ISYE 6669 Deterministic Optimization

Homework 11

April 4th, 2021

1.1.

Consider the following linear programming problem:

$$
\begin{array}{lllllllll}
\max & & x_{12} & & & +x_{22} & +x_{23} & & \\
\text{s.t.} & x_{11} & & & & & +x_{23} & \leq & 12 \\
& x_{11} & +x_{12} & +x_{13} & & & & = & 20 \\
& & & & x_{21} & +x_{22} & +x_{23} & = & 20 \\
& x_{11} & & & +x_{21} & & & = & 10 \\
& & x_{12} & & & +x_{22} & & = & 20 \\
& & & x_{13} & & & +x_{23} & = & 10 \\
& & & x_{ij} \geq 0, & \multicolumn{5}{l}{\text{for all } i = 1,2, j = 1,2,3.}
\end{array}
$$

Figure 1: Original Problem

The polyhedron P, is defined from the 5 easy constraints listed above. Since each of our easy constraints are defined by equalities (i.e. finite numbers), and all of our x_{ij} are non-negative, we know that each constraint can be contained within a form of a hypershere or other shape that can enclose these constraints.

1.2.

Since P is bounded, we can use the extreme point representation forP. The Dantzig-Wolfemaster problem can be written as:

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{N} \lambda_i \left(\boldsymbol{c}^T \boldsymbol{x}^i\right) \\
\text{s.t.} \quad & \sum_{j=1}^{N} \lambda_i \left(\boldsymbol{D}\boldsymbol{x}^i\right) \leq \boldsymbol{b} \\
& \sum_{j=1}^{N} \lambda_i = 1, \\
& \lambda_i \geq 0, \quad \forall i = 1, \ldots, N.
\end{aligned}
$$

Figure 2: Danzig Wolfe-Representation Extreme Points

$c \ = \ [0, \ 1, 0, 0, 1, 1]$

$D = \ [1, 0, 0, 0, 0, 1]$

$b \ = \ 12$

1.3.

You are given the following two extreme points of the polyhedron P:

$$x^1 = (x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}) = (10, 10, 0, 0, 10, 10),$$

and

$$x^2 = (x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}) = (0, 10, 10, 10, 10, 0).$$

Construct the restricted master problem using these two extreme points. Use variables λ_1 and λ_2 for the restricted master problem.

Figure 3: Initial Extreme Points

$$c^T x^1 = [0, 1, 0, 0, 1, 1]\begin{bmatrix}10\\10\\0\\0\\10\\10\end{bmatrix}$$

$$= 30$$

$$c^T x^2 = [0, 1, 0, 0, 1, 1]\begin{bmatrix}0\\10\\10\\10\\10\\0\end{bmatrix}$$

$$= 20$$

$$Dx^1 = [1, 0, 0, 0, 0, 1]\begin{bmatrix}10\\10\\0\\0\\10\\10\end{bmatrix}$$

$$= 20$$

$$Dx^2 = [1, 0, 0, 0, 0, 1]\begin{bmatrix}0\\10\\10\\10\\10\\0\end{bmatrix}$$

$$= 0$$

RMP :

$$\max_{\lambda_1,\lambda_2} \quad 30\lambda_1 + 20\lambda_2$$

$$\begin{aligned} s.t. \quad & 20\lambda_1 \leq 12 \\ & \lambda_1 + \lambda_2 = 1 \\ & \lambda_1, \lambda_2 \geq 0 \end{aligned}$$

1.4.

Therefore, we can see that the optimal solution to the above is:

$\lambda_1 = 0.6, \ \lambda_2 = 0.4$

1.5.

$$Given\ that\ B = \begin{bmatrix} 20 & 0 \\ 1 & 1 \end{bmatrix},\ then\ B^{-1} = \begin{bmatrix} \frac{1}{20} & 0 \\ -\frac{1}{20} & 1 \end{bmatrix}$$

We can confirm the optimal solution from 4) by the following:

$$\begin{aligned} &= B^{-1}b \\ &= \begin{bmatrix} \frac{1}{20} & 0 \\ -\frac{1}{20} & 1 \end{bmatrix} [12, 1] \\ &= \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix} \end{aligned}$$

To calculate the dual variables we'll use the inverse and cost vector as follows:

$$[\hat{y}, \hat{r}] = c_B^T B^{-1} = [30, 20]^T \begin{bmatrix} \frac{1}{20} & 0 \\ -\frac{1}{20} & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 20 \end{bmatrix}$$

1.6.

Solve the following Pricing Problem to compute the minimum reduced cost:

$$\hat{Z} = \min \left(c^T - \hat{y}^T D\right) x - \hat{r}$$

$$s.t.\ x \in P$$

$$= c^T - \hat{y}^T D = [0,1,0,0,1,1] - \left(\frac{1}{2}\right)[1,0,0,0,0,1]$$

$$= \left[-\frac{1}{2}, 1, 0, 0, 1, \frac{1}{2}\right]$$

$$\widehat{Z} = max\left(-\frac{1}{2}x_{11} + x_{12} + x_{22} + \frac{1}{2}x_{23} - 20\right)$$

$s.t.$

$$\begin{array}{cccccccc}
x_{11} & +x_{12} & +x_{13} & & & & = & 20 \\
 & & & x_{21} & +x_{22} & +x_{23} & = & 20 \\
x_{11} & & & +x_{21} & & & = & 10 \\
 & x_{12} & & & +x_{22} & & = & 20 \\
 & & x_{13} & & & +x_{23} & = & 10 \\
 & & \multicolumn{6}{l}{x_{ij} \geq 0, \quad \text{for all } i = 1,2, j = 1,2,3}
\end{array}$$

1.7.

$$\begin{array}{cccccccc}
x_{11} & +x_{12} & +x_{13} & & & & = & 20 \\
 & & & x_{21} & +x_{22} & +x_{23} & = & 20 \\
x_{11} & & & +x_{21} & & & = & 10 \\
 & x_{12} & & & +x_{22} & & = & 20 \\
 & & x_{13} & & & +x_{23} & = & 10 \\
 & & \multicolumn{6}{l}{x_{ij} \geq 0, \quad \text{for all } i = 1,2, j = 1,2,3}
\end{array}$$

Figure 4: Constraints to Original Problem

When you look at the first two constraints, followed by the remaining 3, they denote our total suppy and total demand:

$$\sum_{j=1}^{3} = \begin{bmatrix} x_{11} \\ x_{21} \end{bmatrix} + \begin{bmatrix} x_{12} \\ x_{22} \end{bmatrix} + \begin{bmatrix} x_{13} \\ x_{23} \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$$

$$\sum_{x=1}^{2} = \begin{bmatrix} x_{11} + x_{21} \\ x_{12} + x_{22} \\ x_{13} + x_{23} \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

Which makes sense when thinking back to Week 5, as the summation of the columns refers to demand while the summation of rows refers to supply.

2.1

```
[19]: import numpy as np
      import matplotlib.pyplot as plt
      from numpy.linalg import matrix_rank,svd
      import math

      #1) Load the cloud image from the text file
      X = np.loadtxt('./clownImage.txt')
      plt.imshow(X)

[19]: <matplotlib.image.AxesImage at 0x28e99c0ec08>
```

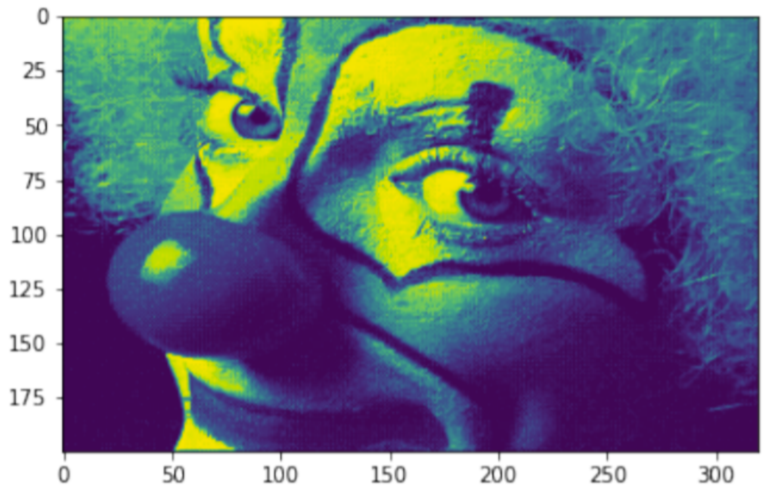

Figure 5: Original Image

2.2

```
#2) Calculate SVD Decomposition
X_u,X_s,X_vh = svd(X)
```

2.3

```
def low_rank_k(u,s,vh,rank_number):
# rank k approx

    u = u[:,:rank_number]
    vh = vh[:rank_number,:]
    s = s[:rank_number]
    s = np.diag(s)
    my_low_rank = np.dot(np.dot(u,s),vh)
    return my_low_rank

low_rank_5 = low_rank_k(X_u,X_s,X_vh,5)
```

```
low_rank_15 = low_rank_k(X_u,X_s,X_vh,15)
low_rank_25 = low_rank_k(X_u,X_s,X_vh,25)
```

2.4

```
f, axarr = plt.subplots(1,3,figsize=(20, 10))
axarr[0].set_title('Rank 5 Approximation')
axarr[0].imshow(low_rank_5)
axarr[1].set_title('Rank 15 Approximation')
axarr[1].imshow(low_rank_15)
axarr[2].set_title('Rank 25 Approximation')
axarr[2].imshow(low_rank_25)
```

```
<matplotlib.image.AxesImage at 0x28e9e63b4c8>
```