Regime-Switching Quadratic Programming in CVXPY

Overview

This document outlines how to implement a regime-switching quadratic programming (QP) problem in `cvxpy`, where the objective function switches per asset depending on whether the weight $w_i > 0$ or $w_i = 0$. This kind of problem is modeled as a Mixed-Integer Quadratic Program (MIQP) using the Big-M method.

Problem Setup

We consider:

- $W \in \mathbb{R}^n$ — portfolio weights (decision variable)
- $z_i \in \{0, 1\}$ — binary variable for regime switching

For each asset i:

$$z_i = \begin{cases} 1 & \text{if } W_i > 0 \quad \text{(Regime 1)} \\ 0 & \text{if } W_i = 0 \quad \text{(Regime 2)} \end{cases}$$

Each regime has a different quadratic objective:

$$\text{Regime 1: } W_i^T Q_1 W_i - c_1^T W_i, \quad \text{Regime 2: } W_i^T Q_2 W_i - c_2^T W_i$$

Modeling in CVXPY

We use auxiliary variables and the Big-M method to select the regime-dependent objective:

```
import cvxpy as cp
import numpy as np

# Problem parameters
n = 4
M = 1000 # Big-M constant

# Variables
W = cp.Variable(n)
z = cp.Variable(n, boolean=True)

# Regime-specific parameters
Q1 = np.eye(n)
Q2 = 5 * np.eye(n)
c1 = np.array([0.2, 0.1, 0.3, 0.1])
c2 = np.array([-0.1, -0.2, -0.1, -0.3])

constraints = []
objective_expr = 0

for i in range(n):
```

```
    w_i = W[i]
    z_i = z[i]

    # Enforce regime logic: w_i > 0  z_i = 1
    constraints += [
        w_i <= M * z_i,
        w_i >= 1e-4 * z_i
    ]

    # Objective per regime
    obj1 = Q1[i, i] * w_i**2 - c1[i] * w_i
    obj2 = Q2[i, i] * w_i**2 - c2[i] * w_i

    t_obj = cp.Variable()
    constraints += [
        t_obj >= obj1 - M * (1 - z_i),
        t_obj >= obj2 - M * z_i
    ]
    objective_expr += t_obj

# Portfolio constraints
constraints += [cp.sum(W) == 1, W >= 0]

# Solve
prob = cp.Problem(cp.Minimize(objective_expr), constraints)
prob.solve(solver=cp.GUROBI)

print("Optimal Weights:", W.value)
print("Regime Indicators:", z.value.round())
print("Objective Value:", prob.value)
```

Interpretation

- The binary variable z_i activates one of the two regime-specific objectives.
- The auxiliary variable t_{obj} ensures that only the active regime's objective contributes.
- The constraint $\sum W = 1$ ensures budget normalization.

Extensions

- Allow different Q_1, Q_2, c_1, c_2 per asset or segment.
- Add constraints or penalties for switching regimes.
- Use transaction cost terms, or regime-based risk limits.