Numerical optimization

Mines Nancy – Fall 2024

session 8 – constrained convex optimization II
Algorithms

Lecturer: Julien Flamant (CNRS, CRAN)

julien.flamant@univ-lorraine.fr

Office 425, FST 1er cycle

Course material:

arche.univ-lorraine.fr/course/view.php?id=74098

github.com/jflamant/mines-nancy-fall24-optimization

Setting: constrained convex optimization

We consider optimizations problems of the form

$$\min_{\mathbf{x}\in\Omega} f_0(\mathbf{x}) \tag{$\mathcal{P}$}$$

where we make the following assumptions:

- the objective $f_0 : \mathcal{D}_0 \subseteq \mathbb{R}^N \to \mathbb{R}$ is convex on its domain $\mathcal{D}_0$
- the set of constraints $\Omega \subset \mathbb{R}^N$ is convex. We assume that Ω is described by M inequality constraints and P equality constraints such that

$$\Omega = \left\{ \mathbf{x} \in \mathbb{R}^N \;\middle|\; \begin{array}{lcl} f_i(\mathbf{x}) \leq 0 & , & i = 1, \ldots, M \\ \mathbf{a}_i^\top \mathbf{x} = b_i & , & i = 1, \ldots, P \end{array} \right\}$$

  where the functions $f_i : \mathbb{R}^N \to \mathbb{R}$ are convex, hence Ω is a convex set. (prove it)

$(\mathcal{P})$ is a constrained convex optimization problem

Agenda

Session 7 - 03/12/24: Theory

- Characterization of solutions
- Lagrangian duality
- Karush-Kuhn-Tucker (KKT) conditions

Session 8 - 10/12/24: Algorithms

- Penalty methods
- Projected gradient
- Dual methods: Uzawa's algorithm
- Disciplined convex programming with CVXPY

Main reference

Convex Optimization, Boyd and Vanderberghe (2004) - Chapters 4, 5

Outline

Penalty method principle

Consider the optimization problem

$$\min_{\mathbf{x}\in\Omega} f_0(\mathbf{x}) \tag{$\mathcal{P}$}$$

where f_0 is the objective and $\Omega \subset \mathbb{R}^N$ is the constraint set.

Penalty method

Replace $(\mathcal{P})$ with the unconstrained optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^N} f_0(\mathbf{x}) + \rho p(\mathbf{x}), \quad \rho > 0 \tag{$\mathcal{P}_\rho$}$$

where $p : \mathbb{R}^N \to \mathbb{R}$ is the penalty function such that

- p is continuous
- $p(\mathbf{x}) > 0$ if $\mathbf{x} \notin \Omega$
- $p(\mathbf{x}) = 0$ if $\mathbf{x} \in \Omega$.

The penalty parameter ρ controls the cost of violating the constraints.

Choosing penalty functions

Recall Ω is often parameterized by equality and inequality constraints.

$$\Omega = \left\{ \mathbf{x} \in \mathbb{R}^N \;\middle|\; \begin{array}{ccc} f_i(\mathbf{x}) \leq 0 & , & i = 1, \ldots, M \\ h_i(\mathbf{x}) = 0 & , & i = 1, \ldots, P \end{array} \right\}$$

Typical choices of penalty functions: use quadratic functions

- inequality constraint $f_i(\mathbf{x}) \leq 0$: take

$$p(\mathbf{x}) = \frac{1}{2} \left[\max(0, f_i(\mathbf{x}))\right]^2 := \frac{1}{2} \left[f_i^+(\mathbf{x})\right]^2$$

- equality constraint $h_i(\mathbf{x}) = 0$: take

$$p(\mathbf{x}) = \frac{1}{2}(h_i(\mathbf{x}))^2$$

for multiple constraints, simply sum all penalties $p(\mathbf{x}) = \sum_i p_i(\mathbf{x})$

Remark: can consider also non-smooth penalties such as ℓ_1 with very interesting properties (outside the scope of this lecture).

Penalty method algorithm

$$\min_{\mathbf{x}\in\Omega} f_0(\mathbf{x}) \tag{$\mathcal{P}$}$$

Solve $(\mathcal{P})$ by a succession of penalized problems

$$\min_{\mathbf{x}\in\mathbb{R}^N} f_0(\mathbf{x}) + \rho_k p(\mathbf{x}), \quad \rho_k > 0 \tag{$\mathcal{P}_{\rho_k}$}$$

with increasing penalties $\rho_1 < \rho_2 < \ldots < \rho_k \to +\infty$

Theorem (Convergence of the penalty method)

Let f_0 and p be continuous. Let $\mathbf{x}^{(k)}$ be a minimizer of $\mathcal{P}_{\rho_k}$. Suppose the sequence $\{\rho_k\}$ is strictly increasing with $\rho_k \to \infty$ as $k \to \infty$. Then $\mathbf{x}^{(k)} \to \mathbf{x}^\star$, where $\mathbf{x}^\star$ is a solution of $(\mathcal{P})$.

Computing solutions of penalized problems

Consider a problem with one inequality constraint $f_1(\mathbf{x}) \leq 0$.
The quadratic penalty is $p(\mathbf{x}) = \frac{1}{2}\left[\max(0, f_1(\mathbf{x}))\right]^2 := \frac{1}{2}\left[f_1^+(\mathbf{x})\right]^2$

How to compute the gradient?

The main issue is that f_1^+ is not differentiable at points $\mathbf{x}$ such that $f_1(\mathbf{x}) = 0$.

However, remark that p is a function of f_1^+ only s.t. $p(\mathbf{x}) = \gamma(f_1^+(\mathbf{x}))$, where γ is the quadratic function. In particular, $\frac{\partial \gamma}{\partial y}(0) = 0$, which implies that $p(\mathbf{x})$ is differentiable whenever f_1^+ is.

Using the chain rule

$$\nabla p(\mathbf{x}) = f_1^+(\mathbf{x}) \nabla f_1(\mathbf{x}) = \begin{cases} f_1(\mathbf{x}) \nabla f_1(\mathbf{x}) & \text{if } f_1(\mathbf{x}) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Example

Using the penalty method, solve the following problem:

$$\begin{aligned} \text{minimize} \quad & x \\ \text{subject to} \quad & x \geq 1 \end{aligned}$$

Example

Using the penalty method, solve the following problem:

$$\begin{aligned} \text{minimize} \quad & x \\ \text{subject to} \quad & x \geq 1 \end{aligned}$$

Solution Write the penalized problem:

$$\text{minimize} \quad x + \frac{\rho}{2}(\max(0, 1 - x))^2 := f_\rho(x), \qquad \rho > 0$$

The objective is convex, with gradient $f'_\rho(x) = 1 - \rho \max(0, 1 - x)$.
Solving $f'_\rho(x) = 0$ gives $x = 1 - 1/\rho$, which converges to $x = 1$ as $\rho \to \infty$.

Another example

Using the penalty method, solve the following problem:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|\mathbf{x}\|_2^2 \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b} \end{array}$$

Solution Write the penalized problem:

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{x}\|^2 + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2, \qquad \rho > 0$$

The objective is convex, with gradient $\nabla f_\rho(x) = \mathbf{x} + \rho \mathbf{A}^\top(\mathbf{A}\mathbf{x} - \mathbf{b})$. Solving $\nabla f_\rho(x) = 0$ gives $\mathbf{x}_\rho = \rho(\mathbf{I} + \rho\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{b}$.

Tedious calculations (e.g., using the SVD) show that, in the limit $\rho \to \infty$, $\mathbf{x}_\rho \to \mathbf{x}^\star$ with $\mathbf{x}^\star = \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-1}\mathbf{b}$.

Practical algorithm

$$\min_{\mathbf{x}\in\Omega} f_0(\mathbf{x}) \qquad (\mathcal{P})$$

Penalty algorithm

Choose a penalty function $p(\mathbf{x})$ for the problem $(\mathcal{P})$, e.g., a quadratic penalty: $p(\mathbf{x}) = \sum_{i=1}^{M}[f_i^+(\mathbf{x})]^2 + \sum_{i=1}^{P} h_i^2(\mathbf{x})$.
Choose a stopping criterion. Then the algorithm reads:

Fix increase parameter $s > 1$ and initial penalty $\rho_0 > 0$.
$k := 0$
while stopping criterion is not satisfied **do**
    $\mathbf{x}^{(k+1)} := \arg\min\left(f_0(\mathbf{x}) + \rho_k p(\mathbf{x})\right)$ (possibly using $\mathbf{x}^{(k)}$)
    $k := k + 1$
    $\rho_{k+1} := s\rho_k$
end while
return (approximate) solution $\mathbf{x}^{(k)}$

Outline

Motivation

Consider the general constrained optimization problem

$$\min_{\mathbf{x}\in\Omega} f_0(\mathbf{x}) \tag{$\mathcal{P}$}$$

goal: design a simple algorithm to solve $(\mathcal{P})$

recall from lecture 4: for unconstrained problems ($\Omega = \mathbb{R}^N$) with differentiable objective, one simple algorithm is gradient descent (GD):

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)}), \quad \alpha_k > 0, \quad k = 0, 1, \ldots$$

for constrained problems, a natural generalization is projected gradient descent (PGD)

combines GD with (Euclidean) projection onto Ω

Projection onto a set

The (Euclidean) projection operator $P_\Omega : \mathbb{R}^N \to \mathbb{R}^N$ onto Ω is defined as

$$P_\Omega(\mathbf{x}_0) = \arg\min_{\mathbf{x}\in\Omega} \|\mathbf{x} - \mathbf{x}_0\|_2^2$$

- computing $P_\Omega(\mathbf{x}_0)$ is itself an optimization problem
- $P_\Omega(\mathbf{x}_0)$ returns the closest point(s) (in Euclidean norm) to $\mathbf{x}_0$ that belongs to the set Ω
- when Ω is closed and convex the projection is unique
- if $\mathbf{x}_0 \in \Omega$, then $P_\Omega(\mathbf{x}_0) = \mathbf{x}_0$
- in many important cases, $P_\Omega(\mathbf{x}_0)$ can be computed explicitly

$$P_{\mathbb{R}_+^N}(\mathbf{x}_0) = [\max(0, [\mathbf{x}_0]_i)]_{i=1}^N \quad \text{(entry-wise maximum)}$$

exercise: compute the projection onto the unit ball in $\mathbb{R}^N$,
$\Omega = \{\mathbf{x} \mid \|\mathbf{x}\| \le 1\}$

Projected gradient method

$$\min_{\mathbf{x} \in \Omega} f_0(\mathbf{x}) \qquad (\mathcal{P})$$

Projected gradient algorithm

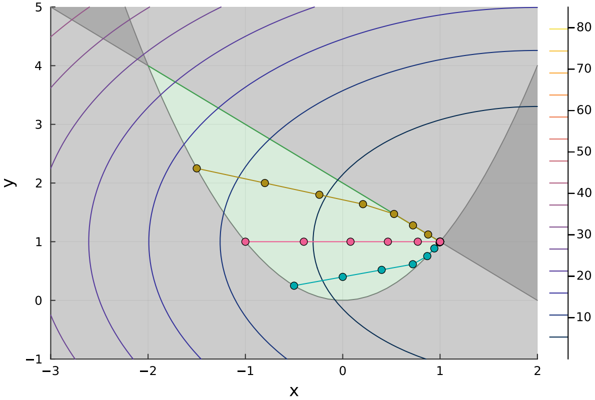
starting point $\mathbf{x}^{(0)} \in \mathbb{R}^N$
$k := 0$
while stopping criterion is not satisfied **do**
    $\mathbf{x}^{(k+1)} := P_\Omega\left(\mathbf{x}^{(k)} - \alpha_k \nabla f_0(\mathbf{x}^{(k)})\right)$
    $k := k + 1$
end while
return solution $\mathbf{x}^{(k)}$

typical step-size strategies:

- fixed step-size $\alpha_k = \alpha$ for all k
- use backtracking strategies to determine α_k at each iteration

Example

$$
\begin{aligned}
\text{minimize} \quad & (x_1 - 2)^2 + (x_2 - 1)^2 \\
\text{subject to} \quad & x_1^2 - x_2 \leq 0 \\
& x_1 + x_2 \leq 2
\end{aligned}
$$



Comments on projected gradient descent

- PGD is particularly interesting when the projection operator is either explicit, easy to solve, or computationally efficient.
- Theoretical analysis of PGD is similar to that of GD (see session 5). In particular, if f_0 is convex and L-smooth, then convergence is in $\mathcal{O}(1/K)$ (like GD)
- PGD is a special case of proximal gradient method, used to solve problems of the form $\min f(\mathbf{x}) + g(\mathbf{x})$, where f is differentiable and g is not. To recover PGD, take g to be the indicator function on Ω
- sufficient decrease condition in backtracking should be adapted to account for the projection step.

Outline

Reminder: primal and dual problems

Primal problem

$$\begin{array}{rll} \text{minimize} & f_0(\mathbf{x}) & \\ \text{subject to} & f_i(\mathbf{x}) \leq 0, & i = 1, \ldots, M \\ & h_i(\mathbf{x}) = 0, & i = 1, \ldots, P \end{array} \tag{$\mathcal{P}$}$$

Dual problem

$$\begin{array}{rl} \text{maximize} & g(\boldsymbol{\lambda}, \boldsymbol{\nu}) := \inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \\ \text{subject to} & \boldsymbol{\lambda} \geq 0 \end{array} \tag{$\mathcal{D}$}$$

where the L is the Lagrangian

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\mathbf{x}) + \sum_{i=1}^{M} \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^{P} \nu_i h_i(\mathbf{x})$$

Principle of Uzawa's algorithm

- can be viewed as dual projected gradient
  $\to$ simple projection of $\boldsymbol{\lambda}$ onto $\mathbb{R}_+^P$: $P_{\mathbb{R}_+^P}(\boldsymbol{\lambda}) = \max(0, \boldsymbol{\lambda})$ (entrywise)
- dual is a maximization problem $\to$ gradient ascent
- when it converges, iterates tends to a saddle point of L (therefore algorithm requires strong duality)

Uzawa's algorithm

1: initial Lagrange multipliers $\boldsymbol{\lambda}^{(0)} \geq 0$ and $\boldsymbol{\nu}^{(0)}$.
2: $k := 0$
3: **while** stopping criterion is not satisfied **do**
4: $\quad \mathbf{x}^{(k+1)} := \arg\min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}^{(k)}, \boldsymbol{\nu}^{(k)})$
5: $\quad \boldsymbol{\lambda}^{(k+1)} = P_{\mathbb{R}_+^P}\left(\boldsymbol{\lambda}^{(k)} + \alpha_k [f_1(\mathbf{x}^{(k+1)}) \cdots f_M(\mathbf{x}^{(k+1)})]^\top\right), \quad \alpha_k > 0$ stepsize
6: $\quad \boldsymbol{\nu}^{(k+1)} = \boldsymbol{\nu}^{(k)} + \beta_k [h_1(\mathbf{x}^{(k+1)}) \cdots h_P(\mathbf{x}^{(k+1)})]^\top, \quad \beta_k > 0$ stepsize
7: **end while**
8: **return** (approximate) solution $\mathbf{x}^{(k)}$

Example

Write Uzawa's iterates for the following problem

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \\ \text{subject to} \quad & \mathbf{x} \geq 0 \end{aligned}$$

where $\mathbf{A}$ is full column rank.

Example

Write Uzawa's iterates for the following problem

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \\ \text{subject to} \quad & \mathbf{x} \geq 0 \end{aligned}$$

where $\mathbf{A}$ is full column rank.

solution The Lagrangian for the problem is $L(\mathbf{x}, \boldsymbol{\lambda}) = \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 - \boldsymbol{\lambda}^\top \mathbf{x}$ with gradient with respect to $\mathbf{x}$ given by $\nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{A}^\top(\mathbf{A}\mathbf{x} - \mathbf{y}) - \boldsymbol{\lambda}$. Therefore we have the iterations

$$\begin{aligned} \mathbf{x}^{(k+1)} &= (\mathbf{A}^\top \mathbf{A})^{-1}\left[\mathbf{A}^\top \mathbf{y} + \boldsymbol{\lambda}^{(k)}\right] \\ \boldsymbol{\lambda}^{(k+1)} &= \left[\boldsymbol{\lambda}^{(k)} - \alpha_k \mathbf{x}^{(k+1)}\right]^+ \end{aligned}$$

until primal and dual convergence.

Outline

1. Penalty method
2. Projected gradient
3. Dual methods: Uzawa's algorithm
4. Disciplined Convex programming with CVXPY

Disciplined Convex Programming (DCP)

What is it?

A set of rules, a “language” that permits to formulate and implement efficiently convex optimization problems.

→ underlying language of numerical solvers, such as CVXPY (Python) or CVX (Matlab)

CVXPY: Python-based solver using DCP

see `https://www.cvxpy.org/` for tutorial and documentation

check also the JMLR paper:

> CVXPY: A Python-Embedded Modeling Language for Convex Optimization, by Diamond and Boyd

CVXPY example [from tutorial]

```
import cvxpy as cp

# Create two scalar optimization variables.
x = cp.Variable()
y = cp.Variable()

# Create two constraints.
constraints = [x + y == 1,
               x - y >= 1]

# Form objective.
obj = cp.Minimize((x - y)**2)

# Form and solve problem.
prob = cp.Problem(obj, constraints)
prob.solve()  # Returns the optimal value.
print("status:", prob.status)
print("optimal value", prob.value)
print("optimal var", x.value, y.value)
```

Exercise

Solve the following problem using CVXPY

$$
\begin{array}{ll}
\text{minimize} & (x_1 - 2)^2 + (x_2 - 1)^2 \\
\text{subject to} & x_1^2 - x_2 \leq 0 \\
& x_1 + x_2 \leq 2
\end{array}
$$