Lecture 4: Monte Carlo Simulation- tavern queues

Daniel Chan

UNSW

Semester 1 2014

Scenario

Business has been bad at The Prancing Pony, ever since the Orcs moved in!

Scenario

Business has been bad at The Prancing Pony, ever since the Orcs moved in!

Barliman Butterbur is thinking of reducing the number of bar tenders.

Question

Is this a good idea?

Scenario

Business has been bad at The Prancing Pony, ever since the Orcs moved in!

Barliman Butterbur is thinking of reducing the number of bar tenders.

Question

Is this a good idea?

One non-mathematical possibility is to try this out for a couple of weeks and see if the queues get too long.

Scenario

Business has been bad at The Prancing Pony, ever since the Orcs moved in!

Barliman Butterbur is thinking of reducing the number of bar tenders.

Question

Is this a good idea?

One non-mathematical possibility is to try this out for a couple of weeks and see if the queues get too long.

Are there any problems with this? DISCUSS

Monte Carlo Simulation

Another possibility is to simulate (by computer or hand if need be) what would happen if you reduced the number of bar tenders. This type of approach is sometimes called *Monte Carlo simulation*.

Monte Carlo Simulation

Another possibility is to simulate (by computer or hand if need be) what would happen if you reduced the number of bar tenders. This type of approach is sometimes called *Monte Carlo simulation*.

For discussion

What type of data should you collect to allow you to approximately simulate queues at the bar?

Monte Carlo Simulation

Another possibility is to simulate (by computer or hand if need be) what would happen if you reduced the number of bar tenders. This type of approach is sometimes called *Monte Carlo simulation*.

For discussion

What type of data should you collect to allow you to approximately simulate queues at the bar?

Let's suppose we monitor a couple of nights at the bar. To simplify things, we assume that only ale is served. We record the following:

- The time it takes to serve a hobbit.
- Each minute, the number of hobbits that go up to the bar.

Monte Carlo Simulation

Another possibility is to simulate (by computer or hand if need be) what would happen if you reduced the number of bar tenders. This type of approach is sometimes called *Monte Carlo simulation*.

For discussion

What type of data should you collect to allow you to approximately simulate queues at the bar?

Let's suppose we monitor a couple of nights at the bar. To simplify things, we assume that only ale is served. We record the following:

- The time it takes to serve a hobbit.
- Each minute, the number of hobbits that go up to the bar.

Suppose the time it takes to serve is fairly constant at 1 minute.

Bar data

Suppose over a 30 minute period the number of hobbits going up to the bar each minute is
0,1,1,2,1,0,0,2,3,1,0,1,3,0,1,2,3,2,1,1,0,3,1,2,2,2,1,0,2,1

Bar data

Suppose over a 30 minute period the number of hobbits going up to the bar each minute is
0,1,1,2,1,0,0,2,3,1,0,1,3,0,1,2,3,2,1,1,0,3,1,2,2,2,1,0,2,1

It's more useful to tabulate this information as follows:

No. hobbits n	0	1	2	3
frequency f_n	7	11	8	4
Probability $p_n = \frac{f}{30}$	0.23	0.37	0.27	0.13
Cumulative probability $p_1 + \ldots + p_n$	.23	.60	.87	1

Bar data

Suppose over a 30 minute period the number of hobbits going up to the bar each minute is
0,1,1,2,1,0,0,2,3,1,0,1,3,0,1,2,3,2,1,1,0,3,1,2,2,2,1,0,2,1

It's more useful to tabulate this information as follows:

No. hobbits n	0	1	2	3
frequency f_n	7	11	8	4
Probability $p_n = \frac{f}{30}$	0.23	0.37	0.27	0.13
Cumulative probability $p_1 + \ldots + p_n$	.23	.60	.87	1

We wish to simulate hobbits queuing up, that is, a sequence of 0,1,2,3s etc. which is statistically similar to what one expects.

Simulating hobbits lining up

Suppose that we look over a 30 minute period. We wish to generate the sequence $n_1, \ldots, n_{30}$ of hobbits going up to the bar each minute.

Simulating hobbits lining up

Suppose that we look over a 30 minute period. We wish to generate the sequence $n_1, \ldots, n_{30}$ of hobbits going up to the bar each minute.

We first generate a sequence $r_1, \ldots, r_{30}$ of 30 random numbers between 0 and 100 using the computer, a table, rolling dice etc.

Simulating hobbits lining up

Suppose that we look over a 30 minute period. We wish to generate the sequence $n_1, \ldots, n_{30}$ of hobbits going up to the bar each minute.

We first generate a sequence $r_1, \ldots, r_{30}$ of 30 random numbers between 0 and 100 using the computer, a table, rolling dice etc.

11,96,0,77,82,87,8,40,26,80,43,91,18,26,15,
14,87,58,55,14,85,62,35,51,40,8,24,12,18,24

Simulating hobbits lining up

Suppose that we look over a 30 minute period. We wish to generate the sequence $n_1, \ldots, n_{30}$ of hobbits going up to the bar each minute.

We first generate a sequence $r_1, \ldots, r_{30}$ of 30 random numbers between 0 and 100 using the computer, a table, rolling dice etc.

11,96,0,77,82,87,8,40,26,80,43,91,18,26,15,
14,87,58,55,14,85,62,35,51,40,8,24,12,18,24

We define n_k by

$$n_k = \begin{cases} 0 & \text{if } 0 \leq r_k < 23 \\ 1 & \text{if } 23 \leq r_k < 60 \\ 2 & \text{if } 60 \leq r_k < 87 \\ 3 & \text{if } 87 \leq r_k < 100 \end{cases}$$

Simulating hobbits lining up

Suppose that we look over a 30 minute period. We wish to generate the sequence $n_1, \ldots, n_{30}$ of hobbits going up to the bar each minute.

We first generate a sequence $r_1, \ldots, r_{30}$ of 30 random numbers between 0 and 100 using the computer, a table, rolling dice etc.

11,96,0,77,82,87,8,40,26,80,43,91,18,26,15,
14,87,58,55,14,85,62,35,51,40,8,24,12,18,24

We define n_k by

$$n_k = \begin{cases} 0 & \text{if } 0 \leq r_k < 23 \\ 1 & \text{if } 23 \leq r_k < 60 \\ 2 & \text{if } 60 \leq r_k < 87 \\ 3 & \text{if } 87 \leq r_k < 100 \end{cases}$$

We obtain in this particular case
0,3,0,2,2,3,0,1,1,2,1,3,0,1,0,0,3,1,1,0,2,2,1,1,1,0,1,0,0,1

Simulating queue behaviour

Let's suppose for starters that we wish to reduce to one bar tender so there is a single queue of length q_k during minute k. Then we can determine q_k using the following

Algorithm

Start with $q_0 = 0$ and suppose inductively that q_k has been determined.

1. If $q_k = 0$ then $q_{k+1} = n_{k+1}$.
2. If $q_k > 0$ then $q_{k+1} = q_k + n_{k+1} - 1$.

Simulating queue behaviour

Let's suppose for starters that we wish to reduce to one bar tender so there is a single queue of length q_k during minute k. Then we can determine q_k using the following

Algorithm

Start with $q_0 = 0$ and suppose inductively that q_k has been determined.

1. If $q_k = 0$ then $q_{k+1} = n_{k+1}$.
2. If $q_k > 0$ then $q_{k+1} = q_k + n_{k+1} - 1$.

Applying this to our sequence gives
0,3,2,3,4,6,5,5,5,6,6,8,7,7,6,5,7,7,7,6,7,8,8,8,8,7,7,6,5,5

Simulating queue behaviour

Let's suppose for starters that we wish to reduce to one bar tender so there is a single queue of length q_k during minute k. Then we can determine q_k using the following

Algorithm

Start with $q_0 = 0$ and suppose inductively that q_k has been determined.

1. If $q_k = 0$ then $q_{k+1} = n_{k+1}$.
2. If $q_k > 0$ then $q_{k+1} = q_k + n_{k+1} - 1$.

Applying this to our sequence gives
0,3,2,3,4,6,5,5,5,6,6,8,7,7,6,5,7,7,7,6,7,8,8,8,8,7,7,6,5,5

Some relevant observations In this case the maximum queue length we observe is 8 and the average is 5.8.

Simulating queue behaviour

Let's suppose for starters that we wish to reduce to one bar tender so there is a single queue of length q_k during minute k. Then we can determine q_k using the following

Algorithm

Start with $q_0 = 0$ and suppose inductively that q_k has been determined.

1. If $q_k = 0$ then $q_{k+1} = n_{k+1}$.
2. If $q_k > 0$ then $q_{k+1} = q_k + n_{k+1} - 1$.

Applying this to our sequence gives
0,3,2,3,4,6,5,5,5,6,6,8,7,7,6,5,7,7,7,6,7,8,8,8,8,7,7,6,5,5

Some relevant observations In this case the maximum queue length we observe is 8 and the average is 5.8.

Repeat We can repeat this procedure as often as we wish to get a feel of what happens in the case of one bar tender.

What if there are two bar tenders?

Let the queue lengths be q_k and q'_k. We need to modify the algorithm. Let's assume that hobbits will join the shortest queue (DISCUSS).

What if there are two bar tenders?

Let the queue lengths be q_k and q'_k. We need to modify the algorithm. Let's assume that hobbits will join the shortest queue (DISCUSS).

New algorithm

1. First reduce q_k, q'_k by one unless they are already 0.
2. If n_{k+1} is the even number $2m$, then add m to each of q_k, q'_k to obtain q_{k+1}, q'_{k+1}
3. If n_{k+1} is the odd number $2m+1$, then add $m+1$ to the smaller of q_k, q'_k and m to the other.

What if there are two bar tenders?

Let the queue lengths be q_k and q'_k. We need to modify the algorithm. Let's assume that hobbits will join the shortest queue (DISCUSS).

New algorithm

1. First reduce q_k, q'_k by one unless they are already 0.
2. If n_{k+1} is the even number $2m$, then add m to each of q_k, q'_k to obtain q_{k+1}, q'_{k+1}
3. If n_{k+1} is the odd number $2m+1$, then add $m+1$ to the smaller of q_k, q'_k and m to the other.

The queue lengths are

q_k 0,2,1,1,1,2,1,1,1,1,1,2,1,1,0,0,2,1,1,0,1,1,1,1,1,0,1,0,0,1

q'_k 0,1,0,1,1,1,0,0,0,1,0,1,0,0,0,0,1,1,0,0,1,1,0,0,0,0,0,0,0,0

What if there are two bar tenders?

Let the queue lengths be q_k and q'_k. We need to modify the algorithm. Let's assume that hobbits will join the shortest queue (DISCUSS).

New algorithm

1. First reduce q_k, q'_k by one unless they are already 0.
2. If n_{k+1} is the even number $2m$, then add m to each of q_k, q'_k to obtain q_{k+1}, q'_{k+1}
3. If n_{k+1} is the odd number $2m+1$, then add $m+1$ to the smaller of q_k, q'_k and m to the other.

The queue lengths are

q_k 0,2,1,1,1,2,1,1,1,1,1,2,1,1,0,0,2,1,1,0,1,1,1,1,1,0,1,0,0,1

q'_k 0,1,0,1,1,1,0,0,0,1,0,1,0,0,0,0,1,1,0,0,1,1,0,0,0,0,0,0,0,0

For discussion Is this the type of result you expect?

For more realistic models

Q What variations should you introduce to make the model more realistic?

For more realistic models

Q What variations should you introduce to make the model more realistic?

Here are some possibilities:

For more realistic models

Q What variations should you introduce to make the model more realistic?

Here are some possibilities:

- Lengthen the time period to the actual bar hours.

For more realistic models

Q What variations should you introduce to make the model more realistic?

Here are some possibilities:

- Lengthen the time period to the actual bar hours.
- There will be busy times and quiet times. Collect statistics for various times.

For more realistic models

Q What variations should you introduce to make the model more realistic?

Here are some possibilities:

- Lengthen the time period to the actual bar hours.
- There will be busy times and quiet times. Collect statistics for various times.
- Perhaps cocktails which take a bit longer will also be served.

On generating random numbers

There's quite an art to generating random numbers. The following middle-square method is due to John von Neumann.

On generating random numbers

There's quite an art to generating random numbers. The following middle-square method is due to John von Neumann.

Algorithm

Start with any 4 digit number as the seed for the algorithm and the first random number.

1. Square the number to obtain an 8 digit number.
2. Take the middle four digits as your next random number.
3. Repeat as needed.

On generating random numbers

There's quite an art to generating random numbers. The following middle-square method is due to John von Neumann.

Algorithm

Start with any 4 digit number as the seed for the algorithm and the first random number.

1. Square the number to obtain an 8 digit number.
2. Take the middle four digits as your next random number.
3. Repeat as needed.

E.g. If the seed is 4877 then

On generating random numbers

There's quite an art to generating random numbers. The following middle-square method is due to John von Neumann.

Algorithm

Start with any 4 digit number as the seed for the algorithm and the first random number.

1. Square the number to obtain an 8 digit number.
2. Take the middle four digits as your next random number.
3. Repeat as needed.

E.g. If the seed is 4877 then
$4877^2 = 23785129 \implies$ next random number is 7851.

On generating random numbers

There's quite an art to generating random numbers. The following middle-square method is due to John von Neumann.

Algorithm

Start with any 4 digit number as the seed for the algorithm and the first random number.

1. Square the number to obtain an 8 digit number.
2. Take the middle four digits as your next random number.
3. Repeat as needed.

E.g. If the seed is 4877 then
$4877^2 = 23785129 \Longrightarrow$ next random number is 7851.
$7851^2 = 61638201 \Longrightarrow$ next random number is 6382.

An historical note

The Monte Carlo method was invented by Stanislaw Ulam in connection with the Manhattan project. It was really about using probabilistic methods to study deterministic problems.

An historical note

The Monte Carlo method was invented by Stanislaw Ulam in connection with the Manhattan project. It was really about using probabilistic methods to study deterministic problems.

One example that is now used extensively is to perform numerical integration.