Nirbhay Sharma (B19CSE114)

Optimization for ML - Lab-3

Que1

matrics for constraints

```
{
    "c":[[20],[25],[22],[28],[15],[18],[23],[17],[19],[17],[21],[24],[25],
[23],[24],[24]],
    "A":[16,16],
    "b":[16,1],
    "aeq":[
        [1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0],
        [0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0],
        [0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0],
        [0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1],
        [1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0],
        [0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0],
        [0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0],
        [0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1]
    ],
    "beq":[[1],[1],[1],[1],[1],[1],[1],[1]],
    "solvers":"glpk"
}
```

Code

```
import numpy as np
import pandas as pd
import cvxopt as cp
import json, sys

json_file = sys.argv[1]
with open(json_file, 'r') as jf:
    data = json.load(jf)

c=np.array(data['c'])
A=-np.eye(*data['A'])
b=np.zeros(data['b'])
aeq = np.array(data['aeq'])
beq = np.array(data['beq'])

sol = cp.solvers.lp(cp.matrix(
    c,tc="d"),
    cp.matrix(A,tc="d"),
    cp.matrix(b,tc="d"),
    cp.matrix(aeq,tc='d'), # error in named parameters
    cp.matrix(beq,tc='d'),
    solver=data['solvers'])

print(sol["x"],sol["primal objective"])

"""
[ 0.00e+00]
[ 0.00e+00]
[ 1.00e+00]
[ 0.00e+00]
[ 1.00e+00]
[ 0.00e+00]
[ 0.00e+00]
[ 0.00e+00]
[ 0.00e+00]
[ 1.00e+00]
[ 0.00e+00]
[ 0.00e+00]
[ 0.00e+00]
[ 0.00e+00]
[ 0.00e+00]
[ 1.00e+00]
 78.0
"""
```

Que2

matrics for constraints

```
{
    "c":[[9.2],[-6],[-1.3],[4.1],[3],[8],[-2.1]],
    "A":[7,7],
    "b":[7,1],
    "aeq":[
        [1,1,0,0,0,0,0],
        [-1,0,-1,1,0,0,0],
        [0,-1,0,0,1,1,0],
        [0,0,1,0,-1,0,-1],
        [0,0,0,-1,0,-1,1]
    ],
    "beq":[[12],[0],[0],[-8],[-4]],
    "solvers":"glpk"
}
```

Code

```
import numpy as np
import pandas as pd
import cvxopt as cp
import json, sys

json_file = sys.argv[1]
with open(json_file, 'r') as jf:
    data = json.load(jf)

c=np.array(data['c'])
A=-np.eye(*data['A'])
b=np.zeros(data['b'])
aeq = np.array(data['aeq'])
beq = np.array(data['beq'])

sol = cp.solvers.lp(cp.matrix(
    c,tc="d"),
    cp.matrix(A,tc="d"),
    cp.matrix(b,tc="d"),
    cp.matrix(aeq,tc='d'), # error in named parameters
    cp.matrix(beq,tc='d'),
    solver=data['solvers'])

print(sol["x"],sol["primal objective"])

"""
OPTIMAL LP SOLUTION FOUND
[ 0.00e+00]
[ 1.20e+01]
[ 4.00e+00]
[ 4.00e+00]
[ 1.20e+01]
[ 0.00e+00]
[ 0.00e+00]
 -24.800000000000004
"""
```

Que3

matrics for constraints

```
{
    "Q":[[6,2],[2,2]],
    "c":[[1],[6]],
    "A":[[-2,-3],[-1,0],[0,-1]],
    "b":[[-4],[0],[0]],
    "const":2
}
```

Code

```
import numpy as np
import pandas as pd
import cvxopt as cp
import json, sys

json_file = sys.argv[1]
with open(json_file, 'r') as jf:
    data = json.load(jf)

Q=np.array(data['Q'])
c=np.array(data['c'])
A=np.array(data['A'])
b = np.array(data['b'])

sol = cp.solvers.qp(
    cp.matrix(Q,tc="d"),
    cp.matrix(c,tc="d"),
    cp.matrix(A,tc="d"),
    cp.matrix(b,tc='d')
)

print(sol["x"])
print(sol['primal objective'] + data['const'])

"""
Optimal solution found.
[ 5.00e-01]
[ 1.00e+00]

11.250000683093692
"""
```

Que4

matrics for constraints

```
{
    "Q":[[2,0],[0,2]],
    "c":[[-2],[-3]],
    "A":[[1,1],[2,1],[-1,0],[0,-1]],
    "b":[[2],[3],[0],[0]],
    "const":0
}
```

Code

```
import numpy as np
import pandas as pd
import cvxopt as cp
import json, sys

json_file = sys.argv[1]
with open(json_file, 'r') as jf:
    data = json.load(jf)

Q=np.array(data['Q'])
c=np.array(data['c'])
A=np.array(data['A'])
b = np.array(data['b'])

sol = cp.solvers.qp(
    cp.matrix(Q,tc="d"),
    cp.matrix(c,tc="d"),
    cp.matrix(A,tc="d"),
    cp.matrix(b,tc='d')
)

print(sol["x"])
print(sol['primal objective'] + data['const'])

"""
Optimal solution found.
[ 7.50e-01]
[ 1.25e+00]

3.1249998233364806
"""
```