Robustness in Machine Learning and Optimization: A Minmax Approach

Asu Ozdaglar
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology

joint work with
Farzan Farnia, Sarath Pattathil (MIT) and Aryan Mokhtari (UT Austin)
Noah Golowich and Costis Daskalakis (MIT)

UT Austin ECE Distinguished Seminar, MIT
November, 2020

Minmax Problems

- We are interested in the following minmax problem:

$$\min_{\mathbf{x}\in\mathbb{R}^m} \max_{\mathbf{y}\in\mathbb{R}^n} f(\mathbf{x},\mathbf{y})$$

These arise in a multitude of applications:

- Worst-case design (robust optimization): We view $\mathbf{y}$ as a parameter and minimize over $\mathbf{x}$ a cost function, assuming the worst possible value of $\mathbf{y}$.
- Duality theory for constrained optimization:
  - Primal problem:

$$\min_{\mathbf{x}\in\mathbb{R}^m} f(\mathbf{x}) \qquad \text{s.t. } g_j(\mathbf{x}) \le 0, \qquad j = 1, \cdots, n$$

  - Given a vector $\mu = (\mu_1, \cdots, \mu_n)$, Lagrangian function is given by:

$$\mathcal{L}(\mathbf{x},\mu) = f(\mathbf{x}) + \sum_{j=1}^{n} \mu_j g_j(\mathbf{x})$$

  - The dual problem is given by:

$$\max_{\mu\ge 0} \min_{\mathbf{x}\in\mathbb{R}^m} \mathcal{L}(\mathbf{x},\mu)$$

  - Thus the dual problem is a minmax problem.

Computing Saddle points

- We are interested in finding saddle points of the function f, i.e., points $(\mathbf{x}^*, \mathbf{y}^*) \in \mathbb{R}^m \times \mathbb{R}^n$ which satisfy:

$$f(\mathbf{x}^*, \mathbf{y}) \leq f(\mathbf{x}^*, \mathbf{y}^*) \leq f(\mathbf{x}, \mathbf{y}^*) \quad \text{for all } \mathbf{x} \in \mathbb{R}^m, \mathbf{y} \in \mathbb{R}^n.$$

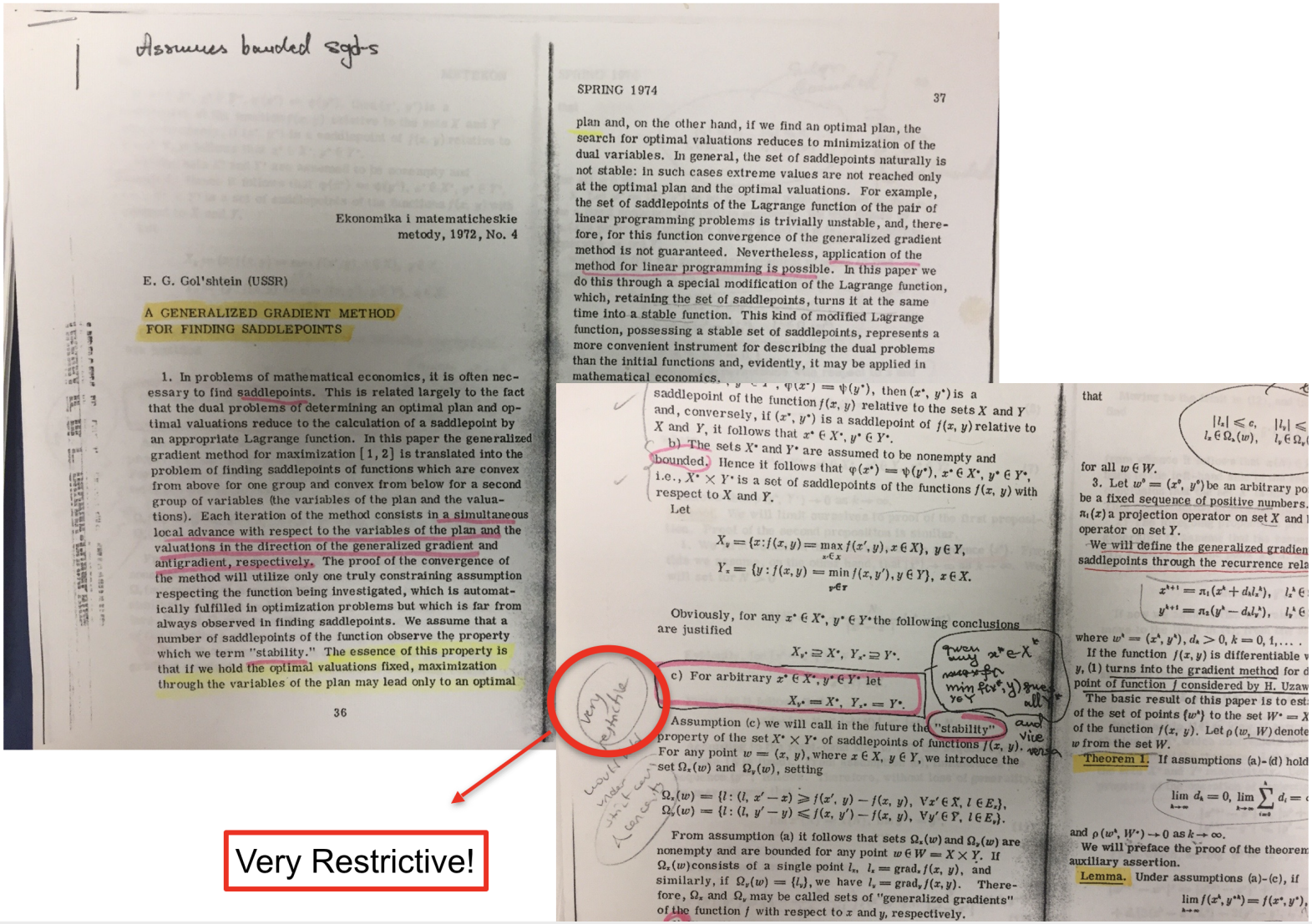
- Standard method for computing saddle points: Gradient Descent Ascent (here $\eta \geq 0$ is a fixed stepsize)

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - \eta \nabla_{\mathbf{x}} f(\mathbf{x}_k, \mathbf{y}_k), \\ \mathbf{y}_{k+1} &= \mathbf{y}_k + \eta \nabla_{\mathbf{y}} f(\mathbf{x}_k, \mathbf{y}_k). \end{aligned}$$

- Some history:
  - [Arrow, Hurwicz, Uzawa 58] proposed CT versions of these methods for convex-concave functions and proved global stability results under strict convexity assumptions.
  - [Uzawa 58] focused on a DT version and showed convergence to a neighborhood under strong convexity assumptions.
  - [Gol'shtein 74] and [Maistroskii 77] provided convergence with diminishing stepsize rules under stability assumptions (weaker than strong convexity).

Earlier Literature

Much of the focus on strong convexity-concavity type assumptions.



Assumes bounded sets

Ekonomika i matematicheskie metody, 1972, No. 4

E. G. Gol'shtein (USSR)

A GENERALIZED GRADIENT METHOD FOR FINDING SADDLEPOINTS

1. In problems of mathematical economics, it is often necessary to find saddlepoints. This is related largely to the fact that the dual problems of determining an optimal plan and optimal valuations reduce to the calculation of a saddlepoint by an appropriate Lagrange function. In this paper the generalized gradient method for maximization [1, 2] is translated into the problem of finding saddlepoints of functions which are convex from above for one group and convex from below for a second group of variables (the variables of the plan and the valuations). Each iteration of the method consists in a simultaneous local advance with respect to the variables of the plan and the valuations in the direction of the generalized gradient and antigradient, respectively. The proof of the convergence of the method will utilize only one truly constraining assumption respecting the function being investigated, which is automatically fulfilled in optimization problems but which is far from always observed in finding saddlepoints. We assume that a number of saddlepoints of the function observe the property which we term "stability." The essence of this property is that if we hold the optimal valuations fixed, maximization through the variables of the plan may lead only to an optimal

36

SPRING 1974 37

plan and, on the other hand, if we find an optimal plan, the search for optimal valuations reduces to minimization of the dual variables. In general, the set of saddlepoints naturally is not stable: in such cases extreme values are not reached only at the optimal plan and the optimal valuations. For example, the set of saddlepoints of the Lagrange function of the pair of linear programming problems is trivially unstable, and, therefore, for this function convergence of the generalized gradient method is not guaranteed. Nevertheless, application of the method for linear programming is possible. In this paper we do this through a special modification of the Lagrange function, which, retaining the set of saddlepoints, turns it at the same time into a stable function. This kind of modified Lagrange function, possessing a stable set of saddlepoints, represents a more convenient instrument for describing the dual problems than the initial functions and, evidently, it may be applied in mathematical economics.

saddlepoint of the function $f(x, y)$ relative to the sets X and Y and, conversely, if (x^*, y^*) is a saddlepoint of $f(x, y)$ relative to X and Y, it follows that $x^* \in X^*$, $y^* \in Y^*$.

b) The sets X^* and Y^* are assumed to be nonempty and bounded. Hence it follows that $\varphi(x^*) = \psi(y^*)$, $x^* \in X^*$, $y^* \in Y^*$, i.e., $X^* \times Y^*$ is a set of saddlepoints of the functions $f(x, y)$ with respect to X and Y.

Let

$$X_y = \{x : f(x, y) = \max_{x' \in X} f(x', y),\ x \in X\},\ y \in Y,$$
$$Y_x = \{y : f(x, y) = \min_{y' \in Y} f(x, y'),\ y \in Y\},\ x \in X.$$

Obviously, for any $x^* \in X^*$, $y^* \in Y^*$ the following conclusions are justified

$$X_{y^*} \supseteq X^*,\ Y_{x^*} \supseteq Y^*.$$

c) For arbitrary $x^* \in X^*$, $y^* \in Y^*$ let

$$X_{y^*} = X^*,\ Y_{x^*} = Y^*.$$

Assumption (c) we will call in the future the "stability" property of the set $X^* \times Y^*$ of saddlepoints of functions $f(x, y)$. For any point $w = (x, y)$, where $x \in X$, $y \in Y$, we introduce the set $\Omega_x(w)$ and $\Omega_y(w)$, setting

$$\Omega_x(w) = \{l : (l, x' - x) \geqslant f(x', y) - f(x, y),\ \forall x' \in X,\ l \in E_x\},$$
$$\Omega_y(w) = \{l : (l, y' - y) \leqslant f(x, y') - f(x, y),\ \forall y' \in Y,\ l \in E_y\}.$$

From assumption (a) it follows that sets $\Omega_x(w)$ and $\Omega_y(w)$ are nonempty and are bounded for any point $w \in W = X \times Y$. If $\Omega_x(w)$ consists of a single point l_x, $l_x = \operatorname{grad}_x f(x, y)$, and similarly, if $\Omega_y(w) = \{l_y\}$, we have $l_y = \operatorname{grad}_y f(x, y)$. Therefore, Ω_x and Ω_y may be called sets of "generalized gradients" of the function f with respect to x and y, respectively.

that

$$|l_x| \leqslant c,\quad l_x \in \Omega_x(w), \qquad |l_y| \leqslant\ l_y \in \Omega_y($$

for all $w \in W$.

3. Let $w^0 = (x^0, y^0)$ be an arbitrary po
be a fixed sequence of positive numbers.
$\pi_1(x)$ a projection operator on set X and
operator on set Y.

We will define the generalized gradien
saddlepoints through the recurrence rela

$$x^{k+1} = \pi_1(x^k + d_k l_x^k),\quad l_x^k \in$$
$$y^{k+1} = \pi_2(y^k - d_k l_y^k),\quad l_y^k \in$$

where $w^k = (x^k, y^k)$, $d_k > 0$, $k = 0, 1, \ldots$.

If the function $f(x, y)$ is differentiable
y, (1) turns into the gradient method for d
point of function f considered by H. Uzaw

The basic result of this paper is to est
of the set of points $\{w^k\}$ to the set $W^* = X$
of the function $f(x, y)$. Let $\rho(w, W)$ denote
w from the set W.

Theorem 1. If assumptions (a)-(d) hold

$$\lim_{k \to \infty} d_k = 0,\ \lim_{k \to \infty} \sum_{i=0}^{k} d_i =$$

and $\rho(w^k, W^*) \to 0$ as $k \to \infty$.

We will preface the proof of the theorem
auxiliary assertion.

Lemma. Under assumptions (a)-(c), if

$$\lim_{k \to \infty} f(x^k, y^{*k}) = f(x^*, y^*),$$

Very Restrictive!

Relaxing Strong Convexity

We considered a subgradient algorithm for generating approximate saddle point solutions for convex-concave functions based on the seminal Arrow-Hurwicz-Uzawa algorithm and the averaging scheme (see [Bruck 77], [Nemirovski, Yudin 78])

$$\hat{\mathbf{x}}_N = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i, \qquad \hat{\mathbf{y}}_N = \frac{1}{N}\sum_{i=1}^{N} \mathbf{y}_i$$

Theorem (Nedic and Ozdaglar, 09')

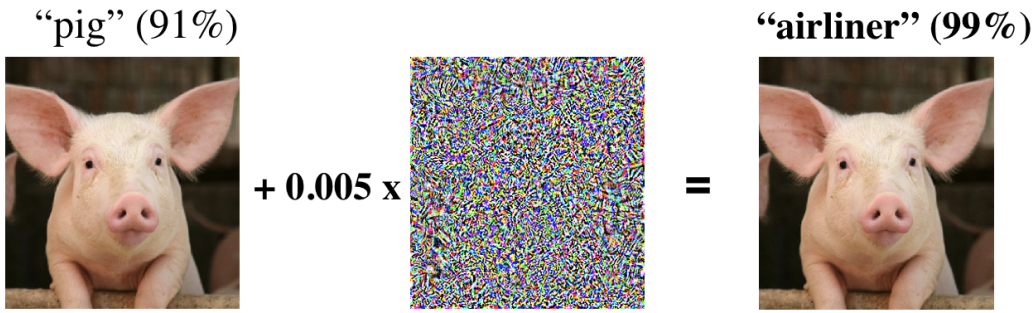
Assume that the (sub)gradients of f are uniformly bounded by L. Then, the iterates generated by GDA with stepsize $\eta \geq 0$ satisfy

$$|f(\hat{\mathbf{x}}_N, \hat{\mathbf{y}}_N) - f(\mathbf{x}^*, \mathbf{y}^*)| \leq \mathcal{O}\left(\frac{1}{\eta N}\right) + \mathcal{O}(\eta L^2)$$

- Boundedness assumption may be restrictive (works for compact domains).

Minmax Problems in Machine Learning

Adversarial robustness through minmax optimization



[Szegedy et al. 2014]: Imperceptible noise (adversarial examples) can fool state-of-the-art classifiers

- Standard training: Selecting model parameters x to minimize expected loss $\mathbb{E}_{(w,\theta)\sim\mathcal{P}}[\ell(w,\theta,x)]$.
- Robust training: Consider inputs with adversarial modifications represented as ℓ_∞-perturbations y of data points w ([Madry et al., 17'])
- The robust learning problem then amounts to choosing x to solve the following minimax problem:

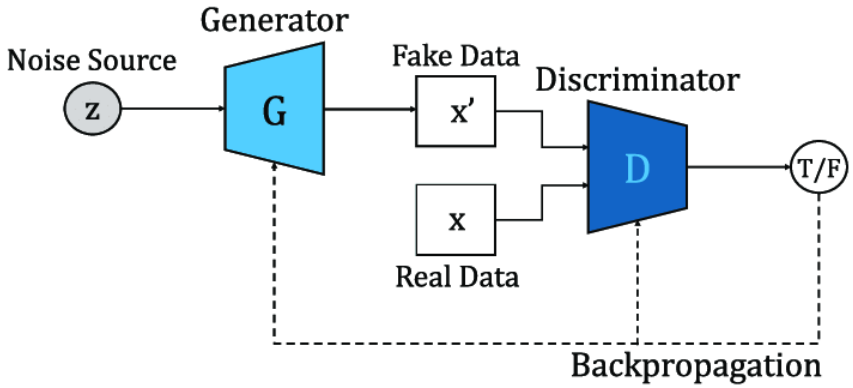
$$\min_x \mathbb{E}_{(w,\theta)\sim\mathcal{P}}\left[\max_{y\in\mathcal{S}} \ell(w+y,\theta,x)\right],$$

where $\mathcal{S}$ denotes allowable perturbations.

Minmax Problems in Machine Learning

Generative Adversarial Networks (GANs)

- Goal: Learn the distribution of observed samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$
- Formulated as a zero-sum game with two players (neural networks):
  - Generator G generating real-like samples from random input $\mathbf{z}$.
  - Discriminator D distinguishing generated samples from real samples.



- Leads to a minmax optimization problem [Goodfellow et al. 14]:

$$\min_G \max_D \frac{1}{n}\sum_{i=1}^{n} \log\big(D(\mathbf{x}_i)\big) + \frac{1}{n}\sum_{i=1}^{n} \log\big(1 - D(G(\mathbf{z}_i))\big)$$

- Nonconvex-nonconcave minmax problem since both the generator and discriminator are neural nets.

Challenges in Training GANs

- GAN training (where the generator and discriminator are trained using first order methods) commonly suffers from:
  - Instability

Iteration 60000

Iteration 65000

Challenges in Training GANs

- GAN training (where the generator and discriminator are trained using first order methods) commonly suffers from:
  - Instability
  - Mode Collapse

Mode Collapse

Regular Learning

Convergence of GDA

- Even for the simple bilinear case, GDA diverges.
- Consider the following bilinear problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \max_{\mathbf{y}\in\mathbb{R}^d} f(\mathbf{x},\mathbf{y}) = \mathbf{x}^\top \mathbf{y}$$

The solution is $(\mathbf{x}^*, \mathbf{y}^*) = (\mathbf{0}, \mathbf{0})$.

- The Gradient Descent Ascent (GDA) updates for this problem:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \underbrace{\mathbf{y}_k}_{\nabla_{\mathbf{x}} f(\mathbf{x}_k,\mathbf{y}_k)}, \qquad \mathbf{y}_{k+1} = \mathbf{y}_k + \eta \underbrace{\mathbf{x}_k}_{\nabla_{\mathbf{y}} f(\mathbf{x}_k,\mathbf{y}_k)}$$

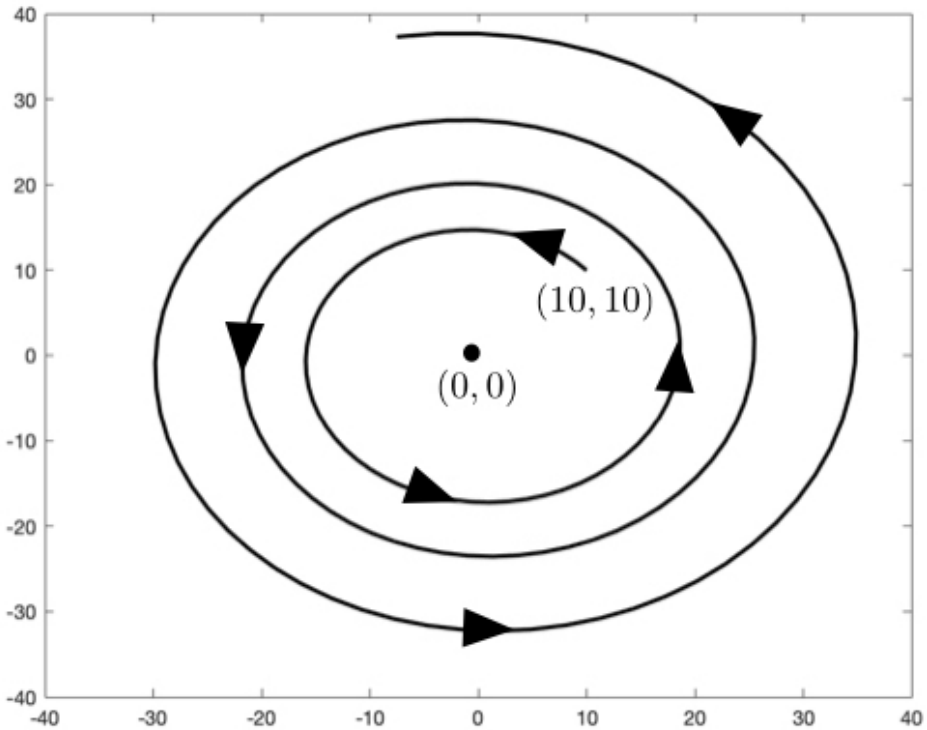
where η is the stepsize.

GDA

- After k iterations of GDA algorithm, we have:

$$\|\mathbf{x}_{k+1}\|^2 + \|\mathbf{y}_{k+1}\|^2 = (1+\eta^2)(\|\mathbf{x}_k\|^2 + \|\mathbf{y}_k\|^2)$$

- GDA diverges as $(1+\eta^2) > 1$



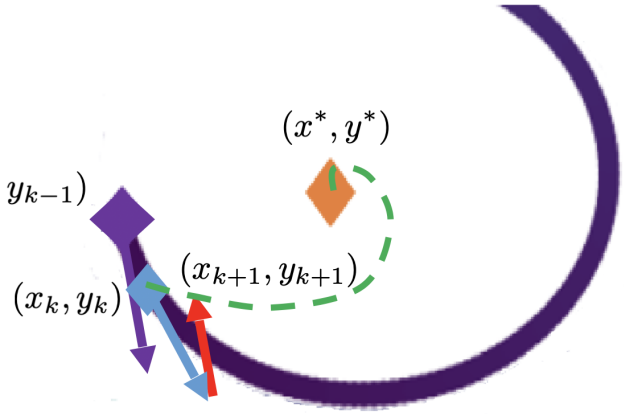
OGDA

- The simple GDA algorithm does not work even for bilinear problems:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \nabla_{\mathbf{x}} f(\mathbf{x}_k, \mathbf{y}_k), \qquad \mathbf{y}_{k+1} = \mathbf{y}_k + \eta \nabla_{\mathbf{y}} f(\mathbf{x}_k, \mathbf{y}_k).$$

- Recent popular variant: Optimistic Gradient Descent Ascent (OGDA) (GDA with negative momentum)

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - 2\eta \nabla_{\mathbf{x}} f(\mathbf{x}_k, \mathbf{y}_k) + \eta \nabla_{\mathbf{x}} f(\mathbf{x}_{k-1}, \mathbf{y}_{k-1}) \\ \mathbf{y}_{k+1} &= \mathbf{y}_k + 2\eta \nabla_{\mathbf{y}} f(\mathbf{x}_k, \mathbf{y}_k) - \eta \nabla_{\mathbf{y}} f(\mathbf{x}_{k-1}, \mathbf{y}_{k-1}) \end{aligned}$$



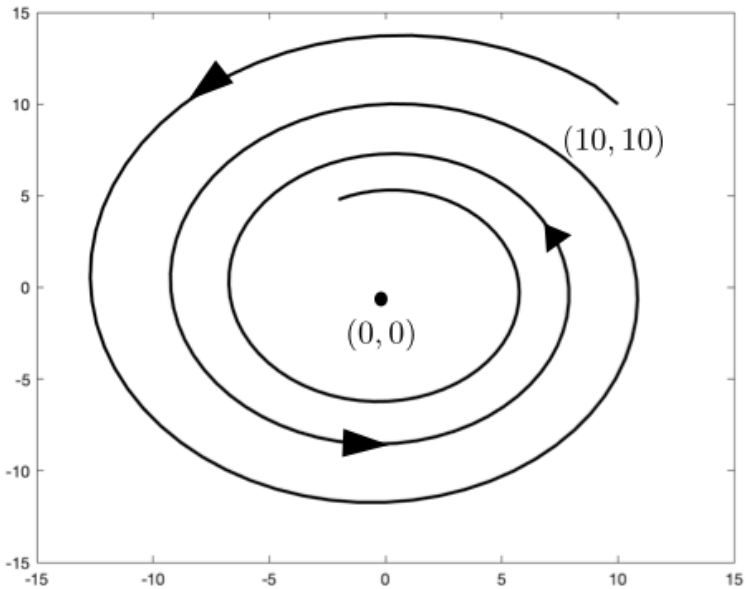
- Introduced in [Popov 80].
- Studied in [Rakhlin et al. 13] in the context of Online Learning.
- Convergence to a neighborhood for bilinear case [Daskalakis et al.18].
- Exact convergence for bilinear case and strongly convex-strongly concave case [Liang and Stokes 19], [Gidel et al. 19], [Mokhtari et al. 19].

OGDA

- OGDA updates for the bilinear problem

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta(\underbrace{2\mathbf{y}_k - \mathbf{y}_{k-1}}_{2\nabla_{\mathbf{x}} f(\mathbf{x}_k,\mathbf{y}_k) - \nabla_{\mathbf{x}} f(\mathbf{x}_{k-1},\mathbf{y}_{k-1})})$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \eta(\underbrace{2\mathbf{x}_k - \mathbf{x}_{k-1}}_{2\nabla_{\mathbf{y}} f(\mathbf{x}_k,\mathbf{y}_k) - \nabla_{\mathbf{y}} f(\mathbf{x}_{k-1},\mathbf{y}_{k-1})})$$



Proximal Point

- The Proximal Point (PP) updates for the same problem:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \underbrace{\mathbf{y}_{k+1}}_{\nabla_{\mathbf{x}} f(\mathbf{x}_{k+1}, \mathbf{y}_{k+1})} \qquad \mathbf{y}_{k+1} = \mathbf{y}_k + \eta \underbrace{\mathbf{x}_{k+1}}_{\nabla_{\mathbf{y}} f(\mathbf{x}_{k+1}, \mathbf{y}_{k+1})}$$

  where η is the stepsize.

- The difference from GDA is that the gradient at the iterate $(\mathbf{x}_{k+1}, \mathbf{y}_{k+1})$ is used for the update instead of the gradient at $(\mathbf{x}_k, \mathbf{y}_k)$.

- Although for this problem it takes a simple form

$$\mathbf{x}_{k+1} = \frac{1}{1+\eta^2}\left(\mathbf{x}_k - \eta \mathbf{y}_k\right), \qquad \mathbf{y}_{k+1} = \frac{1}{1+\eta^2}\left(\mathbf{y}_k + \eta \mathbf{x}_k\right)$$

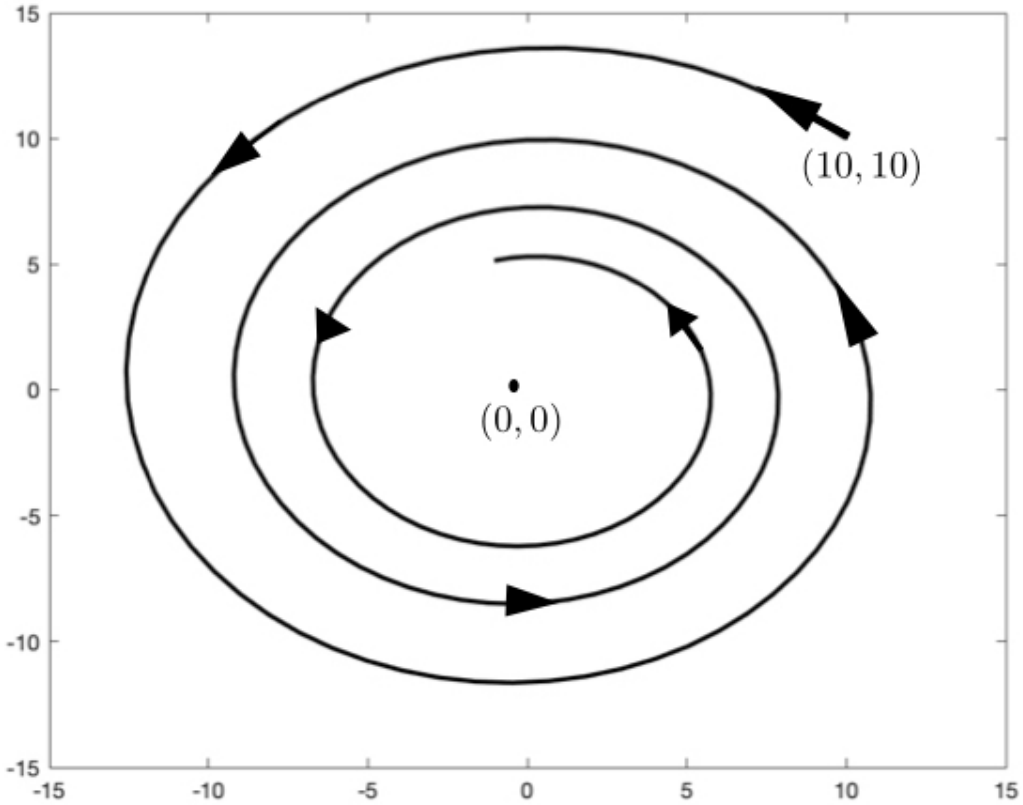
  Proximal Point method in general involves operator inversion and is not easy to implement.

Proximal Point

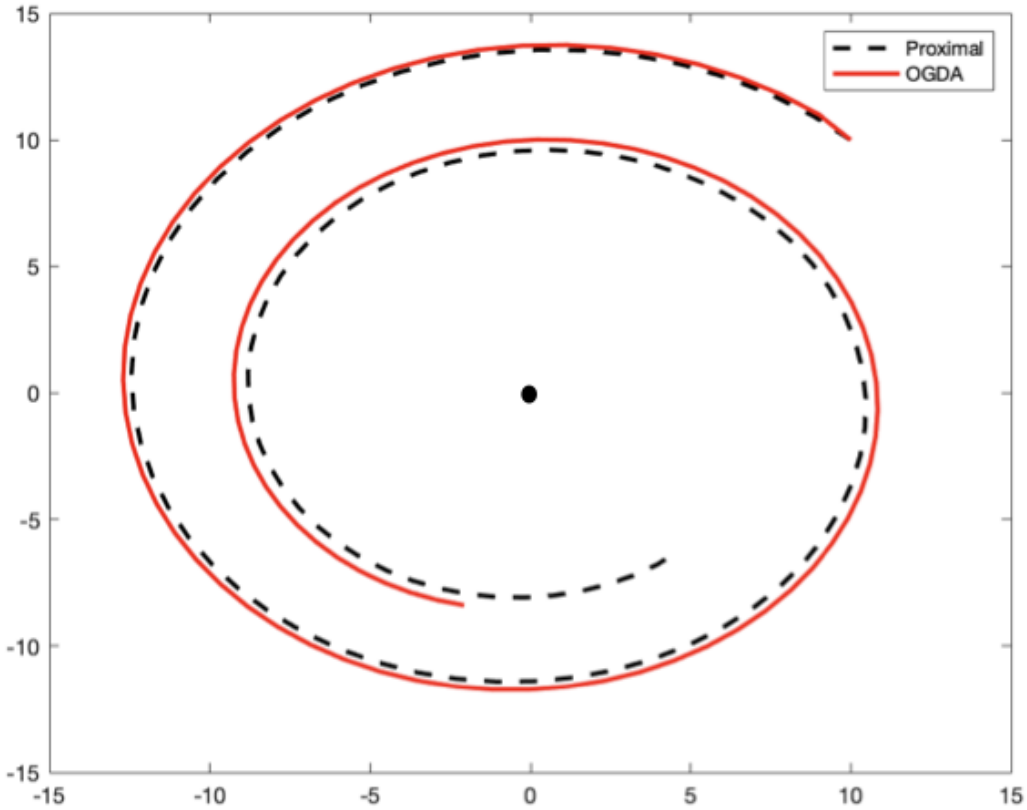
- On running PP, after k iterations we have:

$$\|\mathbf{x}_{k+1}\|^2 + \|\mathbf{y}_{k+1}\|^2 = \frac{1}{1+\eta^2}(\|\mathbf{x}_k\|^2 + \|\mathbf{y}_k\|^2)$$

- PP converges as $1/(1+\eta^2) < 1$



OGDA vs Proximal Point



- It seems like OGDA approximates Proximal Point method!
- Their convergence paths are very similar

Outline

- We view OGDA as an approximation of PP for finding a saddle point.
- We use the PP approximation viewpoint to show that for OGDA
  - The iterates remain bounded.
  - Function value converges at a rate of $\mathcal{O}(1/N)$.
- We revisit the Extra Gradient (EG) method using the same approach
- We then focus on the last iterate convergence of the EG algorithm.
- We show it is provably inferior to the ergodic iterate in the convex-concave setting.
- We finally turn to analysis of generalization properties of gradient based minmax algorithms using algorithmic stability framework defined by [Bousquet and Elisseeff 02].

Problem

- We consider finding the saddle point of the problem:

$$\min_{\mathbf{x}\in\mathbb{R}^m} \max_{\mathbf{y}\in\mathbb{R}^n} f(\mathbf{x},\mathbf{y})$$

- f is convex in $\mathbf{x}$ and concave in $\mathbf{y}$.
- $f(\mathbf{x},\mathbf{y})$ is continuously differentiable in $\mathbf{x}$ and $\mathbf{y}$.
- $\nabla_{\mathbf{x}} f$ and $\nabla_{\mathbf{y}} f$ are Lipschitz in $\mathbf{x}$ and $\mathbf{y}$. L denotes the Lipschitz constant.
- $(\mathbf{x}^*,\mathbf{y}^*) \in \mathbb{R}^m \times \mathbb{R}^n$ is a saddle point if it satisfies:

$$f(\mathbf{x}^*,\mathbf{y}) \le f(\mathbf{x}^*,\mathbf{y}^*) \le f(\mathbf{x},\mathbf{y}^*),$$

for all $\mathbf{x} \in \mathbb{R}^m, \mathbf{y} \in \mathbb{R}^n$. Let $f^* = f(\mathbf{x}^*,\mathbf{y}^*)$.

Proximal Point

- The PP method at each step solves the following:

$$(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}) = \arg \min_{\mathbf{x}\in\mathbb{R}^m} \max_{\mathbf{y}\in\mathbb{R}^n} \left\{ f(\mathbf{x}, \mathbf{y}) + \frac{1}{2\eta}\|\mathbf{x} - \mathbf{x}_k\|^2 - \frac{1}{2\eta}\|\mathbf{y} - \mathbf{y}_k\|^2 \right\}.$$

- Using the first order optimality conditions leads to the following update:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta\nabla_{\mathbf{x}} f(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}), \qquad \mathbf{y}_{k+1} = \mathbf{y}_k + \eta\nabla_{\mathbf{y}} f(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}).$$

Theorem (Convergence of Proximal Point)

The iterates generated by the PP method satisfy

$$\left[\max_{\mathbf{y}:(\hat{\mathbf{x}}_N,\mathbf{y})\in\mathcal{D}} f(\hat{\mathbf{x}}_N, \mathbf{y}) - f^\star\right] + \left[f^\star - \min_{\mathbf{x}:(\mathbf{x},\hat{\mathbf{y}}_N)\in\mathcal{D}} f(\mathbf{x}, \hat{\mathbf{y}}_N)\right] \leq \frac{D}{\eta N}.$$

$D = \|\mathbf{x}_0 - \mathbf{x}^*\|^2 + \|\mathbf{y}_0 - \mathbf{y}^*\|^2$, $\mathcal{D} := \{(\mathbf{x}, \mathbf{y}) \mid \|\mathbf{x} - \mathbf{x}^*\|^2 + \|\mathbf{y} - \mathbf{y}^*\|^2 \leq D\}$.

OGDA updates - How prediction takes place

- One way of approximating the Proximal Point update is as follows

$$\nabla_{\mathbf{x}} f(\mathbf{x}_{k+1}, \mathbf{y}_{k+1})) \approx \nabla_{\mathbf{x}} f(\mathbf{x}_k, \mathbf{y}_k) + (\nabla_{\mathbf{x}} f(\mathbf{x}_k, \mathbf{y}_k) - \nabla_{\mathbf{x}} f(\mathbf{x}_{k-1}, \mathbf{y}_{k-1}))$$
$$\nabla_{\mathbf{y}} f(\mathbf{x}_{k+1}, \mathbf{y}_{k+1})) \approx \nabla_{\mathbf{y}} f(\mathbf{x}_k, \mathbf{y}_k) + (\nabla_{\mathbf{y}} f(\mathbf{x}_k, \mathbf{y}_k) - \nabla_{\mathbf{y}} f(\mathbf{x}_{k-1}, \mathbf{y}_{k-1}))$$

- This leads to the OGDA update

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \nabla_{\mathbf{x}} f(\mathbf{x}_k, \mathbf{y}_k) - \eta(\nabla_{\mathbf{x}} f(\mathbf{x}_k, \mathbf{y}_k) - \nabla_{\mathbf{x}} f(\mathbf{x}_{k-1}, \mathbf{y}_{k-1}))$$
$$\mathbf{y}_{k+1} = \mathbf{y}_k + \eta \nabla_{\mathbf{y}} f(\mathbf{x}_k, \mathbf{y}_k) + \eta(\nabla_{\mathbf{y}} f(\mathbf{x}_k, \mathbf{y}_k) - \nabla_{\mathbf{y}} f(\mathbf{x}_{k-1}, \mathbf{y}_{k-1}))$$

Convergence rates

Theorem (Convex-Concave case)

Let the stepsize η satisfy $0 < \eta \leq 1/2L$. Then, the iterates generated by OGDA satisfy

$$\left[\max_{\mathbf{y}:(\hat{\mathbf{x}}_N,\mathbf{y})\in\mathcal{D}} f(\hat{\mathbf{x}}_N,\mathbf{y}) - f^\star\right] + \left[f^\star - \min_{\mathbf{x}:(\mathbf{x},\hat{\mathbf{y}}_N)\in\mathcal{D}} f(\mathbf{x},\hat{\mathbf{y}}_N)\right] \leq \frac{4D}{\eta N}$$

where $\mathcal{D} := \{(\mathbf{x},\mathbf{y}) \mid \|\mathbf{x}-\mathbf{x}^*\|^2 + \|\mathbf{y}-\mathbf{y}^*\|^2 \leq 2D\}$.

- OGDA has an iteration complexity of $\mathcal{O}(1/N)$.
- This shows that OGDA is an implementable version of PP which enjoys similar convergence guarentees.

Analysis of OGDA

- We analyze it as a Proximal Point method with error:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \nabla_{\mathbf{x}} f(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}) + \varepsilon_k^x$$
$$\mathbf{y}_{k+1} = \mathbf{y}_k + \eta \nabla_{\mathbf{y}} f(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}) - \varepsilon_k^y$$

- Let $\mathbf{z} = [\mathbf{x}; \mathbf{y}]$, $F(\mathbf{z}) = [\nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}); -\nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})]$ and $\varepsilon_k = [\varepsilon_k^x; -\varepsilon_k^y]$.
- For OGDA, the error is given by

$$\varepsilon_k = \eta[(F(\mathbf{z}_{k+1}) - F(\mathbf{z}_k)) - (F(\mathbf{z}_k) - F(\mathbf{z}_{k-1}))].$$

Lemma (Three-Term Equality for PP method with error)

$$F(\mathbf{z}_{k+1})^\top (\mathbf{z}_{k+1} - \mathbf{z})$$
$$= \frac{1}{2\eta}\|\mathbf{z}_k - \mathbf{z}\|^2 - \frac{1}{2\eta}\|\mathbf{z}_{k+1} - \mathbf{z}\|^2 - \frac{1}{2\eta}\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 + \frac{1}{\eta}\varepsilon_k{}^T(\mathbf{z}_{k+1} - \mathbf{z}).$$

Convergence rate - Proof Sketch

- Substituting OGDA error in the three-term equality for PP with error,

$$\sum_{k=0}^{N-1} F(\mathbf{z}_{k+1})^\top (\mathbf{z}_{k+1} - \mathbf{z}) \leq \frac{1}{2\eta}\|\mathbf{z}_0 - \mathbf{z}\|^2 - \frac{1}{2\eta}\|\mathbf{z}_N - \mathbf{z}\|^2 - \frac{L}{2}\|\mathbf{z}_N - \mathbf{z}_{N-1}\|^2 + (F(\mathbf{z}_N) - F(\mathbf{z}_{N-1}))^\top (\mathbf{z}_N - \mathbf{z}).$$

- Using Lipschitz continuity of the operator $F(z)$,

$$\sum_{k=0}^{N-1} F(\mathbf{z}_{k+1})^\top (\mathbf{z}_{k+1} - \mathbf{z}) \leq \frac{1}{2\eta}\|\mathbf{z}_0 - \mathbf{z}\|^2 - \left(\frac{1}{2\eta} - \frac{L}{2}\right)\|\mathbf{z}_N - \mathbf{z}\|^2$$

- By monotonicity, $F(\mathbf{z})^\top (\mathbf{z} - \mathbf{z}^*) \geq 0$ for all $\mathbf{z}$ (and $\eta \leq 1/2L$): iterates are bounded,

$$\|\mathbf{z}_N - \mathbf{z}^*\|^2 \leq 2\|\mathbf{z}_0 - \mathbf{z}^*\|^2$$

Convergence rate - Proof Sketch

- Using convexity-concavity of f, $(F(\mathbf{z}_k)^T(\mathbf{z}_k - \mathbf{z}) \geq f(\mathbf{x}_k, \mathbf{y}) - f(\mathbf{x}, \mathbf{y}_k))$ and averaging which gives

$$\frac{1}{N}\sum_{k=1}^{N} f(\mathbf{x}_k, \mathbf{y}) \geq f(\hat{\mathbf{x}}_N, \mathbf{y}), \qquad \frac{1}{N}\sum_{k=1}^{N} f(\mathbf{x}, \mathbf{y}_k) \leq f(\mathbf{x}, \hat{\mathbf{y}}_N).$$

we have:

$$\begin{aligned}\left[\max_{\mathbf{y}:(\hat{\mathbf{x}}_N,\mathbf{y})\in\mathcal{D}} f(\hat{\mathbf{x}}_N, \mathbf{y}) - f^\star\right] + \left[f^\star - \min_{\mathbf{x}:(\mathbf{x},\hat{\mathbf{y}}_N)\in\mathcal{D}} f(\mathbf{x}, \hat{\mathbf{y}}_N)\right] \\ \leq \max_{z\in\mathcal{D}} \frac{1}{N}\sum_{k=0}^{N-1} F(\mathbf{z}_k)^\top(\mathbf{z}_k - \mathbf{z}) \\ \leq \frac{4D}{\eta N}\end{aligned}$$

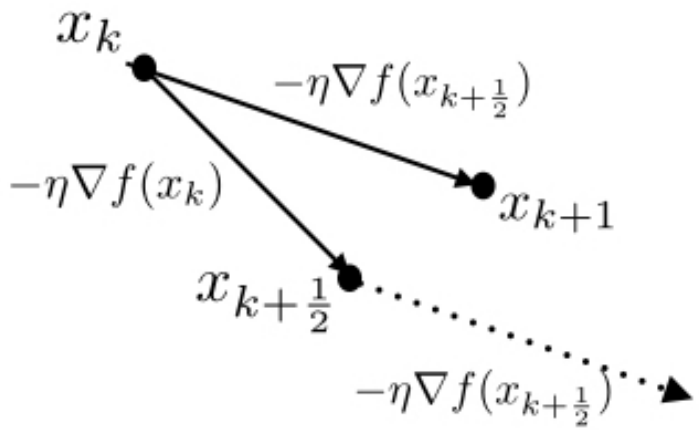
Extragradient Method

- Introduced by [Korpelevich 77] "as a modification of gradient method that uses the idea of extrapolation."
- The updates of EG

$$\mathbf{x}_{k+1/2} = \mathbf{x}_k - \eta \nabla_{\mathbf{x}} f(\mathbf{x}_k, \mathbf{y}_k), \qquad \mathbf{y}_{k+1/2} = \mathbf{y}_k + \eta \nabla_{\mathbf{y}} f(\mathbf{x}_k, \mathbf{y}_k).$$

The gradients evaluated at the midpoints $\mathbf{x}_{k+1/2}$ and $\mathbf{y}_{k+1/2}$ are used to compute the new iterates $\mathbf{x}_{k+1}$ and $\mathbf{y}_{k+1}$ by performing the updates

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \nabla_{\mathbf{x}} f(\mathbf{x}_{k+1/2}, \mathbf{y}_{k+1/2}),$$
$$\mathbf{y}_{k+1} = \mathbf{y}_k + \eta \nabla_{\mathbf{y}} f(\mathbf{x}_{k+1/2}, \mathbf{y}_{k+1/2}).$$



EG updates - How prediction takes place

- The update can also be written as:

$$\begin{aligned}
\mathbf{x}_{k+1/2} &= \mathbf{x}_{k-1/2} - \eta \nabla_{\mathbf{x}} f(\mathbf{x}_{k-1/2}, \mathbf{y}_{k-1/2}) \\
&\quad - \eta \left(\nabla_{\mathbf{x}} f(\mathbf{x}_k, \mathbf{y}_k) - \nabla_{\mathbf{x}} f(\mathbf{x}_{k-1}, \mathbf{y}_{k-1}) \right), \\
\mathbf{y}_{k+1/2} &= \mathbf{y}_{k-1/2} + \eta \nabla_{\mathbf{y}} f(\mathbf{x}_{k-1/2}, \mathbf{y}_{k-1/2}) \\
&\quad + \eta \left(\nabla_{\mathbf{y}} f(\mathbf{x}_k, \mathbf{y}_k) - \nabla_{\mathbf{y}} f(\mathbf{x}_{k-1}, \mathbf{y}_{k-1}) \right).
\end{aligned}$$

- EG tries to predict the gradient using interpolation of the midpoint gradients:

$$\begin{aligned}
\nabla_{\mathbf{x}} f(\mathbf{x}_{k+1/2}, \mathbf{y}_{k+1/2})) &\approx \nabla_{\mathbf{x}} f(\mathbf{x}_{k-1/2}, \mathbf{y}_{k-1/2}) \\
&\quad + \left(\nabla_{\mathbf{x}} f(\mathbf{x}_k, \mathbf{y}_k) - \nabla_{\mathbf{x}} f(\mathbf{x}_{k-1}, \mathbf{y}_{k-1}) \right) \\
\nabla_{\mathbf{y}} f(\mathbf{x}_{k+1/2}, \mathbf{y}_{k+1/2})) &\approx \nabla_{\mathbf{y}} f(\mathbf{x}_{k-1/2}, \mathbf{y}_{k-1/2}) \\
&\quad + \left(\nabla_{\mathbf{y}} f(\mathbf{x}_k, \mathbf{y}_k) - \nabla_{\mathbf{y}} f(\mathbf{x}_{k-1}, \mathbf{y}_{k-1}) \right)
\end{aligned}$$

Convergence rate

Theorem (Convex-Concave case)

Let the stepsize η satisfy $0 < \eta < \frac{1}{L}$. Then the iterates generated by EG satisfy

$$\left[\max_{\mathbf{y}:(\hat{\mathbf{x}}_N,\mathbf{y})\in\mathcal{D}} f(\hat{\mathbf{x}}_N,\mathbf{y}) - f^\star\right] + \left[f^\star - \min_{\mathbf{x}:(\mathbf{x},\hat{\mathbf{y}}_N)\in\mathcal{D}} f(\mathbf{x},\hat{\mathbf{y}}_N)\right] \leq \frac{DL\left(16 + \frac{33}{2(1-\eta^2L^2)}\right)}{N}$$

where $\mathcal{D} := \{(\mathbf{x},\mathbf{y}) \mid \|\mathbf{x}-\mathbf{x}^*\|^2 + \|\mathbf{y}-\mathbf{y}^*\|^2 \leq (2 + \frac{2}{1-\eta^2L^2})D\}$.

- EG has an iteration complexity of $\mathcal{O}(1/N)$ (same as Proximal Point).
- $\mathcal{O}(1/N)$ rate for the convex-concave case was shown in [Nemirovski 04] when the feasible set is compact.
- [Monteiro and Svaiter 10] extended to unbounded sets using a different termination criterion.
- Our result shows a convergence rate of $\mathcal{O}(1/N)$ in terms of primal-dual gap.

Last Iterate Convergence

- For first order algorithms for minmax problems (including OGDA and EG), convergence guarantees in the convex-concave case is only known for the ergodic iterates, i.e.,

$$(\hat{\mathbf{x}}_N, \hat{\mathbf{y}}_N) := \left(\frac{1}{N} \sum_{k=1}^{N} \mathbf{x}_i, \frac{1}{N} \sum_{k=1}^{N} \mathbf{y}_i \right)$$

- Much recent interest in studying last iterate convergence, i.e., $(\mathbf{x}_N, \mathbf{y}_N)$ [Daskalakis et. al 18]:
  - Last iterate guarantees in the convex-concave case may extend to the nonconvex-nonconcave case, which includes the GAN training objective.
  - Last iterate guarantees needed to preserve sparsity.

Last Iterate Convergence - EG

- In recent work, we study the convergence rate of last iterate of EG in the convex-concave setting [Golowich, Pattathil, Daskalakis, Ozdaglar 20].

Theorem (Last iterate EG (Upper bound)- Convex-Concave case)

Suppose f is a convex-concave function that is L-smooth and has a Λ-Lipschitz Hessian. The iterates of EG algorithm with step size $\eta \leq \min\left\{\frac{5}{\Lambda D}, \frac{1}{30L}\right\}$ satisfy

$$\left[\max_{\mathbf{y}:(\mathbf{x}_N,\mathbf{y})\in\mathcal{D}} f(\mathbf{x}_N,\mathbf{y}) - f^\star\right] + \left[f^\star - \min_{\mathbf{x}:(\mathbf{x},\mathbf{y}_N)\in\mathcal{D}} f(\mathbf{x},\mathbf{y}_N)\right] \leq \frac{4\sqrt{2}D}{\eta\sqrt{N}}$$

where $\mathcal{D} := \{(\mathbf{x},\mathbf{y}) \mid \|\mathbf{x}-\mathbf{x}^*\|^2 + \|\mathbf{y}-\mathbf{y}^*\|^2 \leq 2D\}$.

- First characterization of the convergence rate of the last iterate of EG in the general convex-concave setting.

Proof Sketch for Proximal Point

- Recall the three-term equality for PP Lemma. (with $\varepsilon_k = 0$)

$$F(\mathbf{z}_{k+1})^\top(\mathbf{z}_{k+1} - \mathbf{z}) = \frac{1}{2\eta}\|\mathbf{z}_k - \mathbf{z}\|^2 - \frac{1}{2\eta}\|\mathbf{z}_{k+1} - \mathbf{z}\|^2 - \frac{1}{2\eta}\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2$$

- This implies (with $\mathbf{z} = \mathbf{z}^*$ and $\langle F(\mathbf{z}), \mathbf{z} - \mathbf{z}^* \rangle \geq 0$, $\mathbf{z}_{k+1} - \mathbf{z}_k = \eta F(\mathbf{z}_{k+1})$)

$$\sum_{k=0}^{N-1} \frac{\eta}{2}\|F(\mathbf{z}_{k+1})\|^2 \leq \frac{D^2}{2\eta}$$

- Thus, there exists some k^* with $\|F(\mathbf{z}_{k^*})\|^2 \leq \frac{D^2}{\eta^2 N}$.
- We can show $\|F(\mathbf{z}_k)\|$ is decreasing implying $\|F(\mathbf{z}_N)\|^2 \leq \frac{D^2}{\eta^2 N}$
- This translates to a bound on the primal-dual gap.
- Proof extends to EG by noting that $\|F(\mathbf{z}_k)\|^2$ does not increase too much after k^*.

Lower Bound for Last Iterate

- Our upper bound establishes a "quadratic separation" between the last iterate and averaged iterates of EG (a rate of $\mathcal{O}(1/\sqrt{N})$ versus $\mathcal{O}(1/N)$).
- We show that our bound is tight by producing a lower bound, using the Stationary Canonical Linear Iterative (SCLI) algorithm framework of [Arjevani and Shamir 15] and building on [Azizian et al. 19].

Key Idea:

- Instead of the seminal first-order "oracle model" of [Nemirovsky and Yudin 83] for quantifying the computational hardness of optimization problems, [Arjevani and Shamir 15] use a structure based approach.
  - Assume certain dynamics for generating new iterates (which includes a large family of computationally efficient first order algorithms).

Lower Bound for Last Iterate

Theorem (Last iterate EG (Lower bound) - Convex-Concave case)

Consider the EG algorithm for a minmax problem with a bilinear objective function $f(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{M}\mathbf{y} + \mathbf{b}_1^\top \mathbf{x} + \mathbf{b}_2^\top \mathbf{y}$. *Then, there exists matrices* $\mathbf{M}$ *and vectors* $\mathbf{b}_1, \mathbf{b}_2$ *such that*

$$\left[\max_{\mathbf{y}:(\mathbf{x}_N,\mathbf{y})\in\mathcal{D}} f(\mathbf{x}_N, \mathbf{y}) - f^\star\right] + \left[f^\star - \min_{\mathbf{x}:(\mathbf{x},\mathbf{y}_N)\in\mathcal{D}} f(\mathbf{x}, \mathbf{y}_N)\right] \geq \mathcal{O}(1/\sqrt{N})$$

where $\mathcal{D} := \{(\mathbf{x}, \mathbf{y}) \mid \|\mathbf{x} - \mathbf{x}^*\|^2 + \|\mathbf{y} - \mathbf{y}^*\|^2 \leq 2D\}$.

- The lower bound is obtained by considering a bilinear minimax problem.
- Proof follows by showing that the iterates are polynomials of fixed degree and using properties of Chebyshev polynomials to provide lower bounds.

Generalization in Minimax Optimization Problems

- Current studies of minimax learning frameworks focus on convergence [Jin, Netrapalli, Jordan 19] and robustness [Daskalakis and Panageas 18] properties of minimax optimization algorithms.
- However, a successful minimax learner also needs to **generalize** well from empirical training samples to unseen test samples.

Definition

We define the generalization error of model x as the difference between its worst-case population and empirical minimax objectives:

$$\epsilon_{\text{gen}}(x) := \max_y \left\{ \mathbb{E}_{z \sim P_Z} \left[f(x, y; z) \right] \right\} - \max_y \left\{ \frac{1}{n} \sum_{i=1}^{n} f(x, y; z_i) \right\}.$$

We also define the generalization error of Algorithm A as the expected generalization error of $A_x(S)$ defined as A's learned x over random dataset S:

$$\epsilon_{\text{gen}}(A) := \mathbb{E}_S \left[\epsilon_{\text{gen}}(A_x(S)) \right].$$

Stability and Generalization in Minimax Learning

- In [Farnia and Ozdaglar 20], we study the generalization behavior of minimax optimization algorithms through the algorithmic stability framework defined by [Bousquet and Elisseeff 02] and refined in [Hardt, Recht, Singer 16] to study generalization error of iterative algorithms.

Definition

A minimax optimization algorithm A is called ϵ-uniformly stable in minimization if for every two datasets S, S' different in only one sample and every y and data point $\mathbf{z}$ we have:

$$\mathbb{E}_A\big[\, f(A_x(S), y; \mathbf{z}) - f(A_x(S'), y; \mathbf{z}) \,\big] \leq \epsilon.$$

Theorem

Suppose that optimization algorithm A is ϵ-uniformly stable in minimization. Then, A's expected generalization risk is bounded as $\epsilon_{\text{gen}}(A) \leq \epsilon$.

- Similar to [Hardt, Recht, Singer 16] using worst-case minimax objective.
- Relies on decomposition of empirical risk over individual data points.

Stability and Generalization in Minimax Learning

- Learning theory based approaches estimate accuracy of a learning system based on theory of uniform convergence of empirical quantities to their mean [Vapnik 82].
- [Bousquet and Elisseeff 02] provides a different sensitivity-based approach aimed at determining how much variation of the input can influence the output of a learning system.
- Related to seminal results in stochastic optimization, in particular (robust) stochastic approximation or stochastic gradient descent convergence estimates (measures difference from optimal population risk) [Nemirovski, Yudin 78,83] , [Nemirovski, Juditsky, Lan, Shapiro 09]
  - Key Difference: These bounds hold with "fresh samples" at every iteration (hold only for single pass over data)
  - Algorithmic stability enables generalization bounds for multiple epochs of the algorithm over the training set by combining stability and optimization errors.

Stability Analysis of Gradient-based Algorithms

Analyze how two different sequences of update rules diverge when iterated from the same starting point.

Step 1: We first study the expansivity of a gradient-based update rule.

Definition

An update rule $G : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d \times \mathbb{R}^d$ *is* γ*-expansive if:*

$$\left\| G(x, y) - G(x', y') \right\| \leq \gamma \left\| [x, y] - [x', y'] \right\| \qquad \text{for all } x, x', y, y'.$$

- Smoothness will imply that gradient updates cannot be overly expansive.
- For a strongly-convex-concave objective, gradient updates are contractive for a sufficiently small stepsize.
- For a convex-concave objective, gradient updates can be expansive while proximal point method (PPM) updates will be non-expansive.

Stability Analysis of Gradient-based Algorithms

Step 2: Growth lemma analyzes how two sequences of updates diverge.

Lemma (Growth Lemma)

Consider two sequences of γ-expansive updates G_t, G'_t with the same starting point. Define $\delta_t = \|[x_t, y_t] - [x'_t, y'_t]\|$. Then,

$$\delta_{t+1} \leq \min\{1, \gamma\}\delta_t + \max_{u,v} \|[u, v] - G_t[u, v]\| + \max_{u',v'} \|[u', v'] - G'_t[u', v']\|.$$

Step 3: To show the stability of an algorithm, we analyze its output on two datasets S, S' that differ in only one sample. Analysis insightful for SGDA. Two cases:

- Select an example which is identical in S, S' w.p. $1 - \frac{1}{n}$: Iterates diverge with expansivity of the same update.
- Select the non-identical example w.p. $\frac{1}{n}$: Use growth lemma and smoothness of the function.

Generalization of GDA vs. GDmax in Minimax Problems

Definition

For minimax risk function $R(x, y)$ and stepsize parameters α_x, α_y, GDA and GDmax update rules are defined as

$$G_{\text{GDA}}\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x - \alpha_x \nabla_x R(x, y) \\ y + \alpha_y \nabla_y R(x, y) \end{bmatrix}, \; G_{\text{GDmax}}\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x - \alpha_x \nabla_x R(x, y) \\ \text{argmax}_{\tilde{y}} \, R(x, \tilde{y}) \end{bmatrix}$$

- Simultaneous GDA-type methods are typically used for training GANs [Goodfellow et al. 14].
- Non-simultaneous GDmax algorithm is the standard optimization method for adversarial training [Madry et al. 17].
- We compare the generalization properties of GDA and GDmax algorithms in strongly-convex strongly-concave and non-convex strongly-concave minimax settings.

Generalization of GDA vs. GDmax in Strongly-convex Strongly-concave Minimax Problems

Theorem

Let $f(\cdot,\cdot;z)$ be μ-strongly convex-concave and ℓ-smooth for every z. Assume that f is L and L_x-Lipschitz in joint $[x,y]$ and x. Then, GDA and GDmax, which applies gradient descent to the max function, with stepsize $\alpha \leq \frac{\mu}{\ell^2}$ will satisfy the following bounds over T iterations:

$$\epsilon_{\text{gen}}(\text{GDA}) \leq \frac{2LL_x}{(\mu - \frac{\alpha\ell^2}{2})n}, \quad \epsilon_{\text{gen}}(\text{GDmax}) \leq \frac{2L_x^2}{\mu n}.$$

- For sufficiently small α, the generalization bounds for GDA and GDmax are comparable and different by a constant factor L/L_x.
- GDmax result follows by applying gradient descent to the max function and using its convexity and Lipschitzness properties.

Generalization of GDA vs. GDmax in Non-convex Strongly-concave Minimax Problems

Theorem

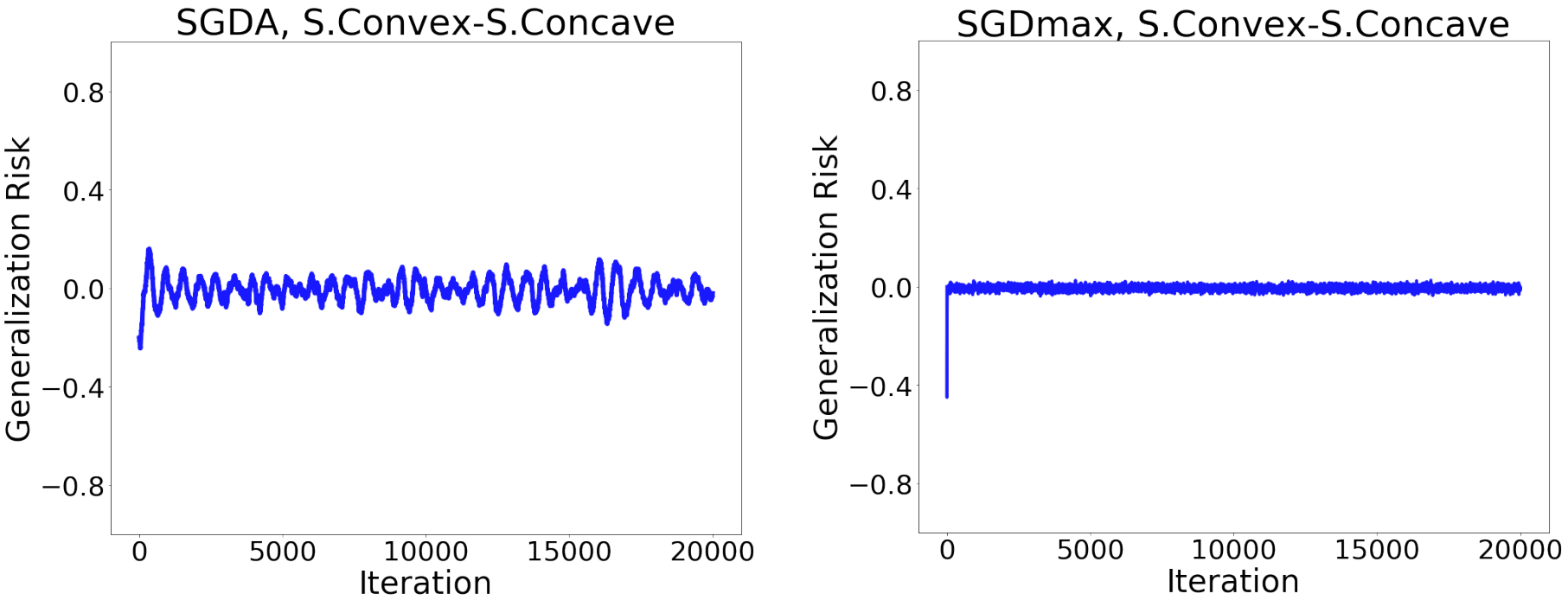
Let $f(\cdot,\cdot;z)$ be non-convex μ-strongly-concave and ℓ-smooth for every z. Then, stochastic GDA and GDmax with min and max stepsizes $\alpha_x = \frac{c}{t}$, $\alpha_y = \frac{cr^2}{t}$ will satisfy the following over T iterations for $\kappa = \frac{\ell}{\mu}$:

$$\epsilon_{\text{gen}}(\text{SGDA}) \leq \mathcal{O}\big(T^{\frac{\ell(r+1)c}{\ell(r+1)c+1}}/n\big), \quad \epsilon_{\text{gen}}(\text{SGDmax}) \leq \mathcal{O}\big(T^{\frac{\ell\kappa c}{\ell\kappa c+1}}/n\big).$$

- For $r \leq \kappa - 1$, the generalization bound for GDA outperforms the generalization bound for GDmax, implying that simultaneous training can lead to better generalization.
- This result theoretically supports the notion of *Implicit Competitive Regularization* introduced in [Schäfer et al. 20] for simultaneous gradient methods in training GANs.

Numerical Analysis for Generalization in Strongly-convex Strongly-concave Problems

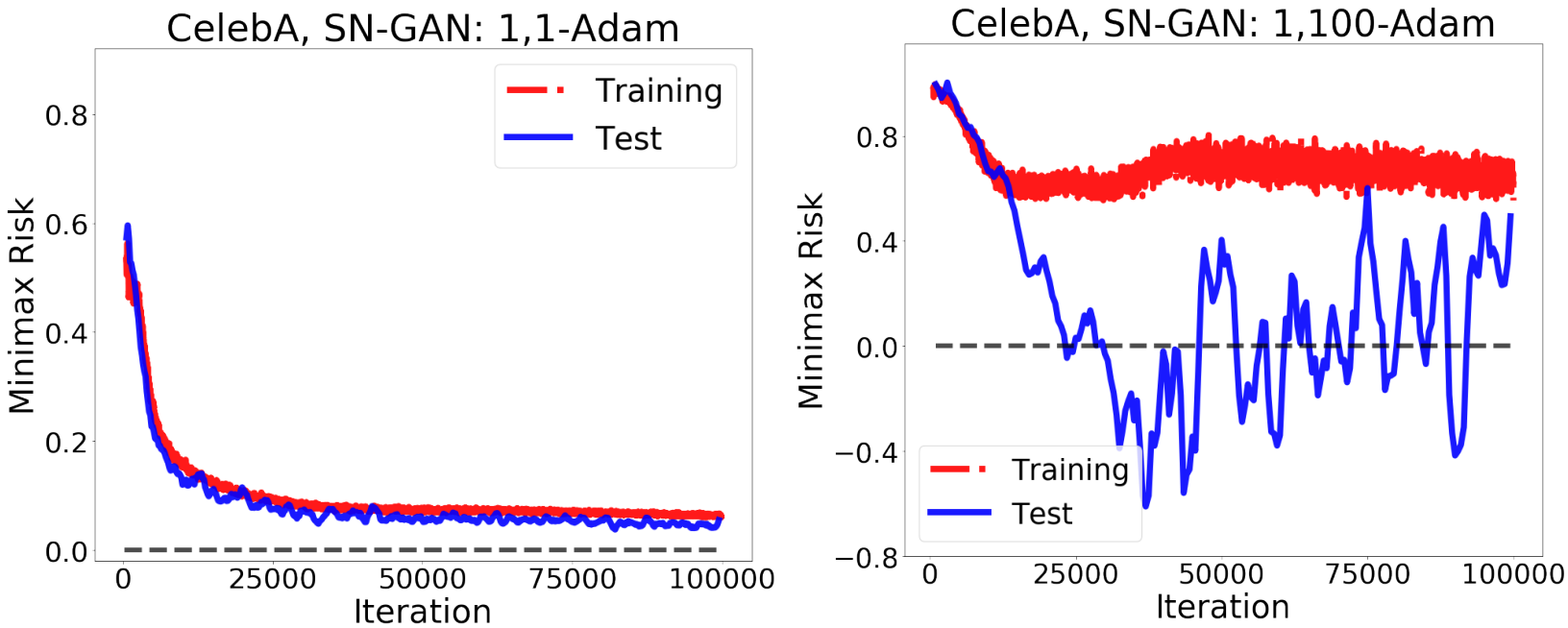
- We analyzed the generalization risk of stochastic GDA vs GDmax in optimizing the cost function $f(x, y; z) = x^\top(z - y) + \frac{1}{10}(\|x\|_2^2 - \|y\|_2^2)$:



Generalization Risk in Stochastic GDA vs. Stochastic GDmax training

Numerical Analysis for Generalization in Non-convex Non-concave GAN Problems

- We analyzed the generalization risk of simultaneous 1,1-Adam vs non-simultaneous 1,100-Adam in optimizing the GAN objective $f(x, y; z) = \log(D_x(z)) + \mathbb{E}_{\nu \sim \mathcal{N}(0,I)}[\log(1 - D_x(G_y(\nu)))]$:



Training and Test Risks in simultaneous vs. non-simultaneous training

Conclusions

- Gradient-based minimax learning algorithms increasingly used in adversarial training and GAN training in machine learning.
- We present our recent results on convergence rate, and generalization analysis for some of the commonly used algorithms and provided insights on their connections and performance.
- Much interest and open questions for both general nonconvex-strongly concave or nonconvex-nonconcave problems as well as minimax problems with structure (that arise in GAN problems).

Thanks!

- Convergence Rate of $\mathcal{O}(1/k)$ for Optimistic Gradient and Extra-gradient Methods in Smooth Convex-Concave Saddle Point Problems
  https://arxiv.org/pdf/1906.01115.pdf
- A Unified Analysis of Extra-gradient and Optimistic Gradient Methods for Saddle Point Problems: Proximal Point Approach
  https://arxiv.org/pdf/1901.08511.pdf
- Last Iterate is Slower than Averaged Iterate in Smooth Convex-Concave Saddle Point Problems
  https://arxiv.org/pdf/2002.00057.pdf
- Train simultaneously, generalize better: Stability of gradient-based minimax learners
  https://arxiv.org/abs/2010.12561.pdf