Deep Dive

LBA: Foot Traffic Modeling

CS113

William Nguyen

Vanya Sofia Villa Soto

Professor Tambasco

April 7th, 2022

Location Based Assignment: Foot Traffic Modeling

Description of the Network

The network we are analyzing is a recreational park with a big lake called Engelbecken (Figure 1). People at weekends often go for a walk, to take pictures and to enjoy the cafe. In our network, we have a big lake in the middle. To the north of the lake, we have a cool cafe where visitors can just chill by the lake, have a sip of coffee and observe swans interacting with pigeons. We have two stairs on both sides of the cafe serving as the entry and exit. These stairs enable reachability between the church, the cafe, and the fountain. When we go up the stairs, we see the church which is a big historical building where people often walk around and take photos. From the cafe, people can also walk around the lake and end up at the fountain to take pictures or go to the bridge. In reality, the bridge is not directly connected to the cafe because people need to go through the fountain (Figure 2).

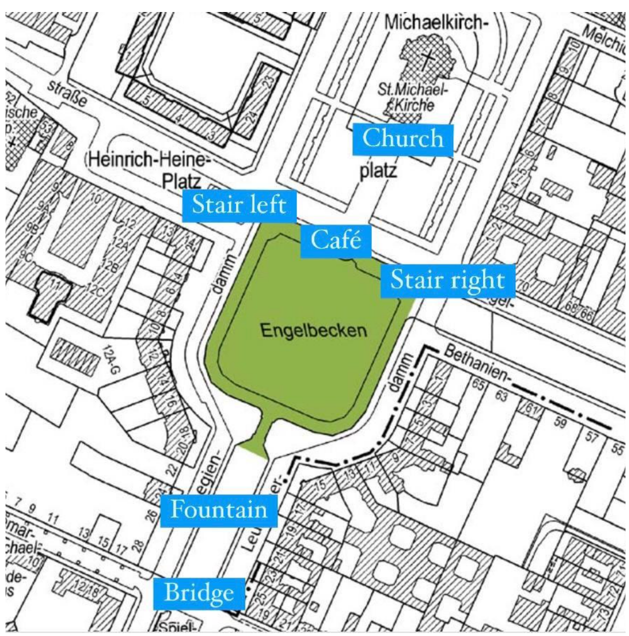

Figure 1. Map of Engelbecken park with sites (Stadtentwicklung, n.d.).

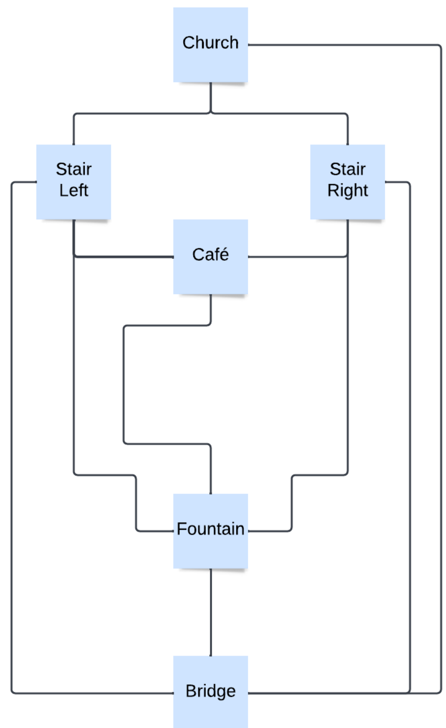

Figure 2. Site map for Engelbecken park in Berlin. Created with LucidChart.

Data Collection procedure including estimation process and assumptions

To collect data, we went to the Engelbecken park together and observed the dynamics of people traffic in multiple 15-minute intervals. Sofia stood in the north side of the network keeping track of people moving around the church, stairs, and the cafe. Meanwhile, William mainly kept an eye on the foot traffic in the south including the fountain and bridge. We collected data about the number of people staying in each node and transiting between nodes in a 15-minute interval. We selected 15 minutes because, except the cafe, most people just walk around the park and don't stay at one place. This timeframe is appropriate to account for enough traffic, while considering the small size of the network. We replicated our observation a few times to observe patterns, then created a distribution based on those observations. We later improved our model considering potential randomness. When we arrived, there were about 25 people in the park, with the following distribution:

Church	Left Stairs	Right Stairs	Cafe	Fountain	Bridge
5	4	4	8	2	2

Table 1. The Table records data at the beginning of our observation

We believe that one-interval data is not representative of the traffic dynamics in the site. Therefore, we replicated our observations for 5 intervals, identified patterns and got averages of transitions between nodes, thus achieving more accurate probabilities. The appendix shows the data, our calculations, and more detailed explanations. Table 2 models our Markov matrix which is an estimation for people traffic. However, it cannot provide an exact demonstration as it is based on only a few replications of our data collection and assumptions.

	Church	Stair left	Stair right	Cafe	Fountain	Bridge
Church	1/10	3/10	3/10	0	0	1/4
Stair left	2/5	0	0	1/5	3/16	1/8
Stair right	2/5	0	0	1/5	3/16	1/8
Cafe	0	2/5	2/5	1/2	1/8	0
Fountain	0	1/5	1/5	1/10	0	1/2
Bridge	1/10	1/10	1/10	0	1/2	0

Table 2. Distribution of people in the park. The columns of the table represent the current state and the rows represent the state after a 15-minute interval.

For this model, we assumed that the number of people in the system would stay constant (25 people), and that people will only relocate once in the 15-minute interval. In other words, we assume that it is impossible to visit multiple nodes within one interval.

Analysis of the Model

Explore Stationary Distribution

To explore the long-term behavior, we left-multiplied powers of M with different initial distributions. The power of M represents the number of intervals that have happened since the original distribution. As the power of M gets higher, we get closer to our stationary distribution.

We can also compute the eigenvector with the corresponding eigenvalue of 1. This eigenvector also represents the analytical stationary distribution. Note that in both strategies, we have to normalize the final vector to represent the true stationary distribution (all sums to 1). Table 3 presents our results about the stationary distribution from the model. The calculations and further explanations are found in the Appendix.

Site	Stationary Distribution	Expected Population (25 people)
Church	0.1362741479020549	3(3.4)
Stair left	0.15457733480989294	4 (3.86)
Stair right	0.15457733480989294	4 (3.86)
Cafe	0.2848529592715639	7 (7.12)
Fountain	0.1501168943029408	4 (3.75)
Bridge	0.11960132890365449	3 (2.99)

Table 3. Stationary Distribution and Expected Population in each node in the long-term.

Analyze the intuition behind Stationary Distribution

As we can observe, we expect more people to be in the cafe at all times, which makes sense because cafe visitors tend to stay there for longer periods than other node visitors. The second most populated nodes are the stairs. We assumed an equal distribution of people between stairs. It seems reasonable that we expect the stairs to be populated because these nodes serve as bridges between others.

The third highest populated is the fountain because people have to go through it to go to the bridge and some people wandering for a walk might stop to take pictures. The church is also less crowded because it is to the north and isolated from most other nodes. The least visited is the bridge because most of the people reaching there come from the fountain, which doesn't tend to be very populated (From Table 2).

However, the difference between the percentage of people in each location is very low. One of the reasons is that the system itself usually has a small population (around 20 people). But a more interesting reason is that the system doesn’t have a centralized network structure and node reachability is pretty dense.

Limitations of the Model

There are many limitations of our model. We made the unrealistic assumption that our network is a roughly closed system. In other words, there will always be 25 people. It is unrealistic to assume no external changes besides people traffic here because there are many confounding variables which possibly influence people traffic, e.g. daytime, weather conditions or park policies.

Possible Improvements

We could improve our model by experimenting with different initial distributions and sample sizes to verify our stationary distribution. Another alternative to account for unpredictability in our system, is to include some randomness to our model (The new Markov matrix can be found in the Appendix). When we add randomness, the difference between nodes is less striking compared to the deterministic model. However, randomness reflects reality better.

Appendix

Matrix A: Average of people in each location based on the observations.

		Church	Stair left	Stair right	Cafe	Fountain	Bridge
	Church	0.5	1.2	1.2	0	0	0.5
	Stair left	2	0	0	1.6	0.375	0.25
A =	Stair right	2	0	0	1.6	0.375	0.25
	Cafe	0	1.6	1.6	4	0.25	0
	Fountain	0	0.8	0.8	0.8	0	1
	Bridge	0.5	0.4	0.4	0	1	0

The entry on the j-th column and i-th row (a_{ij}) represents the number of people that left location j to go to location i. The diagonal entries (a_{ii}) represent the number of people that stayed at the same location.

Note: this matrix represents an average from the multiple observation replications that we did, which is why we observe percentages of people.

Matrix M: Distribution of people in the park

```
church = vector([1/10, 2/5, 2/5, 0, 0, 1/10])
stairleft = vector([3/10, 0, 0, 2/5, 1/5, 1/10])
stairright = vector([3/10, 0, 0, 2/5, 1/5, 1/10])
cafe = vector([0, 1/5, 1/5, 1/2, 1/10, 0])
fountain = vector([0, 3/16, 3/16, 1/8, 0, 1/2])
bridge = vector([1/4, 1/8, 1/8, 0, 1/2, 0])

# the Markov matrix representing the Deterministic Model
M = matrix([church, stairleft, stairright, cafe, fountain, bridge]).transpose()
M
```

```
[1/10 3/10 3/10    0    0  1/4]
[ 2/5    0    0  1/5 3/16  1/8]
[ 2/5    0    0  1/5 3/16  1/8]
[   0  2/5  2/5  1/2  1/8    0]
[   0  1/5  1/5 1/10    0  1/2]
[1/10 1/10 1/10    0  1/2    0]
```

The matrix M was created by normalizing each of the columns of A. As we can observe, all the columns add up to 1 because they represent probabilities. The matrix represents all the possible distributions, which is why this is a Markov matrix.

Eigenvalues and eigenvectors

We used Sage to calculate the eigenvalues and eigenvectors as shown in the attached document.

```
M.eigenvectors_right()
```

```
[(1,
  [
  (1, 1005/886, 1005/886, 926/443, 488/443, 1944/2215)
  ],
  1),
 (0,
  [
  (0, 1, -1, 0, 0, 0)
  ],
  1),
 (-0.642908473380485?,
  [(1, -0.9698879644049803?, -0.9698879644049803?, 0.5677824167575808?, 1.015896291002365?, -0.6439027789499848?)],
  1),
 (-0.4232370648667022?,
  [(1, -1.844894695933550?, -1.844894695933550?, 1.793379463467059?, -1.438389082373670?, 2.334799010773710?)],
  1),
 (0.2314259720385964?,
  [(1, 0.3699785383762458?, 0.3699785383762458?, -0.862055438284305?, -0.5156570345195827?, -0.3622446039486047?)],
  1),
 (0.4347195662085902?,
  [(1, 0.1403593772986793?, 0.1403593772986793?, -2.897782475444794?, 0.6150479615299042?, 1.002015759317531?)],
  1)]
```

Although we get different eigenvalues and eigenvectors, we will focus on the the eigenvalue $\lambda = 1$, with the following eigenvector: (1, 1005/886, 1005/886, 926/443, 488/443, 1944/2215). This is because this is the only eigenvalue that will give us the stationary distribution.

However, to use it, we need to normalize this vector to length 1 to get the stationary distribution. We do this by dividing each of the entries of the vector by the sum of all the vectors. The results are found in Table 4 and the calculations in the attached document.

```
# (1, 1005/886, 1005/886, 926/443, 488/443, 1944/2215)
# normalize eigenvectors
ch = 1
sl = 1005/886
sr =  1005/886
cf = 926/443
f = 488/443
b =  1944/2215

total = ch + sl + sr + cf + f + b
float(ch / total), float(sl / total), float(sr / total), float(cf / total), float(f / total), float(b / total)
```

```
(0.1362741479020549,
 0.15457733480989294,
 0.15457733480989294,
 0.2848529592715639,
 0.1501168943029408,
 0.11960132890365449)
```

Site	Stationary distribution
Church	0.1362741479020549
Stair left	0.15457733480989294
Stair right	0.15457733480989294
Cafe	0.2848529592715639
Fountain	0.1501168943029408
Bridge	0.11960132890365449

Table 4. Stationary distribution

Testing the Deterministic Model

To test the stochastic model we chose three different **starting distributions**, demonstrated by the following vectors. These have the same number of entries as the number of your nodes and their entries sum to one. The first initial condition is based on our raw data. The second is our assumption that from the starting point, every node is equally populated. In the third initial condition, we are curious if people always start from the church, how would the stationary distribution look like?

1. $(\frac{5}{25}, \frac{4}{25}, \frac{4}{25}, \frac{8}{25}, \frac{2}{25}, \frac{2}{25})$

2. $(\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$

3. (1, 0, 0, 0, 0, 0)

We used Sage to calculate the observed long term behavior ($\lim_{n\to\infty} M^n v$) of each of the initial distributions by repeatedly multiplying them by matrix M. The calculations can be found in the Jupyter notebook.

```
# experiment with three different initial conditions
u0 = vector([0.2, float(4/25), float(4/25), float(8/25), float(2/25), float(2/25)])
u1 = vector([float(1/6), float(1/6), float(1/6), float(1/6), float(1/6)])
u2 = vector([1.0, 0.0, 0.0, 0.0, 0.0, 0.0])
# different powers
M*u0, M**5 * u0, M**10 * u0, M**100 * u0
```

```
((0.136000000000000, 0.169000000000000, 0.169000000000000, 0.298000000000000, 0.136000000000000, 0.0920000000000000
),
 (0.135377775000000, 0.155115343750000, 0.155115343750000, 0.285716437500000, 0.149619450000000, 0.119055650000000)
,
 (0.136308360478914, 0.154534721300816, 0.154534721300816, 0.284899339105469, 0.150150267305852, 0.119572590508133)
,
 (0.136274147902055, 0.154577334809893, 0.154577334809893, 0.284852959271564, 0.150116894302941, 0.119601328903655)
)
```

```
M*u1, M**5 * u1, M**10 * u1, M**100 * u1
```

```
((0.15833333333333333, 0.15208333333333335, 0.15208333333333335, 0.23750000000000002, 0.16666666666666669, 0.133333
33333333333),
 (0.1377905729166666, 0.15379959635416668, 0.15379959635416668, 0.283556484375, 0.15142520833333334, 0.11962854166
666666),
 (0.1361847128599772, 0.154674706698291, 0.154674706698291, 0.28476887335668943, 0.15002197340356443, 0.11967502698
318684),
 (0.1362741479020549, 0.15457733480989294, 0.15457733480989294, 0.2848529592715639, 0.1501168943029408, 0.119601328
90365448))
```

```
M*u2, M**5 * u2, M**10 * u2, M**100 * u2
```

```
((0.100000000000000, 0.400000000000000, 0.400000000000000, 0.000000000000000, 0.000000000000000, 0.100000000000000)
,
 (0.112086250000000, 0.179977187500000, 0.179977187500000, 0.267659375000000, 0.127445000000000, 0.132855000000000)
,
 (0.138926010516797, 0.152001017869922, 0.152001017869922, 0.286350782881641, 0.152771829605078, 0.117949341256641)
,
 (0.136274147902055, 0.154577334809893, 0.154577334809893, 0.284852959271564, 0.150116894302941, 0.119601328903654)
)
```

With a big power of 100, the result comes closer to the stationary distribution. The last result of each of 3 code cells above represents the stationary distribution with 3 different initial conditions. We can see that they all have pretty similar results, which shows that even with different initial conditions, the Markov matrix will still head for the same analytical solution found through eigenvectors.

Even with initial distributions, as the power of M gets higher, we approximate more to the same stationary distribution (given by the eigenvector that we found before).

Model with Randomness

To account for uncertainty in our model, we improved our model by including some randomness. To do this we created a new matrix R that represents the randomness. The matrix shows an equal possibility of going to each of reachable nodes (in the case of the church and the cafe, also considering the possibility that people stay based on our observations.

We considered that ⅕ of the transitions will be due to randomness, so we constructed a new model *M_new* that is given by the following equation.

$$M_{new} = \frac{4}{5}M + \frac{1}{5}R$$

```
# Stochastic Model
church_rd = vector([1/4, 1/4, 1/4, 0, 0, 1/4])
stairleft_rd = vector([1/4, 0, 0, 1/4, 1/4, 1/4])
stairright_rd = vector([1/4, 0, 0, 1/4, 1/4, 1/4])
cafe_rd = vector([0, 1/4, 1/4, 1/4, 1/4, 0])
fountain_rd = vector([0, 1/4, 1/4, 1/4, 0, 1/4])
bridge_rd = vector([1/4, 1/4, 1/4, 0, 1/4, 0])

R = matrix([church_rd, stairleft_rd, stairright_rd, cafe_rd, fountain_rd, bridge_rd]).transpose()
A = (4/5)*M + (1/5)*R
A
# all column vectors sum up to 1 --> satisfied!
```

```
[13/100 29/100 29/100      0      0    1/4]
[37/100      0      0 21/100    1/5   3/20]
[37/100      0      0 21/100    1/5   3/20]
[     0 37/100 37/100   9/20   3/20      0]
[     0 21/100 21/100 13/100      0   9/20]
[13/100 13/100 13/100      0   9/20      0]
```

We followed the same procedure as before to calculate the stationary distribution, as shown in Table 5.

```
A.eigenvectors_right()
```

```
[(1,
  [
  (1, 138959/125754, 138959/125754, 562501/314385, 348676/314385, 1440488/1571925)
  ],
  1),
 (0,
  [
  (0, 1, -1, 0, 0, 0)
  ],
  1),
 (-0.5956926308742601?,
  [(1, -1.029901524302070?, -1.029901524302070?, 0.5874189476270644?, 0.9857830880933138?, -0.5133989871162397?)],
  1),
 (-0.4119798067151492?,
  [(1, -2.360195750851207?, -2.360195750851207?, 2.366143466082402?, -1.953486879494192?, 3.307734915114204?)],
  1),
 (0.2012525230650791?,
  [(1, 0.3101822935906685?, 0.3101822935906685?, -0.5232754811999252?, -0.6624762771113771?, -0.4346128288700346?)]
,
  1),
 (0.3864199145243303?,
  [(1, 0.1442519015865398?, 0.1442519015865398?, -2.200666956351387?, 0.2211479067617586?, 0.6910152464165486?)],
  1)]
```

```
# (1, 138959/125754, 138959/125754, 562501/314385, 348676/314385, 1440488/1571925)

ch = 1
sl = 138959/125754
sr = 138959/125754
cf = 562501/314385
f = 348676/314385
b = 1440488/1571925

total = ch + sl + sr + cf + f + b
# Stationary Distribution
float(ch / total), float(sl / total), float(sr / total), float(cf / total), float(f / total), float(b / total)
```

```
(0.14235520168718885,
 0.15730343743539035,
 0.15730343743539035,
 0.25470344737899525,
 0.15788234904172357,
 0.13045212702131165)
```

Site	Stationary distribution	Expected Population (25 people)
Church	0.14235520168718885	4 (3.56)
Stair left	0.15730343743539035	4 (3.93)
Stair right	0.15730343743539035	4 (3.93)
Cafe	0.25470344737899525	6 (6.37)
Fountain	0.15788234904172357	4 (3.94)
Bridge	0.13045212702131165	3 (3.26)

Table 5. Stationary distribution when including randomness

This is a more accurate model, but we still beware of many other limitations as listed above.

System Evolvement over time

We created diagrams to portray how our system is forecasted to evolve over time. Figure 3 shows the initial distributions that analyzed and the long-term distribution that we expect based on our two model (with and without randomness).

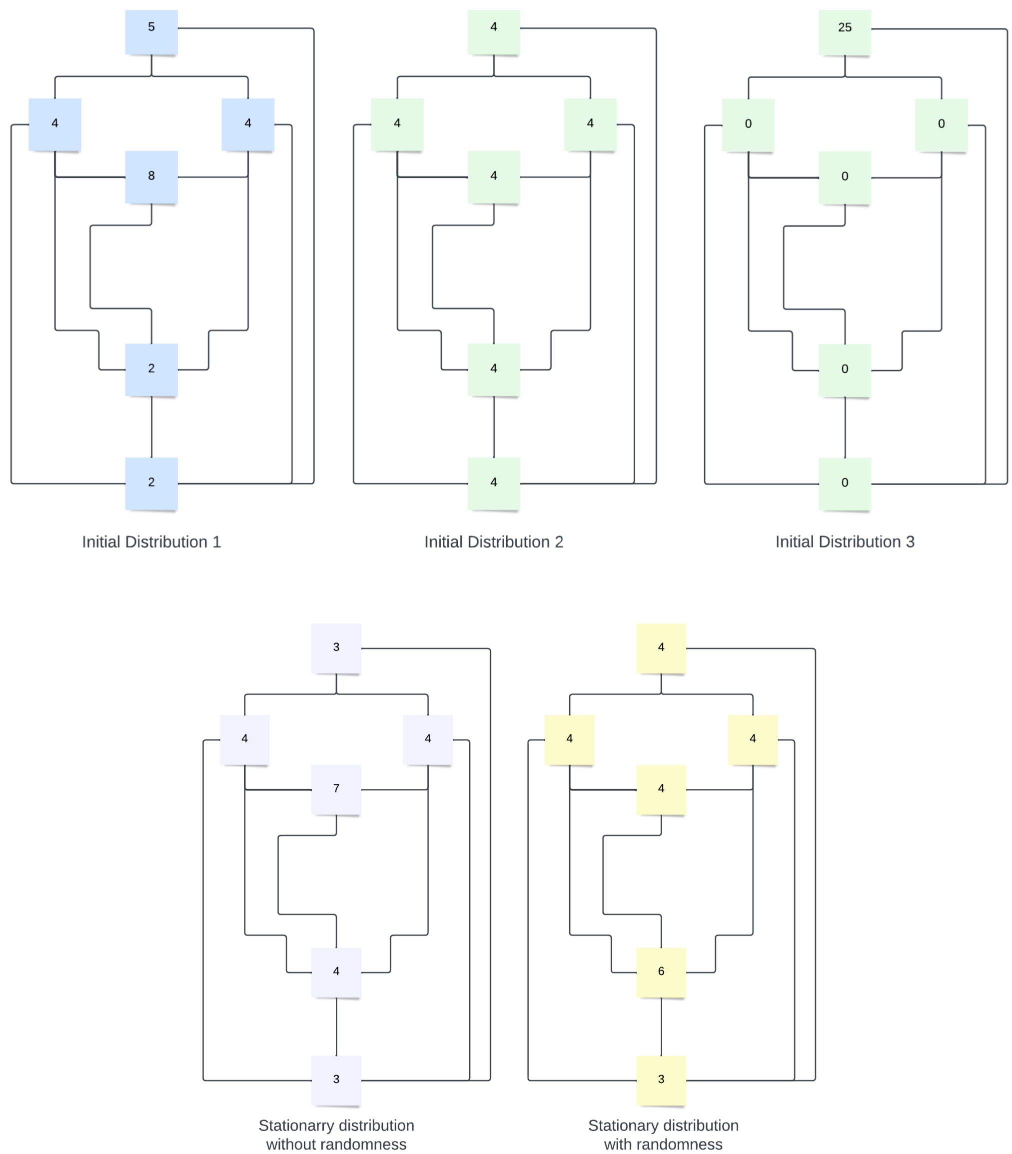

Figure 3. Initial Conditions and Stationary Distributions based on our model

Bibliography

Stadtentwicklung (n.d.). Engelbecken. Retrieved April 3, 2022, from https://www.stadtentwicklung.berlin.de/staedtebau/foerderprogramme/lebendige_zentren/de/gebiete/mit/luisenstadt/engelbecken.shtml