Convex-Concave Minmax Optimization

Applications and Methods

School of Data Science

August 5, 2022

Yilin Gu

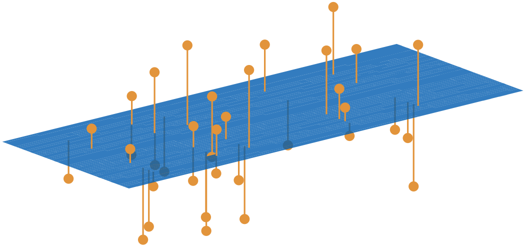
Outline

Robustness in Learning (I)

Standard training: Minimize empirical loss by selecting parameters $\boldsymbol{x}$

$$L(\boldsymbol{x}) := \frac{1}{N} \sum_{i=1}^{N} \ell(a_i, b_i | \boldsymbol{x})$$

(a_i, b_i) is a training sample, a_i is the input and b_i is the expected output



Linear regression: Consider $\ell(a_i, b_i | \boldsymbol{x}) = \|a_i^\top \boldsymbol{x} - b_i\|^2$

$$L(\boldsymbol{x}) := \frac{1}{N} \sum_{i=1}^{N} \|a_i^\top \boldsymbol{x} - b_i\|^2 = \frac{1}{N} \|A\boldsymbol{x} - b\|^2$$

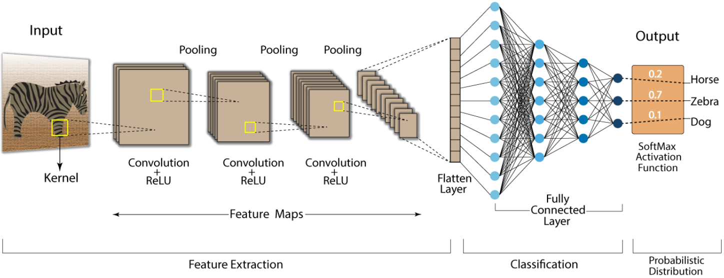
Robustness in Learning (II)

Neural network: Consider $\ell(a_i, b_i|\boldsymbol{x}) = \|\mathcal{M}(a_i|\boldsymbol{x}) - b_i\|^2$

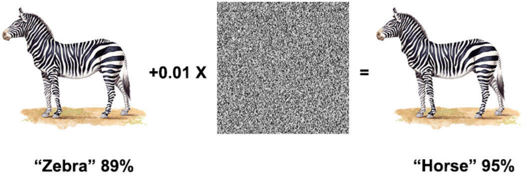
$$L(\boldsymbol{x}) := \frac{1}{N}\sum_{i=1}^{N}\|\mathcal{M}(a_i|\boldsymbol{x}) - b_i\|^2$$

where $\mathcal{M}(\cdot|\boldsymbol{x})$ denotes the model with parameters $\boldsymbol{x}$

Convolution Neural Network (CNN)



Robustness in Learning (III)



- **Robust training:** Consider inputs with modifications represented as perturbations $\boldsymbol{y}$ of data.
- It amounts to choosing $\boldsymbol{x}$ to solve the **minmax problem:**

$$\min_{\boldsymbol{x}\in\mathbb{R}^m} \frac{1}{N} \sum_{i=1}^{N} \max_{\boldsymbol{y}\in\mathcal{S}} \ell(a_i + \boldsymbol{y}, b_i|\boldsymbol{x})$$

where $\mathcal{S}$ denotes allowable perturbations

Outline

Minmax Problems

Consider the following minmax problem:

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

Applications:

- Worst-case design (robust optimization): Minimize over $\boldsymbol{x}$ the loss function with the worst possible value of $\boldsymbol{y}$

Minmax Problems

Consider the following minmax problem:

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

Applications:

- Worst-case design (robust optimization): Minimize over $\boldsymbol{x}$ the loss function with the worst possible value of $\boldsymbol{y}$
- Duality theory for constrained optimization:
  - Primal problem

    $$\min_{\boldsymbol{x}\in\mathbb{R}^m} f(\boldsymbol{x}), \quad \text{s.t. } g(\boldsymbol{x}) \leq 0$$

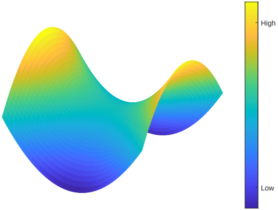
  - Lagarangian function

    $$\mathcal{L}(\boldsymbol{x}, y) = f(\boldsymbol{x}) + yg(\boldsymbol{x}), \quad y \geq 0$$

  - Dual problem is a minmax problem

    $$\max_{y\geq 0} \min_{\boldsymbol{x}\in\mathbb{R}^m} \mathcal{L}(\boldsymbol{x}, y) \quad \Longleftrightarrow \quad -\min_{y\geq 0} \max_{\boldsymbol{x}\in\mathbb{R}^m} -\mathcal{L}(\boldsymbol{x}, y)$$

Convex-concave Functions

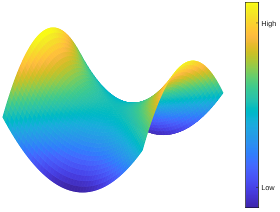


$f(x, y) = x^2 - y^2$

function w.r.t. x

function w.r.t. y

Convex-concave Functions



$f(x, y) = x^2 - y^2$

function w.r.t. x

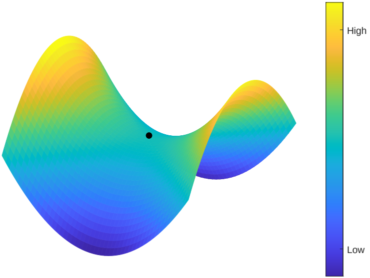
function w.r.t. y

Definition: Convex-concave Function

The function $f(\boldsymbol{x}, \boldsymbol{y})$ is convex-concave if

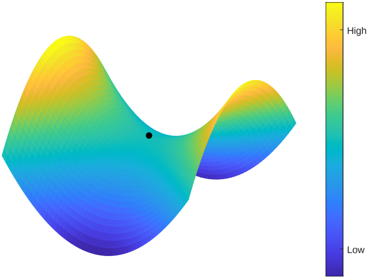
- for any $\boldsymbol{y} \in \mathbb{R}^n$, the function $f(\boldsymbol{x}, \boldsymbol{y})$ is a convex function of $\boldsymbol{x}$; and
- for any $\boldsymbol{x} \in \mathbb{R}^m$, the function $f(\boldsymbol{x}, \boldsymbol{y})$ is a concave function of $\boldsymbol{y}$

Saddle Points



$f(x, y) = x^2 - y^2$ with saddle point $(0, 0)$

Saddle Points



$f(x, y) = x^2 - y^2$ with saddle point $(0, 0)$

Definition: Saddle Points

A saddle point of the minmax problem is a pair $(\boldsymbol{x}^*, \boldsymbol{y}^*) \in \mathbb{R}^m \times \mathbb{R}^n$ that

$$f(\boldsymbol{x}^*, \boldsymbol{y}) \leq f(\boldsymbol{x}^*, \boldsymbol{y}^*) \leq f(\boldsymbol{x}, \boldsymbol{y}^*)$$

for all $\boldsymbol{x} \in \mathbb{R}^m$ and $\boldsymbol{y} \in \mathbb{R}^n$

Outline

Primal-dual Gap

Define the constant D and the neighborhood $\mathcal{S}$ of saddle point $(\boldsymbol{x}^*, \boldsymbol{y}^*)$

$$D := \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 + \|\boldsymbol{y}_0 - \boldsymbol{y}^*\|^2$$

$$\mathcal{S} := \left\{(\boldsymbol{x}, \boldsymbol{y}) : \|\boldsymbol{x} - \boldsymbol{x}^*\|^2 + \|\boldsymbol{y} - \boldsymbol{y}^*\|^2 \leq 2D\right\}$$

Primal-dual Gap

Define the constant D and the neighborhood $\mathcal{S}$ of saddle point $(\boldsymbol{x}^*, \boldsymbol{y}^*)$

$$D := \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 + \|\boldsymbol{y}_0 - \boldsymbol{y}^*\|^2$$

$$\mathcal{S} := \left\{(\boldsymbol{x}, \boldsymbol{y}) : \|\boldsymbol{x} - \boldsymbol{x}^*\|^2 + \|\boldsymbol{y} - \boldsymbol{y}^*\|^2 \leq 2D\right\}$$

Definition: Primal-dual Gap

For fixed $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{y}}$, the primal-dual gap is

$$\max_{\boldsymbol{y}:(\bar{\boldsymbol{x}},\boldsymbol{y})\in\mathcal{S}} f(\bar{\boldsymbol{x}}, \boldsymbol{y}) - \min_{\boldsymbol{x}:(\boldsymbol{x},\bar{\boldsymbol{y}})\in\mathcal{S}} f(\boldsymbol{x}, \bar{\boldsymbol{y}})$$

Remark:

- The primal-dual gap is zero iff $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ is a saddle point
- We also write the primal dual gap as

$$\left[\max_{\boldsymbol{y}:(\bar{\boldsymbol{x}},\boldsymbol{y})\in\mathcal{S}} f(\bar{\boldsymbol{x}}, \boldsymbol{y}) - f(\boldsymbol{x}^*, \boldsymbol{y}^*)\right] + \left[f(\boldsymbol{x}^*, \boldsymbol{y}^*) - \min_{\boldsymbol{x}:(\boldsymbol{x},\bar{\boldsymbol{y}})\in\mathcal{S}} f(\boldsymbol{x}, \bar{\boldsymbol{y}})\right]$$

Monotone Operator

Consider the minmax problem with convex-concave objective function

- Saddle point satisfies the first-order optimality condition

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}^*, \boldsymbol{y}^*) = \mathbf{0} \quad \text{and} \quad \nabla_{\boldsymbol{y}} f(\boldsymbol{x}^*, \boldsymbol{y}^*) = \mathbf{0}$$

- Define $\boldsymbol{z} := [\boldsymbol{x}; \boldsymbol{y}] \in \mathbb{R}^{m+n}$ and the monotone operator

$$F(\boldsymbol{z}) := [\nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{y}); -\nabla_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y})] \quad \Longrightarrow \quad F(\boldsymbol{z}^*) = \mathbf{0}$$

Definition: Monotone Operator

F is a monotone operator if for any $\boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathbb{R}^{m+n}$

$$\langle F(\boldsymbol{z}_1) - F(\boldsymbol{z}_2), \boldsymbol{z}_1 - \boldsymbol{z}_2 \rangle \geq 0$$

Remark: If $h : \mathbb{R}^n \to \mathbb{R}$ is convex, then $\nabla h : \mathbb{R}^n \to \mathbb{R}^n$ is monotone

Outline

GDA (I)

Algorithm: Gradient Descent Ascent

- **Initialization:** $\boldsymbol{x}_0 \in \mathbb{R}^m, \boldsymbol{y}_0 \in \mathbb{R}^n$ and step size $\eta > 0$
- **Iteration:**

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k, \boldsymbol{y}_k) \qquad \text{Gradient Descent}$$

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \eta \nabla_{\boldsymbol{y}} f(\boldsymbol{x}_k, \boldsymbol{y}_k) \qquad \text{Gradient Ascent}$$

GDA (I)

Algorithm: Gradient Descent Ascent

- **Initialization:** $\boldsymbol{x}_0 \in \mathbb{R}^m, \boldsymbol{y}_0 \in \mathbb{R}^n$ and step size $\eta > 0$
- **Iteration:**

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k, \boldsymbol{y}_k) \qquad \text{Gradient Descent}$$

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \eta \nabla_{\boldsymbol{y}} f(\boldsymbol{x}_k, \boldsymbol{y}_k) \qquad \text{Gradient Ascent}$$

- Even for the simplest case, GDA diverges
- Consider the following bilinear problem

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

- The GDA updates for this problem

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta \boldsymbol{y}_k$$

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \eta \boldsymbol{x}_k$$

GDA (II)

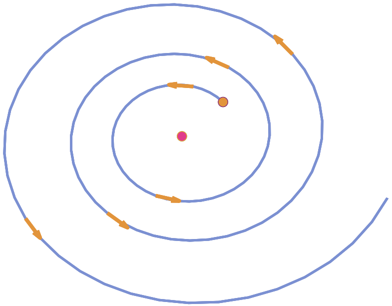
- The GDA updates for this problem

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta \boldsymbol{y}_k \quad \text{and} \quad \boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \eta \boldsymbol{x}_k$$

- At the k-th of GDA, we have

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

- GDA diverges because $1 + \eta^2 > 1$



• Saddle $(0, 0)$ • Initial $(10, 10)$

Outline

PPA (I)

Algorithm: Proximal Point Algorithm

- **Initialization:** $\boldsymbol{x}_0 \in \mathbb{R}^m, \boldsymbol{y}_0 \in \mathbb{R}^n$ and step size $\eta > 0$
- **Iteration:** The pair $(\boldsymbol{x}_{k+1}, \boldsymbol{y}_{k+1})$ is the unique solution to

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

PPA (I)

Algorithm: Proximal Point Algorithm

- **Initialization:** $\boldsymbol{x}_0 \in \mathbb{R}^m, \boldsymbol{y}_0 \in \mathbb{R}^n$ and step size $\eta > 0$
- **Iteration:** The pair $(\boldsymbol{x}_{k+1}, \boldsymbol{y}_{k+1})$ is the unique solution to

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

Remark: Iterative steps of PPA can be written as

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

Different from GDA steps

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

PPA (II)

- PPA for $f(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\mathsf{T}\boldsymbol{y}$

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

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

- At the k-th iteration of PPA, we have

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

PPA (II)

- PPA for $f(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}^\top \boldsymbol{y}$

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

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

- At the k-th iteration of PPA, we have

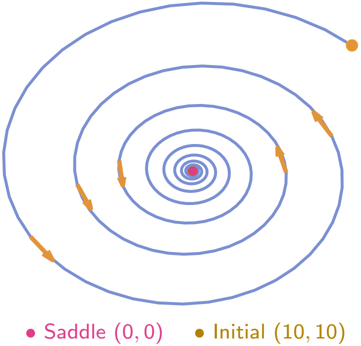
$$\|\boldsymbol{x}_{k+1}\|^2 + \|\boldsymbol{y}_{k+1}\|^2 = \frac{1}{1+\eta^2}(\|\boldsymbol{x}_k\|^2 + \|\boldsymbol{y}_k\|^2)$$

- True iterative steps

$$\boldsymbol{x}_{k+1} = \frac{\boldsymbol{x}_k - \eta \boldsymbol{y}_k}{1+\eta^2}$$

$$\boldsymbol{y}_{k+1} = \frac{\boldsymbol{y}_k + \eta \boldsymbol{x}_k}{1+\eta^2}$$

- PPA converges to saddle point



PPA (III)

- Let iterates $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ be generated by PPA with step size η
- Define the averaged iterates $(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{y}}_k)$ as

$$\bar{\boldsymbol{x}}_k := \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{x}_i \quad \text{and} \quad \bar{\boldsymbol{y}}_k := \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{y}_i$$

Theorem: Convergence of Averaged Iterates

- If f is convex-concave and L-smooth
- Then, we have

$$\max_{\boldsymbol{y}:(\bar{\boldsymbol{x}}_k, \boldsymbol{y}) \in \mathcal{S}} f(\bar{\boldsymbol{x}}_k, \boldsymbol{y}) - \min_{\boldsymbol{x}:(\boldsymbol{x}, \bar{\boldsymbol{y}}_k) \in \mathcal{S}} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_k) \leq \frac{D}{\eta k}$$

Remark: PPA involves operator inversion and is not easy to implement

Require: Efficient algorithms that behave like PPA!

Outline

OGDA (I)

Algorithm: Optimistic Gradient Descent Ascent

- **Initialization:** $\boldsymbol{x}_0 \in \mathbb{R}^m, \boldsymbol{y}_0 \in \mathbb{R}^n$ and step size $\eta > 0$
- **Iteration:**

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

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

OGDA (I)

Algorithm: Optimistic Gradient Descent Ascent

- **Initialization:** $\boldsymbol{x}_0 \in \mathbb{R}^m, \boldsymbol{y}_0 \in \mathbb{R}^n$ and step size $\eta > 0$
- **Iteration:**

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

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

Remark: OGDA can be seen as PPA with error term

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

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

Approximate using linear extrapolation of the previous gradients

$$\nabla f(\boldsymbol{x}_{k+1}, \boldsymbol{y}_{k+1}) \approx \nabla f(\boldsymbol{x}_k, \boldsymbol{y}_k) + [\nabla f(\boldsymbol{x}_k, \boldsymbol{y}_k) - \nabla f(\boldsymbol{x}_{k-1}, \boldsymbol{y}_{k-1})]$$

OGDA (II)

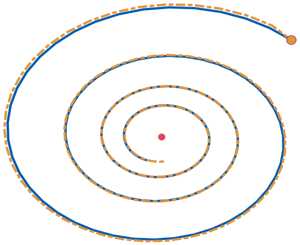
Algorithm: Optimistic Gradient Descent Ascent

- **Initialization:** $\boldsymbol{x}_0 \in \mathbb{R}^m, \boldsymbol{y}_0 \in \mathbb{R}^n$ and step size $\eta > 0$
- **Iteration:**

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

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

- Consider $f(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top \boldsymbol{y}$
- Convergence paths are similar
- OGDA approximates PPA



OGDA (III)

- Let iterates $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ be generated by OGDA with step size η
- Define the averaged iterates $(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{y}}_k)$ as

$$\bar{\boldsymbol{x}}_k := \frac{1}{k}\sum\nolimits_{i=1}^{k} \boldsymbol{x}_i \quad \text{and} \quad \bar{\boldsymbol{y}}_k := \frac{1}{k}\sum\nolimits_{i=1}^{k} \boldsymbol{y}_i$$

Theorem: Convergence of Averaged Iterates

- If f is convex-concave and L-smooth
- Then, we have

$$\max_{\boldsymbol{y}:(\bar{\boldsymbol{x}}_k,\boldsymbol{y})\in\mathcal{S}} f(\bar{\boldsymbol{x}}_k, \boldsymbol{y}) - \min_{\boldsymbol{x}:(\boldsymbol{x},\bar{\boldsymbol{y}}_k)\in\mathcal{S}} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_k) \leq \frac{5D}{\eta k}$$

Remark:

- OGDA is an implementable version of PPA
- OGDA enjoys similar convergence guarantee $\mathcal{O}(1/k)$

Outline

EG (I)

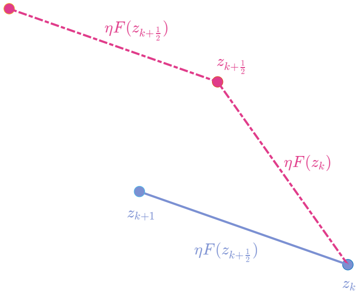
Algorithm: Extragradient Method

- **Initialization:** $\boldsymbol{x}_0 \in \mathbb{R}^m, \boldsymbol{y}_0 \in \mathbb{R}^n$ and step size $\eta > 0$
- **Iteration:**

$$\boldsymbol{z}_{k+\frac{1}{2}} = \boldsymbol{z}_k - \eta F(\boldsymbol{z}_k)$$

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k - \eta F(\boldsymbol{z}_{k+\frac{1}{2}})$$

- Define vector $\boldsymbol{z} := [\boldsymbol{x}; \boldsymbol{y}]$
- Define the operator F as
  $F(\boldsymbol{z}) := [\nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{y}); -\nabla_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y})]$
- EG utilizes the gradient of midpoint to update



EG (II)

Algorithm: Extragradient Method

- **Initialization:** $\boldsymbol{x}_0 \in \mathbb{R}^m, \boldsymbol{y}_0 \in \mathbb{R}^n$ and step size $\eta > 0$
- **Iteration:**

$$\boldsymbol{z}_{k+\frac{1}{2}} = \boldsymbol{z}_k - \eta F(\boldsymbol{z}_k)$$

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k - \eta F(\boldsymbol{z}_{k+\frac{1}{2}})$$

- Updates can be written as

$$\boldsymbol{z}_{k+\frac{1}{2}} = \boldsymbol{z}_{k-\frac{1}{2}} - \eta F(\boldsymbol{z}_{k-\frac{1}{2}}) - \eta[F(\boldsymbol{z}_k) - F(\boldsymbol{z}_{k-1})]$$

EG (II)

Algorithm: Extragradient Method

- **Initialization:** $\boldsymbol{x}_0 \in \mathbb{R}^m, \boldsymbol{y}_0 \in \mathbb{R}^n$ and step size $\eta > 0$
- **Iteration:**

$$\boldsymbol{z}_{k+\frac{1}{2}} = \boldsymbol{z}_k - \eta F(\boldsymbol{z}_k)$$

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k - \eta F(\boldsymbol{z}_{k+\frac{1}{2}})$$

- Updates can be written as

$$\boldsymbol{z}_{k+\frac{1}{2}} = \boldsymbol{z}_{k-\frac{1}{2}} - \eta F(\boldsymbol{z}_{k-\frac{1}{2}}) - \eta[F(\boldsymbol{z}_k) - F(\boldsymbol{z}_{k-1})]$$

- When the variations are close to each other, i.e.,

$$F(\boldsymbol{z}_k) - F(\boldsymbol{z}_{k-1}) \approx F(\boldsymbol{z}_{k+\frac{1}{2}}) - F(\boldsymbol{z}_{k-\frac{1}{2}})$$

  EG method approximates PPA

$$\boldsymbol{z}_{k+\frac{1}{2}} \approx \boldsymbol{z}_{k-\frac{1}{2}} - \eta F(\boldsymbol{z}_{k+\frac{1}{2}})$$

EG (III)

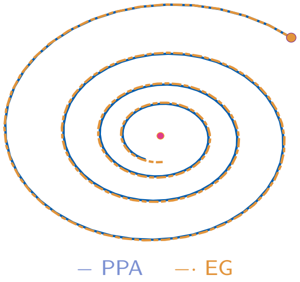
Algorithm: Extragradient Method

- **Initialization:** $\boldsymbol{x}_0 \in \mathbb{R}^m, \boldsymbol{y}_0 \in \mathbb{R}^n$ and step size $\eta > 0$
- **Iteration:**

$$\boldsymbol{z}_{k+\frac{1}{2}} = \boldsymbol{z}_k - \eta F(\boldsymbol{z}_k)$$

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k - \eta F(\boldsymbol{z}_{k+\frac{1}{2}})$$

- Consider $f(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\mathsf{T} \boldsymbol{y}$
- Convergence paths are similar
- EG approximates PPA



EG (IV)

- Let iterates $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ be generated by EG with step size η
- Define the averaged iterates $(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{y}}_k)$ as

$$\bar{\boldsymbol{x}}_k := \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{x}_i \quad \text{and} \quad \bar{\boldsymbol{y}}_k := \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{y}_i$$

Theorem: Convergence of Averaged Iterates

- If f is convex-concave and L-smooth
- Then, we have

$$\max_{\boldsymbol{y}:(\bar{\boldsymbol{x}}_k,\boldsymbol{y})\in\mathcal{S}} f(\bar{\boldsymbol{x}}_k, \boldsymbol{y}) - \min_{\boldsymbol{x}:(\boldsymbol{x},\bar{\boldsymbol{y}}_k)\in\mathcal{S}} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_k) \leq \frac{16D}{\eta k}$$

Remark:

- EG is an implementable version of PPA
- EG enjoys similar convergence guarantee $\mathcal{O}(1/k)$

Last Iterate Convergence

The averaged iterate is not always what we want!

- Imagine that we are seeking for a sparse solution $\boldsymbol{x}^*$
- Assume $\bar{\boldsymbol{x}} := (\boldsymbol{x}_1 + \boldsymbol{x}_2 + \boldsymbol{x}_3)/3$ reaches ε-accuracy

$$\boldsymbol{x}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \boldsymbol{x}_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \quad \boldsymbol{x}_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \bar{\boldsymbol{x}} = \begin{bmatrix} 2/3 \\ 1 \\ 1/3 \end{bmatrix}$$

Last Iterate Convergence

The averaged iterate is not always what we want!

- Imagine that we are seeking for a sparse solution $\boldsymbol{x}^*$
- Assume $\bar{\boldsymbol{x}} := (\boldsymbol{x}_1 + \boldsymbol{x}_2 + \boldsymbol{x}_3)/3$ reaches ε-accuracy

$$\boldsymbol{x}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \boldsymbol{x}_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \quad \boldsymbol{x}_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \bar{\boldsymbol{x}} = \begin{bmatrix} 2/3 \\ 1 \\ 1/3 \end{bmatrix}$$

Theorem: Last Iterate Convergence

- Let iterates $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ be generated by EG/PPA
- If f is convex-concave and L-smooth
- Then, we have

$$\max_{\boldsymbol{y}:(\boldsymbol{x}^k,\boldsymbol{y})\in\mathcal{S}} f(\boldsymbol{x}^k, \boldsymbol{y}) - \min_{\boldsymbol{x}:(\boldsymbol{x},\boldsymbol{y}^k)\in\mathcal{S}} f(\boldsymbol{x}, \boldsymbol{y}^k) = \Theta\left(\frac{1}{\sqrt{k}}\right)$$

Remark: Slower than the averaged iterate results $\mathcal{O}(1/k)$

Conclusion

Motivation

Background

- ▷ Minmax Problems
- ▷ Convergence Measure

Algorithms

- ▷ Gradient Descent Ascent (GDA)
- ▷ Proximal Point Algorithm (PPA)
- ▷ Optimistic Gradient Descent Ascent (OGDA)
- ▷ Extragradient Method (EG)

Last Iterate