Nirbhay Sharma (B19CSE114)

Optimization for ML - Lab-2

Que1

matrics for constraints

```
{
    "c":[[0],[0],[1]],
    "A":[[5,2,-1],[3,7,-1],[-4,-3,0],[1,2,0],[-1,0,0],[0,-1,0]],
    "b":[[0],[0],[-6],[3],[0],[0]],
    "aeq":[[3,1,0]],
    "beq":[[3]],
    "solvers":null
}
```

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.array(data['A'])
b=np.array(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 solution found.
[ 6.00e-01]
[ 1.20e+00]
[ 1.02e+01]
 10.199999996490329
"""
```

Que2

matrics for constraints

```
{
    "c":[[2],[3],[3],[2],[4],[1],[2],[3],[1]],
    "A":[
            [-1,0,0,0,0,0,0,0,0],
            [0,-1,0,0,0,0,0,0,0],
            [0,0,-1,0,0,0,0,0,0],
            [0,0,0,-1,0,0,0,0,0],
            [0,0,0,0,-1,0,0,0,0],
            [0,0,0,0,0,-1,0,0,0],
            [0,0,0,0,0,0,-1,0,0],
            [0,0,0,0,0,0,0,-1,0],
            [0,0,0,0,0,0,0,0,-1]
        ],
    "b":[[0],[0],[0],[0],[0],[0],[0],[0],[0]],
    "aeq":[
        [1,1,1,0,0,0,0,0,0],
        [-1,0,0,1,1,0,0,0,0],
        [0,-1,0,-1,0,1,1,0,0],
        [0,0,0,0,-1,-1,0,1,0],
        [0,0,-1,0,0,0,-1,0,1],
        [0,0,0,0,0,0,0,-1,-1]
    ],
    "beq":[[1],[3],[0],[0],[0],[-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.array(data['A'])
b=np.array(data['b'])
aeq = np.array(data['aeq'])
```

```
beq = np.array(data['beq'])

print(c.shape)
print(A.shape)
print(b.shape)
print(aeq.shape)
print(beq.shape)


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

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

"""
OPTIMAL LP SOLUTION FOUND
[ 0.00e+00]
[ 0.00e+00]
[ 1.00e+00]
[ 3.00e+00]
[ 0.00e+00]
[ 0.00e+00]
[ 3.00e+00]
[ 0.00e+00]
[ 4.00e+00]
 19.0
"""
```

Que3

matrics for constraints

```
{
    "c":[[1],[5],[1],[4],[2]],
    "A":[5,5],
    "b":[[1],[3],[2],[2],[3]],
    "aeq":[
        [1,1,0,0,0],
        [-1,0,1,1,0],
        [0,-1,-1,0,1],
        [0,0,0,-1,-1]
    ],
    "beq":[[2],[2],[-2],[-2]],
    "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.array(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'),
    cp.matrix(beq,tc='d'),
    solver = data["solvers"])

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

"""
OPTIMAL LP SOLUTION FOUND
[ 1.00e+00]
[ 1.00e+00]
[ 2.00e+00]
[ 1.00e+00]
[ 1.00e+00]
 14.0
"""
```

Que4

matrics for constraints

```
{
    "c":[[4],[6],[6],[6],[8],[9],[5],[4],[6],[5],[5],[7],[6],[8],[4],[9],[3],[7],[9],[6]],
    "A":[20,20],
    "b":[20,1],
    "aeq":[
        [1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
        [-1,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
        [0,-1,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0],
        [0,0,-1,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0],
        [0,0,0,-1,0,0,-1,0,0,-1,0,0,1,1,0,0,0,0,0,0],
        [0,0,0,0,-1,0,0,-1,0,0,-1,0,0,0,1,1,0,0,0,0],
        [0,0,0,0,0,-1,0,0,-1,0,0,-1,0,0,0,0,1,1,0,0],
        [0,0,0,0,0,0,0,0,0,0,0,0,-1,0,-1,0,-1,0,1,0],
        [0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,-1,0,-1,0,1],
        [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,-1]],
    "beq":[[1],[0],[0],[0],[0],[0],[0],[0],[0],[-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"])

"""
OPTIMAL LP SOLUTION FOUND
[ 0.00e+00]
[ 1.00e+00]
[ 0.00e+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]
[ 0.00e+00]
[ 1.00e+00]
[ 0.00e+00]
[ 0.00e+00]
[ 0.00e+00]
[ 1.00e+00]
[ 0.00e+00]
 23.0
"""
```

Que5

matrics for constraints

```
{
    "c":[[20],[28],[19],[13],[15],[30],[31],[28],[40],[21],[20],[17],[21],
[28],[26],[12]],
    "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"])

"""
OPTIMAL LP SOLUTION FOUND
[ 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]
 67.0
"""
```

Que6

matrics for constraints

```
{
    "c":[[37.7],[32.9],[33.8],[37.0],[35.4],[43.4],[33.1],[42.2],[34.7],
[41.8],[33.3],[28.5],[38.9],[30.4],[33.6],[29.2],[26.4],[29.6],[28.5],
[31.1]],
    "A":[20,20],
    "b":[20,1],
    "aeq":[
        [1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0],
        [0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0],
        [0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0],
        [0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0],
        [1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
        [0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0],
        [0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0],
        [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1]
    ],
    "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"])

"""
OPTIMAL LP SOLUTION FOUND
[ 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]
[ 0.00e+00]
[ 0.00e+00]
[ 1.00e+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]
 126.2
"""
```