and the above quadratic problem with simplex constraint, $f(x) - f(x^*) \leq \epsilon$ requires $\Omega(\min\{n, 1/\epsilon\})$ optimization steps

M. Jaggi. Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization, ICML 2013

Summary: Comparison of PGD and FW

- Consider convex and smooth problems

	Step Size	Convergence Rate	Iteration Complexity
PGD	$\eta = \frac{1}{L}$	$f(x_t) - f(x^*) \leqslant \frac{3L\|x_0 - x^*\|^2 + \left(f(x_0) - f(x^*)\right)}{t+1} = O(\frac{1}{t})$	$O(\frac{1}{\epsilon})$
FW	$\eta_t = \frac{2}{t+2}$	$f(x_t) - f(x^*) \leqslant \frac{2Ld_C^2}{t+2} = O(\frac{1}{t})$	$O(\frac{1}{\epsilon})$

How About Frank-Wolfe for Non-Convex Functions?

Convergence of Frank-Wolfe for Non-Convex Functions

Theorem Let $f(x)$ and C satisfy the following conditions:

(1) f is a continuously differentiable function (possibly non-convex).

(2) C is convex and compact.

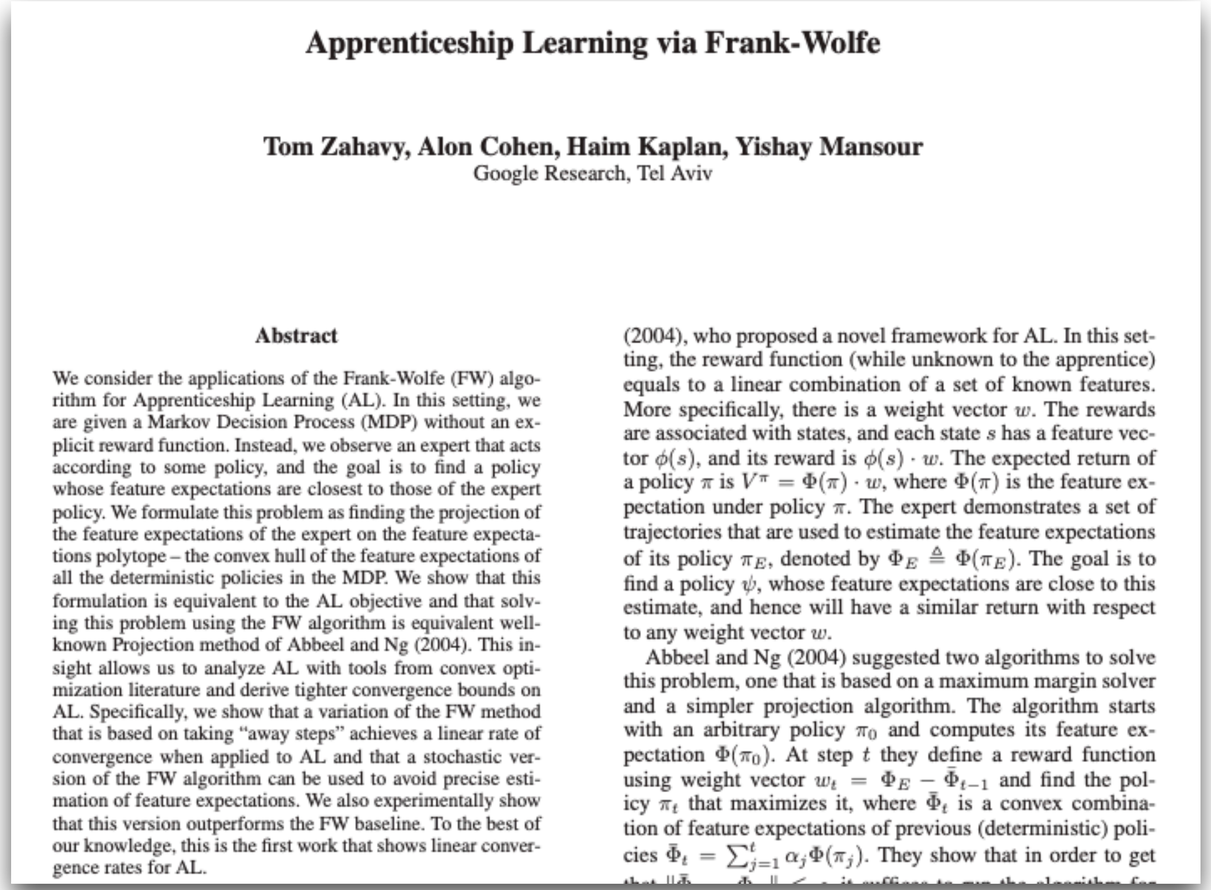
Then, we have $$\min_{0 \le t \le T} G_{FW}(x_t) \le \frac{\max\{2(f(x_0) - f(x^*)), C_f\}}{\sqrt{T+1}}$$

Simon Lacoste-Julien, Convergence Rate of Frank-Wolfe for Non-Convex Objectives, arXiv 2016

Some Applications of Frank-Wolfe Method

- L_p-norm regularization
- Trust-region optimization
- Frank-Wolfe for inverse RL
- Frank-Wolfe for constrained RL

[Zahavy et al., AAAI 2020]



Apprenticeship Learning via Frank-Wolfe

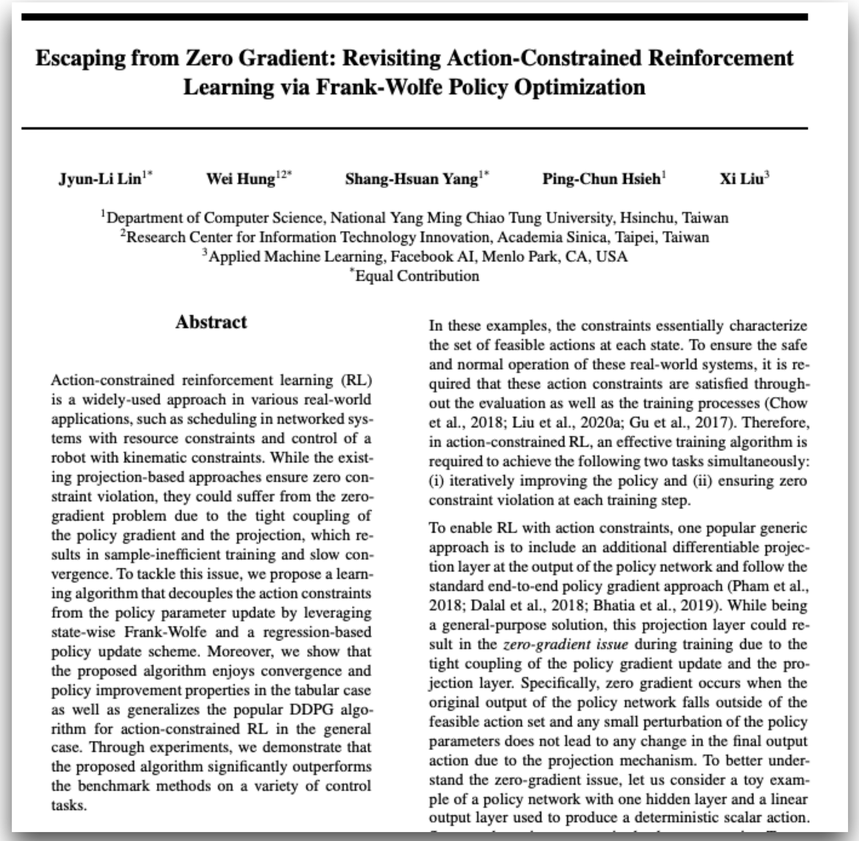
Tom Zahavy, Alon Cohen, Haim Kaplan, Yishay Mansour
Google Research, Tel Aviv

Abstract

We consider the applications of the Frank-Wolfe (FW) algorithm for Apprenticeship Learning (AL). In this setting, we are given a Markov Decision Process (MDP) without an explicit reward function. Instead, we observe an expert that acts according to some policy, and the goal is to find a policy whose feature expectations are closest to those of the expert policy. We formulate this problem as finding the projection of the feature expectations of the expert on the feature expectations polytope – the convex hull of the feature expectations of all the deterministic policies in the MDP. We show that this formulation is equivalent to the AL objective and that solving this problem using the FW algorithm is equivalent well-known Projection method of Abbeel and Ng (2004). This insight allows us to analyze AL with tools from convex optimization literature and derive tighter convergence bounds on AL. Specifically, we show that a variation of the FW method that is based on taking "away steps" achieves a linear rate of convergence when applied to AL and that a stochastic version of the FW algorithm can be used to avoid precise estimation of feature expectations. We also experimentally show that this version outperforms the FW baseline. To the best of our knowledge, this is the first work that shows linear convergence rates for AL.

(2004), who proposed a novel framework for AL. In this setting, the reward function (while unknown to the apprentice) equals to a linear combination of a set of known features. More specifically, there is a weight vector w. The rewards are associated with states, and each state s has a feature vector $\phi(s)$, and its reward is $\phi(s) \cdot w$. The expected return of a policy π is $V^\pi = \Phi(\pi) \cdot w$, where $\Phi(\pi)$ is the feature expectation under policy π. The expert demonstrates a set of trajectories that are used to estimate the feature expectations of its policy π_E, denoted by $\Phi_E \triangleq \Phi(\pi_E)$. The goal is to find a policy ψ, whose feature expectations are close to this estimate, and hence will have a similar return with respect to any weight vector w.

Abbeel and Ng (2004) suggested two algorithms to solve this problem, one that is based on a maximum margin solver and a simpler projection algorithm. The algorithm starts with an arbitrary policy π_0 and computes its feature expectation $\Phi(\pi_0)$. At step t they define a reward function using weight vector $w_t = \Phi_E - \bar{\Phi}_{t-1}$ and find the policy π_t that maximizes it, where $\bar{\Phi}_t$ is a convex combination of feature expectations of previous (deterministic) policies $\bar{\Phi}_t = \sum_{j=1}^{t} \alpha_j \Phi(\pi_j)$. They show that in order to get



Escaping from Zero Gradient: Revisiting Action-Constrained Reinforcement Learning via Frank-Wolfe Policy Optimization

Jyun-Li Lin[1*] **Wei Hung**[12*] **Shang-Hsuan Yang**[1*] **Ping-Chun Hsieh**[1] **Xi Liu**[3]

[1]Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
[2]Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
[3]Applied Machine Learning, Facebook AI, Menlo Park, CA, USA
[*]Equal Contribution

Abstract

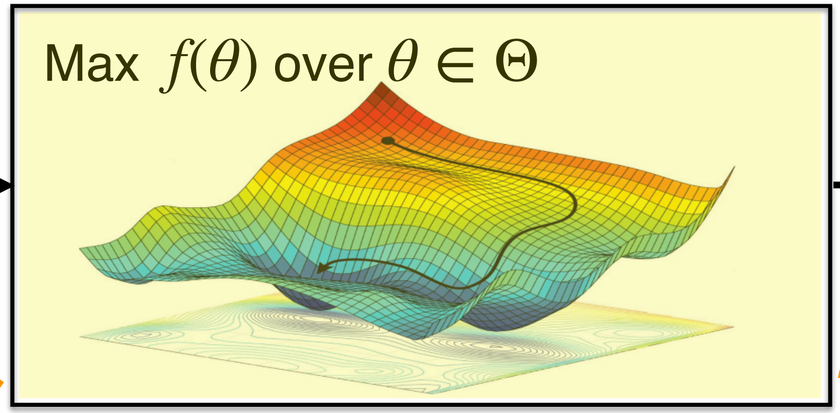
Action-constrained reinforcement learning (RL) is a widely-used approach in various real-world applications, such as scheduling in networked systems with resource constraints and control of a robot with kinematic constraints. While the existing projection-based approaches ensure zero constraint violation, they could suffer from the zero-gradient problem due to the tight coupling of the policy gradient and the projection, which results in sample-inefficient training and slow convergence. To tackle this issue, we propose a learning algorithm that decouples the action constraints from the policy parameter update by leveraging state-wise Frank-Wolfe and a regression-based policy update scheme. Moreover, we show that the proposed algorithm enjoys convergence and policy improvement properties in the tabular case as well as generalizes the popular DDPG algorithm for action-constrained RL in the general case. Through experiments, we demonstrate that the proposed algorithm significantly outperforms the benchmark methods on a variety of control tasks.

In these examples, the constraints essentially characterize the set of feasible actions at each state. To ensure the safe and normal operation of these real-world systems, it is required that these action constraints are satisfied throughout the evaluation as well as the training processes (Chow et al., 2018; Liu et al., 2020a; Gu et al., 2017). Therefore, in action-constrained RL, an effective training algorithm is required to achieve the following two tasks simultaneously: (i) iteratively improving the policy and (ii) ensuring zero constraint violation at each training step.

To enable RL with action constraints, one popular generic approach is to include an additional differentiable projection layer at the output of the policy network and follow the standard end-to-end policy gradient approach (Pham et al., 2018; Dalal et al., 2018; Bhatia et al., 2019). While being a general-purpose solution, this projection layer could result in the *zero-gradient issue* during training due to the tight coupling of the policy gradient update and the projection layer. Specifically, zero gradient occurs when the original output of the policy network falls outside of the feasible action set and any small perturbation of the policy parameters does not lead to any change in the final output action due to the projection mechanism. To better understand the zero-gradient issue, let us consider a toy example of a policy network with one hidden layer and a linear output layer used to produce a deterministic scalar action.

[Lin et al., UAI 2021]

RL as an Optimization Problem



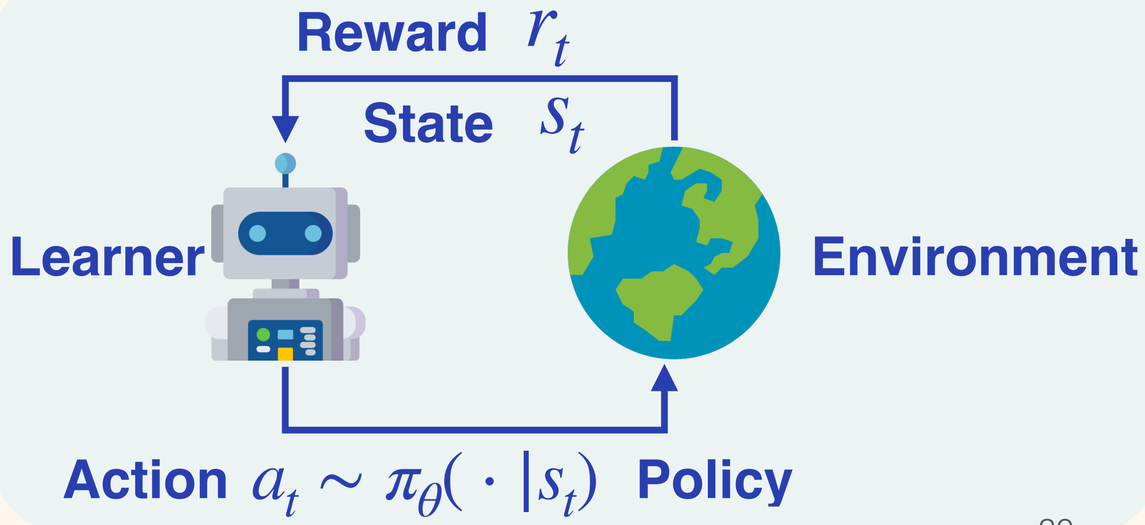
Next iterate θ_t

Information about f at θ_t

Optimization Algorithm

μ : initial state distribution

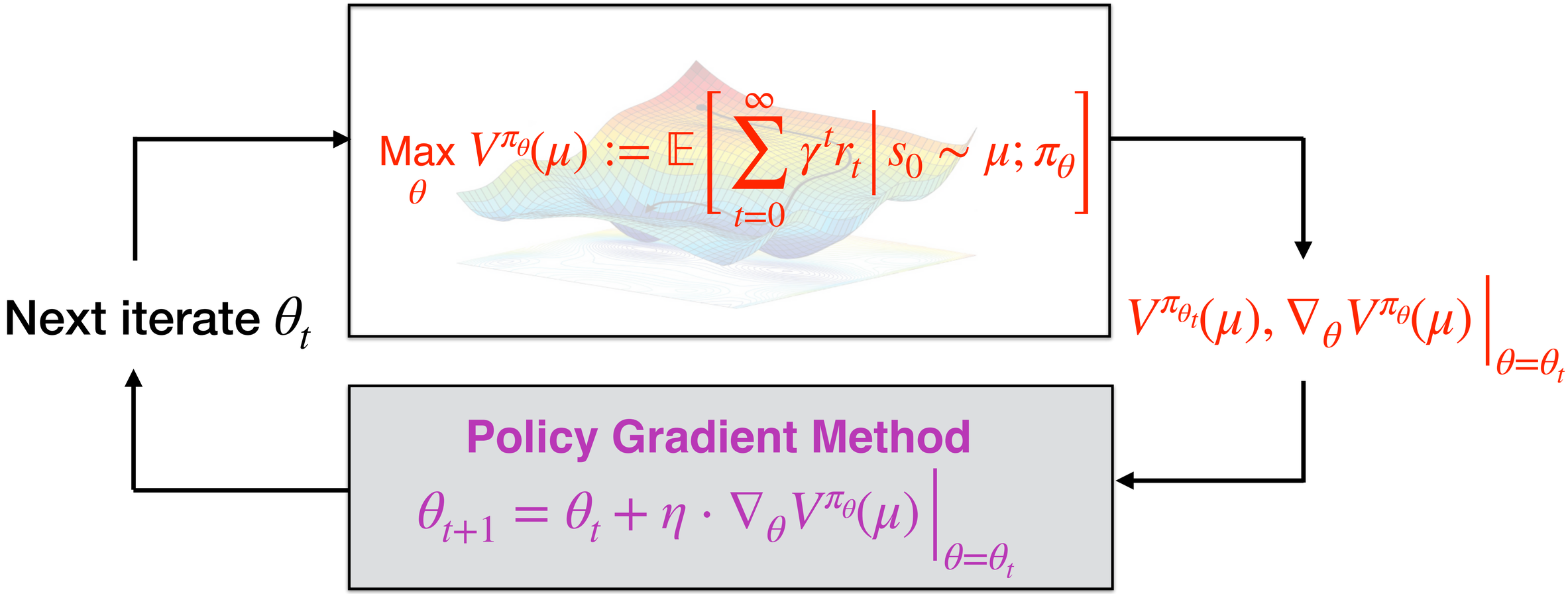
γ : discount factor



RL objective function

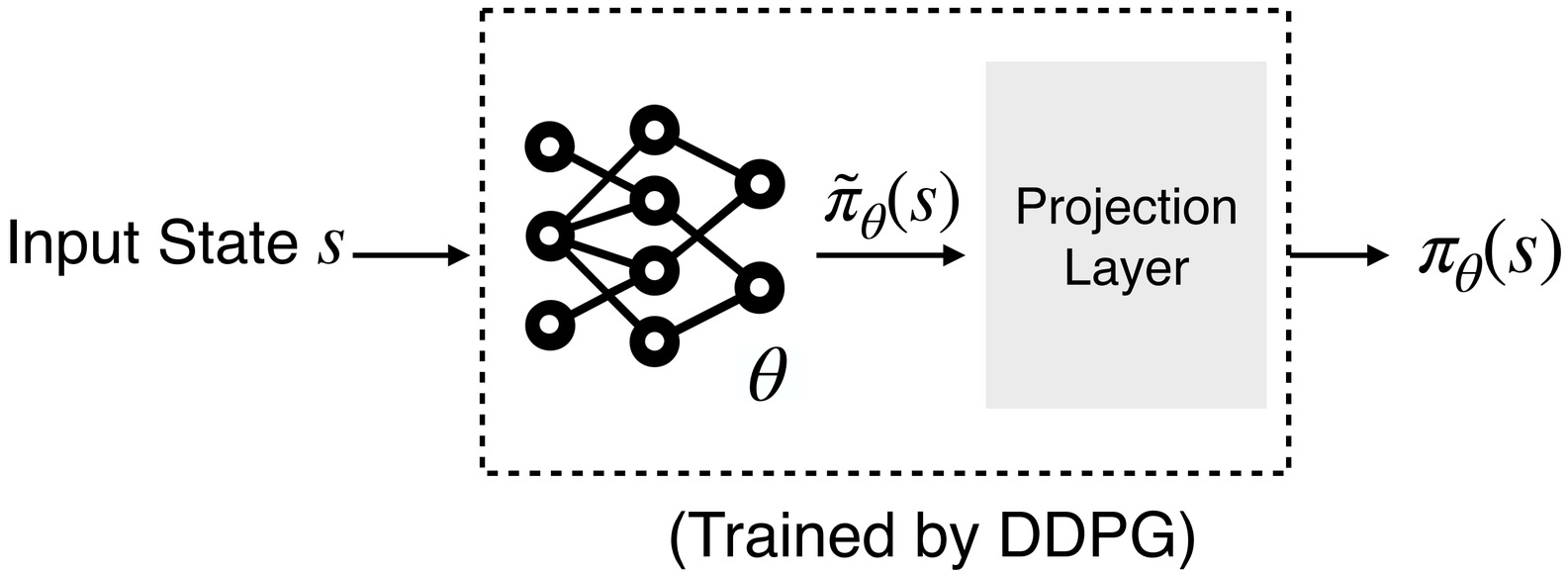
$$f(\theta) \equiv V^{\pi_\theta}(\mu) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 \sim \mu; \pi_\theta\right]$$

Policy Gradient (PG)

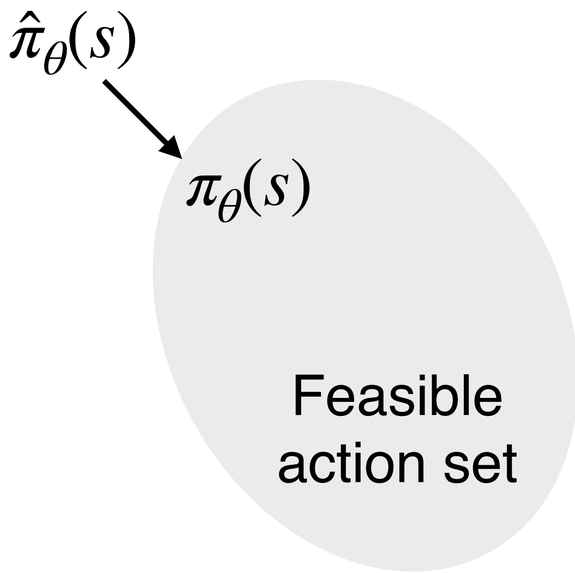


Existing Solution to Action-Constrained RL: Projection Layer

DDPG-OptLayer [Pham et al., 2018]



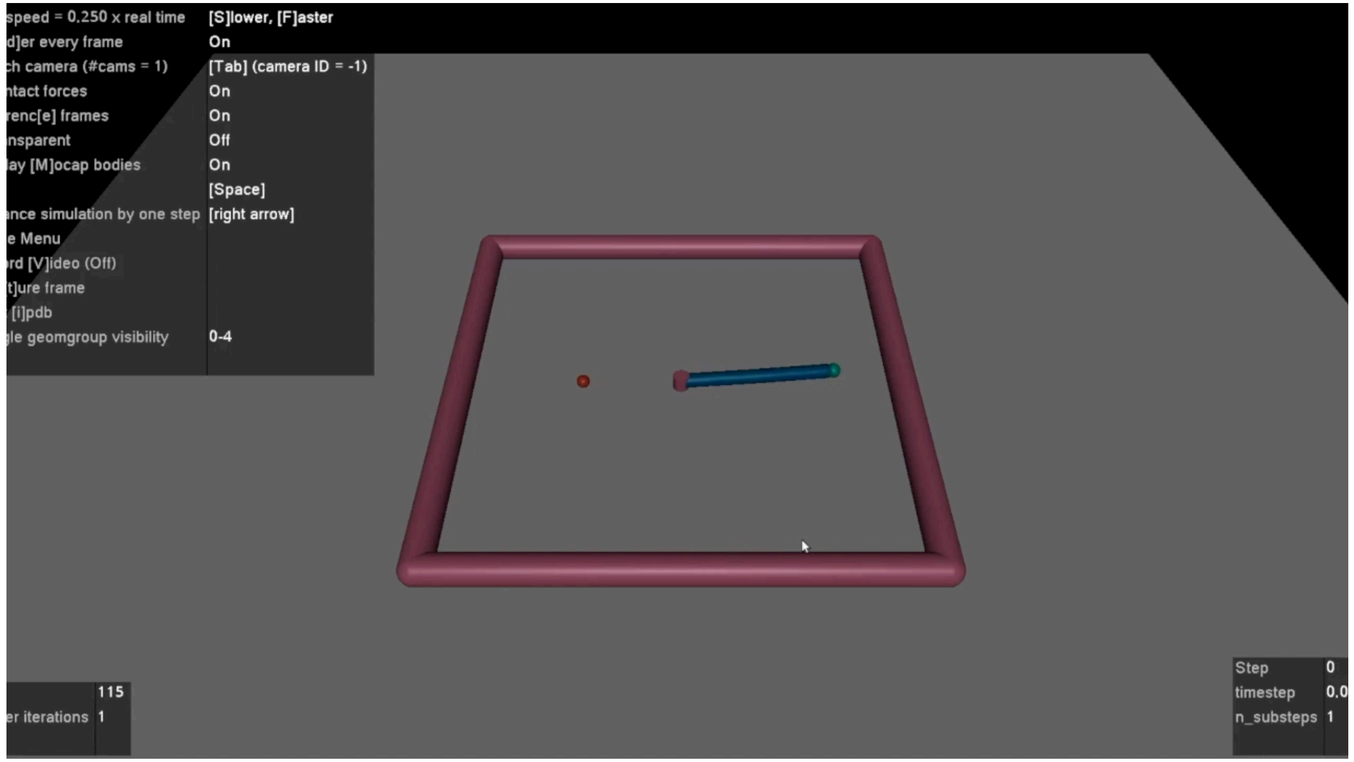
What projection layer does is:



However, this projection layer is problematic

T. Pham, et al., “Optlayer - practical constrained optimization for deep reinforcement learning in the real world,” ICRA 2018

Why is Projection Layer Problematic?
A Motivating Experiment



Zero-gradient issue!

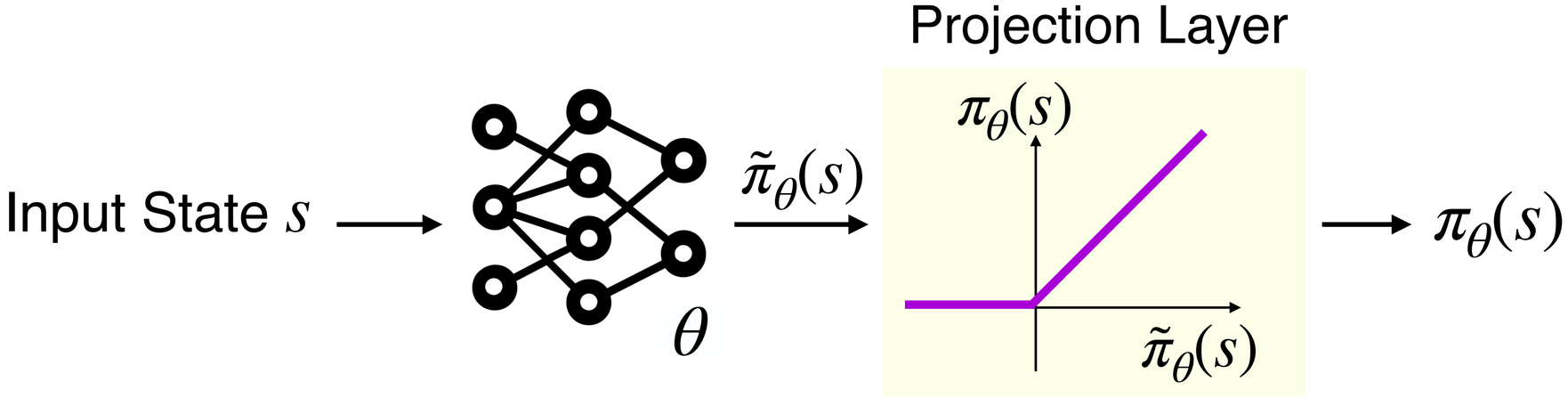
Reacher in MuJoCo

- Two joints with torques $u_1, u_2 \in [-1, 1]$
- Constraints: $|u_1 + u_2| \leq 0.1$, $u_1^2 + u_2^2 \leq 0.02$
- Algorithm: DDPG-OptLayer
- Model trained for 500k steps

What is Zero Gradient in ACRL? A Motivating Example

Suppose output actions $\pi_\theta(s)$ must be nonnegative (i.e., $\pi_\theta(s) \geq 0$)

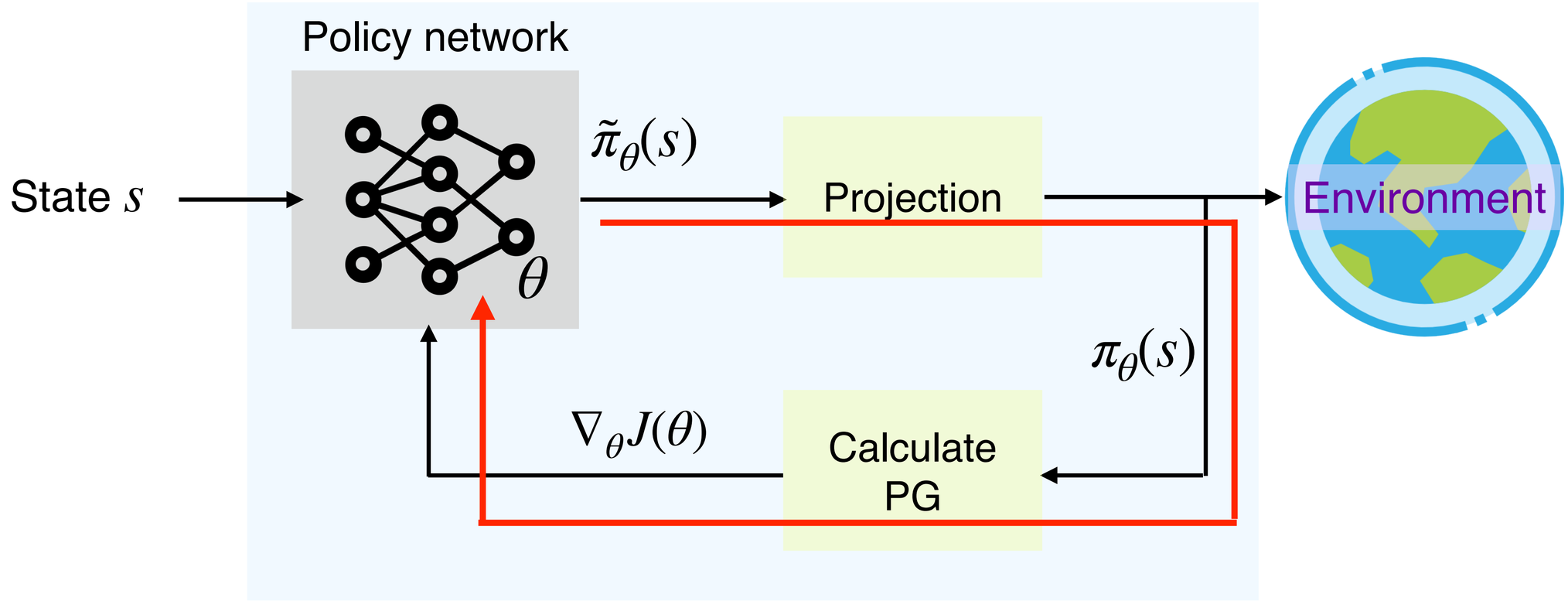
(In this case: Projection = Rectification)



When $\hat{\pi}_\theta(s) < 0$, any small perturbation on θ has no effect!

Zero gradient! (No learning progress at all)

Why DDPG+PGD Fails: Tight Coupling of PG & Projection

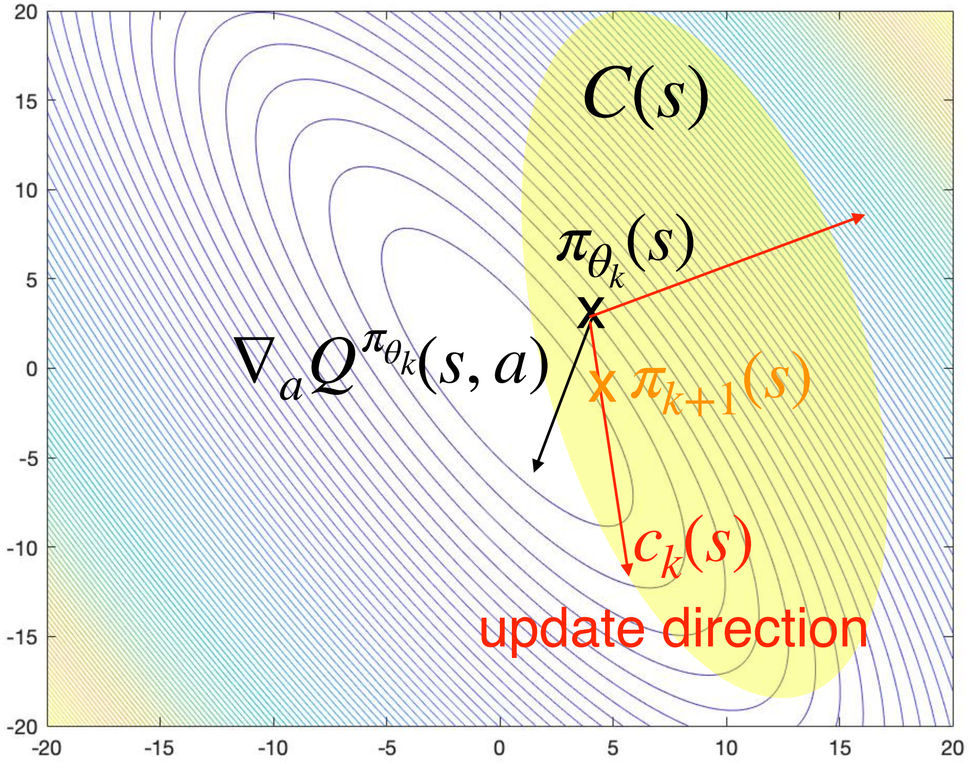


PG and Projection are in the same training loop!

We shall decouple these two components!

Frank-Wolfe Policy Optimization (FWPO)

- FWPO: Search "in the feasible set" for better actions (for each state)
  - Use state-wise Frank-Wolfe for policy improvement



Contour of $Q^{\pi_k}(s,a)$ w.r.t. action a

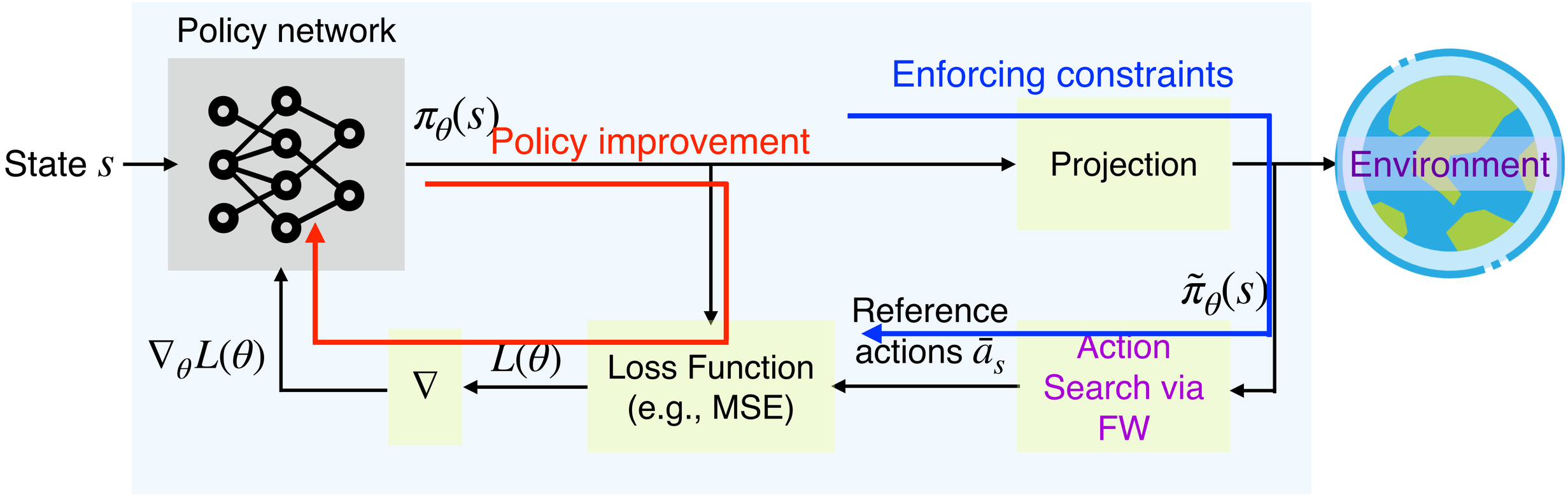
$$c_k(s) = \arg\max_{c \in C(s)} < c - \theta_k(s), \nabla_a Q(s,a;\pi_{\theta_k}) >$$

$$\theta_{k+1}(s) = \theta_k(s) + \alpha_k(s)(c_k(s) - \theta_k(s))$$

Features of FWPO:

1. No constraint violation (always stays in feasible set)
2. No projection layer needed in FWPO
3. No policy gradient in FWPO
4. Provably convergent!

Neural FWPO Framework

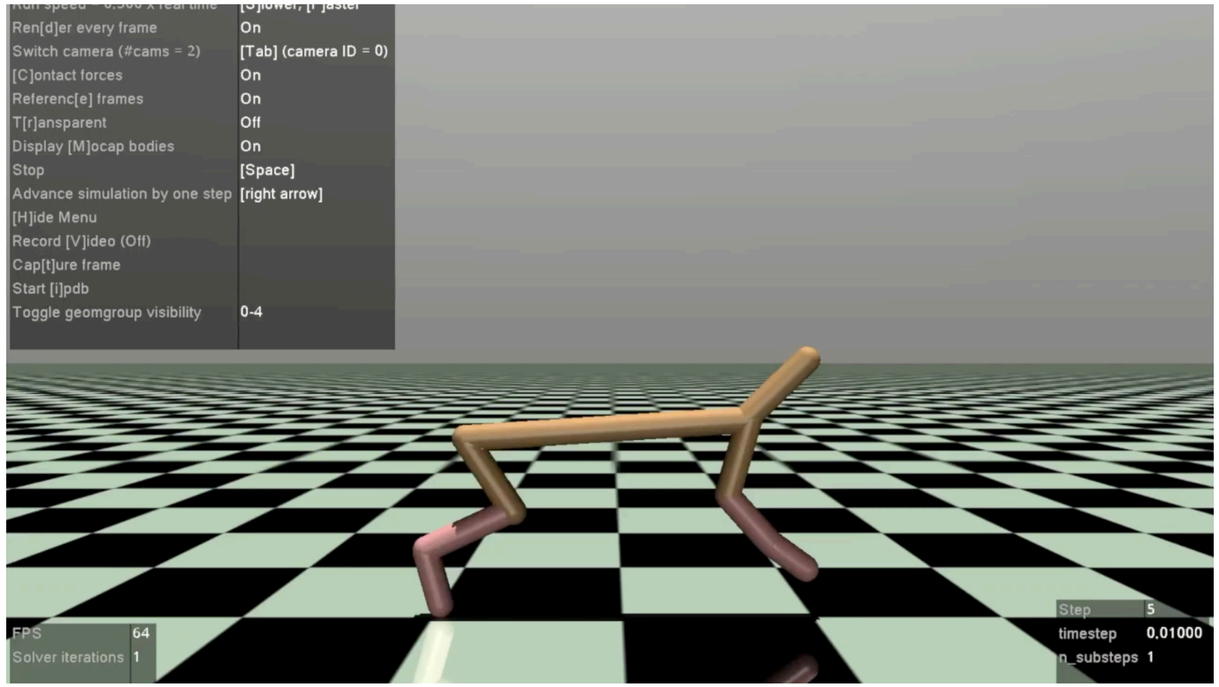


FWPO decouples policy improvement and constraint satisfaction

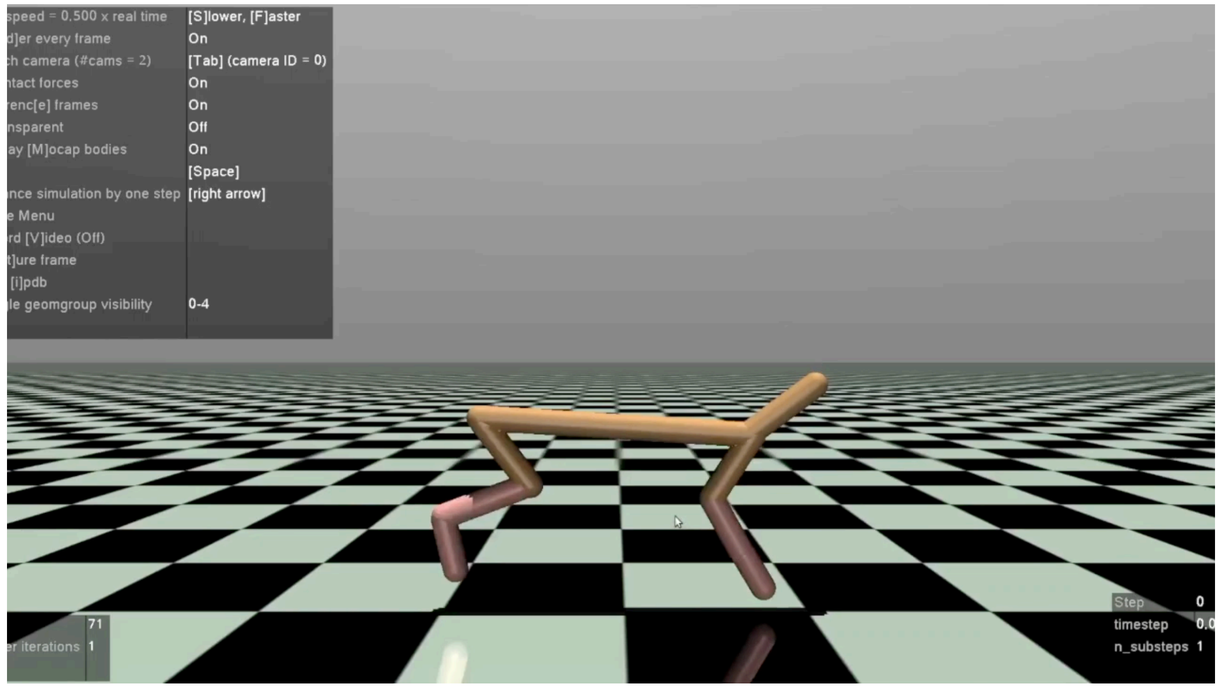
Hence, it completely avoids zero gradient!

Demo: Halfcheetah in MuJoCo

Neural FWPO



DDPG With Projection Layer



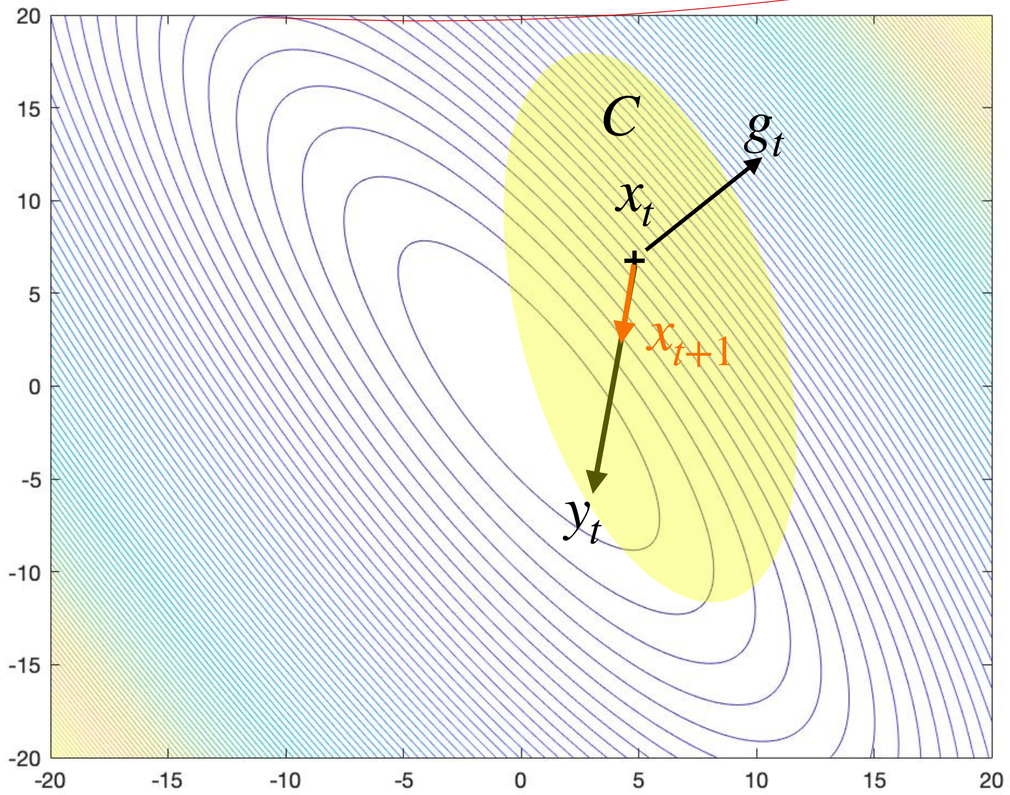
Stochastic Frank-Wolfe

Elad Hazan and Haipeng Luo, “Variance-Reduced and Projection-Free Stochastic Optimization,” ICML 2016

Extending FW to Stochastic FW (SFW)

- Suppose we focus on empirical risk minimization

$$\min_{x \in C} F(x) := \frac{1}{n} \sum_{i=1}^{n} f(x; d_i) \equiv f_i(x) \qquad (\{d_i\}_{i=1}^{n} \text{ are data samples})$$



- Under SFW, the iterates are updated as

$$y_t = \arg\min_{x \in C} g_t^\top x$$

$$x_{t+1} = (1 - \eta_t) x_t + \eta_t y_t$$

where $g_t = \frac{1}{|B_t|} \sum_{i \in B_t} \nabla f(x_t; d_i)$ is an unbiased estimate of $\nabla F(x_t)$ based on $|B_t| \equiv m_t$ i.i.d. samples from the dataset

Convergence Rate of Stochastic FW

Theorem Suppose the following conditions hold:

(1) $F(x)$ is L-smooth and convex

(2) Each $f_i(x)$ is G-Lipschitz

(3) Step size $\eta_t = \frac{2}{t+1}$ and $m_t = \left(\frac{G(t+1)}{L \cdot d_C} \right)^2$

Then, we have $E[F(x_t) - F(x^*)] \leq \frac{4 L d_C^2}{t+2} = O\left(\frac{1}{t}\right)$

Question: Any difference in convergence between SGD and SFW?

Proof of Convergence

Step 1: One-step improvement

$$F(x_{t+1}) \le F(x_t) + \nabla F(x_t)^T (x_{t+1} - x_t) + \frac{L}{2}\|x_{t+1} - x_t\|^2 \quad \cdots \text{(by } L\text{-smoothness)}$$

$$= F(x_t) + \eta_t \cdot \nabla F(x_t)^T (y_t - x_t) + \frac{L}{2}\eta_t^2 \cdot \|y_t - x_t\|^2 \quad \cdots \text{(by FW update)}$$

$$\le F(x_t) + \eta_t \cdot g_t^T (y_t - x_t) + \eta_t \cdot (\nabla F(x_t) - g_t)^T \cdot (y_t - x_t) + \frac{L d_C^2 \eta_t^2}{2} \quad \cdots \text{(Adding } g_t \text{ and subtracting } g_t \text{; Applying the diameter } d_C\text{)}$$

$$\le F(x_t) + \eta_t \cdot g_t^T (x^* - x_t) + \eta_t \cdot (\nabla F(x_t) - g_t)^T \cdot (y_t - x_t) + \frac{L d_C^2 \eta_t^2}{2} \quad \cdots \text{(by the FW update)}$$

$$= F(x_t) + \eta_t \nabla F(x_t)^T (x^* - x_t) + \eta_t (\nabla F(x_t) - g_t)(y_t - x^*) + \frac{L \cdot d_C^2 \eta_t^2}{2}$$

$$\le F(x_t) + \eta_t \left(F(x^*) - F(x_t)\right) + \eta_t \cdot \|\nabla F(x_t) - g_t\| \cdot \|y_t - x^*\| + \frac{L \cdot d_C^2 \cdot \eta_t^2}{2} \quad \cdots \text{(by Cauchy-Schwarz)}$$

Step2: By taking the expectation on both sides,

$$\mathbb{E}[F(x_{t+1})] \le \mathbb{E}[F(x_t)] + \eta_t \cdot \mathbb{E}[F(x^*) - F(x_t)]$$

$$+ \eta_t \cdot d_c \cdot \mathbb{E}[\| \nabla F(x_t) - g_t \|] + \frac{L d_c^2 \eta_t^2}{2}$$

$$\mathbb{E}[\| \nabla F(x_t) - g_t \|] \le \sqrt{\mathbb{E}[\| \nabla F(x_t) - g_t \|^2]} \le \frac{G}{\sqrt{m_t}} \le \frac{L \cdot d_c \cdot \eta_t}{2}$$

by the property of variance (why?) (why?)

Then, we have $\mathbb{E}[F(x_{t+1}) - F(x^*)] \le (1-\eta_t) \cdot \mathbb{E}[F(x_t) - F(x^*)] + L \cdot d_c^2 \cdot \eta_t^2$

(Finally, the proof can be completed by an inductive argument)

n i.i.d. sample

Variance $\sim \frac{1}{n}$

by taking the empirical average of m_t i.i.d. samples

$X_1, X_2, \cdots, X_{m_t}$ are i.i.d. random variables.

Let's define $\overline{X} = \dfrac{X_1 + X_2 + \cdots + X_{m_t}}{m_t}$

$$Var(\overline{X}) = Var\left(\frac{\overbrace{X_1 + X_2 + \cdots + X_{m_t}}^{Z}}{m_t}\right)$$

$$= \frac{1}{m_t^2} \cdot Var(Z)$$

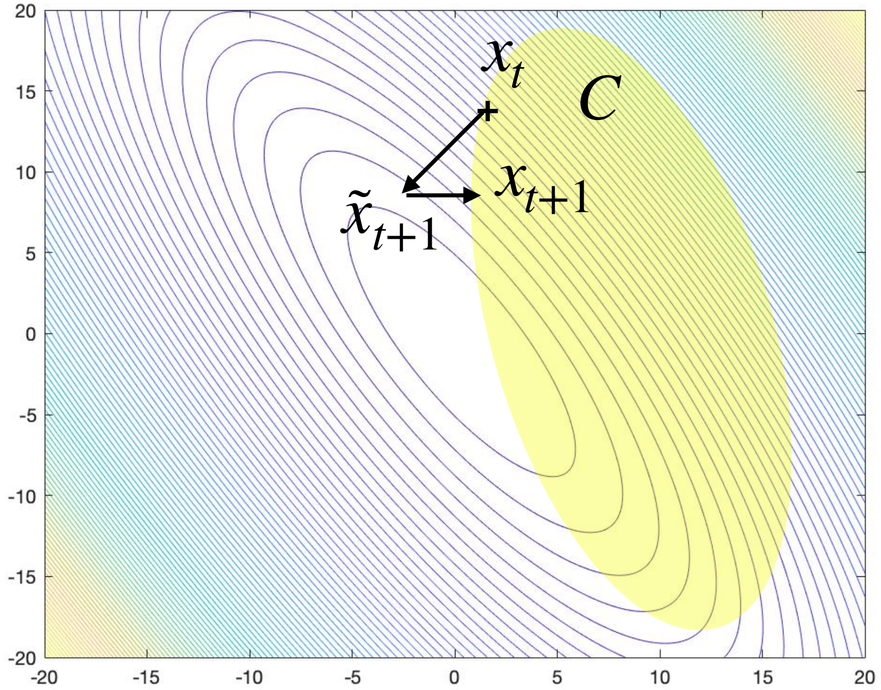
$$= \frac{1}{m_t^2} \cdot \left(Var(X_1) + \cdots + Var(X_{m_t})\right)$$

$$= \frac{1}{m_t} \cdot Var(X_1)$$

Part 2. Mirror Descent

Rethinking Projected GD

$\|v\|^2 = v^T v$



- Under PGD, the iterates are updated as

$$x_{t+1} = \Pi_C(\underbrace{x_t - \eta_t \nabla f(x_t)}_{=:\tilde{x}_{t+1}})$$

- The PGD update can be rewritten as:

$$x_{t+1} = \arg\min_{x \in C} \|x - \underbrace{(x_t - \eta_t \nabla f(x_t))}_{\tilde{x}_{t+1}}\|^2$$

$$= \arg\min_{x \in C} \{ \|x - x_t\|^2 + 2\eta_t (x - x_t)^\top \nabla f(x_t) + \eta_t^2 \|\nabla f(x_t)\|^2 \}$$

(a) (b) (c)

"penalty term"

"proximal term"

$$= \arg\min_{x \in C} \left\{ \frac{1}{2\eta_t} \|x - x_t\|^2 + (x - x_t)^\top \nabla f(x_t) \right\}$$

$$= \arg\min_{x \in C} \left\{ \frac{1}{2\eta_t} \|x - x_t\|^2 + (x - x_t)^\top \nabla f(x_t) + f(x_t) \right\}$$

1st-order approximation w.r.t. x_t

Rethinking Projected GD

- Original viewpoint: $x_{t+1} = \Pi_C(\underbrace{x_t - \eta_t \nabla f(x_t)}_{=:\tilde{x}_{t+1}})$

- "**Proximal**" viewpoint: $x_{t+1} = \arg\min_{x \in C} \{\underbrace{f(x_t) + \nabla f(x_t)^\top (x - x_t)}_{\text{first-order approximation}} + \underbrace{\frac{1}{2\eta_t}\|x - x_t\|^2}_{\text{proximal term}}\}$

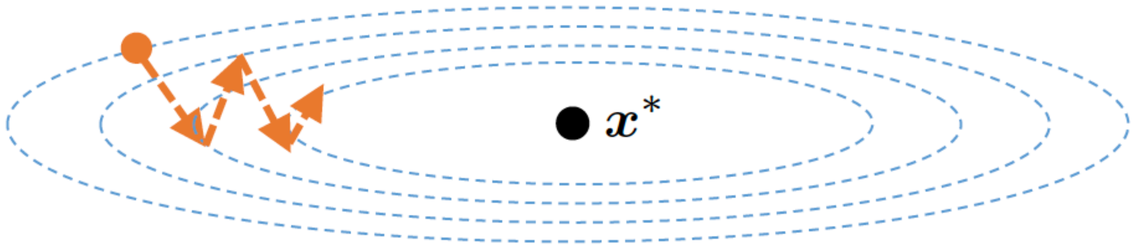
Interpretation: This L_2 proximal term is meant to capture the discrepancy between $f(x)$ and the first-order approximation

Issue: This choice of L_2 proximal term presumes that the local geometry is homogeneous / Euclidean

Question: Can we extend this idea to non-Euclidean geometry?

An Example of Non-Homogeneity

$$\text{minimize}_{x\in\mathbb{R}^2} \quad f(x) := \frac{1}{2}(x-x^*)^\top Q(x-x^*)$$

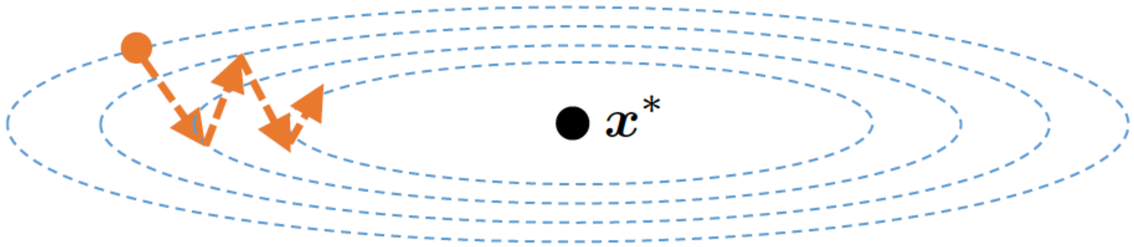


- Suppose $Q = [Q_{11}, 0; 0, Q_{22}]$ is a diagonal matrix with $Q_{11} \gg Q_{22}$

- In this example, the condition number $\kappa \gg 1$ (why?)
- Recall from Lecture 4:
  - Under GD, the convergence rate is $\|x_t - x^*\| \le \left(\frac{\kappa-1}{\kappa+1}\right)^t \|x_0 - x^*\|$
  - Equivalently, the sample complexity is $O(\kappa \log(\frac{1}{\epsilon}))$
- GD converges slowly due to large κ (as GD update does not capture the curvature well)
- Question: How to improve the convergence rate under a large κ?

An Example of Non-Homogeneity: "Scaled" Gradients

$$\text{minimize}_{x \in \mathbb{R}^2} \quad f(x) := \frac{1}{2}(x - x^*)^\top Q (x - x^*)$$



- Suppose $Q = [Q_{11}, 0; 0, Q_{22}]$ is a diagonal matrix with $Q_{11} \gg Q_{22}$

- Idea: Accelerate GD by *scaling the gradient*

$$x_{t+1} = x_t - \eta_t Q^{-1} \nabla f(x_t) = \underline{\qquad\qquad\qquad\qquad}$$

- Insight: The above "scaled GD update" is equivalent to

$$x_{t+1} = \arg\min_{x \in C} \left\{ f(x_t) + \nabla f(x_t)^\top (x - x_t) + \frac{1}{2\eta_t} \underline{(x - x_t)^\top Q (x - x_t)} \right\}$$

a scaled L_2 proximal term

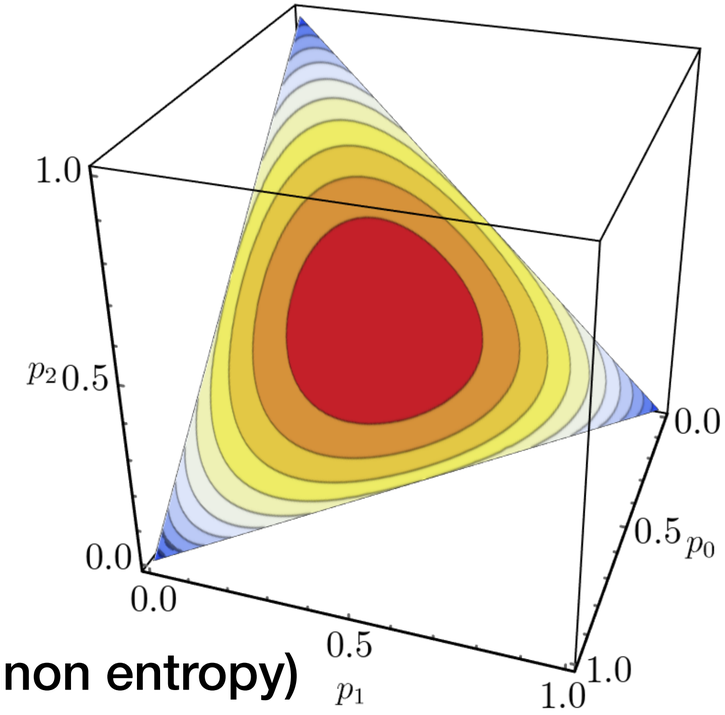
(that better captures the local geometry)

An Example of Non-Euclidean Geometry

$$\text{minimize}_{x \in C} \quad f(x)$$

$$C = \{x \in \mathbb{R}^d : \sum_{i=1}^{d} x_i = 1, x_j \geq 0, \forall j\}$$

(probability simplex)



(Contour of Shannon entropy)

- Question: Is it good to measure "the distance of two probability vectors" by L_2 norm?
- Some popular distance measures of probability vectors:
  - Kullback-Leibler (KL) divergence
  - Total variation (TV) divergence
  - Jensen-Shannon (JS) divergence
  - … and more

Mirror Descent (MD)

Key Idea: Generalize the L_2 proximal term to other metrics!

$$x_{t+1} = \arg\min_{x \in C} \left\{ \underbrace{f(x_t) + \nabla f(x_t)^\top (x - x_t)}_{\text{first-order approximation}} + \frac{1}{\eta_t} \underbrace{D_\phi(x \| x_t)}_{\text{Bregman divergence}} \right\}$$

where $D_\phi(y \| x) := \phi(y) - \phi(x) - \nabla \phi(x)^\top (y - x)$

(with respect to $\phi(\cdot)$ strictly convex and differentiable)

- Remark: Bregman divergence is meant to capture "local geometry" of objective function
- Remark: Bregman divergence is NOT symmetric and hence not a metric
- How about MD in the unconstrained cases?

Bregman Divergence (Formally)

Definition: Given a strictly convex and differentiable function $\phi : X \to \mathbb{R}$, the Bregman divergence $D_\phi(\cdot \| \cdot)$ w.r.t. to ϕ is defined as

$$D_\phi(y\|x) := \phi(y) - \phi(x) - \nabla\phi(x)^\top(y-x)$$

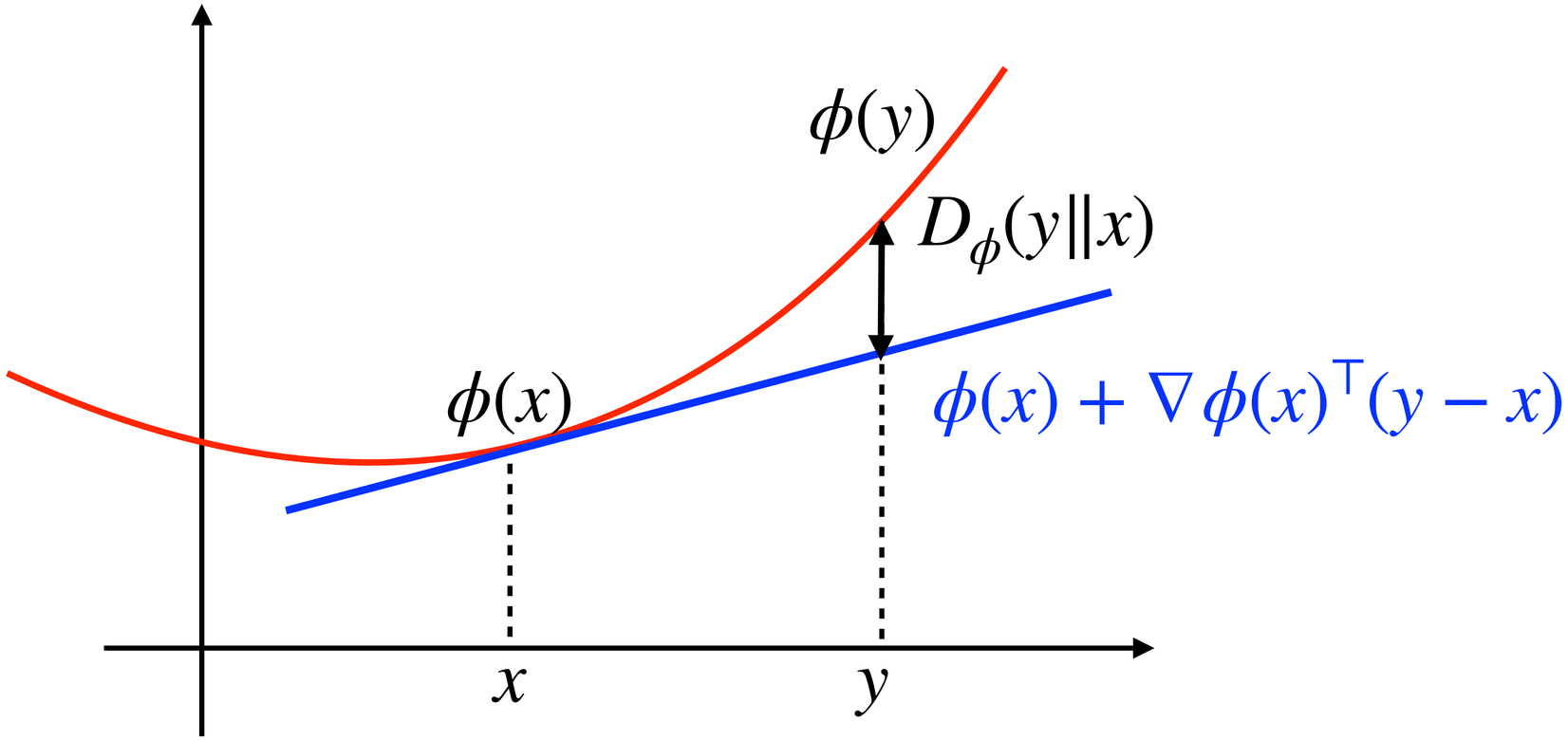
- Intuition: Connection between L_2 norm and Bregman divergence?

$$D_\phi(y\|x) = \phi(y) - \phi(x) - \nabla\phi(x)^\top(y-x) = (y-x)^\top \_\_\_\_\_ (y-x)$$

Bregman divergence is locally a scaled L_2 proximal term

Visualization of Bregman Divergence

$D_\phi(y\|x) = \phi(y) - \phi(x) - \nabla\phi(x)^\top(y-x)$, where $\phi(\cdot)$ is strictly convex



Example #1: Negative Entropy and KL Divergence

Suppose $\phi(x) = \sum_{i=1}^{d} x_i \log x_i - x_i$ (unnormalized negative entropy)

- The resulting Bregman divergence is $D_\phi(y\|x) = \sum_{i=1}^{d} y_i \log \frac{y_i}{x_i}$ (KL divergence)

Example #2: Quadratic Function and Mahalanobis Distance

Suppose $\phi(x) = x^\top A x$ (where A is a positive definite $d \times d$ matrix)

- The resulting Bregman divergence is $D_\phi(y\|x) = (x-y)^\top A(x-y)$

(aka “Mahalanobis distance”)

Example #3: Negative Logarithm and Itakura-Saito Distance

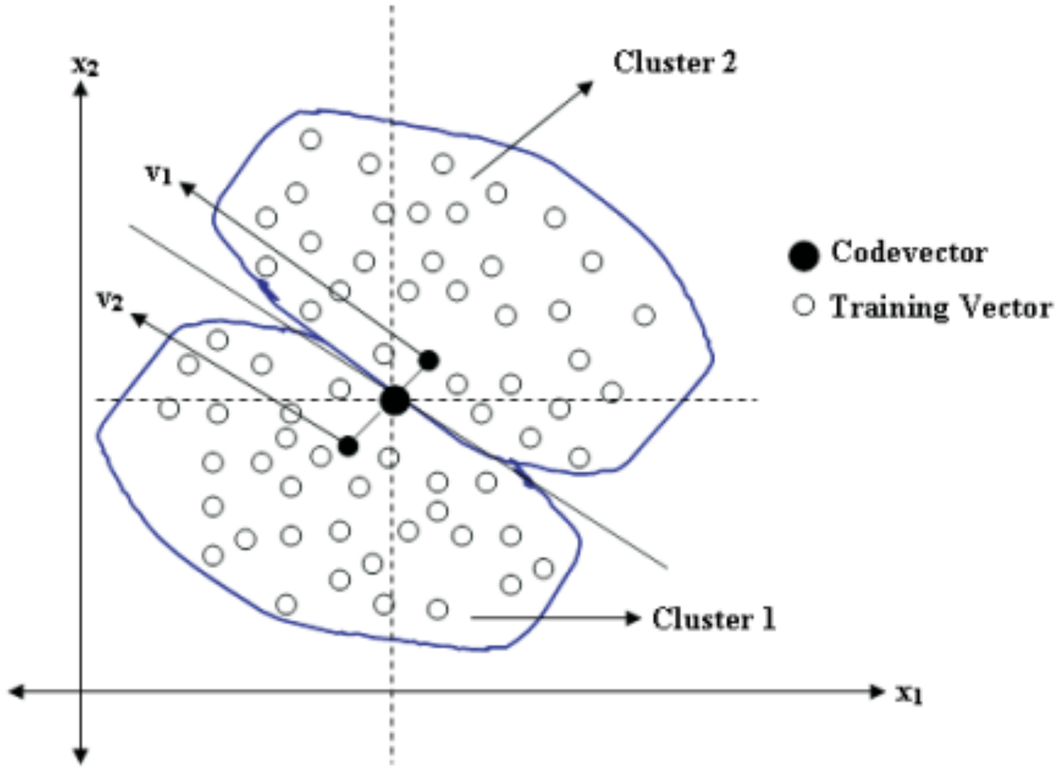
Suppose $\phi(x) = -\log x$

- The resulting Bregman divergence is $D_\phi(y\|x) = \frac{x}{y} - \log\frac{x}{y} - 1$

(aka “Itakura-Saito distance”)

Itakura-Saito Distance

- Itakura-Saito distance is widely used in vector quantization (Linde, Buzo, Gray, 1980)
  - K-means-like clustering in the frequency domain
  - Especially for speech signal coding



Y. Linde, A. Buzo, and R. Gray, “An algorithm for vector quantizer design,” IEEE Transactions on Communications, 1980

H. B. Kekrea, Prachi Natub, and Tanuja Sarodec, “Color Image Compression Using Vector Quantization and Hybrid Wavelet Transform,” Procedia Computer Science, 2016

Popular Choices of Bregman Divergence

Domain	$\phi(\mathbf{x})$	$d_\phi(\mathbf{x},\mathbf{y})$	**Divergence**
$\mathbb{R}$	x^2	$(x-y)^2$	Squared loss
$\mathbb{R}_+$	$x\log x$	$x\log(\frac{x}{y})-(x-y)$	
$[0,1]$	$x\log x+(1-x)\log(1-x)$	$x\log(\frac{x}{y})+(1-x)\log(\frac{1-x}{1-y})$	Logistic loss [3]
$\mathbb{R}_{++}$	$-\log x$	$\frac{x}{y}-\log(\frac{x}{y})-1$	Itakura-Saito distance
$\mathbb{R}$	e^x	$e^x-e^y-(x-y)e^y$	
$\mathbb{R}^d$	$\|\mathbf{x}\|^2$	$\|\mathbf{x}-\mathbf{y}\|^2$	Squared Euclidean distance
$\mathbb{R}^d$	$\mathbf{x}^T A\mathbf{x}$	$(\mathbf{x}-\mathbf{y})^T A(\mathbf{x}-\mathbf{y})$	Mahalanobis distance [4]
d-Simplex	$\sum_{j=1}^d x_j\log_2 x_j$	$\sum_{j=1}^d x_j\log_2(\frac{x_j}{y_j})$	KL-divergence
$\mathbb{R}_+^d$	$\sum_{j=1}^d x_j\log x_j$	$\sum_{j=1}^d x_j\log(\frac{x_j}{y_j})-\sum_{j=1}^d(x_j-y_j)$	Generalized I-divergence

Function name	$\varphi(x)$	dom φ	$D_\varphi(x;y)$
Squared norm	$\frac{1}{2}x^2$	$(-\infty,+\infty)$	$\frac{1}{2}(x-y)^2$
Shannon entropy	$x\log x-x$	$[0,+\infty)$	$x\log\frac{x}{y}-x+y$
Bit entropy	$x\log x+(1-x)\log(1-x)$	$[0,1]$	$x\log\frac{x}{y}+(1-x)\log\frac{1-x}{1-y}$
Burg entropy	$-\log x$	$(0,+\infty)$	$\frac{x}{y}-\log\frac{x}{y}-1$
Hellinger	$-\sqrt{1-x^2}$	$[-1,1]$	$(1-xy)(1-y^2)^{-1/2}-(1-x^2)^{1/2}$
ℓ_p quasi-norm	$-x^p \quad (0<p<1)$	$[0,+\infty)$	$-x^p+p\,xy^{p-1}-(p-1)\,y^p$
ℓ_p norm	$\|x\|^p \quad (1<p<\infty)$	$(-\infty,+\infty)$	$\|x\|^p-p\,x\,\operatorname{sgn}y\,\|y\|^{p-1}+(p-1)\,\|y\|^p$
Exponential	$\exp x$	$(-\infty,+\infty)$	$\exp x-(x-y+1)\exp y$
Inverse	$1/x$	$(0,+\infty)$	$1/x+x/y^2-2/y$

A. Banerjee et al., “Clustering with Bregman Divergences,” JMLR 2005

I. Dillon and J. Tropp, “Matrix Nearness Problems with Bregman Divergences,” SIAM Journal on Matrix Analysis and Applications, 2008

Applications: PPO-KL/TRPO as Mirror Descent with KL Divergence

$$\max_{\theta} \mathbb{E}_{s\sim d_{\mu}^{\pi_{\theta_k}}, a\sim\pi_{\theta_k}(\cdot|s)}\left[\frac{\pi_{\theta}(a\,|\,s)}{\pi_{\theta_k}(a\,|\,s)}A^{\theta_k}(s,a) - \beta_k D_{KL}(\pi_{\theta_k}(\,\cdot\,|\,s)\,||\,\pi_{\theta}(\,\cdot\,|\,s))\right]$$

Neural Proximal/Trust Region Policy Optimization Attains Globally Optimal Policy

Boyi Liu[*†] **Qi Cai**[*‡] **Zhuoran Yang**[§] **Zhaoran Wang**[¶]

Abstract

Proximal policy optimization and trust region policy optimization (PPO and TRPO) with actor and critic parametrized by neural networks achieve significant empirical success in deep reinforcement learning. However, due to nonconvexity, the global convergence of PPO and TRPO remains less understood, which separates theory from practice. In this paper, we prove that a variant of PPO and TRPO equipped with overparametrized neural networks converges to the globally optimal policy at a sublinear rate. The key to our analysis is the global convergence of infinite-dimensional mirror descent under a notion of one-point monotonicity, where the gradient and iterate are instantiated by neural networks. In particular, the desirable representation power and optimization geometry induced by the overparametrization of such neural networks allow them to accurately approximate the infinite-dimensional gradient and iterate.

[NeurIPS 2019]

Adaptive Trust Region Policy Optimization: Global Convergence and Faster Rates for Regularized MDPs

Lior Shani[†], **Yonathan Efroni**[†], **Shie Mannor**
[†] equal contribution
Technion - Israel Institute of Technology
Haifa, Israel

[AAAI 2020]

Abstract

Trust region policy optimization (TRPO) is a popular and empirically successful policy search algorithm in Reinforcement Learning (RL) in which a surrogate problem, that restricts consecutive policies to be 'close' to one another, is iteratively solved. Nevertheless, TRPO has been considered a heuristic algorithm inspired by Conservative Policy Iteration (CPI). We show that the adaptive scaling mechanism used in TRPO is in fact the natural "RL version" of traditional trust-region methods from convex analysis. We first analyze TRPO in the planning setting, in which we have access to the model and the entire state space. Then, we consider sample-based TRPO and establish $\tilde{O}(1/\sqrt{N})$ convergence rate to the global optimum. Importantly, the adaptive scaling mechanism allows us to analyze TRPO in *regularized MDPs* for which we prove fast rates of $\tilde{O}(1/N)$, much like results in convex optimization. This is the first result in RL of better rates when regularizing the instantaneous cost or reward.

In spite of their popularity, much less is understood in terms of their convergence guarantees and they are considered heuristics (Schulman et al. 2015; Papini, Pirotta, and Restelli 2019) (see Figure 1).

TRPO and Regularized MDPs: Trust region methods are often used in conjunction with regularization. This is commonly done by adding the negative entropy to the instantaneous cost (Nachum et al. 2017; Schulman et al. 2017). The intuitive justification for using entropy regularization is that it induces inherent exploration (Fox, Pakman, and Tishby 2016), and the advantage of 'softening' the Bellman equation (Chow, Nachum, and Ghavamzadeh 2018; Dai et al. 2018). Recently, Ahmed et al. (2019) empirically observed that adding entropy regularization results in a smoother objective which in turn leads to faster convergence when the learning rate is chosen more aggressively. Yet, to the best of our knowledge, there is no finite-sample analysis

Basic Properties of Bregman Divergence

Let $\phi : X \to \mathbb{R}$ be a strictly convex and differentiable function

- 1. Non-negativity: $D_\phi(y\|x) \geq 0$
- 2. Distance between a point to itself is zero: $D_\phi(y\|x) = 0$ if and only if $x = y$
- 3. Convexity: $D_\phi(y\|x) = 0$ is convex in y (but not necessarily in x)
- 4. Symmetry not guaranteed: $D_\phi(y\|x) \neq D_\phi(x\|y)$ in general

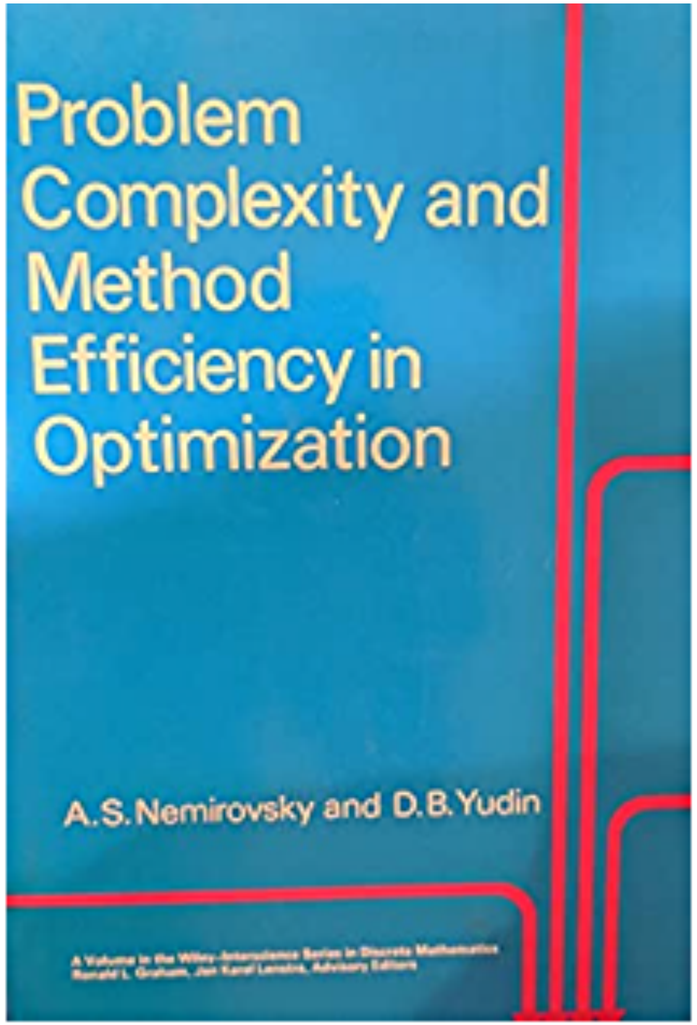
More properties of Mirror Descent will be discussed at the convergence analysis :)

Question: Why using this name “Mirror Descent”?

Mirror Descent involves the “mirror map”

A. Nemirovski and D. Yudin, ”Problem complexity and method efficiency in optimization,” Wiley, 1983.

Arkadi Nemirovski
(Professor @ Georgia Tech)



Warm-Up: MD Algorithm for Unconstrained Problems

- Mirror Descent (unconstrained cases):

$$x_{t+1} = \arg\min_{x \in \mathbb{R}^d} \left\{ \underbrace{f(x_t) + \nabla f(x_t)^\top (x - x_t)}_{\text{first-order approximation}} + \frac{1}{\eta_t} \underbrace{D_\phi(x \| x_t)}_{\text{Bregman divergence}} \right\}$$

- An Equivalent Update Rule of MD:

$$x_{t+1} = \nabla \phi^{-1}\big(\nabla \phi(x_t) - \eta_t \nabla f(x_t) \big)$$

- Does $\nabla \phi^{-1}(\,\cdot\,)$ always exist?
- Which update rule is easier to apply?

Derivation of Equivalent MD Update (Unconstrained Cases)

$$x_{t+1} = \operatorname*{argmin}_{x \in \mathbb{R}^d} \underbrace{\left\{ \nabla f(x_t)^\top (x - x_t) + \frac{1}{\eta_t} D_\phi(x \| x_t) \right\}}_{=: h(x)} \iff x_{t+1} = \nabla \phi^{-1}\left(\nabla \phi(x_t) - \eta_t \cdot \nabla f(x_t) \right)$$

---

1. By the necessary condition of optimality, we have

$$\nabla h(x_{t+1}) = 0, \quad \text{i.e.,} \quad \nabla f(x_t) + \frac{1}{\eta_t}\left(\nabla \phi(x_{t+1}) - \nabla \phi(x_t) \right) = 0$$

2. Equivalently, we have

$$\nabla \phi(x_{t+1}) = \nabla \phi(x_t) - \eta_t \cdot \nabla f(x_t)$$

Examples of MD Algorithms (Unconstrained cases)

$$x_{t+1} = \nabla\phi^{-1}\left(\nabla\phi(x_t) - \eta_t \cdot \nabla f(x_t)\right)$$

Example 1: $\phi(x) = \frac{1}{2}\|x\|^2$

- $\nabla\phi(x) = x$
- $\nabla\phi^{-1}(z) =$ _____

$$x_{t+1} = \nabla\phi(x_t) - \eta_t \cdot \nabla f(x_t)$$

$$= x_t - \eta_t \nabla f(x_t)$$

Example 2: $\phi(x) = \sum_{i=1}^{d} x_i \log x_i - x_i$

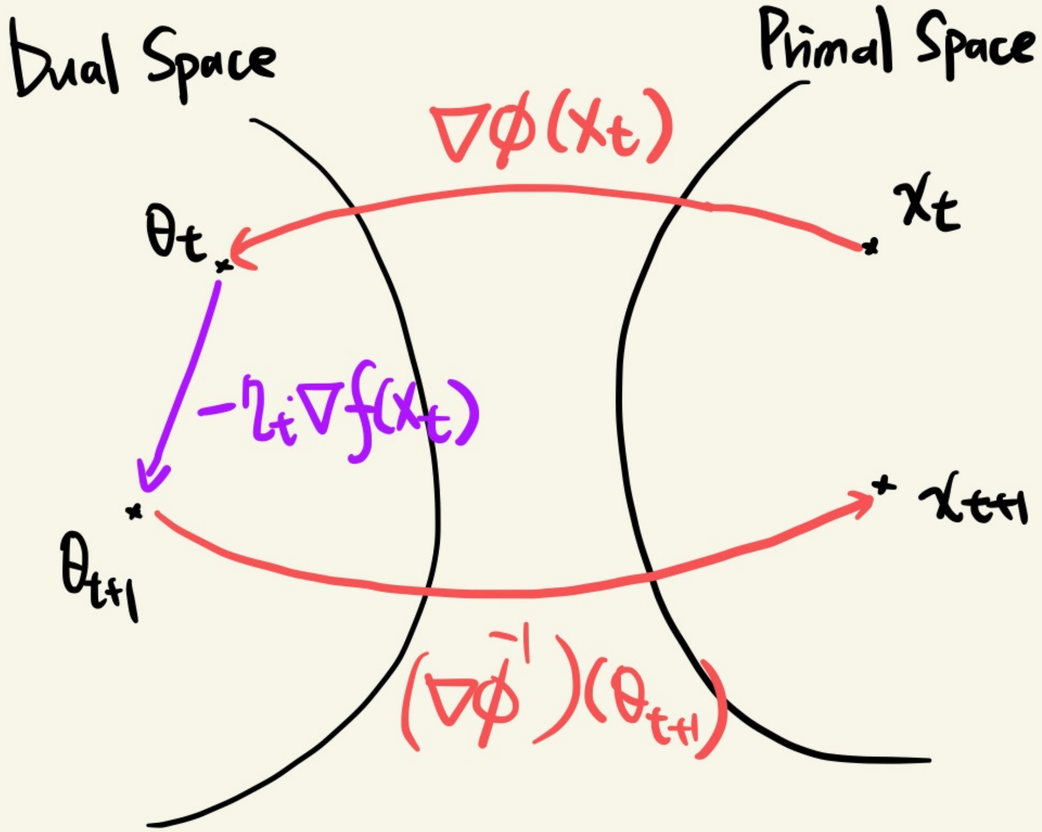
- $\nabla\phi(x) = (\log x_1, \cdots, \log x_d)$
- $\nabla\phi^{-1}(z) =$ _____

$$(x_{t+1})_i = \exp\left(\log(x_t)_i - \eta_t (\nabla f(x_t))_i\right)$$

$$= (x_t)_i \cdot \exp\left(-\eta_t (\nabla f(x_t))_i\right)$$

aka "exponentiated gradient"

Mirror Map Viewpoint (Nemirovski & Yudin, 1983)



Step 1: Map x_t to the "dual space"

$$\theta_t \leftarrow \nabla\phi(x_t)$$

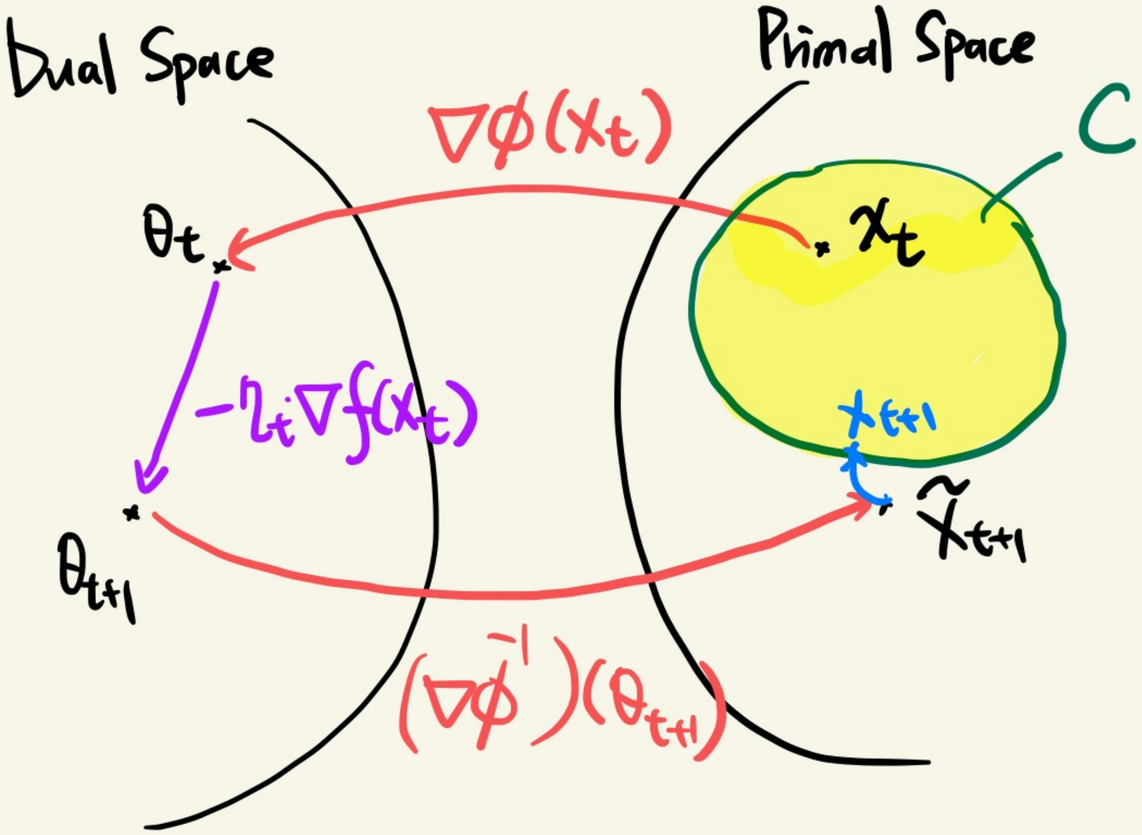
Step 2: Take a gradient step in the "dual space":

$$\theta_{t+1} \leftarrow \theta_t - \eta_t \cdot \nabla f_t(x_t)$$

Step 3: Map θ_{t+1} back to the "primal space":

$$\tilde{x}_{t+1} \leftarrow \nabla\phi^{-1}(\theta_{t+1})$$

Mirror Map Viewpoint (Nemirovski & Yudin, 1983)



Step1: Map x_t to the "dual space"

$$\theta_t \leftarrow \nabla\phi(x_t)$$

Step2: Take a gradient step in the "dual space":

$$\theta_{t+1} \leftarrow \theta_t - \eta_t \cdot \nabla f_t(x_t)$$

Step3: Map θ_{t+1} back to the "primal space":

$$\tilde{x}_{t+1} \leftarrow \nabla\phi^{-1}(\theta_{t+1})$$

Step4: Bregman Projection

$$x_{t+1} \leftarrow \min_{x \in C} D_\phi(x \| \tilde{x}_{t+1})$$

Equivalence Between "Mirror Map Viewpoint" and "Proximal Viewpoint"

$$x_{t+1} = \operatorname*{argmin}_{x \in C} D_\phi(x \| \tilde{x}_{t+1}) \quad \cdots\cdots \quad (\quad)$$

$$= \operatorname*{argmin}_{x \in C} \phi(x) - \phi(\tilde{x}_{t+1}) - \nabla\phi(\tilde{x}_{t+1})^T (x - \tilde{x}_{t+1}) \cdots \quad (\quad)$$

$$= \operatorname*{argmin}_{x \in C} \phi(x) - \nabla\phi(\tilde{x}_{t+1})^T x \quad \cdots \quad (\quad)$$

$$= \operatorname*{argmin}_{x \in C} \phi(x) - (\nabla\phi(x_t) - \eta_t \cdot \nabla f(x_t))^T x \quad \cdots \quad (\quad)$$

$$= \operatorname*{argmin}_{x \in C} \eta_t \cdot \nabla f(x_t)^T x + D_\phi(x \| x_t) \quad \cdots \quad (\quad)$$