Convex Optimization - HW2

IGOR LUCINDO CARDOSO

October 5, 2025

Disclaimer

This work was assisted by ChatGPT in the following ways: improvement of text content, clarification of mathematical definitions, and rapid generation of graphs and tables.

Problem 1

We present two equivalent definitions of linear independence.

Definition 1. A set of vectors $\{v_i\}_{i\in[n]}$ in a vector space V is *linearly independent* if, for scalars $\{\lambda_i\}_{i\in[n]}$,

$$\sum_{i\in[n]} \lambda_i v_i = 0 \implies \lambda_i = 0 \quad \forall i \in [n].$$

Definition 2. A set of vectors $\{v_i\}_{i\in[n]}$ in a vector space V is *linearly independent* if, for every $i \in [n]$, there do not exist scalars $\{\lambda_j\}_{j\in[n]\setminus\{i\}}$ such that

$$v_i = \sum_{j\in[n]\setminus\{i\}} \lambda_j v_j.$$

Claim 1. Definitions 1 and 2 are equivalent.

Proof. We prove both directions.

($\Rightarrow$)
Assume Definition 1 holds. Suppose, for contradiction, that there exists some $i \in [n]$ and scalars $\{\lambda_j\}_{j\in[n]\setminus\{i\}}$ such that

$$v_i = \sum_{j\in[n]\setminus\{i\}} \lambda_j v_j.$$

Rearranging gives

$$\sum_{j\in[n]\setminus\{i\}} \lambda_j v_j - v_i = 0.$$

This is a nontrivial linear combination of the v_j's equaling 0, since the coefficient of v_i is $-1 \neq 0$. This contradicts Definition 1. Hence, no such representation of v_i exists, which establishes Definition 2.

($\Leftarrow$)
Assume Definition 2 holds. Suppose, for contradiction, that Definition 1 fails. Then there exist scalars $\{\lambda_i\}_{i\in[n]}$, not all zero, such that

$$\sum_{i\in[n]} \lambda_i v_i = 0.$$

Let $j \in [n]$ be an index with $\lambda_j \neq 0$. Rearranging,

$$v_j = \sum_{i\in[n]\setminus\{j\}} \frac{-\lambda_i}{\lambda_j} v_i.$$

This expresses v_j as a linear combination of the other vectors, contradicting Definition 2. Therefore, Definition 1 must hold.

Since each definition implies the other, they are equivalent. $\square$

Problem 2

We present two equivalent definitions of affine independence.

Definition 3. A set of vectors $\{v_i\}_{i\in[n]}$ in a vector space V is *affinely independent* if, for any $j \in [n]$, the set $\{v_i - v_j\}_{i\in[n]\setminus\{j\}}$ is linearly independent.

Definition 4. A set of vectors $\{v_i\}_{i\in[n]}$ in a vector space V is *affinely independent* if

$$\sum_{i\in[n]} \lambda_i v_i = 0 \quad \text{and} \quad \sum_{i\in[n]} \lambda_i = 0 \quad \Longrightarrow \quad \lambda_i = 0 \quad \forall i \in [n].$$

Claim 2. Definitions 3 and 4 are equivalent.

Proof. We prove both directions.

($\Rightarrow$)
Assume Definition 3 holds. Suppose scalars $\{\lambda_i\}_{i\in[n]}$ satisfy

$$\sum_{i\in[n]} \lambda_i v_i = 0 \quad \text{and} \quad \sum_{i\in[n]} \lambda_i = 0.$$

Fix $j \in [n]$. Then

$$0 = \sum_{i\in[n]} \lambda_i v_i - \Big(\sum_{i\in[n]} \lambda_i\Big) v_j = \sum_{i\in[n]} \lambda_i (v_i - v_j).$$

Since the coefficient of $(v_j - v_j)$ is 0, this simplifies to

$$\sum_{i\in[n]\setminus\{j\}} \lambda_i (v_i - v_j) = 0.$$

But by Definition 3, the vectors $\{v_i - v_j\}_{i\in[n]\setminus\{j\}}$ are linearly independent. Hence,

$$\lambda_i = 0 \quad \forall i \in [n] \setminus \{j\}.$$

Since this holds for every $j \in [n]$, it follows that all $\lambda_i = 0$. Thus Definition 4 holds.

($\Leftarrow$)
Assume Definition 4 holds. Suppose, for contradiction, that Definition 3 fails. Then there exists $j \in [n]$ and scalars $\{\lambda_i\}_{i\in[n]\setminus\{j\}}$, not all zero, such that

$$\sum_{i\in[n]\setminus\{j\}} \lambda_i (v_i - v_j) = 0.$$

Define new scalars λ_i' by

$$\lambda_i' = \lambda_i \;\; (i \neq j), \qquad \lambda_j' = -\sum_{i\in[n]\setminus\{j\}} \lambda_i.$$

Then

$$\sum_{i\in[n]} \lambda_i' v_i = \sum_{i\in[n]\setminus\{j\}} \lambda_i (v_i - v_j) = 0,$$

and

$$\sum_{i\in[n]} \lambda_i' = 0.$$

By Definition 4, it follows that $\lambda_i' = 0$ for all $i \in [n]$. In particular, $\lambda_i = 0$ for $i \neq j$, contradicting our assumption. Hence Definition 3 must hold.

Since each definition implies the other, they are equivalent. $\square$

Problem 3

3.1

Proof. Take any $x_1, x_2 \in f^{-1}(U)$ and any $\alpha \in [0,1]$. By definition of preimage, $f(x_1), f(x_2) \in U$. Since f is affine there exists a linear map T and $w \in W$ with $f(v) = T(v) + w$ for all v, hence

$$f\big(\alpha x_1 + (1-\alpha)x_2\big) = T\big(\alpha x_1 + (1-\alpha)x_2\big) + w = \alpha T(x_1) + (1-\alpha)T(x_2) + w = \alpha f(x_1) + (1-\alpha)f(x_2).$$

Because U is convex and $f(x_1), f(x_2) \in U$, we have

$$\alpha f(x_1) + (1-\alpha)f(x_2) \in U$$

Therefore $f(\alpha x_1 + (1-\alpha)x_2) \in U$, so $\alpha x_1 + (1-\alpha)x_2 \in f^{-1}(U)$. As $x_1, x_2 \in f^{-1}(U)$ and $\alpha \in [0,1]$ were arbitrary, it follows that $f^{-1}(U)$ is convex. $\square$

3.2 - Let $a = (a_1, \dots, a_n)^\top$.

(a) Computing $f^{-1}(0)$: by definition of f we require

$$0 = Av + a,$$

so that

$$Av = -a.$$

Since A is diagonal with entries $A_{ii} = 2^i$ for $i \in [n]$, this implies

$$v_i = -\frac{a_i}{2^i} \quad \forall i \in [n].$$

The coordinates $v_{n+1}, \dots, v_m$ do not appear in Av and are therefore unconstrained. Hence,

$$f^{-1}(0) = \{v \in \mathbb{R}^m : v_i = -\tfrac{a_i}{2^i},\ \forall i \in [n]\}.$$

(b) Now we compute $f^{-1}(U)$. By definition of f we require

$$u = Av + a \quad u \in U,$$

which means

$$u_i = 2^i v_i + a_i \quad \forall i \in [n].$$

Since $u \in U$, we must have

$$\|u\|_2^2 = \sum_{i=1}^{n} u_i^2 = 1.$$

Substituting the expression for u_i, this becomes

$$\sum_{i=1}^{n} (2^i v_i + a_i)^2 = 1.$$

Therefore,

$$f^{-1}(U) = \{v \in \mathbb{R}^m : \sum_{i \in [n]} (2^i v_i + a_i)^2 = 1\}.$$

Problem 4

Proof. In order to prove that the Cartesian product $U \times V$ is convex, with convex sets U, V, we will show that, for arbitrary $(u_1, v_1), (u_2, v_2) \in U \times V$ and for $\alpha \in [0, 1]$,

$$\alpha(u_1, v_1) + (1 - \alpha)(u_2, v_2) \in U \times V.$$

First, we develop the above expression as follows.

$$\alpha(u_1, v_1) + (1 - \alpha)(u_2, v_2) = (\alpha u_1 + (1 - \alpha)u_2, \alpha v_1 + (1 - \alpha)v_2).$$

Since U, V are convex, $\alpha u_1 + (1-\alpha)u_2 \in U$ and $\alpha v_1 + (1-\alpha)v_2 \in V$. Hence, the pair on the right-hand side belongs to the Cartesian product $U \times V$. As the choice of $(u_1, v_1), (u_2, v_2)$ and α was arbitrary, $U \times V$ is convex. $\square$

Problem 5

First we compute $0.5U$ and $0.5V$ as follows.

$$0.5U = \{0.5x : x \in \mathbb{R}^2,\ \|x\|_2 \leq 1\} = \{x \in \mathbb{R}^2 : \|x\|_2 \leq 0.5\}$$
$$0.5V = \{0.5x : x \in \mathbb{R}^2,\ \|x-1\|_2 \leq 1\} = \{x \in \mathbb{R}^2 : \|x-0.5\|_2 \leq 0.5\}.$$

Each is a disk of radius 0.5; their centers are $c_1 = (0,0)$ and $c_2 = (0.5, 0)$.

Claim 3. Let

$$B_1 = \{x \in \mathbb{R}^n : \|x - c_1\|_2 \leq r_1\}, \qquad B_2 = \{x \in \mathbb{R}^n : \|x - c_2\|_2 \leq r_2\},$$

with $r_1, r_2 \geq 0$. Set

$$B = \{x \in \mathbb{R}^n : \|x - (c_1 + c_2)\|_2 \leq r_1 + r_2\}.$$

Then

$$B_1 + B_2 = B.$$

Proof. We prove the two inclusions.

($\subseteq$)
Take $x \in B_1 + B_2$. Then $x = x_1 + x_2$ with $x_1 \in B_1$ and $x_2 \in B_2$. Thus, by the triangle inequality:

$$\|x - (c_1 + c_2)\|_2 = \|(x_1 - c_1) + (x_2 - c_2)\|_2 \leq \|x_1 - c_1\|_2 + \|x_2 - c_2\|_2 \leq r_1 + r_2.$$

Which means that $x \in B$.

($\supseteq$)
If $r_1 + r_2 = 0$ then $r_1 = r_2 = 0$ and the claim is trivial (each ball is a single point and $B = c_1 + c_2$). So assume $r_1 + r_2 > 0$ and let $x \in B$ be an arbitrary vector. Define

$$d := x - (c_1 + c_2),$$

and set

$$x_1 = c_1 + \frac{r_1}{r_1 + r_2} d, \qquad x_2 = c_2 + \frac{r_2}{r_1 + r_2} d.$$

Then

$$x_1 + x_2 = c_1 + c_2 + \frac{r_1 + r_2}{r_1 + r_2} d = c_1 + c_2 + d = x.$$

Moreover,

$$\|x_1 - c_1\|_2 = \frac{r_1}{r_1 + r_2} \|d\|_2 \leq \frac{r_1}{r_1 + r_2} (r_1 + r_2) = r_1,$$

and similarly $\|x_2 - c_2\|_2 \leq r_2$. Therefore $x_1 \in B_1$ and $x_2 \in B_2$, so $x \in B_1 + B_2$.

Combining the two inclusions yields $B_1 + B_2 = B$, as claimed. □

Hence, by Claim 3, we obtain

$$0.5U + 0.5V = \{x \in \mathbb{R}^2 : \|x - 0.5\|_2 \leq 1\}.$$

Problem 6

6.1 - Consider the vector $(x', y', u', v') = (0.5, 0.5, 1, 2) \in \mathcal{P}$. We check that it satisfies all inequalities strictly:

$$0 < x' < 1, \quad 0 < y < 1, \quad u > 0, \quad v > 0, \quad u - v < 0, \quad x + y < 2.$$

Hence (x', y', u', v') is an interior point of $\mathcal{P}$, which proves that $\mathcal{P}$ is full-dimensional in $\mathbb{R}^4$.

6.2 - The recession cone of a polyhedron $\{x : Ax \geq b\}$ is given by $\{d : Ad \geq 0\}$. Applying this to $\mathcal{P}$ gives

$$\operatorname{rec}(\mathcal{P}) = \{(x, y, u, v) : 0 \leq x \leq 0,\ 0 \leq y \leq 0,\ u \geq 0,\ v \geq 0,\ u - v \leq 0,\ x + y \leq 0\}$$

Thus $x = 0$ and $y = 0$, while the remaining conditions are

$$u \geq 0, \quad v \geq 0, \quad v \geq u.$$

Therefore

$$\operatorname{rec}(\mathcal{P}) = \{(0, 0, u, v) : u \geq 0,\ v \geq u\}.$$

6.3

(a) To find the extreme points, we look for feasible solutions where a maximal number of inequalities are tight. Since $\mathcal{P}$ is bounded in the (x, y) coordinates but unbounded in (u, v), the extreme points will occur when $u = v = 0$ and the box constraints $0 \leq x \leq 1$, $0 \leq y \leq 1$ are at their corners. Thus the extreme points are

$$(0, 0, 0, 0), \quad (0, 1, 0, 0), \quad (1, 0, 0, 0), \quad (1, 1, 0, 0).$$

(b) To compute the extreme rays, recall that

$$\operatorname{rec}(\mathcal{P}) = \{(0, 0, u, v) : u \geq 0,\ v \geq u\}.$$

This is a cone in the (u, v) plane, generated by its boundary directions. The two boundary directions are

$$(0, 0, 0, 1) \quad \text{and} \quad (0, 0, 1, 1).$$

Therefore the set of extreme rays of $\mathcal{P}$ is

$$\{\lambda(0, 0, 0, 1) : \lambda \geq 0\} \cup \{\lambda(0, 0, 1, 1) : \lambda \geq 0\}.$$

(c) The facet-defining inequalities of $\mathcal{P}$ are those that cannot be omitted without changing the polyhedron. The redundant inequalities are $x + y \leq 2$ and $v \geq 0$, so the facet-defining inequalities are

$$\begin{aligned} x &\geq 0, \quad x \leq 1, \\ y &\geq 0, \quad y \leq 1, \\ u &\geq 0, \quad u - v \leq 0. \end{aligned}$$

Thus we determine the facets as follows.

$$\begin{aligned} F_1 &= \mathcal{P} \cap \{(x, y, u, v) \in \mathbb{R}^4 : x = 0\} \\ F_2 &= \mathcal{P} \cap \{(x, y, u, v) \in \mathbb{R}^4 : x = 1\} \\ F_3 &= \mathcal{P} \cap \{(x, y, u, v) \in \mathbb{R}^4 : y = 0\} \\ F_4 &= \mathcal{P} \cap \{(x, y, u, v) \in \mathbb{R}^4 : y = 1\} \\ F_5 &= \mathcal{P} \cap \{(x, y, u, v) \in \mathbb{R}^4 : u = 0\} \\ F_6 &= \mathcal{P} \cap \{(x, y, u, v) \in \mathbb{R}^4 : u = v\} \end{aligned}$$

(d) As discussed in part (c), the redundant constraint are $x + y \leq 2$ and $v \geq 0$. In fact, from $x \leq 1$ and $y \leq 1$ it is guaranteed that $x + y \leq 2$, moreover, from $u \geq 0$ and $u - v \leq 0$, it follows that $v \geq u \geq 0$.

6.4 -

Python code

```
from ppl import Variable, Constraint, Constraint_System, C_Polyhedron

# Define variables (order: x, y, u, v)
x = Variable(0)
y = Variable(1)
u = Variable(2)
v = Variable(3)

# Build the constraint system Ax >= b for polyhedron P
cs = Constraint_System()

# Constraints:
cs.insert(x >= 0)
cs.insert(x <= 1)
cs.insert(y >= 0)
cs.insert(y <= 1)
cs.insert(u >= 0)
cs.insert(v >= 0)
cs.insert(u - v <= 0)
cs.insert(x + y <= 2)

# Construct polyhedron
P = C_Polyhedron(cs)

# Get extreme points and extreme rays
gen = P.generators()

extreme_points = [g for g in gen if g.is_point()]
extreme_rays   = [g for g in gen if g.is_ray()]

print("Extreme points:")
for p in extreme_points:
    print(p)

print("\nExtreme rays:")
for r in extreme_rays:
    print(r)

# Get minimal (facet-defining) constraints
facets = P.minimized_constraints()

print("\nFacet-defining inequalities:")
for f in facets:
    print(f)
```

1. **Define variables:** Each coordinate (x, y, u, v) is represented by a `Variable(i)` object, where the index i determines the coordinate.

2. **Build the constraint system:** We construct a `Constraint_System()` object and insert the inequalities that define $\mathcal{P}$:

$$0 \leq x \leq 1, \quad 0 \leq y \leq 1, \quad u \geq 0, \quad v \geq 0, \quad u - v \leq 0, \quad x + y \leq 2.$$

3. **Construct the polyhedron:** The polyhedron object $\mathcal{P}$ is created using `C_Polyhedron(cs)`.

4. **Enumerate generators:** Calling `P.generators()` returns all generators of $\mathcal{P}$ (points, rays, or lines). We separate them using `is_point()` and `is_ray()`.

5. **Compute facets:** The function `P.minimized_constraints()` eliminates redundant constraints and returns only the facet-defining inequalities.

Running the above code produces the following:

- **Extreme points:** $(0, 0, 0, 0)$, $(0, 1, 0, 0)$, $(1, 0, 0, 0)$, $(1, 1, 0, 0)$.
- **Extreme rays:** $(0, 0, 0, 1)$ and $(0, 0, 1, 1)$.
- **Facet-defining inequalities:**

$$x \geq 0, \quad x \leq 1, \quad y \geq 0, \quad y \leq 1, \quad u \geq 0, \quad u - v \leq 0.$$