Assignment 1

EE5121: Convex Optimisation

By

Mohammed Nawfal

ME21B046

Course Instructor:

Prof. Pravin Nair

Indian Institute of Technology, Madras

September, 2025

Contents

Chapter 1

Assignment 1

1.1 LINK TO CVXPY SOLUTIONS

The repository includes a Jupyter Notebook (.ipynb) containing well-documented CVXPY implementations for this assignment, ensuring reproducibility.
Link: https://github.com/mohammednawfal/EE5121-Assignment-1

1.2 QUESTION 1

1.2.1 l_0 sparse solution

Let $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Assume the linear system $Ax = b$ is consistent and admits multiple solutions. We want to check whether the problem

$$\min_{x \in \mathbb{R}^n} \|x\|_0 \quad \text{s.t.} \quad Ax = b$$

is a convex optimisation problem.

Here, $\|x\|_0$ counts the number of non-zero components of x. For the above to qualify as a convex optimisation problem, both the objective function and the constraints need to be convex.

- The constraint $Ax = b$ is convex (it defines an affine set).
- Thus, we only need to check whether the function $f(x) = \|x\|_0$ is convex.

We shall prove that $f(x) = \|x\|_0$ is **not convex**, and hence the above optimisation problem is not convex. To prove that a function $f : \mathbb{R}^n \to \mathbb{R}$ is non-convex, it suffices to find some $x_1, x_2 \in \mathbb{R}^n$ and some $\lambda \in [0, 1]$ such that

$$f(\lambda x_1 + (1 - \lambda)x_2) > \lambda f(x_1) + (1 - \lambda)f(x_2).$$

Proof by Counterexample

Choose $x_1, x_2 \in \mathbb{R}^n$ such that

$$x_1 = e_1, \quad x_2 = e_2,$$

where e_1 and e_2 are standard basis vectors in $\mathbb{R}^n$. Let us choose $\lambda = \frac{1}{2}$. Then,

$$f(x_1) = 1, \qquad f(x_2) = 1.$$

Let x_λ be the convex combination of x_1 and x_2. Therefore,

$$x_\lambda = \lambda x_1 + (1 - \lambda)x_2 = \frac{x_1 + x_2}{2} = \frac{(1, 0, 0, \ldots, 0) + (0, 1, 0, \ldots, 0)}{2} = \left(\tfrac{1}{2}, \tfrac{1}{2}, 0, \ldots, 0\right).$$

From the above, we have

$$f(x_\lambda) = 2. \tag{1.1}$$

Now, compute the convex combination of $f(x_1)$ and $f(x_2)$:

$$\begin{aligned} \lambda f(x_1) + (1 - \lambda)f(x_2) &= \tfrac{1}{2}(1) + \tfrac{1}{2}(1) \\ &= 1. \end{aligned} \tag{1.2}$$

From (1.1) and (1.2), we clearly see that

$$f(x_\lambda) > \lambda f(x_1) + (1 - \lambda)f(x_2).$$

Thus, we have shown that $f(x) = \|x\|_0$ is **non-convex** for some choice of x_1, x_2, and λ. Since $\|x\|_0$ is non-convex, the above optimisation problem is also **non-convex**.

1.2.2 l_1 **proxy**

The l_1 relaxation of the previous problem is

$$\min_{x \in \mathbb{R}^n} \|x\|_1 \quad \text{s.t.} \quad Ax = b,$$

where $\|x\|_1 = \sum_{i=1}^{n} |x_i|$. We wish to check if the above problem is a convex optimisation problem. For that, we need to show that $f(x) = \|x\|_1$ is a convex function for all $x_1, x_2 \in \mathbb{R}^n$ and all $\lambda \in [0, 1]$. (As mentioned in the previous subpart, $Ax = b$ is a convex constraint).

Proof of Convexity of $||x||_1$

Let $f : \mathbb{R}^n \to \mathbb{R}$ be defined as $f(x) = \|x\|_1$, where

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|.$$

Choose any $x_1, x_2 \in \mathbb{R}^n$ and $\lambda \in [0, 1]$. Then,

$$\begin{aligned} f(\lambda x_1 + (1-\lambda)x_2) &= \|\lambda x_1 + (1-\lambda)x_2\|_1 \\ &= \sum_{i=1}^{n} |\lambda x_1^i + (1-\lambda)x_2^i|, \end{aligned} \tag{1.3}$$

where x_1^i and x_2^i denote the i^{th} components of x_1 and x_2.

Using the scalar triangle inequality $|a + b| \leq |a| + |b|$, we obtain

$$\begin{aligned} |\lambda x_1^i + (1-\lambda)x_2^i| &\leq |\lambda x_1^i| + |(1-\lambda)x_2^i| \\ &= \lambda |x_1^i| + (1-\lambda)|x_2^i|. \end{aligned}$$

Summing over all components $i = 1, \dots, n$, we get

$$\begin{aligned} \sum_{i=1}^{n} |\lambda x_1^i + (1-\lambda)x_2^i| &\le \sum_{i=1}^{n} \big(\lambda|x_1^i| + (1-\lambda)|x_2^i|\big) \\ &= \lambda \sum_{i=1}^{n} |x_1^i| + (1-\lambda) \sum_{i=1}^{n} |x_2^i| \\ &= \lambda f(x_1) + (1-\lambda) f(x_2). \end{aligned} \tag{1.4}$$

From (1.3) and (1.4), we have

$$f(\lambda x_1 + (1-\lambda)x_2) \le \lambda f(x_1) + (1-\lambda) f(x_2),$$

which is the condition for convexity.

Since our choice of x_1, x_2 was arbitrary, this holds for all $x_1, x_2 \in \mathbb{R}^n$ and all $\lambda \in [0, 1]$. Hence, $f(x) = \|x\|_1$ is a **convex** function, and the corresponding ℓ_1-relaxed optimisation problem is a **convex optimisation problem**.

1.2.3 l_1 as a linear program

The l_1 relaxation can be written as a linear program using auxiliary variables $u \in \mathbb{R}^n$:

$$\min_{x,u} \; \mathbf{1}^\top u \quad \text{s.t.} \quad Ax = b, \; -u \le x \le u, \; u \ge 0$$

We wish to show that these two problems are equivalent.

Problem Equivalence

Let problems P1 and P2 be as follows:

P1:

$$\min_{x \in \mathbb{R}^n} \; \|x\|_1 \quad \text{s.t.} \quad Ax = b,$$

P2:

$$\min_{x,u} \; \mathbf{1}^\top u \quad \text{s.t.} \quad Ax = b, \; -u \le x \le u, \; u \ge 0$$

To show that these two problems are equivalent, it is sufficient to show the following:

1. An optimal solution of **P1** can be mapped to a feasible and optimal solution for **P2**.
2. An optimal solution of **P2** can be mapped to a feasible and optimal solution for **P1**.
3. At optimality, the values of the objective functions for both problems are equal.

Proof for 1:

Let x^* be the optimal solution to **P1**. Define $u^* \in \mathbb{R}^n$ such that

$$u_i^* = |x_i^*|, \quad i = 1, 2, \ldots, n.$$

The solution (x^*, u^*) satisfies the constraint set of **P2**, i.e.,

1. $u^* \geq 0$: Follows directly from the definition of u^*.
2. $-u^* \leq x^* \leq u^*$: This holds by definition of u^*, and the bounds are tight.
3. $Ax^* = b$: Since x^* is optimal for **P1**, it must satisfy the equality constraint.

Therefore, (x^*, u^*) is a feasible solution. The value of the objective function of **P2** is

$$\mathbf{1}^\top u^* = \sum_{i=1}^{n} u_i^* = \sum_{i=1}^{n} |x_i^*|,$$

which is equal to the objective function of **P1** at optimality.

—

To show that u^* is optimal, consider any other candidate solution $\tilde{u}$:

1. $\tilde{u}_i < |x_i^*|$: This choice does not necessarily satisfy the first constraint $u \geq 0$ and violates the second constraint $-u \leq x \leq u$. Hence, it is infeasible and cannot be optimal.

2. $\tilde{u}_i > |x_i^*|$: This solution is feasible, but the objective value becomes

$$\mathbf{1}^\top \tilde{u} = \sum_{i=1}^{n} \tilde{u}_i > \sum_{i=1}^{n} |x_i^*| = \mathbf{1}^\top u^*,$$

which is strictly larger than the objective at (x^*, u^*). Hence, $\tilde{u}$ cannot be optimal (we could decrease $\tilde{u}_i$ to $|x_i^*|$ without violating any constraints).

Therefore, $u^* = |x^*|$ is the optimal choice, and (x^*, u^*) forms the optimal solution to **P2**.

Proof for 2:

Let $(\tilde{x}, \tilde{u})$ be the optimal solution for **P2**. The value of the objective function at this solution is

$$\mathbf{1}^\top \tilde{u} = \sum_{i=1}^{n} \tilde{u}_i.$$

From the constraints, for all $i = 1, 2, \ldots, n$, we have

$$\tilde{u}_i \geq 0,$$
$$-\tilde{u}_i \leq \tilde{x}_i \leq \tilde{u}_i \quad \Rightarrow \quad |\tilde{x}_i| \leq \tilde{u}_i.$$

Therefore, the least value each $\tilde{u}_i$ can take without violating feasibility is $|\tilde{x}_i|$. At optimality, we must have

$$\tilde{u}_i = |\tilde{x}_i|.$$

Any solution where $\tilde{u}_i > |\tilde{x}_i|$ can be decreased to $|\tilde{x}_i|$ without violating the constraints, which reduces the objective. Hence, at optimality,

$$\mathbf{1}^\top \tilde{u} = \sum_{i=1}^{n} \tilde{u}_i = \sum_{i=1}^{n} |\tilde{x}_i|.$$

Since $(\tilde{x}, \tilde{u})$ is optimal for **P2**, it satisfies $A\tilde{x} = b$, which is also the constraint for **P1**. Therefore, the objective value of **P1** at $\tilde{x}$ is

$$\|\tilde{x}\|_1 = \sum_{i=1}^{n} |\tilde{x}_i|,$$

which is equal to the objective value of **P2** at optimality. Thus, $\tilde{x}$ is the optimal solution of **P1**.

Proof for 3:

From both proofs 1 and 2, we have shown that at optimality for both **P1** and **P2**, they both have the same objective value.

From these three proofs, we can conclude that **P1** and **P2** are equivalent problems.

1.2.4 Application

Fewest-Pigments Formulation

We would like to pose the fewest-pigments formulation as a sparse recovery problem. Consider $A \in \mathbb{R}^{3\times 3}$ and $b \in \mathbb{R}^3$ such that $Ax = b$ is feasible and admits multiple solutions. The solution vector $x \in \mathbb{R}^3$ gives the proportion of each pigment used from P_1, P_2, and P_3.

We consider the following conditions:

- Since we want to use the fewest number of pigments, it is equivalent to minimising the number of non-zero components in x. Thus, the objective is to minimise $\|x\|_0$.
- The solution x should satisfy the constraint $Ax = b$.
- The proportion of each pigment cannot be negative, so we require $x_i \geq 0$ for $i = 1, 2, 3$, i.e., $x \geq 0$.

Hence, the problem can be formulated as

$$\begin{aligned} \min_{x\in\mathbb{R}^3} \quad & \|x\|_0 \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0. \end{aligned}$$

Where A and b are given as,

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 2 \end{bmatrix}, \qquad b = \begin{bmatrix} 0.2 \\ 0.8 \\ 1.0 \end{bmatrix}$$

To solve $Ax = b$, we perform the following row operations.

$R_3 \leftarrow R_3 - R_1$

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.8 \\ 0.8 \end{bmatrix}$$

$R_3 \leftarrow R_3 - R_2$

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.8 \\ 0 \end{bmatrix}$$

We see that x_1, x_2 are determined by x_3, which is a free variable. Specifically,

$$x_1 = 0.2 - x_3, \quad x_2 = 0.8 - x_3, \quad x_3 \in \mathbb{R}.$$

Thus, the set of all solutions to $Ax = b$ is

$$S = \left\{ \begin{bmatrix} 0.2 - t \\ 0.8 - t \\ t \end{bmatrix} : t \in \mathbb{R} \right\} = \left\{ \begin{bmatrix} 0.2 \\ 0.8 \\ 0 \end{bmatrix} + t \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} : t \in \mathbb{R} \right\}.$$

Since the objective is to minimise $||x||_0$, we consider the following values for t:

1. $t = 0$: The solution that we get is $x_1^* = (0.2, 0.8, 0)$. This solution also satisfies $x \geq 0$. Hence, it is a feasible solution.

2. $t = 0.2$: The solution that we get is $x_2^* = (0, 0.6, 0.2)$. This solution also satisfies all constraints and is feasible.

3. $t = 0.8$: The solution that we get is $x_3^* = (-0.6, 0, 0.8)$. This solution violates the non-negativity constraint $(-0.6 < 0)$. Hence, it is not feasible.

Evaluating the feasible solutions on the objective,

$$||x_1^*||_0 = 2$$
$$||x_2^*||_0 = 2$$

We see that both x_1^* and x_2^* achieve the same objective value. Thus,

$$x_1^* = \begin{bmatrix} 0.2 \\ 0.8 \\ 0 \end{bmatrix} \quad \text{and} \quad x_2^* = \begin{bmatrix} 0 \\ 0.6 \\ 0.2 \end{bmatrix}.$$

are the optimum solutions to this problem, achieving an objective value of 2. This implies that only 2 pigments are required to synthesize the target reflectance vector.
Had we solved for the l_1 relaxation, the values of the objective would have been,

$$||x_1^*||_1 = 1$$
$$||x_2^*||_1 = 0.8$$

We see that the objective is minimum at x_2^*. Thus,

$$x_2^* = \begin{bmatrix} 0 \\ 0.6 \\ 0.2 \end{bmatrix}.$$

is the optimum solution to the l_1 relaxation of the problem, achieving an objective value of 0.8.

Solve with CVXPY

The original sparse recovery problem cannot be solved using CVXPY as it is non-convex. Thus, we shall use the l_1 relaxation as a proxy for the sparse recovery problem and solve it using CVXPY.

The required .ipynb file has been attached along with this report. The code contains all documentation for reproducibility. Upon solving the l_1 relaxation problem with CVXPY, the optimiser x^* that it achieved is (after rounding):

$$x^* = \begin{bmatrix} 0 \\ 0.6 \\ 0.2 \end{bmatrix}.$$

It's support is $S = \{2, 3\}$ (in 0-indexing like python, it is $\{1, 2\}$).
Link to code (incase .ipynb file is not working): https://github.com/mohammednawfal/EE5121-Assignment-1

1.2.5 Infeasible Constraint

Modified Formulation

Suppose the linear system $Ax = b$ is **infeasible**, i.e., no exact solution exists. This implies that $b \notin \text{CS}(A)$, where $\text{CS}(A)$ denotes the column space of A. To handle this, we relax the equality constraint as follows:

- The solution vector x should make Ax the closest approximation to b.
- The closeness can be measured by the Euclidean norm $\|Ax - b\|_2$. Minimising this is equivalent to projecting b onto $\text{CS}(A)$.
- This error can be rewritten as a constraint:

  $$\|Ax - b\|_2 \leq t$$

  for some slack variable $t \geq 0$.

The constraint $\|Ax - b\|_2 \leq t$ is convex because:

1. $f(x) = \|Ax - b\|_2$ is a convex function since all norms are convex (they satisfy the triangle inequality).
2. The set $\{x : \|Ax - b\|_2 \leq t\}$ is the t-sublevel set of $f(x)$. Sublevel sets of convex functions are convex.

Thus, the modified sparse recovery problem becomes:

$$\begin{aligned}
\min_{x\in\mathbb{R}^3,\, t\in\mathbb{R}} \quad & \|x\|_0 + t \\
\text{s.t.} \quad & \|Ax - b\|_2 \leq t, \\
& x \geq 0, \\
& t \geq 0.
\end{aligned}$$

This problem is still **non-convex** due to the l_0-norm term. Using the l_1-relaxation, we obtain a convex formulation:

$$\begin{aligned}
\min_{x\in\mathbb{R}^3,\, t\in\mathbb{R}} \quad & \|x\|_1 + t \\
\text{s.t.} \quad & \|Ax - b\|_2 \leq t, \\
& x \geq 0, \\
& t \geq 0.
\end{aligned}$$

which is **convex** because the objective is a sum of convex functions and the constraints are convex. The slack variable t is minimised because it is equivalent to minimising the error $||Ax - b||_2$. At optimality, $t^* = ||Ax^* - b||_2$. This can be proven via contradiction. Let $(x^*, \tilde{t})$ be an optimal solution such that $\tilde{t} > ||Ax^* - b||_2$. Then the value of $\tilde{t}$ can be reduced to $||Ax^* - b||_2$ without violating the constraint.

$$||x^*||_1 + \tilde{t} > ||x^*||_1 + ||Ax^* - b||_2$$

This in turn would reduce the objective value, thus contradicting the assumption that $\tilde{t}$ is optimal. Hence, we have proven that at optimality, $t^* = ||Ax^* - b||_2$.

Solve with CVXPY

- We solve the modified l_1 relaxation problem using CVXPY. The original modified problem is not solvable as it is non-convex. The optimiser x^* it achieved is (after rounding):
$$x^* = \begin{bmatrix} 0 \\ 0.55 \\ 0.2 \end{bmatrix}.$$
- It's support is $S = \{2, 3\}$ ($\{1, 2\}$ in python indexing).
- The achieved residual norm is: 0.049.

Link to code (incase .ipynb file is not working): https://github.com/mohammednawfal/EE5121-Assignment-1

1.3 QUESTION 2

1.3.1 Rank Minimisation

Let $A \in \mathbb{R}^{m \times p}$, $X \in \mathbb{R}^{p \times n}$, and $B \in \mathbb{R}^{m \times n}$. Assume that the linear matrix equation $AX = B$ is consistent and admits multiple solutions (p $>$ m). We want to check whether the problem

$$\min_{X \in \mathbb{R}^{p \times n}} \ \text{rank}(X) \quad \text{s.t.} \quad AX = B$$

is a convex optimisation problem. For the above to qualify as a convex optimisation problem, both the objective function and the constraints need to be convex.

- The constraint $AX = B$ is convex since the solution set is an affine space (given in the question).
- Thus, we need to check if $f(X) = \text{rank}(X)$ is convex.

We shall prove that $f(X) = \text{rank}(X)$ is **not convex**, and hence the above optimisation problem is not convex.

Proof by Counterexample

Choose $X_1, X_2 \in \mathbb{R}^{p \times n}$ such that

- The first column of X_1 is e_1 and all other columns are zero.
- The second column of X_2 is e_2 and all other columns are zero.

Here, e_1 and e_2 are the standard basis vectors in $\mathbb{R}^p$. Let us choose $\lambda = \frac{1}{2}$. Then,

$$f(X_1) = 1, \qquad f(X_2) = 1.$$

Let X_λ be the convex combination of X_1 and X_2. Therefore,

$$X_\lambda = \lambda X_1 + (1-\lambda) X_2 = \frac{X_1 + X_2}{2} = \frac{(e_1, 0, 0, \dots, 0) + (0, e_2, 0, \dots, 0)}{2} = \left(\tfrac{1}{2}e_1, \tfrac{1}{2}e_2, 0, \dots, 0\right).$$

From the above, we have

$$f(X_\lambda) = 2. \tag{1.5}$$

Now, compute the convex combination of $f(X_1)$ and $f(X_2)$:

$$\begin{aligned} \lambda f(X_1) + (1-\lambda) f(X_2) &= \tfrac{1}{2}(1) + \tfrac{1}{2}(1) \\ &= 1. \end{aligned} \tag{1.6}$$

From (1.5) and (1.6), we clearly see that

$$f(X_\lambda) > \lambda f(X_1) + (1-\lambda) f(X_2).$$

This violates the definition of convexity of a function. Thus, we have shown that

$$f(X) = \text{rank}(X)$$

is **non-convex** for some choice of X_1, X_2, and λ. Since $\text{rank}(X)$ is non-convex, the above optimisation problem is also **non-convex**.

1.3.2 Nuclear Norm Minimisation

The nuclear norm minimisation problem is given by

$$\min_{X \in \mathbb{R}^{p \times n}} \|X\|_* \quad \text{s.t.} \quad AX = B,$$

where $\|X\|_* = \sum_i \sigma_i(X)$ is the nuclear norm (sum of singular values). It can also be written as

$$\|X\|_* = \sup_{\|Y\|_2 \leq 1} \langle X, Y \rangle,$$

where $\|\cdot\|_2$ is the spectral norm and $\langle X, Y \rangle = \text{Trace}(X^\top Y)$. We wish to show that $||X||_*$ is convex and the above problem is a convex optimisation problem.

Proof of Convexity of $||X||_*$

Let $f : \mathbb{R}^{p \times n} \to \mathbb{R}$ be defined as

$$f(X) = \|X\|_* = \sum_i \sigma_i(X).$$

Choose any $X_1, X_2 \in \mathbb{R}^{p \times n}$ and $\lambda \in [0, 1]$. Then,

$$f(\lambda X_1 + (1 - \lambda) X_2) = \|\lambda X_1 + (1 - \lambda) X_2\|_*.$$

Using the trace definition of the nuclear norm, for any $Y \in \mathbb{R}^{p \times n}$ we get

$$\begin{aligned} f(\lambda X_1 + (1 - \lambda) X_2) &= \sup_{\|Y\|_2 \leq 1} \langle \lambda X_1 + (1 - \lambda) X_2,\, Y \rangle \\ &= \sup_{\|Y\|_2 \leq 1} \text{Tr}\big((\lambda X_1 + (1 - \lambda) X_2)^\top Y\big). \end{aligned} \tag{1.7}$$

Consider the RHS term in (1.7). Using transpose properties and linearity of the trace,

$$\operatorname{Tr}\big((\lambda X_1 + (1-\lambda)X_2)^\top Y\big) = \lambda \ \operatorname{Tr}(X_1^\top Y) + (1-\lambda) \ \operatorname{Tr}(X_2^\top Y). \tag{1.8}$$

Now, define the set

$$S = \{Y \in \mathbb{R}^{p\times n} : \|Y\|_2 \leq 1\}.$$

From the definition of supremum, for all $Y \in S$,

$$\operatorname{Tr}(X_1^\top Y) \leq \sup_{Y\in S} \operatorname{Tr}(X_1^\top Y), \quad \operatorname{Tr}(X_2^\top Y) \leq \sup_{Y\in S} \operatorname{Tr}(X_2^\top Y).$$

Since λ and $(1-\lambda)$ are positive constants, substituting into (1.8), we get

$$\operatorname{Tr}\big((\lambda X_1 + (1-\lambda)X_2)^\top Y\big) \leq \lambda \sup_{Y\in S} \operatorname{Tr}(X_1^\top Y) + (1-\lambda) \sup_{Y\in S} \operatorname{Tr}(X_2^\top Y).$$

Since this holds for all $Y \in S$, it also holds for the supremum:

$$\sup_{Y\in S} \operatorname{Tr}\big((\lambda X_1 + (1-\lambda)X_2)^\top Y\big) \leq \lambda \sup_{Y\in S} \operatorname{Tr}(X_1^\top Y) + (1-\lambda) \sup_{Y\in S} \operatorname{Tr}(X_2^\top Y). \tag{1.9}$$

From (1.7), (1.9), and the definition of the nuclear norm, we conclude

$$\begin{aligned}
\|\lambda X_1 + (1-\lambda)X_2\|_* &\leq \lambda\|X_1\|_* + (1-\lambda)\|X_2\|_* \\
f(\lambda X_1 + (1-\lambda)X_2) &\leq \lambda f(X_1) + (1-\lambda) f(X_2).
\end{aligned}$$

This is exactly the condition for convexity. Since our choice of X_1, X_2 was arbitrary, the inequality holds for all $X_1, X_2 \in \mathbb{R}^{p\times n}$ and all $\lambda \in [0, 1]$. Hence, $f(X) = \|X\|_*$ is a **convex** function, and the corresponding problem is a **convex optimisation problem**.

1.3.3 Multi-experiment System Identification

Solve with CVXPY

The Multi-experiment System Identification problem was solved on CVXPY as a Nuclear Norm Minimisation problem. The original rank minimization problem is non-convex; hence, CVXPY cannot solve it directly. The optimiser X^* CVXPY achieved is (after rounding):

$$X^* = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 2 & 0 & 1 & -1 & 0 \\ 1 & 1 & 1 & -1 & -1 \\ 2 & -2 & 0 & 0 & 2 \\ -2 & 0 & -1 & 1 & 0 \end{bmatrix}$$

Link to code (incase .ipynb file is not working): https://github.com/mohammednawfal/EE5121-Assignment-1

Modified formulation for infeasible constraint and Solve with CVXPY

Suppose B is replaced with $B + N$ where $N \in \mathbb{R}^{5x5}$ and $N[i,j] \sim \mathcal{N}(0, 0.1)$ if $i = j$ and 0 when $i \neq j$. Then the constraint $AX = B$ may not have an exact match, i.e., it is infeasible. To handle this, we relax the equality constraint in a manner similar to that in Question 1.

- We would want to find an X such that AX is the closest approximation to B.
- The closeness can be measured by the Frobenius norm $||AX - B||_F$. If $X \in \mathbb{R}^{p \times n}$ then $||X||_F$ is given by,

$$||X||_F = \sqrt{\sum_{i=1}^{p} \sum_{j=1}^{n} X_{ij}^2}$$

- This error can be rewritten as a constraint:

$$||AX - B||_F \leq t$$

for some slack variable $t \geq 0$.

- The constraint $||AX - B||_F \leq t$ is the matrix analogue of the constraint $||Ax - b||_2 \leq t$. Here, $||AX - B||_F$ is equal to the l_2 norm of a matrix that has been vectorised. And since we have proven that $||Ax - b||_2 \leq t$ is a convex constraint, $||AX - B||_F \leq t$ is also a convex constraint.

Thus the modified rank minimisation problem becomes:

$$\min_{X \in \mathbb{R}^{p \times n}, t \in \mathbb{R}} \quad \text{rank}(X) + t \quad \text{s.t.} \quad ||AX - B||_F \leq t, \quad t \geq 0$$

This problem is still **non-convex** due to the rank(X) term. The nuclear norm minimisation problem becomes:

$$\min_{X \in \mathbb{R}^{p \times n}, t \in \mathbb{R}} \quad \|X\|_* + t \quad \text{s.t.} \quad ||AX - B||_F \leq t, \quad t \geq 0$$

This problem is convex because the sum of two convex functions is convex, and the constraints are convex too. Similar to the reasoning in Question 1, the slack variable t is minimised because it is equivalent to minimised the error $||AX - B||_F$. At optimality, $t^* = ||AX^* - B||_F$.

Solving the modified nuclear norm minimisation problem with CVXPY, the optimiser X^* is (rounded off to first 3 decimal places):

$$X^* = \begin{bmatrix} 0.969 & -0.631 & 0.169 & -0.089 & 0.618 \\ 0.917 & 0.002 & 0.493 & -0.425 & -0.005 \\ 1.053 & 0.098 & 0.619 & -0.542 & -0.100 \\ 0.630 & -0.701 & -0.051 & 0.107 & 0.687 \\ -0.093 & -0.342 & -0.240 & 0.237 & 0.337 \end{bmatrix}$$

Link to code (incase .ipynb file is not working): https://github.com/mohammednawfal/EE5121-Assignment-1

1.4 QUESTION 3

1.4.1 Feasibility via a 0-1 formulation

Let $G = (V, E)$ be a simple undirected graph with $|V| = n$. A K-coloring assigns to each vertex $v \in V$ a color in $\{1, \ldots, K\}$ such that adjacent vertices receive different colors. The smallest K for which a K-coloring exists is called the chromatic number of G, denoted by $\chi(G)$. We introduce variables $x_{vk} \in \{0, 1\}$ indicating that vertex v uses color k. Consider the feasibility system for a fixed K:

$$
\begin{aligned}
\sum_{k=1}^{K} x_{vk} &= 1 && \forall v \in V, \\
x_{uk} + x_{vk} &\leq 1 && \forall (u, v) \in E,\ \forall k = 1, \ldots, K, \\
x_{vk} &\in \{0, 1\} && \forall v \in V,\ \forall k = 1, \ldots, K.
\end{aligned}
$$

We would like to show that the above constraint set C is **non-convex**. To show that it is non-convex, it is sufficient to show that for some $x_1, x_2 \in C$ and for some $\lambda \in [0, 1]$, $\lambda x_1 + (1 - \lambda) x_2 \notin C$.

Proof by Counterexample

Let C be the constraint set defined as

$$
\begin{aligned}
C = \Big\{ x \in \mathbb{R}^{nK} \ : \ & \sum_{k=1}^{K} x_{vk} = 1, && \forall v \in V, \\
& x_{uk} + x_{vk} \leq 1, && \forall (u, v) \in E,\ \forall k = 1, \ldots, K, \\
& x_{vk} \in \{0, 1\}, && \forall v \in V,\ \forall k = 1, \ldots, K \Big\}.
\end{aligned}
\tag{1.10}
$$

Let $x^{(1)}, x^{(2)} \in C$ be two feasible solutions such that for a fixed $v \in V$,

$$
\begin{aligned}
x^{(1)}_{v1} &= 1, \quad x^{(1)}_{vk} = 0, \quad \forall k \neq 1, \\
x^{(2)}_{v2} &= 1, \quad x^{(2)}_{vk} = 0, \quad \forall k \neq 2.
\end{aligned}
$$

Let $\lambda = \frac{1}{2}$. Taking the convex combination of $x^{(1)}$ and $x^{(2)}$, we get

$$
\begin{aligned}
x^{(\lambda)} &= \lambda x^{(1)} + (1-\lambda)x^{(2)}, \\
x^{(\lambda)} &= \frac{x^{(1)} + x^{(2)}}{2}, \\
x_{v1}^{(\lambda)} &= \frac{x_{v1}^{(1)} + x_{v1}^{(2)}}{2} = 0.5, \\
x_{v2}^{(\lambda)} &= \frac{x_{v2}^{(1)} + x_{v2}^{(2)}}{2} = 0.5.
\end{aligned}
$$

We see that both $x_{v1}^{(\lambda)}, x_{v2}^{(\lambda)} = 0.5$, which violates the constraint that for all $v \in V$ and $k = 1, \ldots, K$, $x_{vk} \in \{0, 1\}$. Thus, $x^{(\lambda)}$ is not a feasible solution, i.e., $x^{(\lambda)} \notin C$.

Hence, we have shown that C is **non-convex** for some choice of $x^{(1)}, x^{(2)}$, and λ.

Convexity without binary requirement

We now drop the binary requirement and replace it with $0 \leq x_{vk} \leq 1$. We need to show that under this relaxation, the constraint set C is **convex**. Let C be defined as

$$
\begin{aligned}
C &= \{x \in \mathbb{R}^{nK} : x \in C_1 \cap C_2 \cap C_3\}, \\
C_1 &= \left\{x \in \mathbb{R}^{nK} : \sum_{k=1}^{K} x_{vk} = 1,\ \forall v \in V\right\}, \\
C_2 &= \left\{x \in \mathbb{R}^{nK} : x_{uk} + x_{vk} \leq 1,\ \forall (u,v) \in E,\ \forall k = 1, \ldots, K\right\}, \\
C_3 &= \left\{x \in \mathbb{R}^{nK} : 0 \leq x_{vk} \leq 1,\ \forall v \in V,\ \forall k = 1, \ldots, K\right\}.
\end{aligned}
$$

To show that C is convex, it is sufficient to show C_1, C_2, and C_3 are convex. Convexity of C will then follow from the intersection of convex sets being convex.

Convexity of C_1:

Let $x, y \in C_1$ and $\lambda \in [0, 1]$. then for all $v \in V$,

$$
\sum_{k=1}^{K} x_{vk} = 1, \quad \sum_{k=1}^{K} y_{vk} = 1 \tag{1.11}
$$

Taking the convex combination of x and y,

$$x^{\lambda} = \lambda x + (1 - \lambda)y$$

Now, for all $v \in V$,

$$\sum_{i=1}^{K} x^{\lambda}_{vk} = \sum_{i=1}^{K}(\lambda x_{vk} + (1 - \lambda)y_{vk})$$
$$\sum_{i=1}^{K} x^{\lambda}_{vk} = \lambda \sum_{i=1}^{K} x_{vk} + (1 - \lambda) \sum_{i=1}^{K} y_{vk}$$

From (1.11), we get,

$$\sum_{i=1}^{K} x^{\lambda}_{vk} = \lambda \cdot 1 + (1 - \lambda) \cdot 1$$
$$\sum_{i=1}^{K} x^{\lambda}_{vk} = 1$$

Therefore, $x^{\lambda} \in C_1$. Since the choice of x, y was arbitrary, this is true for all $x, y \in C_1$ and all $\lambda \in [0, 1]$. Hence, C_1 is a convex constraint.

Convexity of C_2:

Let $x, y \in C_2$ and $\lambda \in [0, 1]$. Then for all $(u, v) \in E$ and $k = 1, \cdots, K$,

$$x_{uk} + x_{vk} \leq 1, \quad y_{uk} + y_{vk} \leq 1 \tag{1.12}$$

taking the convex combination of x and y,

$$x^{\lambda} = \lambda x + (1 - \lambda)y$$

Now, for all $(u, v) \in E$ and $k = 1, \cdots, K$,

$$x^{\lambda}_{uk} + x^{\lambda}_{vk} = \lambda x_{uk} + (1 - \lambda)y_{uk} + \lambda x_{vk} + (1 - \lambda)y_{vk}$$
$$x^{\lambda}_{uk} + x^{\lambda}_{vk} = \lambda(x_{uk} + x_{vk}) + (1 - \lambda)(y_{uk} + y_{vk})$$

From (1.12), and since $\lambda, (1-\lambda)$ are positive constants,

$$\lambda(x_{uk} + x_{vk}) + (1-\lambda)(y_{uk} + y_{vk}) \leq 1$$
$$x^{\lambda}_{uk} + x^{\lambda}_{vk} \leq 1$$

Therefore $x^{\lambda} \in C_2$. Since the choice of x, y was arbitrary, this is true for all $x, y \in C_2$ and all $\lambda \in [0,1]$. Hence, C_2 is a convex constraint.

Convexity of C_3:

Let $x, y \in C_3$ and $\lambda \in [0,1]$. Then for all $v \in V$ and $k = 1, \cdots, K$,

$$0 \leq x_{vk} \leq 1, \quad 0 \leq y_{vk} \leq 1 \tag{1.13}$$

taking the convex combination of x and y,

$$x^{\lambda} = \lambda x + (1-\lambda)y$$

Now, for all $v \in V$ and $k = 1, \cdots, K$, we have,

$$x^{\lambda}_{vk} = \lambda x_{vk} + (1-\lambda)y_{vk}$$

From (1.13) and since $\lambda, (1-\lambda)$ are positive constants,

$$0 \leq \lambda x_{vk} + (1-\lambda)y_{vk} \leq \lambda + (1-\lambda)$$
$$0 \leq \lambda x_{vk} + (1-\lambda)y_{vk} \leq 1$$
$$0 \leq x^{\lambda}_{vk} \leq 1$$

Therefore $x^{\lambda} \in C_3$. Since the choice of x, y was arbitrary, this is true for all $x, y \in C_3$ and all $\lambda \in [0,1]$. Hence, C_3 is a convex constraint.

Since we have shown that C_1, C_2,and C_3 are convex constraints, their intersection is also a convex constraint set. Therefore, C is proven to be a convex constraint set.

1.4.2 SDP relaxation

Associate each vertex $v \in V$ with a unit vector $u_v \in \mathbb{R}^n$, and let $G \succeq 0$ be the Gram matrix with entries

$$G_{uv} = u_u^\top u_v.$$

For a fixed $K \geq 2$, the vector K-coloring feasibility semidefinite program (SDP) can be written as:

$$\begin{aligned}
\text{find } & G \in \mathbb{S}^n \\
\text{s.t. } & G \succeq 0, \\
& G_{vv} = 1, && \forall v \in V, \\
& G_{uv} \leq -\frac{1}{K-1}, && \forall (u,v) \in E.
\end{aligned}$$

Interpretation of feasibility

A one line interpretation of feasibility and non-feasibility is given below:

- **Feasible**: For a given K, there exists a set of n unit vectors such that for all $(u,v) \in E$, the angle between u_u and u_v is atleast $\arccos(-\frac{1}{K-1})$ and hence, G exists and is Positive - Semi Definite.
- **Non-feasible**: There does not exist an assignment of unit vectors (and hence no PSD Gram matrix with ones on the diagonal) exists for the given K.

Equivalence to Optimisation Problem

We need to show that the above feasibility problem is equivalent to the optimization problem minimizing the edge-wise inner-product bound ρ:

$$\begin{aligned}
\min_{G,\rho} \quad & \rho \\
\text{s.t.} \quad & G \succeq 0, \\
& G_{vv} = 1, \quad \forall v \in V, \\
& G_{uv} \leq \rho, \quad \forall (u, v) \in E,
\end{aligned}$$

To show that these two are equivalent, it is sufficient to show that they solve the same problem. This can be proven in the following manner:

- The original problem asks about the existence/feasibility of a set of n unit vectors that obey the constraints (and equivalently the existence of the Gram Matrix G, which corresponds to whether the graph is Vector - K colourable).
- For a fixed K, the constraint $G_{uv} \leq -\frac{1}{K-1}$ was fixed.
- Instead of fixing the upper bound, we can relax it to an optimisation problem of finding the smallest possible value that bounds G_{uv} for all $(u, v) \in E$. Thus, we introduce the variable ρ which we minimise such that $G_{uv} \leq \rho$ for all $(u, v) \in E$. In essence, the optimisation problem asks, what is the smallest maximum edge-wise inner-product bound?
- Suppose ρ^* is the optimal value for the optimisation problem. Then, if $\rho^* \leq -\frac{1}{K-1}$, then it implies that the edge-wise inner product bound for the feasibility problem is also satisfied. Hence, a matrix G exists that satisfies the feasibility criteria.
- On the other hand, if $\rho^* > -\frac{1}{K-1}$, then the edge-wise inner product bound is not satisfied in the original problem, and hence, no matrix G exists, and the SDP is infeasible.

- Thus, the optimisation problem is equivalent to the original SDP feasibility problem and at the same time, gives an intuition of how far we are from achieving feasibility for a fixed K.

Interpretation of Chromatic Number

From the above problem equivalence, we can state the following about the chromatic number of the graph:
For a given graph and a fixed K, if $\rho^* \leq -\frac{1}{K-1}$, where ρ^* is the optimal solution to the optimisation problem, then the graph is Vector-K colourable, i.e., $\chi(G) \leq K$. If $\rho^* > -\frac{1}{K-1}$, the graph is not Vector-K colourable, i.e., $\chi(G) > K$.

1.4.3 Application in your field

A real-world application of graph colouring in mechanical engineering arises in the form of scheduling of machining operations. Typically, in workshops and manufacturing pipelines, a set of operations is performed on the raw material (such as molding, heat treatment, machining, surface finishing, assembling, etc.) Each of these operations has a machine allotted to it, a time window for the operation, and some sort of interaction between the machinery, the workpiece, and sometimes neighbouring machinery. A common issue that arises in this setup is a time slot conflict, where two operations require the same machine in a given time slot.

It is required to assign a time slot for each operation, such that conflicting operations are not assigned the same time slot. At the same time, the number of time slots shouldn't be too large, as it would lengthen the duration of the entire sequence of operations. This can be formulated as a graph colouring problem where:

- A vertex is an operation.
- An edge between two operations if they cannot be scheduled in the same time slot
- A colour designates a time slot.

The aim is to minimise the number of time slots, which is equivalent to finding the chromatic number $\chi(G)$ of the graph. The problem can be formally stated as follows:

- Given a set of operations $V = \{1, 2, \cdots, n\}$, we construct a graph $G = (V, E)$ such that for all $(u, v) \in E$ iff u and v are conflicting operations, $u, v \in V$.

- For the graph G, what is the chromatic number $\chi(G)$?, or equivalently, what is the least number of time slots required to run the operations without any conflict?

If instead of finding $\chi(G)$, we ask if a certain number of time slots K is practical for scheduling operations without any conflict, then it becomes a K-colouring problem. We shall now formulate this using the 0-1 formulation, the SDP relaxation (Vector-K colouring), and the optimisation form.

0-1 Formulation:

We introduce indicator variables $x_{vk} \in \{0, 1\}$ for each vertex v to indicate whether it uses colour k or not. For a fixed number of time slots K, a feasible colouring scheme would have to satisfy the following:

1. Each operation should be assigned only one time slot: $\sum_{k=1}^{K} x_{vk} = 1$, for all $v \in V$.

2. All pairs of conflicting operations should be assigned different time slots: $x_{uk}+x_{vk} \leq 1$ for all $(u, v) \in E$ and $k = 1, \cdots, K$

3. The indicator variable x_{vk} which denotes if the time slot k is chosen for vertex v has to always be 0 or 1.

This formulation is combinatorial in nature and is not convex. The relaxation provides a feasibility set that is convex.

SDP Relaxation (Vector - K Colouring):

For a fixed K, we now assign a unit vector u_v to each operation (vertex). Let $G \succeq 0$, be the Gram Matrix such that $G_{uv} = u_u^T u_v$. We would want to find a $G \in \mathbb{S}^n$ such that:

$$
\begin{aligned}
&G \succeq 0, \\
&G_{vv} = 1, && \forall v \in V, \\
&G_{uv} \leq -\frac{1}{K-1}, && \forall (u,v) \in E.
\end{aligned}
$$

Equivalent Optimisation:

We can reformulate the SDP feasibility problem as an equivalent optimisation problem, as seen in the previous sub-question. We can pose it as:

$$
\begin{aligned}
\min_{G,\rho} \quad & \rho \\
\text{s.t.} \quad & G \succeq 0, \\
& G_{vv} = 1, \quad \forall v \in V, \\
& G_{uv} \leq \rho, \quad \forall (u,v) \in E,
\end{aligned}
$$

Thus, we have successfully formulated the scheduling of machining operations as a graph colouring problem. The task of finding the least number of time slots for a practical scheduling with no conflicts is equivalent to finding the chromatic number of the graph.