1 CVXPY

Lab Objective: *CVXPY is a package of Python functions and classes designed for the purpose of convex optimization. In this lab we use these tools for linear and quadratic programming. We will solve various optimization problems using CVXPY and optimize eating healthily on a budget.*

Linear Programs

A *linear program* is a linear constrained optimization problem. Such a problem can be stated in several different forms, one of which is

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x} \\ \text{subject to} \quad & G\mathbf{x} \preceq \mathbf{h} \\ & A\mathbf{x} = \mathbf{b}. \end{aligned}$$

The symbol $\preceq$ denotes that the components of $G\mathbf{x}$ are less than the components of $\mathbf{h}$. In other words, if $\mathbf{x} \preceq \mathbf{y}$, then $x_i < y_i$ for all $x_i \in \mathbf{x}$ and $y_i \in \mathbf{y}$. CVXPY accepts $\leq, \geq$, and $=$ in its constraints as long as the equations satisfy convexity requirements described later in this chapter, so we can reformulate this problem in yet another form:

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x} \\ \text{subject to} \quad & G\mathbf{x} \preceq \mathbf{h} \\ & P\mathbf{x} \succeq \mathbf{q} \\ & A\mathbf{x} = \mathbf{b}. \end{aligned}$$

CVXPY accepts NumPy arrays and SciPy sparse matrices for the constraints, but the variable $\mathbf{x}$ must be a CVXPY `Variable`.

Consider the following example:

$$\begin{aligned} \text{minimize} \quad & -4x_1 - 5x_2 \\ \text{subject to} \quad & x_1 + 2x_2 \leq 3 \\ & 2x_1 + x_2 = 3 \\ & x_1, x_2 \geq 0 \end{aligned}$$

We can solve this problem using the following code. Note that `cvxpy.Problem()` accepts the constraints as a single list and that >= represents both standard and elementwise greater than or equal to. The symbols <= and == are similarly versatile.

```
>>> import cvxpy as cp
>>> import numpy as np

#First we'll initialize the objective
#We can declare x with its size and sign
>>> x = cp.Variable(2, nonneg = True)
>>> c = np.array([-4, -5])
>>> objective = cp.Minimize(c.T @ x)

#Then we'll write the constraints
>>> A = np.array([2, 1])
>>> G = np.array([1, 2])
>>> P = np.eye(2)
>>> constraints = [A @ x == 3, G @ x <= 3, P @ x >= 0] #This must be a list

#Assemble the problem and then solve it
>>> problem = cp.Problem(objective, constraints)
>>> print(problem.solve())
-8.999999999850528
>>> print(x.value)
array([1., 1.])
```

Achtung!

If you are having trouble with `pip install cvxpy` or `conda install cvxpy` check the following:

- CVXPY requires a C++ compiler, most MacOs ad Linux Systems have them built in. If you are running Windows, make sure that you have the "C++ builder tools" from the Visual Studio Build Tools installed.
- CVXPY requires specific versions of packages in order to run, check that you have the right version of your packages. The most common is NumPy is not up to date.

Problem 1. Solve the following convex optimization problem:

$$\begin{aligned} \text{minimize} \quad & 2x_1 + x_2 + 3x_3 \\ \text{subject to} \quad & x_1 + 2x_2 \leq 3 \\ & x_2 - 4x_3 \leq 1 \\ & 2x_1 + 10x_2 + 3x_3 \geq 12 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \\ & x_3 \geq 0 \end{aligned}$$

Return the minimizer $\mathbf{x}$ and the primal objective value.

l_1 Norm

The l_1 norm is defined

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

An l_1 minimization problem is minimizing a vector's l_1 norm, while fitting certain constraints. It can be written in the following form:

$$\begin{array}{rl} \text{minimize} & \|\mathbf{x}\|_1 \\ \text{subject to} & A\mathbf{x} = \mathbf{b}. \end{array}$$

CVXPY includes the l_1 norm and many other useful functions. To specify a norm in CVXPY, use the syntax `cp.norm(x, a)` where a represents your choice of norm (1 in this case).

Problem 2. Write a function called `l1Min()` that accepts a matrix A and vector $\mathbf{b}$ as NumPy arrays and solves the l_1 minimization problem. Return the minimizer $\mathbf{x}$ and the primal objective value.

To test your function consider the matrix A and vector $\mathbf{b}$ below.

$$A = \begin{bmatrix} 1 & 2 & 1 & 1 \\ 0 & 3 & -2 & -1 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 7 \\ 4 \end{bmatrix}$$

The linear system $A\mathbf{x} = \mathbf{b}$ has infinitely many solutions. Use `l1Min()` to verify that the solution which minimizes $||\mathbf{x}||_1$ is approximately $\mathbf{x} = [0., 2.571, 1.857, 0.]^T$ and the minimum objective value is approximately 4.429.

The Transportation Problem

Consider the following transportation problem: A piano company needs to transport thirteen pianos from their three supply centers (denoted by 1, 2, 3) to two demand centers (4, 5). Transporting a piano from a supply center to a demand center incurs a cost, listed in Table 1.3. The company wants to minimize shipping costs for the pianos while meeting the demand.

Supply Center	Number of pianos available
1	7
2	2
3	4

Table 1.1: Number of pianos available at each supply center

Demand Center	Number of pianos needed
4	5
5	8

Table 1.2: Number of pianos needed at each demand center

Supply Center	Demand Center	Cost of transportation	Number of pianos
1	4	4	p_1
1	5	7	p_2
2	4	6	p_3
2	5	8	p_4
3	4	8	p_5
3	5	9	p_6

Table 1.3: Cost of transporting one piano from a supply center to a demand center

A system of constraints can be defined using the variables p_1, p_2, p_3, p_4, p_5, and p_6. First, there cannot be a negative number of pianos transported along any route. Next, use tables 1.1 and 1.2 and the variables $p_1 ... p_6$ to define a supply or demand constraint for each location. You may want to format this as a matrix. Finally, the objective function is the number of pianos shipped along each route multiplied by the respective costs (Table 1.3).

Note

Since our answers must be integers, in general this problem turns out to be an NP-hard problem. There is a whole field devoted to dealing with integer constraints, called *integer linear programming*, which is beyond the scope of this lab. Fortunately, we can treat this particular problem as a standard linear program and still obtain integer solutions.

Problem 3. Solve the piano transportation problem. Return the minimizer $\mathbf{x}$ and the primal objective value.

Quadratic Programming

Quadratic programming is similar to linear programming, but the objective function is quadratic rather than linear. The constraints, if there are any, are still of the same form. Thus, $G, \mathbf{h}, A$, and $\mathbf{b}$ are optional. The formulation that we will use is

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\mathbf{x}^\mathsf{T} Q\mathbf{x} + \mathbf{r}^\mathsf{T}\mathbf{x} \\ \text{subject to} \quad & G\mathbf{x} \preceq \mathbf{h} \\ & A\mathbf{x} = \mathbf{b}, \end{aligned}$$

where Q is a positive semidefinite symmetric matrix.

As an example, consider the quadratic function

$$f(x_1, x_2) = 2x_1^2 + 2x_1x_2 + x_2^2 + x_1 - x_2.$$

There are no constraints, so we only need to initialize the matrix Q and the vector $\mathbf{r}$. To find these, we first rewrite our function to match the formulation given above. If we let

$$Q = \begin{bmatrix} a & b \\ b & c \end{bmatrix}, \qquad \mathbf{r} = \begin{bmatrix} d \\ e \end{bmatrix}, \qquad \text{and} \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

then

$$\begin{aligned}\frac{1}{2}\mathbf{x}^{\mathsf{T}}Q\mathbf{x}+\mathbf{r}^{\mathsf{T}}\mathbf{x} &= \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^{\mathsf{T}}\begin{bmatrix}a & b\\b & c\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}+\begin{bmatrix}d\\e\end{bmatrix}^{\mathsf{T}}\begin{bmatrix}x_1\\x_2\end{bmatrix}\\ &= \frac{1}{2}ax_1^2+bx_1x_2+\frac{1}{2}cx_2^2+dx_1+ex_2\end{aligned}$$

Thus, we see that the proper values to initialize our matrix Q and vector $\mathbf{r}$ are:

$$\begin{aligned} a &= 4 & d &= 1\\ b &= 2 & e &= -1\\ c &= 2 \end{aligned}$$

Now that we have the matrix Q and vector $\mathbf{r}$, we are ready to use the CVXPY function for quadratic programming, `cp.quad_form()`.

```
>>> Q = np.array([[4, 2],[2, 2]])
>>> r = np.array([1, -1])
>>> x = cp.Variable(2)
>>> prob = cp.Problem(cp.Minimize(.5 * cp.quad_form(x, Q) + r.T @ x))
>>> print(prob.solve())
[-1. 1.5]
>>> print(x.value)
-1.25
```

Problem 4. Find the minimizer and minimum of

$$g(x_1,x_2,x_3)=\frac{3}{2}x_1^2+2x_1x_2+x_1x_3+2x_2^2+2x_2x_3+\frac{3}{2}x_3^2+3x_1+x_3$$

(Hint: Write the function g to match the formulation given above before coding.)

So far we have only dealt with affine constraints. When working with non-affine constraints, be aware that CVXPY comes with some Disciplined Convex Programming (DCP) rules. A minimization problem requires a convex objective function; similarly, a maximization problem requires a concave objective function. Equality constraints (`==`) must be affine. Less-than constraints (`<=`) must have the left side convex and the right side concave. Greater-than constraints (`>=`) must have the left side concave and the right side convex. This webpage provides a list of which of CVXPY's functions are concave or convex. `https://www.cvxpy.org/tutorial/functions/index.html`

Problem 5. Write a function that accepts a matrix A and vector $\mathbf{b}$ and solves the following problem.

$$\begin{aligned}\text{minimize}\quad & \|A\mathbf{x}-\mathbf{b}\|_2\\ \text{subject to}\quad & \|\mathbf{x}\|_1 = 1\\ & \mathbf{x}\succeq 0\end{aligned}$$

To test your function, use the matrix A and vector $\mathbf{b}$ from Problem 2. The minimizer is approximately $\mathbf{x} = [0, 1, 0, 0]$ with objective value 5.099. Hint: `norm()` is a convex function, so you will have to think of a different way to take the 1-norm.

Eating on a Budget

In 2009, the inmates of Morgan County jail convinced Judge Clemon of the Federal District Court in Birmingham to put Sheriff Barlett in jail for malnutrition. Under Alabama law, in order to encourage less spending, "the chief lawman could go light on prisoners' meals and pocket the leftover change."[1]. Sheriffs had to ensure a minimum amount of nutrition for inmates, but minimizing costs meant more money for the sheriffs themselves. Judge Clemon jailed Sheriff Barlett until a plan was made to use all allotted funds, \$1.75 per inmate, to feed prisoners more nutritious meals. While this case made national news, the controversy of feeding prisoners in Alabama continues as of 2019[2].

The problem of minimizing cost while reaching healthy nutritional requirements can be approached as a convex optimization problem. Rather than viewing this problem from the sheriff's perspective, we view it from the perspective of a college student trying to minimize food cost in order to pay for higher education, all while meeting standard nutritional guidelines.

The file `food.npy` contains a dataset with nutritional facts for 18 foods that have been eaten frequently by college students working on this text. A subset of this dataset can be found in Table 1.4, where the "Food" column contains the list of all 18 foods.

The columns of the full dataset are:

Column 1: p, price (dollars)
Column 2: s, servings per container
Column 3: c, calories per serving
Column 4: f, fat per serving (grams)
Column 5: $\hat{s}$, sugar per serving (grams)
Column 6: $\hat{c}$, calcium per serving (milligrams)
Column 7: $\hat{f}$, fiber per serving (grams)
Column 8: $\hat{p}$, protein per serving (grams)

[1]Nossiter, Adam, 8 Jan 2009, "As His Inmates Grew Thinner, a Sheriff's Wallet Grew Fatter", *New York Times*,`https://www.nytimes.com/2009/01/09/us/09sheriff.html`

[2]Sheets, Connor, 31 January 2019, "Alabama sheriffs urge lawmakers to get them out of the jail food business", `https://www.al.com/news/2019/01/alabama-sheriffs-urge-lawmakers-to-get-them-out-of-the-jail-food-business.html`

Food	**Price** p dollars	**Servings** s	**Calories** c	**Fat** f g	**Sugar** $\hat{s}$ g	**Calcium** $\hat{c}$ mg	**Fiber** $\hat{f}$ g	**Protein** $\hat{p}$ g
Ramen	6.88	48	190	7	0	0	0	5
Potatoes	0.48	1	290	0.4	3.2	53.8	6.9	7.9
Milk	1.79	16	130	5	12	250	0	8
Eggs	1.32	12	70	5	0	28	0	6
Pasta	3.88	8	200	1	2	0	2	7
Frozen Pizza	2.78	5	350	11	5	150	2	14
Potato Chips	2.12	14	160	11	1	0	1	1
Frozen Broccoli	0.98	4	25	0	1	25	2	1
Carrots	0.98	2	52.5	0.3	6.1	42.2	3.6	1.2
Bananas	0.24	1	105	0.4	14.4	5.9	3.1	1.3
Tortillas	3.48	18	140	4	0	0	0	3
Cheese	1.88	8	110	8	0	191	0	6
Yogurt	3.47	5	90	0	7	190	0	17
Bread	1.28	6	120	2	2	60	0.01	4
Chicken	9.76	20	110	3	0	0	0	20
Rice	8.43	40	205	0.4	0.1	15.8	0.6	4.2
Pasta Sauce	3.57	15	60	1.5	7	20	2	2
Lettuce	1.78	6	8	0.1	0.6	15.5	1	0.6

Table 1.4: Subset of table containing food data

According to the FDA[1] and US Department of Health, someone on a 2000 calorie diet should have no more than 2000 calories, no more than 65 grams of fat, no more than 50 grams of sugar[2], at least 1000 milligrams of calcium[1], at least 25 grams of fiber, and at least 46 grams of protein[2] per day.

We can rewrite this as a convex optimization problem below.

[1]urlhttps://www.accessdata.fda.gov/scripts/InteractiveNutritionFactsLabel/pdv.html
[2]https://www.today.com/health/4-rules-added-sugars-how-calculate-your-daily-limit-t34731
[1]26 Sept 2018, `https://ods.od.nih.gov/factsheets/Calcium-HealthProfessional/`
[2]`https://www.accessdata.fda.gov/scripts/InteractiveNutritionFactsLabel/protein.html`

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{18} p_i x_i, \\
\text{subject to} \quad & \sum_{i=1}^{18} c_i x_i \leq 2000, \\
& \sum_{i=1}^{18} f_i x_i \leq 65, \\
& \sum_{i=1}^{18} \hat{s}_i x_i \leq 50, \\
& \sum_{i=1}^{18} \hat{c}_i x_i \geq 1000, \\
& \sum_{i=1}^{18} \hat{f}_i x_i \geq 25, \\
& \sum_{i=1}^{18} \hat{p}_i x_i \geq 46, \\
& x_i \geq 0.
\end{aligned}$$

Problem 6. Read in the file `food.npy`. Use CVXPY to identify how much of each food item a college student should each to minimize cost spent each day given these simplified nutrition requirements. Return the minimizing vector and the total amount of money spent.

According to this problem, what is the food you should eat most each day? What are the three foods you should eat most each week?

(Hint: Each nutritional value must be multiplied by the number of servings to get the nutrition value of the whole product).

You can learn more about CVXPY at `https://www.cvxpy.org/index.html`.