City Growth Model Simulation

VIGRE Spring 2012
Bowen Huang
Faculty Advisor: David Aldous

Introduction:

Urbanization is a process that many countries go through to be a developed country. The process is defined as populations migrate from low-density area to form high-density areas. It is estimated there are about 3000 cities in the world today[1] with a population over 100,000, with many of them cluster together and become what is known as metropolitan areas, such as the San Francisco Bay Area, New York Area, the Five Lake region. Currently, many cities in the world, especially those in under developed countries, are forming in a non-systematic fashion. A model that can describe the dynamics of each cities and their relationship to one another will be invaluable to help city planner's understanding of the cities. A better understanding of the cities can solve many city and regional planning problems optimally as well as control the city's behavior in a desirable way. Examples of problems can be solved better includes but not limit to: predicting which neighboring cities suburban residents will go to work and shop, finding the optimal path for building highways without interfering each cities' growth, assigning the best zoning code for each region within the city etc.

This paper presents a mathematical model for mimicking the growth of cities. This model, however, can be also generalized to many other fields and applications due to its interesting mathematical properties. Some examples are: to find the safest path for robots to traverse through a dangerous terrain, to calculate the precipitation of an area from point measurements, saving both time and costs. Theoretical mathematical analysis of the behavior of the model is very difficult and produces limited results. This paper attempts to utilize simulation to investigate the model behavior and compare them to the theoretical results.

The Model:

The model has three major parts: model specifications, model step function and model properties due to different specifications.

At a specific time t=1, 2, 3… there are cities at position $(x,y)_i$ within a unit square $[1,1]^2$, with population N_i (t) > 1. Total population is t. Three parameters of the models are $0 < C < \infty$, $0 < a < \infty$, $b > 0$. The influence of each city to some point is defined by the function:

$$I(n,r) = C\, n^a\, r^b$$

Where n is the population and r is the direct distance from the center of the city.

The sphere of influence of a city is then:

[1] Mongabay

$S(i,t) = \{\text{all } P \text{ such that } I(N(t), |P-x|) = I(P,t)\}$

Where P is any points on the unit square and x is the center of specific city.

This model will then evolve stochastically with time. U represents a new immigrant uniform randomly arrive at some point within the unit square:

i) With probability $1/(1+I(U,t))$ a new city is found with population 1.
ii) With probability $I(U,t)/(1+I(U,t))$ joins the city I for which U contains in $S(i,t)$, therefore increases the population by 1.

Where $I(U,t)$ is the maximum of $\{S(i,t) \text{ for all } i\}$.

The model has different properties depending on the specification[2]:

1) If $0 < a < 1$ and $b > 2a$, the cities exhibit balanced growth behavior, meaning no single city can grow too fast relative to others.
2) If $a > 1$ or $b < 2a$, one of the city will have dominant effect over all other cities with large population, which is the unbalance grow scenario.
3) The critical boundary is when $a = 1$ and $b > 2a$.

Simulation Method and Data:

The main program we used for the simulation is Matlab. This program has many powerful features that allow users to easily create the data generating process and eventually visualize the result using 2-D graphics.

The entire simulation is done on a user-defined object in Matlab called *points*, which contains all the pertinent information about the unit square, including city positions and their population at t. The following example is a *points* object at t=20 for a = 0.2, b=0.9 and c =1:

```
points =
  Columns 1 through 7
    0.5469    0.3922    0.7060    0.0344    0.7655    0.7094    0.6797
    0.9575    0.6555    0.0318    0.4387    0.7952    0.7547    0.6551
    9.0000    2.0000    2.0000    2.0000    1.0000    1.0000    1.0000
  Columns 8 through 10
    0.3404    0.7513    0.5472
    0.5853    0.2551    0.1386
    1.0000    1.0000    1.0000
```

The first row is the x position of the cities within the unit square. The second row is the y position of the unit square. Lastly, the integers in third row represent the population at the specific time for each of the cities.

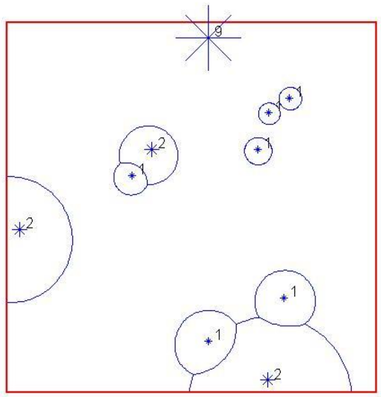
[2] David Aldous, "*City Growth Model*", working paper.

A user-defined function is then used to help simulate the stochastic evolving process of the *points* object. It is called the *newpoint* function, which follows the model step function by adding an uniform random point sampled from the unit square into the system. Then, with probability 1/(1+I(U,t)), this point will found a new city (add a new column to *points* with the position and population of 1), where I(U,t) is the highest influence neighboring city's influence on the point's position. Alternatively, it can join an existing city (+1 to third row). For the above example, the next step t=21, a new city is founded:

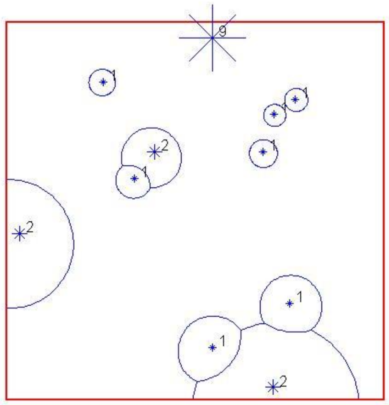
```
points =
 Columns 1 through 7

   0.5469    0.3922    0.7060    0.0344    0.7655    0.7094    0.6797
   0.9575    0.6555    0.0318    0.4387    0.7952    0.7547    0.6551
   9.0000    2.0000    2.0000    2.0000    1.0000    1.0000    1.0000
 Columns 8 through 11
   0.3404    0.7513    0.5472    0.2575
   0.5853    0.2551    0.1386    0.8407
   1.0000    1.0000    1.0000    1.0000
```

One can verify this by noticing the new column in the object.

Even though the entire simulation process can be done without any graphical supports, it is extremely difficult to have an intuitive understand of the behavior of the system with only matrix representation. Therefore, we create a graphic function to visualize the changes. This function is a modified version Andrew Kwok's original *voronoi* function. The two graphs below shows the changes of the system from the data above:



t = 20



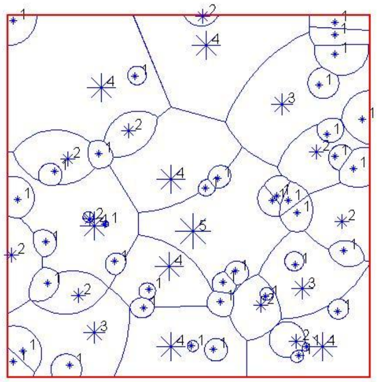
t = 21 with newly founded city

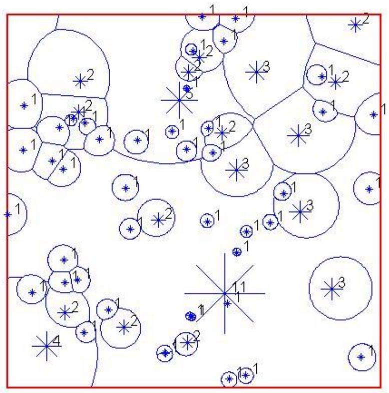
The numbers next to the dots in the graphs represents the population number for that city.

Simulation and Analysis:

Applying the method from the last section, we run 6 simulations of various model specifications and t = 100 to verify the theoretical properties of the models.

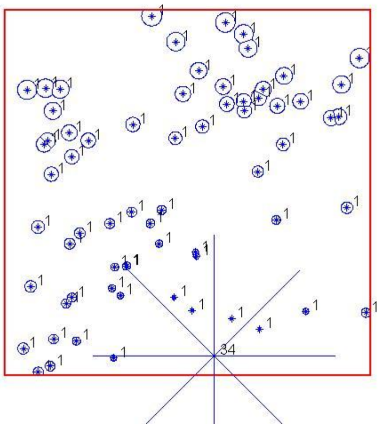
The first simulation sets a = 0.2, b = 0.9 and c = 1. This is the balance growth scenario according to the theoretical derivation. Here are the resulting regions at t=100 for 2 different tries:

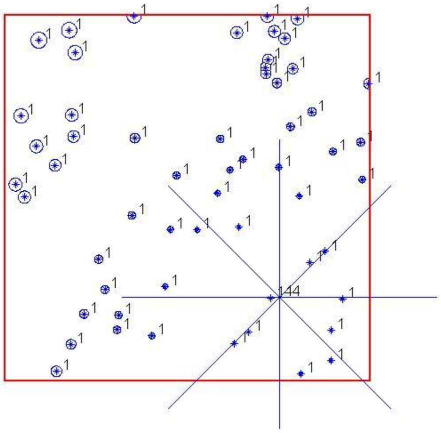




The simulation confirms the theoretical hypothesis that for a < 1 and b > 2a, more large cities are likely to form and none of a single city has too much influence as all the cities have a relative equal chance of growing. In the second graph above, the highest influence city has a population of 11 while the first one has only 5. These graphics closely mimic the famous Voronoi Diagram. Indeed, when two cities have the equal population, the area between them will be the Voronoi bisection.

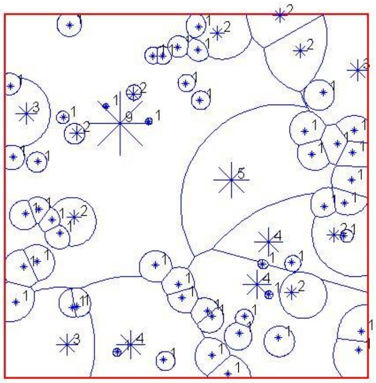
The next two tries set a = 1.2 and b = 0.5, c is still 1. Same as last time, the terminating time t in this case is still 100. The graphs below are the results of the two simulations:

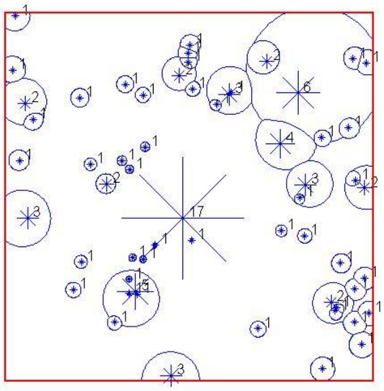




As one can easily see from the two graphs, in both simulations, one single city dominates the entire region---in our simulation, the first one having a population of 54 and the second one 44. This happens because the first city has a much higher chance of attracting the new arrivals, and this result in its enhancing ability to retain the other new arrivals in the future. All other cities, on the other hand, can form but having difficulties retaining new immigrants.

The last simulation here is the critical case, when a = 1 and b = 2a. C is still set to be 1 for simplicity. The results are below:





The boundary case is particular interesting because it exhibits the properties of both the unbalance growth case as well as the balance growth case---depending on the random position of the cities. The first graph above is closely related to the balance growth case: highest influence city has population of 9 and there are some other large cities—4, 4, 5 etc. However, in the second case, we also can tell the traces of the unbalance growth case: one of the cities has a population of 17 in the center, dominate the entire area. This differ from the unbalance case in that there are some other smaller cities with population higher than 1, such as the city with population 6 on the upper right hand corner of the second picture above.

Conclusion:

With the limited numbers of runs, all the results appear to confirm the theoretical derivation of categorizing the model into balance, unbalance and critical categories, depending on the model parameter a, b and c. For this model, when a is less than 1 and b is larger than 2a, the region exhibit balance growth. When a is larger than 1 and b less than 2a, the one city has majority of the influence. In the boundary case, either scenario can play out depending on the random selection outcome. The model parameter c is less meaningful as a scaling factor for the influences.

Future Works:

This paper is by no mean a complete study of the behavior of the mathematical model above. It intends to provide some ground work for verifying the theoretical behaviors only. Further investigations are required for thorough understanding of the model before any actual applications.

Acknowledgements:

This paper cannot exist without the generous support and guidance of Professor David Aldous of University of California at Berkeley. He provides the original idea as well as the underlying model used in this simulation. I like to also thank Andrew Kwok of University of California at San Diego for providing the sample code for the graphical function in this simulation, and Mike Bostock, Nicolas Belmonte for their helps with my exploration and understanding of creating graphics from data.

References:

David Aldous, "*City Growth Model*", working paper.
D3 JavaScript Documentation: *http://mbostock.github.com/d3/api/*
MatLab Central: http://www.mathworks.com/help/techdoc/ref/axis.html
Nicolas Garcia Belmonte, Voronoi Tessellation: http://blog.thejit.org/2010/02/12/voronoi-tessellation/
Andrew Kwok, Generalized Voronoi Partition in Matlab:
http://tintoretto.ucsd.edu/andrew/voronoi.html

Appendix:

Matlab Code Files List

1) Newpoint.m
2) Mwvoronoi.m
3) Makeregions.m
4) Intersect.m
5) Drawregions.m
6) Drawarc.m
7) Simulation.m