Problem 3 Code & Results

Want to solve the following SOC problem

$$\begin{aligned} \max \quad & z \\ \text{s.t} \quad & \boldsymbol{\mu}\{k\}^\top \mathbf{x} - \lambda \|\mathbf{V}\{k\}\mathbf{x}\|_2 \geq z\,, \\ & \textstyle\sum_{i=1}^n (x_i^+ - x_i^-) = 1\,, \\ & \textstyle\sum_{i=1}^n x_i^- \leq 4\,, \\ & x_k^+, x_k^- \geq 0\,, \quad \forall k = 1, \ldots, n \end{aligned}$$

[1]:
```
import numpy as np
import scipy.io as sio
import cvxpy as cvx
import gurobi as grb
import matplotlib.pyplot as plt
%matplotlib inline
```

[2]:
```
mat_contents = sio.loadmat('multiscenario.mat')
```

[3]:
```
_mu = mat_contents['mu']
_V = mat_contents['V']
_lambda = mat_contents['lambda']

# convert into numpy arrays
m = _mu.shape[0] # number of scenarios
d = _mu[0][0].shape[0] # dimension of the problem

mu = np.empty((m,d))
V = np.empty((m,d,d))
for i in range(m):
    mu[i,:] = np.ravel(_mu[i][0])
    V[i,:,:] = _V[i][0]
```

[4]:
```
#r = np.max(mu.ravel())/2
#B = 10

x = cvx.Variable(d)
beta = cvx.Variable(1)
x_pos = cvx.Variable(d)
x_neg = cvx.Variable(d)
x = x_pos - x_neg

objective = cvx.Maximize(beta)
constraints = []

# volatility constriants
for k in range(m):
    constraints += [mu[k,:].T*x - _lambda*cvx.norm(V[k,:,:]*x) >= beta];

# mean constraints
#for k in range(m):
#    constraints += [mu[k,:].T*x >= r];

# portfolio constraint
```

```
constraints += [sum(x_pos) - sum(x_neg) == 1]
constraints += [sum(x_neg) <= 4]
constraints += [x_pos >= 0, x_neg >=0]

# absolute value constraints
#constraints += [cvx.abs(x) <= B]

prob = cvx.Problem(objective,constraints)
prob.solve()
```

[4]: 5.982651221281714