Convex Optimization - HW4

IGOR LUCINDO CARDOSO

November 17, 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

1.1 — We set up a formulation to minimize the negative entropy.

$$\min \sum_{i\in[n]} p_i \log p_i \tag{1a}$$
$$\text{s.t.} \sum_{i\in[n]} i p_i = m_1 \tag{1b}$$
$$\sum_{i\in[n]} i^2 p_i = m_2 \tag{1c}$$
$$\sum_{i\in[n]} p_i = 1 \tag{1d}$$
$$p_i \geq 0 \qquad \forall\, i \in [n].$$

Constraints (1b)–(1d) enforce the known moments and normalization of the probability distribution. The objective (1a) represents the negative entropy, which we seek to minimize.

1.2 — Introducing Lagrange multipliers $u, v, w \in \mathbb{R}$ for the equality constraints, the Lagrangian is:

$$\mathcal{L}(p,u,v,w) = \sum_{i\in[n]} p_i \log p_i + u\left(m_1 - \sum_{i\in[n]} i p_i\right) + v\left(m_2 - \sum_{i\in[n]} i^2 p_i\right) + w\left(1 - \sum_{i\in[n]} p_i\right).$$

1.3 - The KKT conditions are as follows.

- Stationary: $\nabla_p \mathcal{L}(p^*, u^*, v^*, w^*) = 0$,
- Primal feasibility: $\sum_{i\in[n]} i p_i = m_1, \quad \sum_{i\in[n]} i^2 p_i = m_2, \quad \sum_{i\in[n]} p_i = 1, \quad p_i \geq 0$,
- Dual feasibility: $u, v, w \in \mathbb{R}$,
- Complementary slackness:

$$(m_1 - \sum_{i\in[n]} i p_i)u = 0, \quad (m_2 - \sum_{i\in[n]} i^2 p_i)v = 0, \quad (1 - \sum_{i\in[n]} p_i)w = 0.$$

1.4 - The objective $\sum_i p_i \log p_i$ is strictly convex on the positive orthant, since

$$\frac{d^2}{dp_i^2}(p_i \log p_i) = \frac{1}{p_i} > 0 \quad \text{for } p_i > 0.$$

The equality constraints are affine, and if there exists a strictly feasible point, on this case $p_i > 0$, the Slater's condition holds for this problem, ensuring strong duality.

1.5 — To obtain the dual, we first compute the minimizing p_i by setting the stationarity condition $\nabla_p \mathcal{L}(p, u, v, w) = 0$:

$$\frac{\partial \mathcal{L}}{\partial p_i} = \log p_i + 1 - ui - vi^2 - w = 0.$$

Hence,

$$\log p_i = ui + vi^2 + w - 1 \quad \Rightarrow \quad p_i = \exp(ui + vi^2 + w - 1).$$

Using the normalization constraint $\sum_i p_i = 1$, we define

$$Z(u, v) = \sum_{j \in [n]} \exp(uj + vj^2),$$

so that

$$p_i^* = \frac{\exp(ui + vi^2)}{Z(u, v)}.$$

We now substitute this expression into $\mathcal{L}(p, u, v, w)$:

$$\begin{aligned} \mathcal{L}(p^*, u, v, w) &= \sum_i p_i^* \log p_i^* + u\left(m_1 - \sum_i i p_i^*\right) + v\left(m_2 - \sum_i i^2 p_i^*\right) + w\left(1 - \sum_i p_i^*\right) \\ &= -\log Z(u, v) + u m_1 + v m_2, \end{aligned}$$

since at the optimum $\sum_i p_i^* = 1$.
Thus, the **dual problem** is:

$$\max_{u, v \in \mathbb{R}} \left\{ g(u, v) = u m_1 + v m_2 - \log\left(\sum_{i \in [n]} \exp(ui + vi^2) \right) \right\}.$$

The corresponding primal optimum is given by:

$$p_i^* = \frac{\exp(u^* i + v^* i^2)}{\sum_{j \in [n]} \exp(u^* j + v^* j^2)}.$$

Problem 2

2.1 — We write the constraint as

$$\langle w, w \rangle \leq R^2$$

Introducing multiplier $u \geq 0$, the Lagrangian is

$$\mathcal{L}(w,b,u) = \sum_{i=1}^{d} \log(\sigma(y_i(\langle x_i, w\rangle + b))) \; - \; u \left(\langle w, w\rangle - R^2\right).$$

2.2 - The KKT conditions are as follows.

- Stationary: $\nabla_w \mathcal{L}(w,b,u) = 0, \quad \frac{\partial}{\partial b}\mathcal{L}(w,b,u) = 0,$
- Primal feasibility: $\langle w, w\rangle \leq R^2, \quad w \in \mathbb{R}^n, \quad b \in \mathbb{R},$
- Dual feasibility: $u \geq 0,$
- Complementary slackness: $u\left(\langle w, w\rangle - R^2\right) = 0.$

2.3 - Slater's criterion requires existence of a strictly feasible point for the inequality constraints for a convex optimization problem. Hence we need to have that

$$\|w\|_2 < R.$$

Therefore

$$R > 0.$$

2.4 - Let

$$t_i = y_i(\langle x_i, w\rangle + b), \qquad \sigma(t) = \frac{1}{1+e^{-t}}.$$

The Lagrangian is

$$\mathcal{L}(w,b,u) = \sum_{i=1}^{d} \log\left(\sigma(t_i)\right) - u(\langle w, w\rangle - R^2).$$

Now from the stationarity conditions, using $\frac{d}{dt}\log\sigma(t) = 1 - \sigma(t)$, the gradients are

$$\nabla_w \mathcal{L}(w,b,u) = \sum_{i=1}^{d} y_i x_i (1 - \sigma(t_i)) - 2uw = 0,$$

$$\frac{\partial}{\partial b}\mathcal{L}(w,b,u) = \sum_{i=1}^{d} y_i (1 - \sigma(t_i)) = 0.$$

Therefore, for $R > 0$, that is satisfying the Slater's criterion, we have the dual problem.

$$\begin{aligned}
\min_{u}\max_{w,b} \; & \sum_{i=1}^{d} \log(\sigma(y_i(\langle x_i, w\rangle + b))) \; - \; u\left(\langle w, w\rangle - R^2\right) \\
\text{s.t.} \; & \sum_{i=1}^{d} y_i x_i \big(1 - \sigma(y_i(\langle x_i, w\rangle + b))\big) - 2uw = 0 \\
& \sum_{i=1}^{d} y_i \big(1 - \sigma(y_i(\langle x_i, w\rangle + b))\big) = 0 \\
& w \in \mathbb{R}^n \\
& b \in \mathbb{R} \\
& u \geq 0.
\end{aligned}$$

Problem 3

3.1 - First, consider the following decision variables

$$x_i = \begin{cases} 1, & \text{if roll } i \text{ is used,} \\ 0, & \text{otherwise.} \end{cases}$$

$$y_{ij} \in \mathbb{Z}_{\geq 0}$$

Now, let n denote the total number of rolls. We develop a formulation to minimize the amount of of rolls used.

$$\begin{aligned}
\min \quad & \sum_{i \in [n]} x_i \\
\text{s.t.} \quad & \sum_{i \in [n]} y_{ij} \geq d_j && \forall j \in [4] \\
& \sum_{j \in [4]} l_j y_{ij} \leq L x_i && \forall i \in [n] \\
& x_i \in \{0, 1\} && \forall i \in [n] \\
& y_{ij} \in \mathbb{Z}_{\geq 0} && \forall i \in [n], j \in [4].
\end{aligned}$$

3.3 - We reformulate the problem using the Dantzig–Wolfe decomposition. First, let a_j^p denote how many pieces of type j are cut from a single roll, satisfying the constraint

$$\sum_{j=1}^{4} l_j a_j^p \leq L.$$

Let $\mathcal{P}$ be the set of all patterns p, we also define a variable $\lambda_p \geq 0$ representing how many rolls are cut using pattern p. The Dantzig–Wolfe master problem is given as follows.

$$\begin{aligned}
\min_{\lambda} \quad & \sum_{p \in \mathcal{P}} \lambda_p \\
\text{s.t.} \quad & \sum_{p \in \mathcal{P}} a_j^p \lambda_p \geq d_j && \forall j \in [4] \\
& \lambda_p \geq 0, && \forall p \in \mathcal{P}.
\end{aligned}$$

Now, let π_j denote the dual variables associated with the demand constraints. The column generation is performed by solving the pricing subproblem, which searches for a new cutting pattern with negative reduced cost. The reduced cost of a pattern p is

$$1 - \sum_{j \in [4]} \pi_j a_j^p.$$

To find a pattern with negative reduced cost, we solve the following sub-problem:

$$\begin{aligned}
\max \quad & \sum_{j \in [4]} \pi_j a_j \\
\text{s.t.} \quad & \sum_{j \in [4]} l_j a_j \leq L \\
& a_j \in \mathbb{Z}_{\geq 0} && j \in [4].
\end{aligned}$$

3.5 - The results of both approaches are given by the table below.

Table 1: Comparison of solution quality and runtime: full MIP vs. column generation

Method	Objective (rolls)	Time (s)
Full integer MIP	19.0	0.0148
Column generation (LP)	18.2	0.00316

The column-generation approach produced a lower bound of 18.2 in about 0.00316s, which was faster than solving the full integer MIP, while the true integer optimum is 19. The LP relaxation is therefore optimistic; rounding or a final integer solve on the generated patterns is needed to recover a feasible integer solution.

Problem 4

4.1 - First, define the decision variables $x_i \geq 0$, denoting the fraction of initial wealth invested in asset i, and $b_s \geq 0$, denoting the borrowing amount in scenario s. Thus, we have the formulation for the two-stage problem:

$$
\begin{aligned}
\min\ & c^\top x + \sum_{s\in[3]} p_s k_s b_s && && (2)\\
\text{s.t.}\ & \sum_{i\in[3]} x_i = 1 && && (3)\\
& \sum_{i\in[3]} G_{si} x_i + b_s \geq L_s && \forall s \in [3] && (4)\\
& x_i \geq 0 && \forall i \in [3] &&\\
& b_s \geq 0 && \forall s \in [3]. &&
\end{aligned}
$$

The objective function (2) minimizes the total expected cost: the investment cost $c^\top x$ plus the expected borrowing cost $\sum_s p_s k_s b_s$. The constraint (3) enforces that all initial wealth is allocated among the three assets. The constraint (4) ensures that, in each scenario, the portfolio return plus borrowing is sufficient to meet the liability L_s.

4.2 - For each scenario s, the subproblem can be written as

$$
\begin{aligned}
\min\ & p_s k_s b_s\\
\text{s.t.}\ & \sum_{i\in[3]} G_{si} x_i + b_s \geq L_s\\
& b_s \geq 0.
\end{aligned}
$$

Since this is a one-variable linear program, the optimal solution is

$$b_s^* = \max\Big\{0,\ L_s - \sum_{i\in[3]} G_{si} x_i\Big\},$$

giving the scenario cost function

$$Q_s(x) = p_s k_s\, b_s^* = p_s k_s\ \max\Big\{0,\ L_s - \sum_{i\in[3]} G_{si} x_i\Big\}.$$

Dual formulation. Introducing dual variables $\pi_s \geq 0$ for the first constraint and $\mu_s \geq 0$ for the non-negativity of b_s, the dual problem is

$$
\begin{aligned}
\max\ & (L_s - \sum_{i\in[3]} G_{si} x_i)\, \pi_s\\
\text{s.t.}\ & \pi_s + \mu_s = p_s k_s\\
& \pi_s \geq 0\\
& \mu_s \geq 0.
\end{aligned}
$$

The dual variables indicate whether borrowing is needed:

$$\pi_s = \begin{cases} p_s k_s & \text{if } L_s - \sum_i G_{si} x_i > 0,\\ 0 & \text{otherwise,} \end{cases} \qquad \mu_s = p_s k_s - \pi_s.$$

This dual is used to generate Benders cuts in the master problem.

4.3 - We define the Benders master problem by introducing auxiliary variables θ_s that estimate the scenario recourse costs $Q_s(x)$. The master problem is

$$
\begin{aligned}
\min\ & c^\top x + \sum_{s=1}^{3} p_s \theta_s(x) \\
\text{s.t.}\ & \sum_{i\in[3]} x_i = 1 \\
& x_i \geq 0 && \forall i \in [3] \\
& \theta_s \geq 0.
\end{aligned}
$$

The rows generated in Benders decomposition are Benders cuts, which enforce that each θ_s underestimates the true scenario cost $Q_s(x)$. For scenario s, a cut has the form

$$
\theta_s \geq \pi_s \left(L_s - \sum_{i\in[3]} G_{si} x_i \right),
$$

where π_s is the dual multiplier from the subproblem. These cuts iteratively refine the master problem until convergence.

4.4 - The problem was solved using Benders decomposition. The Python implementation converged in two iterations. The optimal portfolio and scenario recourse results are summarized in the table below.

Table 2: Benders decomposition solution for the two-stage investment problem

Scenario / Asset	Asset 1	Asset 2	Asset 3
Optimal x^*	0.4286	0.0	0.5714
Optimal b_s^*	0.0	0.0100	0.0
Total expected cost		0.003789	