Projection onto convex sets

Andersen Ang

U.Southampton UK

angms.science

March 13, 2025

1st ver. MM DD, 2025

Content

Can we relax projection in projected gradient descent?

POCS: Projection Onto Convex Sets

Toy example in $\mathbb{R}^2$

Why relaxation is bad

Convergence of POCS

Acceleration

Convex Feasibility Problem

Extra topics

Contents

Can we relax projection in projected gradient descent?

POCS: Projection Onto Convex Sets

Toy example in $\mathbb{R}^2$

Why relaxation is bad

Convergence of POCS

Acceleration

Convex Feasibility Problem

Extra topics

Can we relax projection in Projected gradient descent?

- Goal: solve

$$\underset{\boldsymbol{x} \in C}{\operatorname{argmin}} \; f(\boldsymbol{x})$$

by projected gradient descent

$$\boldsymbol{x}_{k+1} = \operatorname{proj}_C\Big(\boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k)\Big)$$

- Here is a question: performing proj_C in each iteration is expensive, why don't we do this

Algorithm 1: "Relaxed PGD"

```
Get y = argmin_{x ∈ R^n} f(x)      // relxation: ignore C
Get x* = proj(y)                   // project back onto C
```

Get $\boldsymbol{y} = \underset{\boldsymbol{x} \in \mathbb{R}^n}{\operatorname{argmin}} \; f(\boldsymbol{x})$ // relxation: ignore C

Get $\boldsymbol{x}^* = \operatorname{proj}(\boldsymbol{y})$ // project back onto C

This algorithm will not work. We will explain why.

Contents

Can we relax projection in projected gradient descent?

POCS: Projection Onto Convex Sets

Toy example in $\mathbb{R}^2$

Why relaxation is bad

Convergence of POCS

Acceleration

Convex Feasibility Problem

Extra topics

POCS: Projection Onto Convex Sets

- Consider projection onto $C = C_1 \cap C_2$

$$\operatorname{proj}_C(\boldsymbol{x}) \;=\; \underset{\boldsymbol{\xi} \in C_1 \cap C_2}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{\xi} - \boldsymbol{x}\|_2^2$$

  assume

  - both C_1, C_2 are nonempty convex closed subset of $\mathbb{R}^n$
  - $\operatorname{proj}_{C_1}$ and $\operatorname{proj}_{C_2}$ can be computed cheap
  - $C_1 \cap C_2 \neq \varnothing$, otherwise we project to empty-set
  - $C_1 \nsubseteq C_2$ and $C_2 \nsubseteq C_1$, otherwise $C = C_1$ or C_2 and the problem is trivial

- Problem $\operatorname{proj}_C(\boldsymbol{x})$ looks innocent, this problem is actually not simple at all !!!
  - This problem is actually a key challenge in research.

Contents

Can we relax projection in projected gradient descent?

POCS: Projection Onto Convex Sets

Toy example in $\mathbb{R}^2$

Why relaxation is bad

Convergence of POCS

Acceleration

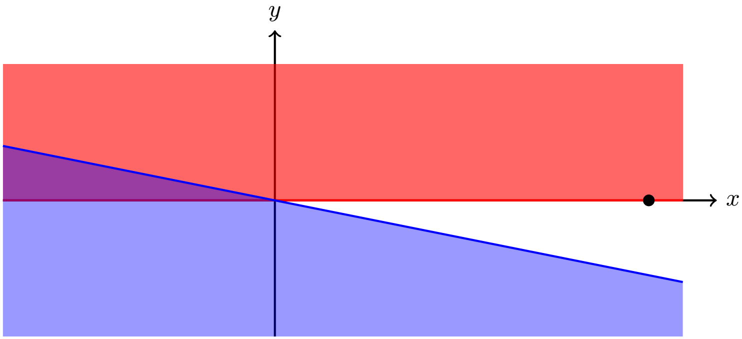
Convex Feasibility Problem

Extra topics

Toy example in $\mathbb{R}^2$ – setup

- We consider $\mathbb{R}^2$ and we have $\boldsymbol{x} \in (x, y)$

- $C = C_1 \cap C_2$, where $C_1 = \left\{(x, y) \ : \ x \leq -5y\right\}$ and $C_2 = \left\{(x, y) \ : \ y \geq 0\right\}$



C_1 is a halfspace, it is a convex set
C_2 is a halfspace, it is a convex set
C is a convex set

- By looking at the figure, we see that $\text{proj}_C(10, 0) = (0, 0)$

Toy example in $\mathbb{R}^2$ – POCS algorithm

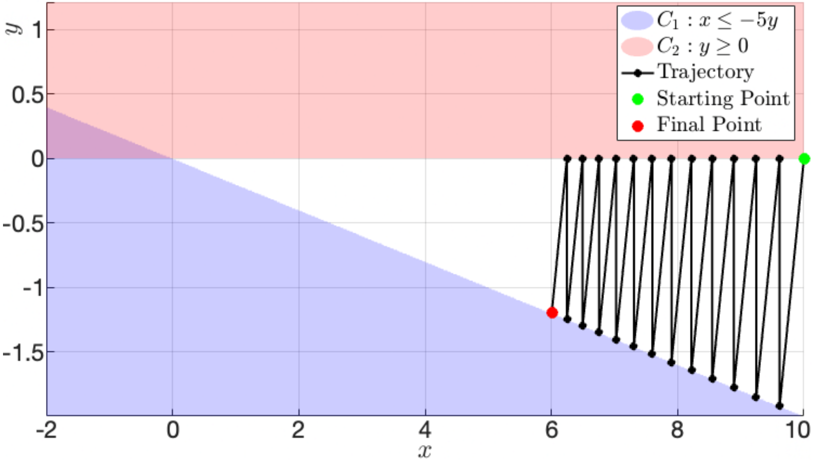
Algorithm 2: POCS

for $k = 1, 2, \ldots$ **do**

$$\boldsymbol{x}_{k+\frac{1}{2}} = \text{proj}_{C_1}(\boldsymbol{x}_k)$$

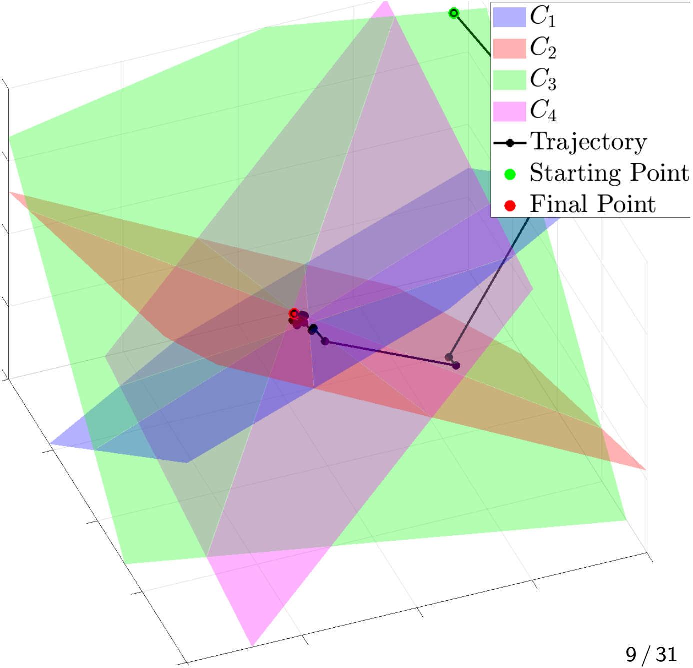
$$\boldsymbol{x}_{k+1} = \text{proj}_{C_2}\left(\boldsymbol{x}_{k+\frac{1}{2}}\right)$$

If 2 sets in $\mathbb{R}^2$ is already this slow, image the complexity of the general case of $\mathbb{R}^n$ with m sets



Example in $\mathbb{R}^3$, 4 hyperplanes

Projection onto the intersection of hyperplane is called Kaczmarz method in linear algebra



Contents

Can we relax projection in projected gradient descent?

POCS: Projection Onto Convex Sets

Toy example in $\mathbb{R}^2$

Why relaxation is bad

Convergence of POCS

Acceleration

Convex Feasibility Problem

Extra topics

Why relaxation is bad

- We want to solve $\operatorname*{argmin}_{\boldsymbol{x} \in C} f(\boldsymbol{x})$.
  - this is same as finding a point in the set $\mathcal{X}_{\mathbb{R}^n \cap C} := \operatorname*{argmin}_{\boldsymbol{x} \in C} f(\boldsymbol{x})$

- Define the set $\mathcal{X}_{\mathbb{R}^n} := \operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x})$
  - note that it is unconstrained
  - Theorem: f is convex $\implies$ sublevel set $\operatorname{lev}_{\leq \alpha} f$ is a convex set
  - Definition: $\mathcal{X}_{\mathbb{R}^n}$ is a special case of $\operatorname{lev}_{\leq \alpha} f$ with $\alpha = \inf f$
  - thus $\mathcal{X}_{\mathbb{R}^n}$ is a convex subset of $\mathbb{R}^n$

- Now we can view $\operatorname*{argmin}_{\boldsymbol{x} \in C} f(\boldsymbol{x})$ as $\operatorname{proj}_{\mathcal{X}_{\mathbb{R}^n \cap C}}(\boldsymbol{x}_0)$

- Relaxed PGD is just POCS with one iteration.
  - $\operatorname{proj}_{\mathcal{X}_{\mathbb{R}^n}}$ means we solve $\operatorname{argmin} f$ ignoring C
  - proj_C means we project onto C ignoring $\mathcal{X}_{\mathbb{R}^n}$
  - the operation proj_C can drag the point $\operatorname{proj}_{\mathcal{X}_{\mathbb{R}^n}}(\boldsymbol{x}_0)$ far away from $\mathcal{X}_{\mathbb{R}^n \cap C}$

Contents

Can we relax projection in projected gradient descent?

POCS: Projection Onto Convex Sets

Toy example in $\mathbb{R}^2$

Why relaxation is bad

Convergence of POCS

Acceleration

Convex Feasibility Problem

Extra topics

Math details: Convergence of POCS

- Back to the $\mathbb{R}^2$ example: project $(10,0)$ to $C_1 = \left\{(x,y) \;:\; x \leq -5y\right\}$ and $C_2 = \left\{(x,y) \;:\; y \geq 0\right\}$

- The iteration is

$$\boldsymbol{x}_{k+1} = \mathrm{proj}_{C_2}\Big(\mathrm{proj}_{C_1}(\boldsymbol{x}_k)\Big)$$

- proj_{C_1} and proj_{C_2} are projection onto halfspaces

$$\mathrm{proj}_H(\boldsymbol{x}) \coloneqq \underset{\boldsymbol{\xi}}{\mathrm{argmin}}\ \frac{1}{2}\|\boldsymbol{\xi} - \boldsymbol{x}\|_2^2 \quad \text{s.t.} \quad \langle \boldsymbol{\xi}, \boldsymbol{a}\rangle \leq b$$

  with the solution

$$\boldsymbol{\xi} = \begin{cases} \boldsymbol{x} + \dfrac{b - \langle \boldsymbol{a}, \boldsymbol{x}\rangle}{\|\boldsymbol{a}\|_2^2}\boldsymbol{a} & \boldsymbol{x} \notin H \\ \boldsymbol{x} & \boldsymbol{x} \in H \end{cases}$$

- For C_1 we have $\boldsymbol{a} = (1,5)$, and for C_2 we have $\boldsymbol{a} = (0,-1)$

Simplifying $\boldsymbol{x}_{k+1} = \mathbf{proj}_{C_2}\Big(\mathbf{proj}_{C_1}(\boldsymbol{x}_k)\Big)$

- The formula of projection onto halfspace is

$$\mathrm{proj}_H(\boldsymbol{x}) \coloneqq \underset{\boldsymbol{\xi}}{\mathrm{argmin}}\ \frac{1}{2}\|\boldsymbol{\xi} - \boldsymbol{x}\|_2^2 \quad \text{s.t.} \quad \langle \boldsymbol{\xi}, \boldsymbol{a} \rangle \le b, \qquad \boldsymbol{\xi} = \begin{cases} \boldsymbol{x} + \dfrac{b - \langle \boldsymbol{a}, \boldsymbol{x} \rangle}{\|\boldsymbol{a}\|_2^2}\boldsymbol{a} & \boldsymbol{x} \notin H \\ \boldsymbol{x} & \boldsymbol{x} \in H \end{cases}$$

- The set $C = C_1 \cap C_2$ can be expressed as $\boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}$

$$\begin{bmatrix} 1 & 5 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \le \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \text{do not confuse coordinate } x, y \text{ with vector } \boldsymbol{x}$$

- We have $b = 0$ for proj_{C_1} and proj_{C_2}

$$\mathrm{proj}_H(\boldsymbol{x}) \coloneqq \underset{\boldsymbol{\xi}}{\mathrm{argmin}}\ \frac{1}{2}\|\boldsymbol{\xi} - \boldsymbol{x}\|_2^2 \quad \text{s.t.} \quad \langle \boldsymbol{\xi}, \boldsymbol{a} \rangle \le 0, \qquad \boldsymbol{\xi} = \begin{cases} \boldsymbol{x} - \langle \hat{\boldsymbol{a}}, \boldsymbol{x} \rangle \hat{\boldsymbol{a}} & \boldsymbol{x} \notin H \\ \boldsymbol{x} & \boldsymbol{x} \in H \end{cases}, \quad \hat{\boldsymbol{a}} = \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|}$$

Simplifying $\mathrm{proj}_{C_1}(\boldsymbol{x}_k)$

$$\mathrm{proj}_H(\boldsymbol{x}) := \underset{\boldsymbol{\xi}}{\mathrm{argmin}} \ \frac{1}{2}\|\boldsymbol{\xi} - \boldsymbol{x}\|_2^2 \quad \text{s.t.} \quad \langle \boldsymbol{\xi}, \boldsymbol{a} \rangle \leq 0, \qquad \boldsymbol{\xi} = \begin{cases} \boldsymbol{x} - \langle \hat{\boldsymbol{a}}, \boldsymbol{x} \rangle \hat{\boldsymbol{a}} & \boldsymbol{x} \notin H \\ \boldsymbol{x} & \boldsymbol{x} \in H \end{cases}, \qquad \hat{\boldsymbol{a}} = \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|}$$

- Consider proj_{C_1}

$$\mathrm{proj}_{C_1}(\boldsymbol{x}) = \begin{cases} \boldsymbol{x} - \langle \hat{\boldsymbol{a}}_1, \boldsymbol{x} \rangle \hat{\boldsymbol{a}}_1 & \boldsymbol{x} \notin C_1 \\ \boldsymbol{x} & \boldsymbol{x} \in C_1 \end{cases}, \qquad (*)$$

  which is a nonlinear expression because it has two parts

- We simplify the nonlinear expression $(*)$ by geometry (this is not a math proof but only an explanation): from the previous figure, we know that after proj_{C_2}, the point must be outside C_1, hence the second case in $(*)$ will never occur, and thus

$$\mathrm{proj}_{C_1}(\boldsymbol{x}) \ = \ \boldsymbol{x} - \langle \hat{\boldsymbol{a}}_1, \boldsymbol{x} \rangle \hat{\boldsymbol{a}}_1 \ \overset{\mathsf{T}}{=} \ \boldsymbol{x} - (\hat{\boldsymbol{a}}_1 \otimes \hat{\boldsymbol{a}}_1)\boldsymbol{x} \ = \ (\boldsymbol{I} - \hat{\boldsymbol{a}}_1 \hat{\boldsymbol{a}}_1^\top)\boldsymbol{x} \ = \ \boldsymbol{A}_1 \boldsymbol{x}$$

  step T is a tensor product formula in multilinear algebra

Simplifying proj_{C_2}

$$\mathrm{proj}_H(\boldsymbol{x}) := \underset{\boldsymbol{\xi}}{\mathrm{argmin}}\ \frac{1}{2}\|\boldsymbol{\xi} - \boldsymbol{x}\|_2^2 \quad \text{s.t.} \quad \langle \boldsymbol{\xi}, \boldsymbol{a} \rangle \leq 0, \qquad \boldsymbol{\xi} = \begin{cases} \boldsymbol{x} - \langle \hat{\boldsymbol{a}}, \boldsymbol{x} \rangle \hat{\boldsymbol{a}} & \boldsymbol{x} \notin H \\ \boldsymbol{x} & \boldsymbol{x} \in H \end{cases}, \qquad \hat{\boldsymbol{a}} = \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|}$$

$$\mathrm{proj}_{C_1}(\boldsymbol{x}) \;=\; (\boldsymbol{I} - \hat{\boldsymbol{a}}_1 \hat{\boldsymbol{a}}_1^\top)\boldsymbol{x} \;=\; \boldsymbol{A}_1 \boldsymbol{x}$$

- Consider proj_{C_2}.

  From the previous figure, we know that after proj_{C_1}, the point must be outside C_2, so y-part is negative, so

$$\mathrm{proj}_{C_2}(\boldsymbol{x}) \;=\; \boldsymbol{x} - \langle \hat{\boldsymbol{a}}_2, \boldsymbol{x} \rangle \hat{\boldsymbol{a}}_2 \;=\; \begin{bmatrix} x \\ y \end{bmatrix} - \left\langle \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \begin{bmatrix} x \\ y \end{bmatrix} \right\rangle \begin{bmatrix} 0 \\ -1 \end{bmatrix} \;=\; \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ -y \end{bmatrix} \;=\; \begin{bmatrix} x \\ 0 \end{bmatrix}.$$

  Meaning: if y is negative, replace it with zero. We do not touch x.

Non-expansive and firmly non-expansive

- Projection onto a set Q is non-expansive[1] (1-Lipschitz)

$$\left\|\mathrm{proj}_Q(\boldsymbol{x}) - \mathrm{proj}_Q(\boldsymbol{y})\right\| \;\le\; \left\|\boldsymbol{x} - \boldsymbol{y}\right\|$$

nonexpansiveness comes from the Bourbaki-Cheney-Goldstein inequality.

- Now we have

$$\begin{aligned}
\left\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\right\| &= \left\|\mathrm{proj}_{C_2}\mathrm{proj}_{C_1}(\boldsymbol{x}_k) - \mathrm{proj}_{C_2}\mathrm{proj}_{C_1}(\boldsymbol{x}^*)\right\| \\
&\le \left\|\mathrm{proj}_{C_1}(\boldsymbol{x}_k) - \mathrm{proj}_{C_1}(\boldsymbol{x}^*)\right\| \\
&\le \left\|\boldsymbol{x}_k - \boldsymbol{x}^*\right\| \\
&= \left\|\mathrm{proj}_{C_2}\mathrm{proj}_{C_1}(\boldsymbol{x}_{k-1}) - \mathrm{proj}_{C_2}\mathrm{proj}_{C_1}(\boldsymbol{x}^*)\right\| \\
&\le \left\|\mathrm{proj}_{C_1}(\boldsymbol{x}_{k-1}) - \mathrm{proj}_{C_1}(\boldsymbol{x}^*)\right\| \\
&\le \left\|\boldsymbol{x}_{k-1} - \boldsymbol{x}^*\right\| \\
&\;\;\vdots \\
&\le \left\|\boldsymbol{x}_0 - \boldsymbol{x}^*\right\|
\end{aligned}$$

[1] Actually we have stronger result here: Projection onto a convex set Q is firmly non-expansive $\left\|\mathrm{proj}_Q(\boldsymbol{x}) - \mathrm{proj}_Q(\boldsymbol{y})\right\| \;\le\; \left\langle \mathrm{proj}_Q(\boldsymbol{x}) - \mathrm{proj}_Q(\boldsymbol{y}),\; \boldsymbol{x} - \boldsymbol{y}\right\rangle$.

Convergence of projected sequence

- $\text{proj}_{C_1}(\boldsymbol{x}) = (\boldsymbol{I} - \hat{\boldsymbol{a}}_1 \hat{\boldsymbol{a}}_1^\top)\boldsymbol{x} = \boldsymbol{A}_1 \boldsymbol{x}$

- $\text{proj}_{C_2}(\boldsymbol{x}) = \begin{bmatrix} x \\ 0 \end{bmatrix}.$

- $\boldsymbol{x}_{k+1} = \text{proj}_{C_2} \text{proj}_{C_1}(\boldsymbol{x}_k)$

- We have $0 \leq \left\| \boldsymbol{x}_k - \boldsymbol{x}^* \right\| \leq \cdots \leq \left\| \boldsymbol{x}_0 - \boldsymbol{x}^* \right\|$, this is known as Fejér monotonicity

- Math fact (real analysis): if a sequence is monotonically decreasing and bounded below, it is convergent.

- $\lim_{k \to \infty} \left\| \boldsymbol{x}_k - \boldsymbol{x}^* \right\| = 0 \quad \Longrightarrow \quad \boldsymbol{x}_k \to \boldsymbol{x}^*$, here we know $\boldsymbol{x}^*$ exists and $\boldsymbol{x}^* = (0, 0)$

Convergence speed

- With starting point $(10, 0)$, the iteration $\boldsymbol{x}_{k+1} = \text{proj}_{C_2}\text{proj}_{C_1}(\boldsymbol{x}_k)$ always has $y_k = 0$ so we focus on x_k

$$\begin{aligned} x_{k+1} = (\boldsymbol{A}_1\boldsymbol{x})_1 &= \left(\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \frac{1}{26} \begin{bmatrix} 1 & 5 \\ 5 & 25 \end{bmatrix} \right) \begin{bmatrix} x_k \\ 0 \end{bmatrix} \right)_1 \\ &= \left(\begin{bmatrix} \frac{25}{26} & -\frac{5}{26} \\ -\frac{5}{26} & \frac{1}{26} \end{bmatrix} \begin{bmatrix} x_k \\ 0 \end{bmatrix} \right)_1 \\ &= \frac{25}{26} x_k = \left(\frac{25}{26}\right)^2 x_{k-1} = \dots \end{aligned}$$

the fraction $\frac{25}{26} = 1 - \sin^2\theta$ is the contraction factor, and $\sin\theta = \frac{1}{\sqrt{26}}$ is the angle between constraints

- Thus reaching ϵ-precision to 0 will requires

$$\left(\frac{25}{26}\right)^k \leq \epsilon \implies k \underbrace{\log \frac{25}{26}}_{<0} \leq \log \epsilon \implies k \geq \frac{\log \epsilon}{\left|\log \frac{25}{26}\right|} \implies k = \lceil 58 \log \epsilon \rceil$$

Example: $\epsilon = 10^{-12}$ gives $k \geq 705$ iterations.

Convergence

- **Theorem (von Neumann 1933)**[2]
  $C_1, C_2 \subset \mathbb{R}^n$ closed subsets, then for any $\boldsymbol{x}_0$, the sequence $\boldsymbol{x}_{k+1} = \mathrm{proj}_{C_2}\Big(\mathrm{proj}_{C_1}(\boldsymbol{x}_k)\Big)$ converges in norm to $\mathrm{proj}_{C_1 \cap C_2}(\boldsymbol{x}_k)$

- **Theorem (Bregman 1965)**
  $C_1, C_2 \subset \mathbb{R}^n$ closed convex subsets with, then for any $\boldsymbol{x}_0$, the sequence $\boldsymbol{x}_{k+1} = \mathrm{proj}_{C_2}\Big(\mathrm{proj}_{C_1}(\boldsymbol{x}_k)\Big)$ converges weakly to a point $\boldsymbol{y} \in C_1 \cap C_2$

[2] von Neumann actually prove it for the general notion in Hilbert space

Contents

Can we relax projection in projected gradient descent?

POCS: Projection Onto Convex Sets

Toy example in $\mathbb{R}^2$

Why relaxation is bad

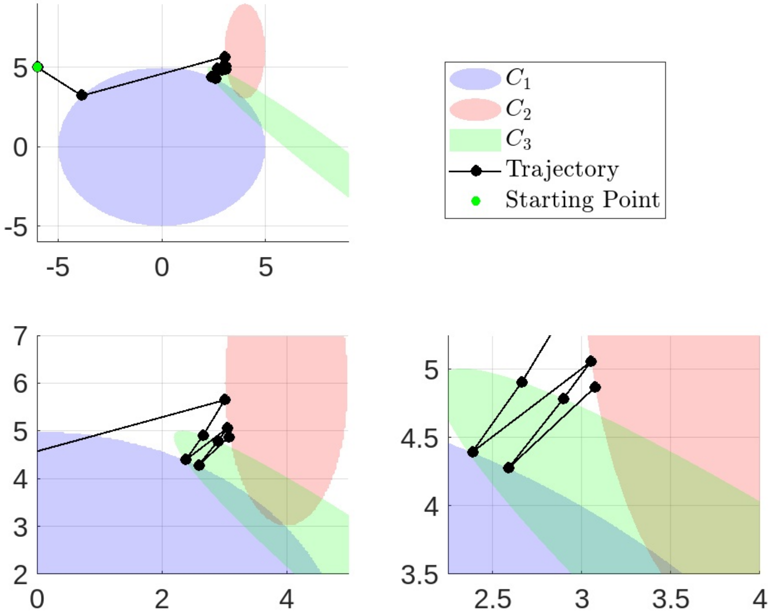
Convergence of POCS

Acceleration

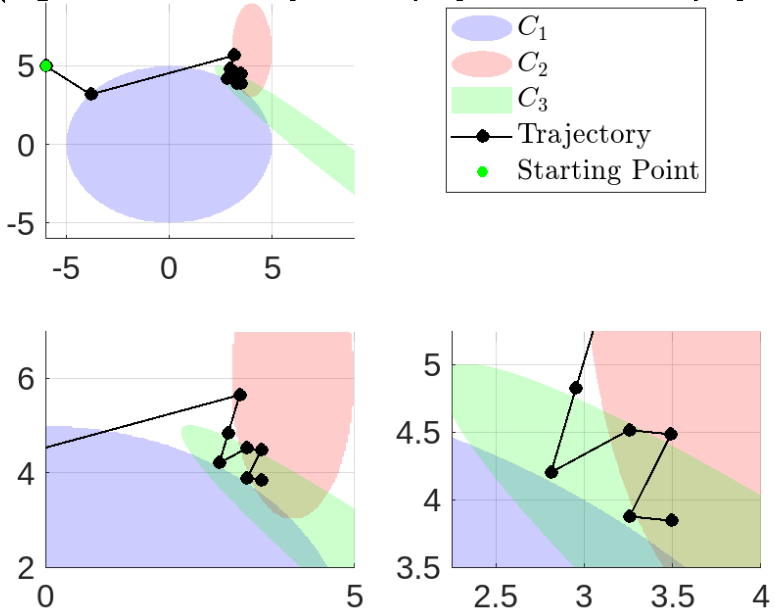
Convex Feasibility Problem

Extra topics

Example in $\mathbb{R}^2$ with 3 sets: slow



Acceleration (e.g. momentum, Dykstra's projection, inexact projection)



Contents

Can we relax projection in projected gradient descent?

POCS: Projection Onto Convex Sets

Toy example in $\mathbb{R}^2$

Why relaxation is bad

Convergence of POCS

Acceleration

Convex Feasibility Problem

Extra topics

Convex Feasibility Problem (CFP)

- Given convex sets $C_1, C_2, \ldots, C_N$ that are all nonempty closed subset of $\mathbb{R}^n$, the CFP is

$$\text{find } \boldsymbol{x} \in \bigcap_i C_i \tag{CFP}$$

- CFP is a special case of convex optimization

$$\begin{aligned} \underset{\boldsymbol{x}}{\operatorname{argmin}} \quad & 0 \\ \text{s.t.} \quad & \boldsymbol{x} \in C := \bigcap_i C_i \end{aligned}$$

  the objective function here is zero all the time, zero gradient, zero Hessian, zero Lipschitz ...

- The message:

  any "find a point problem" is an optimization problem

  Now you should understand why I said this problem is actually not simple at all in page 5

Any "find a point problem" is an optimization problem in the form of CFP

- Given a broken image $\boldsymbol{b}$, we want to recover the image as

$$\text{find } \boldsymbol{x} \in \bigcap_i C_i$$

- $C_1 : \{\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 \leq \epsilon\}$ the recovered image $\boldsymbol{A}\boldsymbol{x}$ "looks like" the broken image
- $C_2 : \{\max_i (\boldsymbol{A}\boldsymbol{x})_i \leq u\}$ the brightest pixel in the recovered image cannot be brighter than a limit u
- $C_3 : \{\max_i (\boldsymbol{A}\boldsymbol{x})_i \geq l\}$ the darkest pixel in the recovered image cannot be darker than a limit l
- $C_4 : \{\max_i \|\nabla \boldsymbol{x}\| \geq \sigma\}$ the recovered image cannot be too spiky

- Solve this CFP you solve image recovery problem

Contents

Can we relax projection in projected gradient descent?

POCS: Projection Onto Convex Sets

Toy example in $\mathbb{R}^2$

Why relaxation is bad

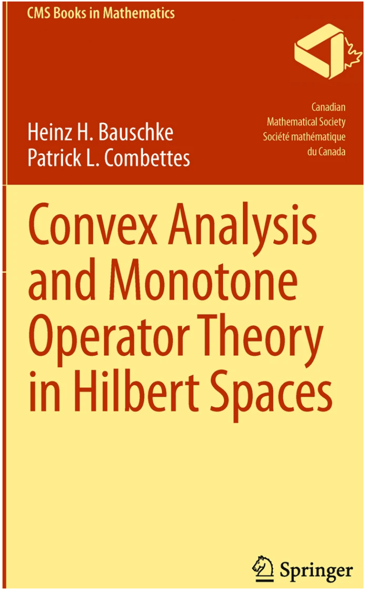
Convergence of POCS

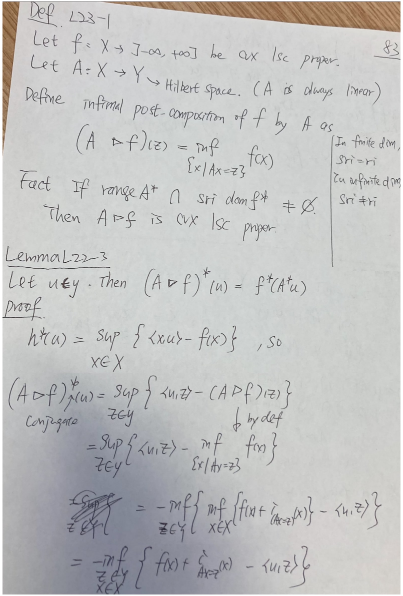
Acceleration

Convex Feasibility Problem

Extra topics

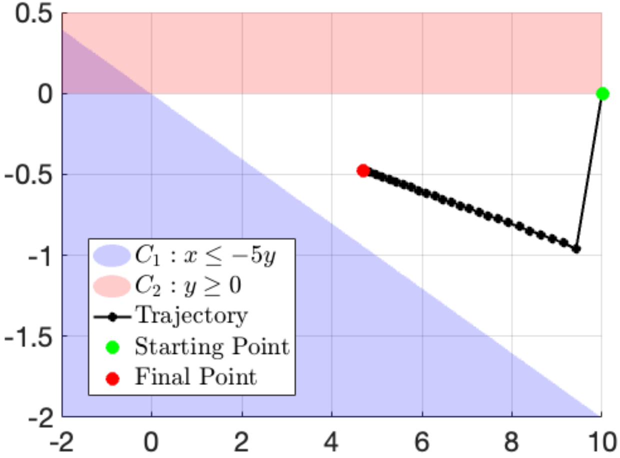
Don't think that projection is something trivial: this is a research area





Douglas-Rachford operator (1950s)

$$\text{reflect}_C(\boldsymbol{x}) := 2\text{proj}_C(\boldsymbol{x}) - \boldsymbol{x}, \qquad \text{DR}_{C_1,C_2}(\boldsymbol{x}) := \frac{\boldsymbol{x} + \text{reflect}_{C_2}\left(\text{reflect}_{C_1}(\boldsymbol{x}_k)\right)}{2}$$



Dykstra's method is a special case of Douglas-Rachford

Consensus optimization

- Big-sum minimization

$$\underset{\boldsymbol{x}}{\operatorname{argmin}}\ f_1(\boldsymbol{x}) + f_2(\boldsymbol{x}) + \ldots + f_N(\boldsymbol{x})$$

  we have one variable $\boldsymbol{x}$ shared over N functions

- consensus optimization

$$\begin{aligned} \underset{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N}{\operatorname{argmin}} \quad & f_1(\boldsymbol{x}_1) + f_2(\boldsymbol{x}_2) + \ldots + f_N(\boldsymbol{x}_N) \\ \text{s.t.} \quad & (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N) \in C \coloneqq \Big\{ \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N \ : \ \boldsymbol{v}_1 = \boldsymbol{v}_2 = \ldots = \boldsymbol{v}_N \Big\} \end{aligned}$$

  Distributed algorithm

  - step 1: N parallel updates
  - step 2: coordination
  - step 3: go back to step 1

Last page: summary

Can we relax projection in projected gradient descent?

POCS: Projection Onto Convex Sets

Toy example in $\mathbb{R}^2$

Why relaxation is bad

Convergence of POCS

Acceleration

Convex Feasibility Problem

Extra topics

End of document