SCO Project 1 - Group 20

Aadithya Anandaraj, Anisha Alluru, Khyathi Balusu

2/9/2020

Q1

Part 1

The decision variables are the amounts invested in each campaign.

The objective at hand is to maximize the ROI of the investment which is given by

$(0.031*Print)+(0.049*TV)+(0.024*SEO)+(0.039*AdWords)+(0.016*Facebook)+(0.024*LinkedIn)+(0.046*Instagram)+$

The constraints are as follows: a.

$$Print + TV <= Facebook + Email$$

$$Print + TV - Facebook - Email <= 0$$

b.
$$Facebook + LinkedIn + Instagram + Snapchat + Twitter >= 2*(SEO + AdWords)$$

$$Facebook + LinkedIn + Instagram + Snapchat + Twitter - 2*SEO - 2*AdWords >= 0$$

c.
$$Print <= 3, TV <= 3, SEO <= 3.....Email <= 3$$

d. Budget Constraint
$$Print + TV + SEO + ... + Twitter + Email <= 10$$

e.
$$Print, TV, SEO, ..., Twitter, Email >= 0$$

```
library(lpSolve)
```

```
## Warning: package 'lpSolve' was built under R version 3.6.2
```

```
c=rep(c(0.031, 0.049, 0.024, 0.039, 0.016, 0.024, 0.046, 0.026, 0.033, 0.044))
A=matrix(0,13,10)
A[1,1:10]=c(1,1,0,0,-1,0,0,0,0,-1)
A[2,1:10]=c(0,0,-2,-2,1,1,1,1,1,0)
A[3:12,1:10]=diag(10)
A[13,1:10]=rep(1,10)
A
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]    1    1    0    0   -1    0    0    0    0    -1
##  [2,]    0    0   -2   -2    1    1    1    1    1     0
##  [3,]    1    0    0    0    0    0    0    0    0     0
##  [4,]    0    1    0    0    0    0    0    0    0     0
##  [5,]    0    0    1    0    0    0    0    0    0     0
##  [6,]    0    0    0    1    0    0    0    0    0     0
##  [7,]    0    0    0    0    1    0    0    0    0     0
##  [8,]    0    0    0    0    0    1    0    0    0     0
##  [9,]    0    0    0    0    0    0    1    0    0     0
## [10,]    0    0    0    0    0    0    0    1    0     0
## [11,]    0    0    0    0    0    0    0    0    1     0
## [12,]    0    0    0    0    0    0    0    0    0     1
## [13,]    1    1    1    1    1    1    1    1    1     1
```

```r
dir=c("<=",'>=','<=',rep('<=',10))
B=c(0,0,rep(3,10),10)
```

Part 2

```r
alc1=lp("max",c,A,dir,B)
alc1$solution
```

```
##  [1] 0 3 0 1 0 0 3 0 0 3
```

```r
alc1$objval
```

```
## [1] 0.456
```

Part 3

```r
library(base)
#setwd('D:/ANISHA/DOCUMENTS/Studies/MSBA/Course Work/Spring 2019/Stochastic Control and Optimization/Gr
source('allocation_g20.R')
result = allocation(c(0.031, 0.049, 0.024, 0.039, 0.016, 0.024, 0.046, 0.026, 0.033, 0.044), 3, 10)
result$objval
```

```
## [1] 0.456
```

```r
result$solution
```

```
##  [1] 0 3 0 1 0 0 3 0 0 3
```

Part 4

```r
library(base)
#setwd('D:/ANISHA/DOCUMENTS/Studies/MSBA/Course Work/Spring 2019/Stochastic Control and Optimization/Gr
#source(allocation_g20.R)
alc2 = allocation(c(0.031, 0.049, 0.024, 0.039, 0.016, 0.024, 0.046, 0.026, 0.033, 0.044), 10, 10)
cat('Without upper bound constraint:\n')
```

```
## Without upper bound constraint:
```

```r
alc2$objval
```

```
## [1] 0.465
```

```r
alc2$solution
```

```
##  [1] 0 5 0 0 0 0 0 0 0 5
```

```r
cat('With upper bound constraint:\n')
```

```
## With upper bound constraint:
```

```r
alc1$objval
```

```
## [1] 0.456
```

```r
alc1$solution
```

```
##  [1] 0 3 0 1 0 0 3 0 0 3
```

```r
cat('The difference in both objective values is ', alc2$objval-alc1$objval)
```

```
## The difference in both objective values is  0.009
```

Q2

#Part 1

Using the new ROI vector,

```r
library(base)
#setwd('D:/ANISHA/DOCUMENTS/Studies/MSBA/Course Work/Spring 2019/Stochastic Control and Optimization/Gr
#source(allocation_g20.R)
alc3 = allocation(c(0.049, 0.023, 0.024, 0.039, 0.044, 0.046, 0.026, 0.019, 0.037, 0.026), 3, 10)
cat('With upper bound constraint and new ROI:\n')
```

```
## With upper bound constraint and new ROI:
```

```r
alc3$objval
```

```
## [1] 0.456
```

```r
alc3$solution
```

```
##  [1] 3 0 0 1 3 3 0 0 0 0
```

The objective value is the same as their counterparts using the previous ROI vector, but the allocation is different.

#Part 2

```r
library(base)
#setwd('D:/ANISHA/DOCUMENTS/Studies/MSBA/Course Work/Spring 2019/Stochastic Control and Optimization/Gr
#source(allocation_g20.R)
alc1_new_obj=((0.023*3)+(0.039*1)+(0.026*3)+(0.026*3))
alc2_new_obj=((0.023*5)+(0.026*5))
disappointment_alc1 = alc1$objval - alc1_new_obj
disappointment_alc2 = alc2$objval - alc2_new_obj

cat('Disappointment alc1')
```

```
## Disappointment alc1
```

```r
disappointment_alc1
```

```
## [1] 0.192
```

```r
cat('Disappointment alc2')
```

```
## Disappointment alc2
```

```r
disappointment_alc2
```

```
## [1] 0.22
```

The 3rd constraint is valuable as relaxing the upper bound has a higher disappointment.

#Part 3

```r
library(base)
#setwd('D:/ANISHA/DOCUMENTS/Studies/MSBA/Course Work/Spring 2019/Stochastic Control and Optimization/Gr
#source(allocation_g20.R)

alc1_avg=(alc1$objval+alc1_new_obj)/2
alc2_avg=(alc2$objval+alc2_new_obj)/2

alc3_old_obj=((0.031*3)+(0.039*1)+(0.016*3)+(0.024*3))
alc3_avg=(alc3_old_obj+alc3$objval)/2
```

```r
#tweaking the ROI vector
old_ROI=c(0.031, 0.049, 0.024, 0.039, 0.016, 0.024, 0.046, 0.026, 0.033, 0.044)
new_ROI=c(0.049, 0.023, 0.024, 0.039, 0.044, 0.046, 0.026, 0.019, 0.037, 0.026)
avg_ROI=(old_ROI+new_ROI)/2
avg_ROI
```

```
##  [1] 0.0400 0.0360 0.0240 0.0390 0.0300 0.0350 0.0360 0.0225 0.0350 0.0350
```

```r
alc=allocation(c(avg_ROI), 4, 10)
cat('The solution is')
```

```
## The solution is
```

```r
alc$solution
```

```
##  [1] 4.0000000 0.0000000 0.0000000 0.6666667 0.0000000 0.0000000 1.3333333
##  [8] 0.0000000 0.0000000 4.0000000
```

```r
cat('Objective value')
```

```
## Objective value
```

```r
alc$objval
```

```
## [1] 0.374
```

```r
cat('Average objective value of alc1')
```

```
## Average objective value of alc1
```

```r
alc1_avg
```

```
## [1] 0.36
```

```r
cat('Average objective value of alc2')
```

```
## Average objective value of alc2
```

```r
alc2_avg
```

```
## [1] 0.355
```

```r
cat('Average objective value of alc3')
```

```
## Average objective value of alc3
```

```
alc3_avg
```

```
## [1] 0.354
```

By tweaking the ROI vector to be the average of the old and the new ROI vector and changing the upper bound to 4, we obtained the allocation above that dominates the average objective values of the previous (alc1,alc2,alc3).

Q3

Part1

```
monthBudget = 10

A<-matrix(0,13,10)
A[1,]<-c(1,1,0,0,-1,0,0,0,0,-1)
A[2,]<-c(0,0,2,2,-1,-1,-1,-1,-1,0)
A[3,]<-c(rep(1,10))
A[4:13,]<-diag(10)

dir<-c(rep("<=",13))


outputMatrix = matrix(0,12,10)

for (i in 1:12){
  #objective coefficients
  monthROI<-c(ROI_mat[i,]/100)

  b<-c(0,0,monthBudget,rep(3,10))

  monthAllocation=lp("max",monthROI,A,dir,b,compute.sens = 1)


  monthSolution=monthAllocation$solution
  monthObjective=monthAllocation$objval

  outputMatrix[i,] = monthSolution
  monthBudget = monthBudget + 0.5*monthObjective
}

print (outputMatrix)
```

```
##          [,1] [,2] [,3]      [,4]     [,5]      [,6]     [,7] [,8]
##  [1,] 3.000000    0    0 1.3333333 0.000000 0.0000000 2.666667    0
##  [2,] 3.000000    0    0 2.3955000 3.000000 0.0000000 0.000000    0
##  [3,] 0.000000    0    0 3.0000000 0.000000 3.0000000 1.389648    0
##  [4,] 0.000000    0    0 3.0000000 0.000000 3.0000000 3.000000    0
##  [5,] 1.804100    0    0 0.0000000 0.000000 0.0000000 3.000000    0
```

```
##  [6,] 3.000000    0    0 0.0000000 0.000000 0.0000000 3.000000    0
##  [7,] 0.000000    0    0 3.0000000 2.247555 0.0000000 3.000000    0
##  [8,] 3.000000    0    0 1.8272941 0.000000 0.6545882 0.000000    0
##  [9,] 1.362933    0    0 3.0000000 0.000000 3.0000000 0.000000    0
## [10,] 0.000000    0    0 3.0000000 0.000000 3.0000000 3.000000    0
## [11,] 3.000000    0    0 2.0564210 0.000000 1.1128419 3.000000    0
## [12,] 3.000000    3    0 0.4279507 3.000000 0.0000000 0.000000    0
##           [,9]    [,10]
##  [1,] 0.000000 3.000000
##  [2,] 1.791000 0.000000
##  [3,] 3.000000 0.000000
##  [4,] 1.596856 0.000000
##  [5,] 3.000000 3.000000
##  [6,] 2.020172 3.000000
##  [7,] 3.000000 0.000000
##  [8,] 3.000000 3.000000
##  [9,] 3.000000 1.362933
## [10,] 0.000000 2.955475
## [11,] 0.000000 3.000000
## [12,] 0.000000 3.000000
```

Part2

Multi period allocation is the iteration run over monthly budget of the previous month. This would make the first year's investment same irrespective of multi period or previous problem's single problem allocation.

Part 3

With the newly added constraints, this solution would not hold good. If the monthly change should be no more than 1M, for a stable relationship with each platform given, then with the constrains, the first year solutions would wary.