Understanding Linear Programming in R

Vicky Crasto

March 14, 2018

Basic concepts in Linear Programming

Definition

Linear programming is a simple technique where we depict complex relationships through linear functions and then find the optimum points. Linear Programming is used for solving optimization (Maximise/Minimise) problem in analytics.

Linear Programming can be of two types

- Integers Linear Programming
- Mixed Integers Linear Programming (Branch & Bound Method, Cutting Plane Method etc.)

Structure of a Linear Programming Model:

A formulation of a linear program in its canonical form of maximum is:

Max $z = c_1x_1 + c_2x_2 + . + c_nx_n$

Subject to:

$a_{11}x_1 + a_{12}x_2 + . + a_{1n}x_n <= b_1$

$a_{21}x_1 + a_{22}x_2 + . + a_{2n}x_n <= b_2$

.

$a_{m1}x_1 + a_{m2}x_2 + . + a_{mn}x_n <= b_m$

$x_i >= 0$

Objective Function

It is defined as the objective of making decisions (Maximise or Minimise)

Decision Variables

It is the variables which decides the output. They are affected by the cost coefficients (cj).

Constraints

It restricts or limits the value of the decision variables. A set of m constraints, in which a linear combination of the variables affected by coefficients aij has to be less or equal than its right-hand side value bi (constraints with signs greater or equal or equalities are also possible)

Non-negative restrictions

Decision variables should have bounds. (Non-negative)

So, Linear programming problem can be alternatively defined as follows:

- It consists of optimising (minimize or maximize) the value of a linear objective function of a vector of decision variables, considering that the variables can only take the values defined by a set of linear constraints.
- Linear programming is a case of mathematical programming, where objective function and constraints are linear.
- Constraints of Linear Programming defines a feasible region. No. of variables determines the shape of feasible region. For a Linear Programming of n variables, the shape of the feasible region would be n dimensional convex polytope. (A convex polytope is a special case of a polytope, having the additional property that it is also a convex set of points in the n-dimensional space Rn.

Steps of Problem Formulation

- Identifying the decision variables
- Writing the objective function
- Writing the constraints
- Writing the non-negativity restrictions

Duality in Linear Programming:

It states that every LPP has another LPP related to it and so can be derived from it. Original LPP is called "Primal" & the derived LPP is called "Dual."

Let's consider a MAX LPP in its canonical form:

Primal

MAX z = c'x

s.t. Ax <= b

x >= 0

Dual

MIN w = u'b

s.t. u'A >= c'

u >= 0

Important Points

- Each variable of dual is linked with a constraint of primal
- Each constraint of dual is linked with a variable of the primal

Properties of Primal & Dual Relationship:

- Dual of dual is a primal
- If a linear program has a bounded optimum, its primal has also a bounded optimum and both have the same value

Solving LP problem using Simplex Method

Algebric Method - Simplex Method

Step 1: Add Slack Variable or subtract non-negative Surplus Variable to the constraints to convert it into equality constraint.

Let us take a problem and illustrate the steps:

Maximise

$Z = 6X_1 + 5X_2$

Subject to

$X_1 + X_2 <= 5$

$3X_1 + 2X_2 <= 12$

$X_1, X_2 >= 0$

Maximise

$Z = 6X_1 + 5X_2 + 0X_3 + 0X_4$

$X_1 + X_2 + X_3 = 5$ .. (eq 1)

$3X_1 + 2X_2 + X_4 = 12$ .. (eq 2)

$X_1, X_2, X_3, X_4 >= 0$

Step 2: An initial basic feasible solution is derived by substituting 0 for X_1 & X_2 in the eq 1 & eq 2. Also, express the basic feasible variables by non-basic variables

Iteration 1:

$X_3 = 5 - X_1 - X_2$

$X_4 = 12 - 3X_1 - 2X_2$

$Z = 6X_1 + 5X_2$

Step 3: Currently, Z = 0. As both the co-efficient of X_1 & X_2 are positive, we will increase any one of the variables X_1 & X_2 to increase Z. Here, we will increase the variable X_1 as the rate of change of Z is more with variable X_1. We can increase the variable X_1 maximum to 4. (As, increasing it beyond 4 will make the variable X_4 negative)

Iteration 2:

$3X_1 = 12 - 2X_2 - X_4$ .. from (eq 2)

$X_1 = 4 - 2/3\ X_2 - 1/3\ X_4$

$X3 = 1 - 1/3\ X_2 - 1/3\ X_4$ ... from (eq 1)

$Z = 24 + X_2 - 2X_4$

From here, we get another **Basic Feasible Solution:** $X_1 = 4, X_3 = 1, X_2, X_4 = 0$

So, next X_2 will be increased to increase Z. X_2 can be increased upto 3.

Iteration 3:

$1/3\ X_2 = 1 - X_3 + 1/3\ X_4$

$X_2 = 3 - 3X_3 + X_4$

$X_1 = 2 + 2X_3 - X_4$

$Z = 24 + (3 - 3X_3 + X_4) - 2X_4$

$Z = 27 - 3X_3 - X_4$

$X_1 = 2, X_2 = 3$ & $Z = 27$

Tabular Form of Simplex Method:

Maximise

$Z = 6X_1 + 5X_2 + 0X_3 + 0X_4$

$X_1 + X_2 + X_3 = 5$ .. (eq 1)

$3X_1 + 2X_2 + X_4 = 12$ .. (eq 2)

$X_1, X_2, X_3, X_4 >= 0$

Step 1: Co-efficient of the variable in the objective function is written on the top row. Co-efficient of the variables on the constraints are written on the corresponding rows. RHS of the constraints are written the RHS Column of the corresponding Row.

Step 2: C_j - Co-efficient of the variable. Z_j - Sum product of 1st Column & the Corresponding Variable Column. So, $C_j - Z_j$ will be as follows

e.g. 6 - (0*1+0*3) = 6, 5 - (0*1+0*2) = 5

	$X_1 = 6$	$X_2 = 5$	$X_3 = 0$	$X_4 = 0$	RHS	Theta
X_3 (0)	1	1	1	0	5	5
X_4 (0)	3	2	0	1	12	4 (Pivot Row)
$C_j - Z_j$	6	5	0	0	0	0
X_3 (0)	0	1/3	1	-1/3	1	3 (Pivot Row)
X_1 (6)	1	2/3	0	1/3	4	6
$C_j - Z_j$	0	1	0	-2	24	
X_2 (5)	0	1	3	-1	3	
X_1 (6)	1	0	-2	1	2	
$C_j - Z_j$	0	0	-3	-1	27	

Step 3: Select the variable with the highest $C_j - Z_j$ value. Here it is 6, variable X_1.

Now, we will calculate the (Theta) by dividing RHS with the corresponding co-efficient of X_1.

Now, we will select the minimum value of the Theta as the limiting value. Corresponding row will become the Pivot Row & the Co-efficient of X_1 in the Pivot Row will be the Pivot Element (3).

In the next step the Pivot Row will be replaced by X_1.

Step 4: Row operations should be performed in such a way that the variables under consideration is equal to the identity matrix.

For X_3, (X_3 -1/3X_4) & for X_1, $X_4/3$. Again, we calculate the $C_j - Z_j$, and find out the pivot row, pivot element & the limiting value.

Step 5: After calculating the 3rd $C_j - Z_j$, there is no way to maximise the RHS, without violating the non-negativity condition.

So, the final answer will be $X_1 = 3, X_2 = 2$ & $Z = 27$.

Solving linear programming problem in R using lpsolve

A car company produces 2 models, model A and model B. Long-term projections indicate an expected demand of at least 100 model A cars and 80 model B cars each day. Because of limitations on production capacity, no more than 200 model A cars and 170 model B cars can be made daily. To satisfy a shipping contract, a total of at least 200 cars much be shipped each day. If each model A car sold results in a $2000 loss, but each model B car produces a $5000 profit, how many of each type should be made daily to maximize net profits?

Formulating the problem

Let the A and B be the number of cars of the two models to be produced.

Objective function : Maximize (-2000A + 5000B)

Constraints :

Demand Constraint A>= 100 , B >=80

Production Constraint A<= 200 , B<=170

Shipping Constraint A+B >=200

Using lpSolveAPI to solve this problem in R

Install package the IpSolve and IpSovleAPI

```
library(lpSolve)
```

```
## Warning: package 'lpSolve' was built under R version 3.3.2
```

```
library(lpSolveAPI)
```

```
## Warning: package 'lpSolveAPI' was built under R version 3.3.2
```

Creating the model

```
model <- make.lp(ncol=2)

m1 <-lp.control(model,sense="max", verbose="neutral")
```

In the lp.control function, we specify sense="max" indicating that we need to maximize the objective function. verbose = "neutral" is to suppress any error report.

Setting the objective function

```
m2 <- set.objfn(model,obj=c(-2000,5000))
```

Incorporating the constraints

```
# Demand constraint
m3 <- set.bounds(model, lower=c(100,80))

#Production constraint
m4 <- set.bounds(model, upper = c(200,170))
```

The set.bounds is used to fix the lower and upper limits for the variables in the model.

The shipping constraint is added by the add.constraint function

```
m5 <- add.constraint(model, c(1,1),">=",200)
```

Print Model

To help generating the model with the constraints in a readable format we add column and row names.

```
rownames <- c("Shipping constraint")
colnames <- c("A", "B")
dimnames(model) <- list(rownames, colnames)
name.lp(model, "Maximize profit")
print(model)
```

```
## Model name: Maximize profit
##                           A      B
## Maximize              -2000   5000
## Shipping constraint       1      1  >=  200
## Kind                    Std    Std
## Type                   Real   Real
## Upper                   200    170
## Lower                   100     80
```

Solving the model

The solve function help us the find the optimized solution for the model.

```
solve(model)
```

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

The output zero indicates that the model has been resolved and a solution is generated.

** Optimum values of decision variables**

get. variables provide the most optimum values of A and B

```
get.variables(model)
```

```
## [1] 100 170
```

The optimum values of A and B are 100 and 170 respectively.

Maximized Objective

get. objective provides the maximized profit

```
get.objective(model)
```

```
## [1] 650000
```

The maximized profit is $650000

Application of Linear Programming

Shelf Space Optimization

This is practical problem faced by super markets to decided which product must be displayed in the shelves with maximum customer visibility and very high probability of being purchased in the customer even if it is not there in their shopping list.

However, the retailer needs to consider various factor before deciding the products to be displayed

- No of products and the brands to be displayed
- Profit margin of the product
- Position of the product in the shelve
- Promotional offer to be considered

- Inventory cost
- Demand
- Expiration date. All the factor acts constraints for the LP problem with objective to maximize the revenue.

Partner matching in Dating Sites

Online dating sites use linear programming to match opposite sex partners which satisfy the maximum number of preferences specified by the members.

Each member has a preference score that is generated based the response to the questionarie provided at the time of signing up.

An optimal dating equilibrium consist of a pairing of couples such that there is no other allocation of females to males that are feasible. In other words, the best allocation is one where people get the best partner they can and are able to get.

Here the constraint for females is to be matched up with males whose score is relatively larger than their own score. In simple words, it means they have been able to get male who surpasses their expectation. Whereas in case of male it is to minimize the score which would mean the relative attractiveness of the female exceeds their expectation. Promotion of ads on Television channels In this case the objective function is maximize the audience viewership. On the other hands, the constraints could be in the form of budget, maximum number of slot for promotion, time slots etc.

Other areas are

- Airlines Revenue Management
- Allocation of budget to an advertisement campaign on different medium.
- Transportation problem
- Call centre staffing and shift management.

References

https://www.analyticsvidhya.com/blog/2016/09/a-beginners-guide-to-shelf-space-optimization-using-linear-programming/

https://masterofeconomics.org/2009/08/15/simplex-algorithm-and-how-dating-web-sites-match-singles/

https://businessjargons.com/duality-in-linear-programming.html

https://businessjargons.com/linear-programming.html

https://www.youtube.com/watch?v=a2QgdDk4Xjw&list=PLAD23E7AEFE221F70

https://www.r-bloggers.com/linear-programming-in-r-an-lpsolveapi-example/

https://www.r-bloggers.com/modeling-and-solving-linear-programming-with-r-free-book/

http://lpsolve.sourceforge.net/5.5/R.htm