Stochastic Modeling Mini Project 1

Ji Yihong
yihong_ji@mymail.sutd.edu.sg
Li Tiantian
tiantian_li@mymail.sutd.edu.sg

April 2020

1 Simulation Scheme

Initialization is done by setting every store to order at week 0. Renewal time is defined as the weeks when three stores all order. The set of renewal time is identified by

$$RenewalTimeList = \{ \text{ t: } (q_{t1} > 0) \wedge (q_{t2} > 0) \wedge (q_{t1} > 0)\}$$

Algorithm 1 would return an array of size (1280, 4) where the (i,j) entry represents the order quantity of entity j at week i.

Algorithm 1 General Simulation

N: number of weeks to be simulated
q_{N4}: an array of size N to store the order history of direct customers
q_{Ni}: an array of size N to store the order history of i^{th} retail store
p_i: The ordering probability distribution of store i
procedure SIMULATE N WEEKS(N)
  $q_{N4} \leftarrow (X_1, X_2, ..X_N)$ where $X_i \sim$ Binomial(160, $\frac{1}{4}$)
  for each store $i \in \{1, 2, 3\}$ **do**
    $q_{Ni} \leftarrow$ SIMULATE STORE(N, p_i, $lotsize$)
  return $(q_{N1}, q_{N2}, q_{N3}, q_{N4})^T$ ▷ This output would be referred as $Performance$
procedure SIMULATE STORE(N, p_i, $lotsize$)
  $StoreOrder \leftarrow \vec{0} \in \mathbb{R}^N$ ▷ The i^{th} element indicates order quantity at week i
  $RenewalTime \leftarrow 0$
  while $RenewalTime \leq N$ **do**
    $Increment \leftarrow$ sample a realization from distribution p_i
    $RenewalTime \leftarrow RenewalTime + Increment$
    if $RenewalTime \leq N$ **then**
      $(StoreOrder$ at $RenewalTime) \leftarrow lotsize$
  return $StoreOrder$

Algorithm 2 is designed to identify renewal time by our definition. With this list of renewal time, $(q_{N1}, q_{N2}, q_{N3}, q_{N4})^T$ could be partitioned according to renewal time. Note that the last interval is discarded because it is usually an incomplete cycle. Name this partitioned $(q_{N1}, q_{N2}, q_{N3}, q_{N4})^T$ as $PartionedPerformance$.

Algorithm 2 Identify Renewal Time

$Performance$: $(q_{N1}, q_{N2}, q_{N3}, q_{N4})^T$ ▷ Output of Algorithm 1 with size (1280, 4)
$RenewalTimeList \leftarrow \varnothing$ ▷ A list to store renewal time
procedure Identify Renewal Time($Performance$)
    for $WeeklyOrder$ in $Performance$ **do**
        if The first three components of $WeeklyOrder$ are all nonzero **then**
            This week is considered as renewal time therefore **do**
            $RenewalTimeList \cup \{Thisweek\}$
    return $RenewalTimeList$

With $PartionedPerformance$, we could iterate through each cycle in it and compute each cycle's reward function and cycle length. The reward function for each question is tabulated under the column of Performance Measure in the output table.

Algorithm 3 Compute Steady State Statistics

$PartionedPerformance$: $Perforamce$ partioned by $RenewalTimeList$
$RewardList \leftarrow \varnothing$ ▷ A list to store reward for each cycle
$CycleList \leftarrow \varnothing$ ▷ A list to store length of cycle for each cycle
f: The performance measure function e.g. $f(\mathbf{q}) = q_1 + q_2 + q_3 + q_4$
N: number of weeks to be simulated
procedure compute steady state($PartionedPerformance$, f, N)
    for $PerformancePerCycle$ in $PartionedPerformance$ **do**
        $TotalReward \leftarrow 0$
        for $WeeklyPerformance$ in $PerformancePerCycle$ **do**
            $reward \leftarrow f(WeeklyPerformance)$ ▷ f is defined in the output table
            $TotalReward \leftarrow TotalReward + reward$
        $RewardList \cup \{TotalReward\}$
        $CycleList$ ∪ {Length of this cycle}
    $C.I. \leftarrow ConstructConfidenceInterval(RewardList, CycleList, N)$
    $PointEstimate \leftarrow$ Mean of $RenewalList$/ Mean of $CycleList$
    return $PointEstimate$, $C.I.$

While iterating through each cycle, reward and cycle length are stored separately into two lists, and these two lists are then passed to the function $ConstructingConfidenceInterval$ to compute C.I. using formulas from slide.

Algorithm 4 Constructing Confidence Interval

R: $RewardList$ invoked in line 13 of **Algorithm 3**
C: $CycleList$ invoked in line 13 of **Algorithm 3**
N: number of weeks to be simulated
procedure COMPUTE VARIANCES(R,C,N)
    $s_{11}^2 \leftarrow \frac{1}{N-1}\sum_{n=1}^{N}(R_i - \bar{R})^2$
    $s_{22}^2 \leftarrow \frac{1}{N-1}\sum_{n=1}^{N}(C_i - \bar{C})^2$
    $s_{12}^2 \leftarrow \frac{1}{N-1}\sum_{n=1}^{N}(C_i - \bar{C})(R_i - \bar{R})$
    return $s_{11}^2, s_{22}^2, s_{12}^2$
procedure CONSTRUCT CONFIDENCE INTERVAL(s_{11}^2,s_{22}^2,s_{12}^2,N)
    $s^2 \leftarrow s_{11}^2 - 2\frac{\bar{R}}{\bar{C}}s_{12}^2 + (\frac{\bar{R}}{\bar{C}})^2 s_{22}^2$
    return $(\frac{\bar{R}}{\bar{C}} - \frac{s z_{\alpha/2}}{\bar{C}\sqrt{N}}, \frac{\bar{R}}{\bar{C}} + \frac{s z_{\alpha/2}}{\bar{C}\sqrt{N}})$

2 Output

Table 1: Results

Part	Performance Measure	Point Estimate	Confidence Interval
a)	$f_1(q) = g(q) + q_4$	40.77	[40.38, 41.16]
b)	$f_2(q) = q_1 + q_2 + q_3 + q_4$	70.14	[69.39, 70.89]
c)	$f_3(q) = \begin{cases} 1 & q_1 + q_2 + q_3 + q_4 > 75 \\ 0 & otherwise \end{cases}$	0.44	[0.42, 0.45]
d)	$f_4(q) = \begin{cases} 1 & q_1 = q_2 = q_3 = 0 \\ 0 & otherwise \end{cases}$	0.42	[0.4, 0.44]
e)	$f_5(q) = 5 + 0.1(g(q) + q_4) + 0.1(q_1 + q_2 + q_3 + q_4) + 100(q_1 + q_2 + q_3 + q_4)^{1/4}$	300.28	[299.28, 301.29]

3 Code

See code in my repository.