Mathematical Optimization Model

Final Project

Melissa Paniagua

MIS 64018 Quantitative Management Modeling
Professor Chaojiang (CJ) Wu
Kent State University

12 December 2020

Abstract

The objective of this assignment is to define, formulate, and solve a mathematical optimization model.

I. Introduction

The current project is about building groups for a class project, and the main goal is to ensure to maximize the chance that each group will do well on a class project.

The class consists of 12 students, and we will form 4 groups of 3 members in each group. The following characteristics of students are some factors that affect the success of groups projects:

- GPA: It has information about historical performance. The scale is 0-4.
- Gender: Determines if the student is male or female. In our dataset 0 mean male, and 1 refers to a female.
- Hours worked on the project: It measures the number of hours worked individually on the group project. In other words, it measures if the student (individually) started working on the project on their own. The GroupProj dataset has a range of 0 to 15 hours. **NOTE: Hours individually worked before starting the project.**
- Current Course Grade: It measures the current course points obtained. It evaluates the current student performance in the course.

The following is the GroupProj dataset that was created to be able to define, formulate, and solve a mathematical optimization model:

	Person	GPA	Gender	HoursWorkedProj	CurrentCourseGrade
1	1	3.5	0	5	80
2	2	2.5	1	12	70
3	3	3.0	1	4	65
4	4	4.0	1	10	40
5	5	3.3	0	9	30
6	6	3.1	0	13	50
7	7	3.8	1	8	55
8	8	2.7	1	11	25
9	9	3.2	0	3	75
10	10	3.0	0	9	60
11	11	3.3	0	15	45
12	12	2.9	1	4	75

In order to formulate the integer programming problem, it is essential to describe some assumptions:

- My definition of success is based on the **current points obtained** by each person on the course. It is specified in the "CurrentCourseGrade" variable and has a scale of 0-100. I believe that the "CurrentCourseGrade" variable is a factor that contributes to my definition of success because it is related to the "HourWorkedProj" variable and the "GPA" variable. Based on HourWorkedProj, if a student is performing not too well, the student should feel pressure to perform very well on the final project. If not, he or she could reprove the course. Regarding the GPA, the "CurrentCourseGrade" is related to it, in the sense that if a student has a very low GPA, it is likely that she or he is not interested to perform in the overall program. If the GPA is high, it infers that the student likes to try the best to succeed. On the other hand, gender is independent in the sense that gender does not influence if a group will succeed or not.
- The data of the GroupProj dataset shown above was randomly generated. For this reason, it could be the scenario that some students have high GPA and low CurrentCourseGrade performance, or vise versa. In this case, we will assume that the student was performing very well in the program, but in the current course, the student had a personal situation that is affecting his performance.
- I assumed that GPA needs to be 11 for group 1, 10 for group 2, 9 for group 3, 8 for group 4. Additionally, the hours worked on the project must we 40 for group 1, 35 for group 2, 30 for group 3, 25 for group 4.

II. Formulate the Problem

Let:

$x_{ij} = 1$ if person j belong to group i, and 0 if not.

- j refers to person = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
- i means group = 1, 2, 3, 4

Objective function:

$$\begin{aligned} Max\ Z = 80\sum_{i=1}^{4} x_{i1} + 70\sum_{i=1}^{4} x_{i2} + 65\sum_{i=1}^{4} x_{i3} + 40\sum_{i=1}^{4} x_{i4} + 30\sum_{i=1}^{4} x_{i5} + 50\sum_{i=1}^{4} x_{i6} \\ + 55\sum_{i=1}^{4} x_{i7} + 25\sum_{i=1}^{4} x_{i8} + 75\sum_{i=1}^{4} x_{i9} + 60\sum_{i=1}^{4} x_{i10} + 45\sum_{i=1}^{4} x_{i11} + 75\sum_{i=1}^{4} x_{i12} \end{aligned}$$

S.T:

Groups

$$\sum_{j=1}^{12} x_{ij} = 3, where\ i = 1, 2, 3, 4\ and\ j = 1, 2, 3, 4, 5, ..., 12$$

GPA

$$\sum_{j=1}^{12} GPA_j x_{ij} \geq 12 - i, where\ i = 1, 2, 3, 4\ and\ j = 1, 2, 3, 4, 5, ..., 12$$

Gender

$$\sum_{j=1}^{12} G_j x_{ij} \geq 1, where\ G = gender,\ i = 1, 2, 3, 4\ and\ j = 1, 2, 3, 4, 5, ..., 12$$

Hours Worked on the Project

$$\sum_{j=1}^{12} H_j x_{ij} \geq 45 - 5i, where\ H = hours,\ i = 1, 2, 3, 4\ and\ j = 1, 2, 3, 4, 5, ..., 12$$

Person must be part on one and only one group

$$\sum_{i=1}^{4} x_{ij} = 1, where\ i = 1, 2, 3, 4\ and\ j = 1, 2, 3, 4, 5, ..., 12$$

Integer constrain

$$x_{ij}\ is\ integer,\ for\ i = 1, 2, 3, 4\ and\ j = 1, 2, 3, 4, 5, ..., 12.$$

III. Solve the Problem

First, we will create the dataset.

```
GroupProj <- data.frame(Person = c(1:12),
                        GPA = c(3.5, 2.5, 3.0, 4.0, 3.3, 3.1, 3.8, 2.7, 3.2, 3.0, 3.3, 2.9),
                        Gender = c(0,1,1,1,0, 0,1,1,0,0,0,1),
                        HoursWorkedProj = c(5, 12, 4, 10, 9, 13, 8, 11, 3, 9, 15, 4),
                        CurrentCourseGrade = c(80, 70, 65, 40, 30, 50, 55, 25, 75, 60, 45, 75))
GroupProj
```

```
   Person GPA Gender HoursWorkedProj CurrentCourseGrade
1       1 3.5      0               5                 80
2       2 2.5      1              12                 70
3       3 3.0      1               4                 65
4       4 4.0      1              10                 40
5       5 3.3      0               9                 30
6       6 3.1      0              13                 50
7       7 3.8      1               8                 55
8       8 2.7      1              11                 25
9       9 3.2      0               3                 75
10     10 3.0      0               9                 60
11     11 3.3      0              15                 45
12     12 2.9      1               4                 75
```

Now, let's write the integer programming problem utilizing R Studio.

```
# Import the lpSolve package.
library(lpSolveAPI)

# Create an lp object named lppoint with 0 constraints and 48 decision variables.
lppoint <- make.lp(0, 48)

# Set the problem as an integer programming
set.type(lppoint, 48, "integer")

# Set the objective function.
set.objfn(lppoint, rep(c(80,70,65,40,30,50,55,25,75,60,45,75),4))

# Here we change the direction to set our problem to maximize.
lp.control(lppoint,sense='max')
```

```
# Add the 12 constraints based on the plant's number and products made on those plants.
add.constraint(lppoint, c(rep(1,12),rep(0,36)), "=", 3)
add.constraint(lppoint, c(rep(0,12),rep(1,12),rep(0,24)), "=", 3)
add.constraint(lppoint, c(rep(0,24),rep(1,12),rep(0,12)), "=", 3)
add.constraint(lppoint, c(rep(0,36),rep(1,12)), "=", 3)

add.constraint(lppoint, rep(GroupProj$GPA,4), ">=", 11)
add.constraint(lppoint, rep(GroupProj$GPA,4), ">=", 10)
add.constraint(lppoint, rep(GroupProj$GPA,4), ">=", 9)
add.constraint(lppoint, rep(GroupProj$GPA,4), ">=", 8)
```

```r
add.constraint(lppoint, rep(GroupProj$Gender,4), ">=", 1)
add.constraint(lppoint, rep(GroupProj$Gender,4), ">=", 1)
add.constraint(lppoint, rep(GroupProj$Gender,4), ">=", 1)
add.constraint(lppoint, rep(GroupProj$Gender,4), ">=", 1)

add.constraint(lppoint, rep(GroupProj$HoursWorkedProj,4), ">=", 40)
add.constraint(lppoint, rep(GroupProj$HoursWorkedProj,4), ">=", 35)
add.constraint(lppoint, rep(GroupProj$HoursWorkedProj,4), ">=", 30)
add.constraint(lppoint, rep(GroupProj$HoursWorkedProj,4), ">=", 25)

add.constraint(lppoint, rep(c(1,rep(0,11)),4), "=", 1)
add.constraint(lppoint, rep(c(rep(0,1),1,rep(0,10)),4), "=", 1)
add.constraint(lppoint, rep(c(rep(0,2),1,rep(0,9)),4), "=", 1)
add.constraint(lppoint, rep(c(rep(0,3),1,rep(0,8)),4), "=", 1)
add.constraint(lppoint, rep(c(rep(0,4),1,rep(0,7)),4), "=", 1)
add.constraint(lppoint, rep(c(rep(0,5),1,rep(0,6)),4), "=", 1)
add.constraint(lppoint, rep(c(rep(0,6),1,rep(0,5)),4), "=", 1)
add.constraint(lppoint, rep(c(rep(0,7),1,rep(0,4)),4), "=", 1)
add.constraint(lppoint, rep(c(rep(0,8),1,rep(0,3)),4), "=", 1)
add.constraint(lppoint, rep(c(rep(0,9),1,rep(0,2)),4), "=", 1)
add.constraint(lppoint, rep(c(rep(0,10),1,rep(0,1)),4), "=", 1)
add.constraint(lppoint, rep(c(rep(0,11),1),4), "=", 1)

# To prove if the model solved
solve(lppoint)
```

```
[1] 0
```

As we can confirm, the solver is giving 0, which means it is finding a solution.

```r
# It will give us the maximun success for the optimal solution.
get.objective(lppoint)
```

```
[1] 670
```

Here, we are maximizing the chance of success for all groups on a class project. As we can see, the maximum chance of success is 670, which is the current overall course grade of the class.

```r
# It shows the constraints
get.constraints(lppoint)
```

```
 [1]   3.0   3.0   3.0   3.0  38.3  38.3  38.3  38.3   6.0   6.0   6.0   6.0
[13] 103.0 103.0 103.0 103.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0
[25]   1.0   1.0   1.0   1.0
```

It confirms that all constrains are satisfied.

```r
# It will give us the optimal group number that each student belongs to.
get.variables(lppoint)
```

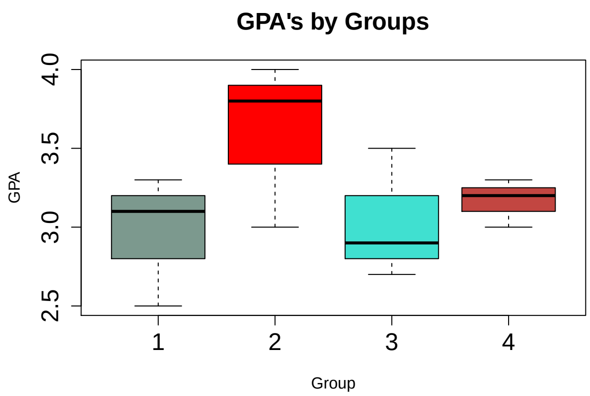
```
 [1] 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0
[39] 1 0 1 0 0 0 1 0 0 0
```

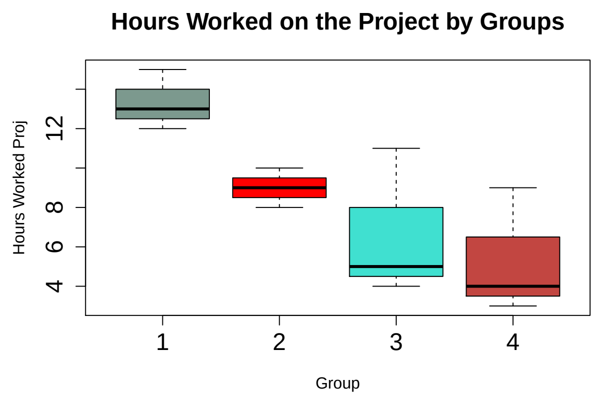
This result provides the optimal mix of group members to be successful in a class project. In other words, of the first 12 entries, it must have only three 1's, and the remaining entries must be 0. It is because the position of the "1" will determine whether the person and the group number she or he belongs to.

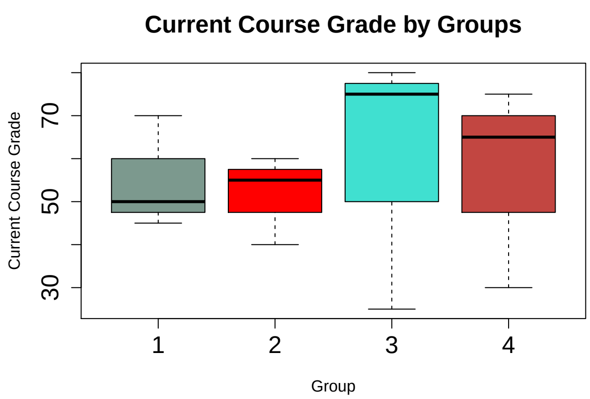
The following output shows the GroupProj dataset with the corresponding group number that each person belongs to.

	Person	GPA	Gender	HoursWorkedProj	CurrentCourseGrade	Group
1	1	3.5	0	5	80	3
2	2	2.5	1	12	70	1
3	3	3.0	1	4	65	4
4	4	4.0	1	10	40	2
5	5	3.3	0	9	30	4
6	6	3.1	0	13	50	1
7	7	3.8	1	8	55	2
8	8	2.7	1	11	25	3
9	9	3.2	0	3	75	4
10	10	3.0	0	9	60	2
11	11	3.3	0	15	45	1
12	12	2.9	1	4	75	3

In order to identify the best group obtained from the integer linear programming, we will plot the data by groups the identify the best group.







After analyzing the previous box plots, we can see the following:

- Group 1 is the group that most hours have spent on the project and group 4 with the lowest. It has a median of 4 hours.

- Group 2 has a higher GPA's. Regarding the hours worked individually on the group project, it is the only group that had 100% of the data concentrated in the small range between 8 to 10.
- Group 3 has a lot of variability on the current course grade factor. It has the minimum and also the maximum, which are the outliers, of the dataset in the same group.

Based on the previous information, we can see that group 2 seems to have the best students in the same group.

IV. Conclusions

Integer linear programming is a mathematical optimization problem that helps to find the optimal solution and restricting the output to be an integer. This model fits very well to the problem explained in this project because the objective function is the maximize the success of assigning people to four different groups.

References

Module 11: Integer programming. IP Additional examples.

Hillier, F. S., & Lieberman, G. J. (2015). Introduction to operations research. 10th Edition. ISBN: 1259614891