FOOT TRAFFIC MODELING

Deep Dive LBA

CS113: Theory and Applications of Linear Algebra

Minerva University

Duc Nguyen, Ga Eun Lee, Liudmyla Serohina

December 2021

Model Analysis Report

Network description

The network we analyze in this report corresponds to the eastern wing of the Altes Museum in Berlin. There are five main locations in the eastern wing, namely the Court, Cyprus, Glass Gems, Bronze Implements and Pergamon rooms. Each location represents a node in our network with edges between the nodes. Based on the arrangement in the museum, some rooms are not directly connected to one of the others (e.g., the Pergamon room is not directly connected to Cyprus and Glass Gems). However, people can still move between those rooms by passing through other rooms (e.g., getting from the Court to the Pergamon by passing through the Bronze Implements room) because we only count if people stay in a given room for about 15 minutes. On the other hand, by only observing the eastern wing, which is not directly connected to the main entrance, we can minimize the number of new people entering the observed rooms.

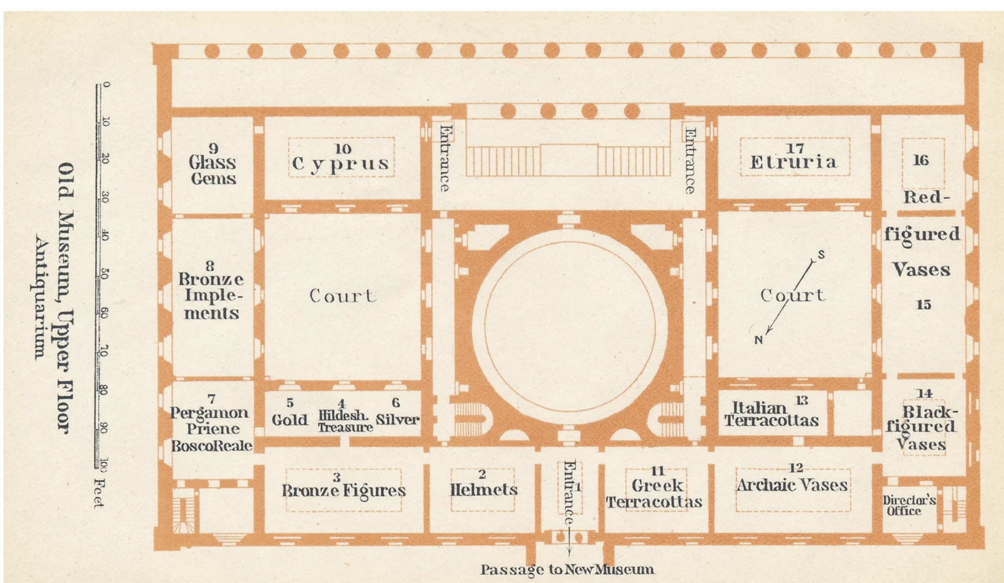
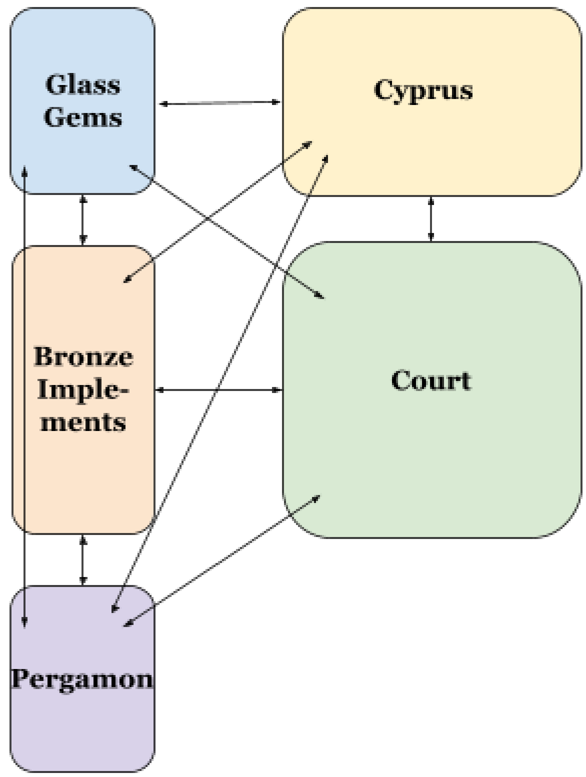


Figure 1: The map of Altes Museum in Berlin.

Network diagram



Data collection procedure

We collected our data at the Altes Museum in Berlin. Because we found the museum map online beforehand, we set the team members' locations to observe people at the same time: (1) Duc stood between Court and Bronze Implements; (2) Liuda stood between Cyprus and Glass Gems; (3) Ga Eun stood between Pergamon and Bronze Implements. We recorded the number of people in each room after every 15-minute period with 4 rounds of observations.

The observation process:

1. Count the total number of people in five rooms at the beginning of observations. This number is the total population that would not change over time.

2. Record the number of people moving to other rooms over time.
3. After every 15 minutes, re-count the number of people in each room and double-check with the number of people that moved to other rooms.
4. After 4 rounds of observations, collect all the data and calculate the average number of people staying and leaving each room. These averages are used for calculating the Markov matrix in this report.

Assumptions:

1. The population is unchanged throughout the process. We mitigate and ignore the new people entering the rooms.
2. The time everyone spends in each room is 15 minutes. We assume that people who left the room before 15 minutes ended count as those who spent the full 15 minutes there.
3. There is no direction guideline or path that people have to follow. Thus, people have the freedom to choose the rooms by themselves.
4. Edges between our nodes are bidirectional since there are ways of getting from one room to another by passing through a room between them (e.g., getting from the Court to the Pergamon by passing through the Bronze Implements room).

The result of data collection

Places	Glass Gems(9)	Cyprus(10)	Court	Bronze Implements(8)	Pergamon(7)
Glass Gems(9)	2	1	2	7	3
Cyprus(10)	2	2	6	2	3
Court	3	3	6	5	7
Bronze Implements(8)	1	3	4	2	3
Pergamon(7)	1	4	3	0	4
Initial Number of people	9	13	21	16	20
Total number of people	79				

Table 1: *This table represents the result of data collection that is used to create the Markov matrix. The rows represent the room people moved from and the columns represent the room people moved to.*

Model analysis

For our Markov model analysis, we first converted the table from the previous section into a Markov matrix by dividing each column by the initial number of people in each room, so that each column sums up to 1 and each element at row *i* and column *j* represents the proportion of people moving from room *i* to room *j*. Then, we assumed three possible initial distributions of population across the five rooms to analyze how these different starting distributions evolve over time, using the same traffic flow model described by the Markov matrix. The three assumed initial distributions were: (1) the original distribution, (2) an approximately equal distribution, (3) distribution in the ratios of about: 3 : 3 : ⅔ : ⅔ : ⅔ (Appendix A).

According to Sage calculations, the Markov matrix resulted in five eigenvectors each with five eigenvalues, but only one of them was a real eigenvector with an eigenvalue of 1. After normalizing the eigenvector with an eigenvalue of 1, we obtained the stationary long-term distribution of our Markov model 0.1801 : 0.2000 : 0.2977 : 0.1670 : 0.1551 (Appendix B). Our calculations for the evolution of the museum's traffic flow distribution also showed that all three initial distributions became similar to this stationary distribution. After multiplying one additional Markov matrix by the initial distribution matrix each time, we obtained the traffic flow distribution across the five rooms after every 15 minutes. They slowly converged to the long-term distribution we obtained. After 60 minutes, the traffic flow distribution almost precisely achieved it(Appendix C). The distributions converged to the same distribution obtained by the eigenvector because regardless of the different initial distributions, our Markov matrix represents the same traffic model that will lead to the same converging static distribution.

We can interpret the stationary distribution that in the long-term or even after 60 minutes, most people (about 30%) will be at the Court and the rest of people will be almost equally distributed across the four rooms (with slight differences) in the order of Cyprus > Glass Gem > Bronze Implements > Pergamon. We would also need to add assumptions that this is a closed system where new people would not enter or leave and that the Markov model stays consistent in the time period. Such assumptions make our Markov model quite unrealistic. Nonetheless, the model may hold in different circumstances such as when a special performance that lasts for more than 60 minutes and regulates entries and exits is held in the East Wing.

Conclusion

When completing the LBA, we got to observe museum attendees' behavior and then translate it to the language of linear algebra. Some of the main takeaways from our experience are:

- The Court tends to gather the most people since it is the central place, from which people can go to other rooms.
- The next most popular room is Cyprus. Its popularity is justified because there are great Ancient Greek, Etruscan, and Roman sculptures exhibited in this room. This is one of the most attractive and well-recognized exhibitions at the Altes Museum in Berlin.
- The third most visited room is the Glass Gem. A newly established exhibition, called "Eye Tunes: The Sound of Music in Ancient Greece" was presented in this room on the day we visited the Altes Museum. This might explain its relative popularity.
- Bronze Implements and Pergamon have approximately the same amount of visitors because they did not have any special features or new exhibitions. Both showcase some artifacts from new artists that followed the Ancient Greek and Roman legacy.

Considering the limited resources at hand, there are possible improvements we could implement to create a more representative model in the future:

- Have more than 3 surveyors to collect more representative data.
- Spend more time at the museum and decrease the intervals between data collection to get more precise observations.

- Attend the museum on different days of the week (e.g., a weekday and a weekend) to compare if the foot traffic differs.
- Increase the number of observed rooms to check if any interesting relationships arise.

Appendix

Appendix A. Markov matrix and matrix representing the three initial distributions

```
#CS113 deep dive Appendix calculations
#markov matrix (order of rows and columns: Glass Gems, Cyprus, Court, Bronze Implements, Pergamon)
markov = matrix([[2/9, 2/9, 3/9, 1/9, 1/9],
                 [1/13, 2/13, 3/13, 3/13, 4/13],
                 [2/21, 6/21, 6/21, 4/21, 3/21],
                 [7/16, 2/16, 5/16, 2/16, 0/16],
                 [3/20, 3/20, 7/20, 3/20, 4/20]]).transpose()

#original initial distribution
initial1 = vector([9/79, 13/79, 21/79, 16/79, 20/79])
#example of a possible initial distribution, almost equally distributed
initial2 = vector([15/79, 16/79, 16/79, 16/79, 16/79])
#example of a possible inital distribution,
#distributed approximately in the ratio 3:3:(2/3):(2/3):(2/3)
initial3 = vector([30/79, 30/79, 7/79, 6/79, 6/79])

#matrix of the above 3 initial distributions
initial=matrix([initial1, initial2, initial3]).transpose()

n(markov, digits=2), n(initial,digits=2)
```

```
(
[ 0.22 0.077 0.095  0.44  0.15]  [ 0.11  0.19  0.38]
[ 0.22  0.15  0.29  0.12  0.15]  [ 0.16  0.20  0.38]
[ 0.33  0.23  0.29  0.31  0.35]  [ 0.27  0.20 0.089]
[ 0.11  0.23  0.19  0.12  0.15]  [ 0.20  0.20 0.076]
[ 0.11  0.31  0.14  0.00  0.20], [ 0.25  0.20 0.076]
)
```

Appendix B. Eigenvectors and eigenvalues of Markov matrix

```
markov.eigenvectors_right()
```

```
[(1,
  [
  (1, 163891/147582, 40663/24597, 68432/73791, 63550/73791)
  ],
  1),
 (-0.03942675012343404? - 0.1688887433890988?*I,
  [(1, -0.4268093714039557? + 0.7885998794547450?*I, 0.5117274542497284? + 0.2159829661735057?*I, -0.3993590185550520? - 0.3458527
027336570?*I, -0.6855590642907207? - 0.6587301428945938?*I)],
  1),
 (-0.03942675012343404? + 0.1688887433890988?*I,
  [(1, -0.4268093714039557? - 0.7885998794547450?*I, 0.5117274542497284? - 0.2159829661735057?*I, -0.3993590185550520? + 0.3458527
027336570?*I, -0.6855590642907207? + 0.6587301428945938?*I)],
  1),
 (0.03281808101476494? - 0.0661655595125261?*I,
  [(1, 0.4024061710637296? + 0.02726619018327642?*I, -0.4553487899208907? - 0.2399020060066399?*I, -0.1215071424906100? - 0.268906
1601218347?*I, -0.8255502386522289? + 0.4815419759451981?*I)],
  1),
 (0.03281808101476494? + 0.0661655595125261?*I,
  [(1, 0.4024061710637296? - 0.02726619018327642?*I, -0.4553487899208907? + 0.2399020060066399?*I, -0.1215071424906100? + 0.268906
1601218347?*I, -0.8255502386522289? - 0.4815419759451981?*I)],
  1)]
```

```
#eigen vector with eigen value of 1
eigenvector=vector([1, 163891/147582, 40663/24597, 68432/73791, 63550/73791])

#sum all elements of the eigen vector
sum = 1 + 163891/147582 + 40663/24597 + 68432/73791 + 63550/73791

#divide the eigen vector by its sum to normalize the vector
#and obtain static or long term distribution
n(eigenvector/sum, digits=4)
```

```
(0.1801, 0.2000, 0.2977, 0.1670, 0.1551)
```

Appendix C. Evolution of our traffic system

```
#after 15 minutes
after15 = markov*initial
#after 30 minutes
after30 = markov^2*initial
#after 45 minutes
after45 = markov^3*initial
#after 60 minutes
after60 = markov^4*initial

#after 1500 minutes (approximating long term distribution)
after1500 = markov^100*initial

n(after15, digits=4), n(after30, digits=4), n(after45, digits=4), n(after60, digits=4), n(after1500, digits=4)
```

```
(
[0.1899 0.1960 0.1667]  [0.1805 0.1806 0.1805]  [0.1797 0.1795 0.1800]
[0.1899 0.1869 0.1890]  [0.2016 0.2018 0.1979]  [0.2002 0.2002 0.2003]
[0.3038 0.3021 0.2898]  [0.2985 0.2990 0.2998]  [0.2975 0.2975 0.2979]
[0.1646 0.1621 0.1676]  [0.1661 0.1656 0.1663]  [0.1672 0.1672 0.1669]
[0.1519 0.1529 0.1869], [0.1533 0.1530 0.1555], [0.1554 0.1555 0.1549],

[0.1801 0.1801 0.1800]  [0.1801 0.1801 0.1801]
[0.2000 0.1999 0.2000]  [0.2000 0.2000 0.2000]
[0.2977 0.2977 0.2977]  [0.2977 0.2977 0.2977]
[0.1670 0.1671 0.1671]  [0.1670 0.1670 0.1670]
[0.1551 0.1552 0.1552], [0.1551 0.1551 0.1551]
)
```

Appendix D. Data collection sheet: ***LINK***