assign3_ANSWER

October 20, 2018

1 Simulation in Sociology, Moretti (2002)

1.1 (a)

The author describes validity as an important criteria of simulation techniques in sociology. Validity is determined by how the model and the computational tools are representative of the reality. Two of the simulation techniques described by the author, multiagent systems and cellular automata, both have some weaknesses in validity.

- Multiagent systems In applications in social anthropology based on emergence of social norms, the definition of agents is based on the concept of bounded rationality developed by Simon(1957). In this concept of bounded rationality, the limited knowledge and abilities of the decision maker are taken into account. However, the models of rationality should be extended to learning and adaptation to be more realistic. And aside from the properties of individuals that currently are included in multiagent system models, there are many other aspects of psychological theories that should be incorporated, such as emotions, motivations, desire, intent, consciousness. Another major challenge in the development of multiagent systems is the formalization of knowledge. What kinds of knowledge are formalizable and how to best formalize knowledge still remain for future reserach.
- Cellular automata One weakness of cellular automata with regard to validity is the use of synchronous updates of states. It is assumed that all cells update simultaneously according to a global clock, which is unrealistic because individuals modify their attitudes and opinions at different moments in actual social processes. Another important weakness comes from the restrictions imposed by spatial structures, where each individual interacts only with a subset of the whole population. This assumption itself is acceptable since it would be unrealistic for individuals to interact with the whole population. However, the limitation based on spatial distance is not. With the help of media, people can interact with people physically far away. And since the range of neighborhood changes over time, it is not realistic to have a static model of the neighborhood.

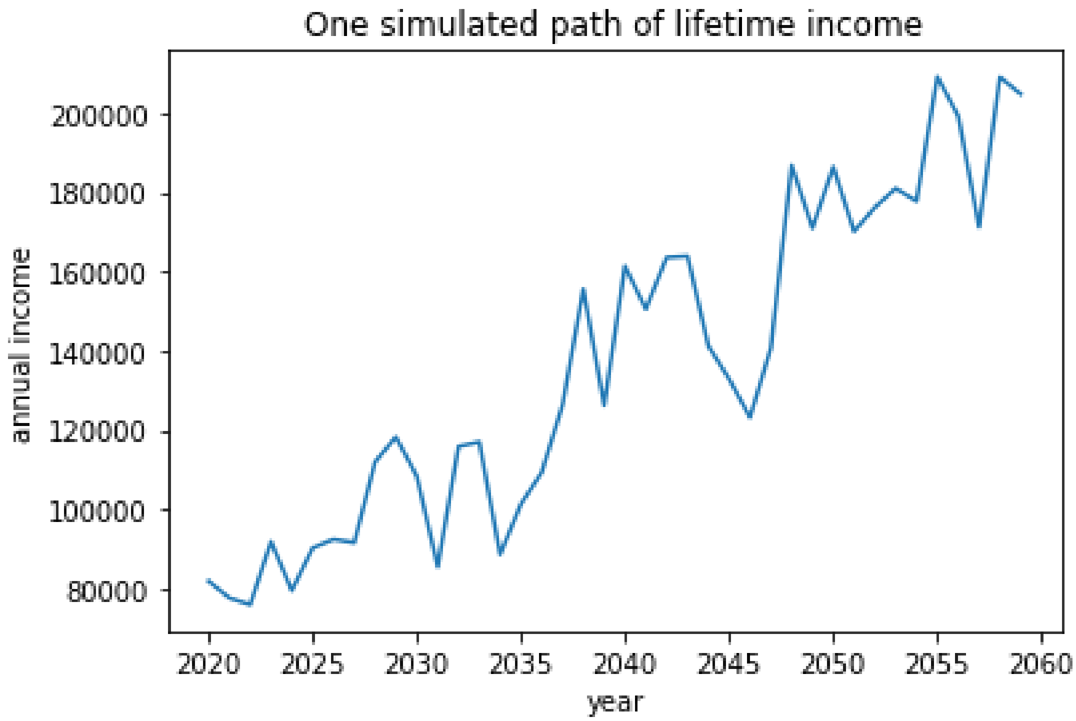
The author also highlights “dynamic feedback” as a key characteristic that computer simulation is good at modeling. One example of dynamic feedback the author cites from sociology is the model in Hanneman, Collins, & Mordt (1995). In this legitimacy-seeking model, the motivation of rulers to initiate external conflict is directly proportional to the difference between their current legitimacy and the goal of maximum legitimacy. After the conflict is initiated, the result of it leads to change in prestige in the status order of political communities, which in turn changes the current legitimacy level. Then the difference between the current legitimacy and the goal of maximum legitimacy is changed, so that a new conflict may be initiated.

A possible research question on a political science topic would be how policy changes affect the political attitudes and behaviors among members of the public, and how this in turn influences policy.

2 Simulating your income

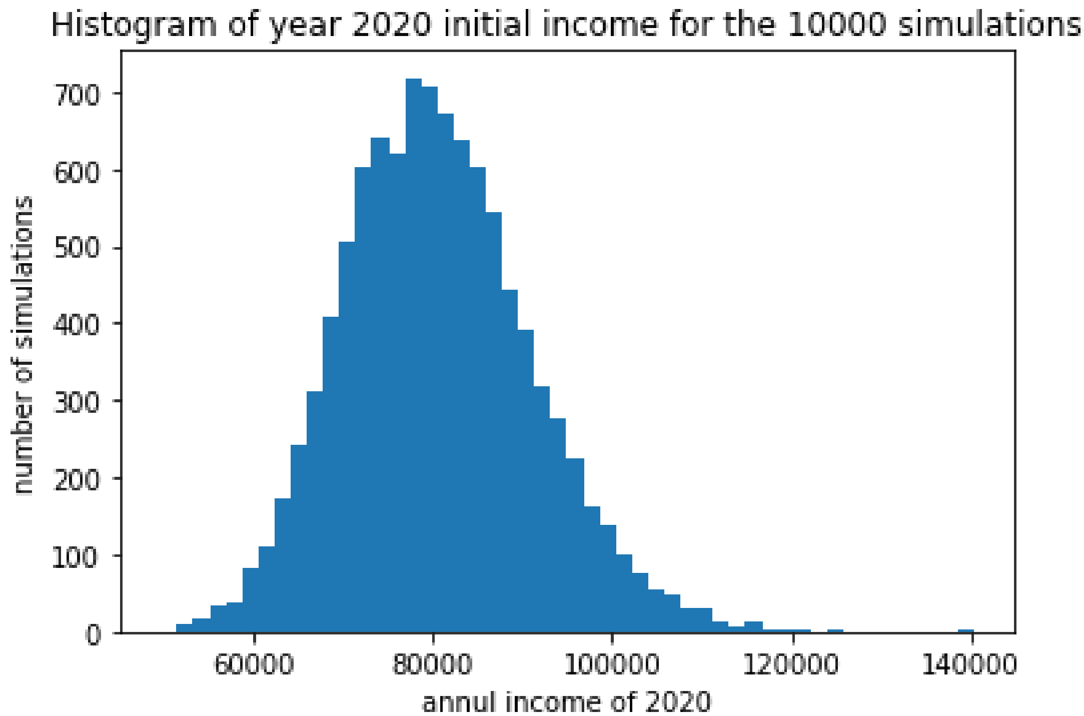
2.1 (a)

```
In [26]: import numpy as np
         epsilon = s = np.random.normal(0, 0.13, (10000,40))

         inc_0 = np.full(10000, 80000)

         ln_inc_0 = np.log(inc_0)

         ln_inc = np.zeros((10000, 40))

         ln_inc[:,0] = ln_inc_0 + epsilon[:,0]

         for i in range(1,40):
             ln_inc[:,i] = (1-0.4)*(ln_inc_0+0.025*i) + 0.4*ln_inc[:,i-1] + epsilon[:,i]

         import matplotlib.pyplot as plt
         years = np.array(range(2020, 2060))

         plt.plot(years, np.exp(ln_inc[0,:]))
         plt.xlabel('year')
         plt.ylabel('annual income')
         plt.title('One simulated path of lifetime income')

Out[26]: Text(0.5,1,'One simulated path of lifetime income')
```



2.2 (b)

```
In [28]: plt.hist(np.exp(ln_inc[:,0]), 50)
         plt.xlabel('annul income of 2020')
         plt.ylabel('number of simulations')
         plt.title('Histogram of year 2020 initial income for the 10000 simulations')

Out[28]: Text(0.5,1,'Histogram of year 2020 initial income for the 10000 simulations')
```




```
In [31]: print((np.exp(ln_inc[:,0])>100000).sum()/10000)
         print((np.exp(ln_inc[:,0])<70000).sum()/10000)
```

```
0.0418
0.1516
```

4.18% of the class will earn more than $100000 in the first year out of the program. And 15.16% of the class will earn less than $70000. This distribution is not normally distributed since it has a fat right tail. ## (c)

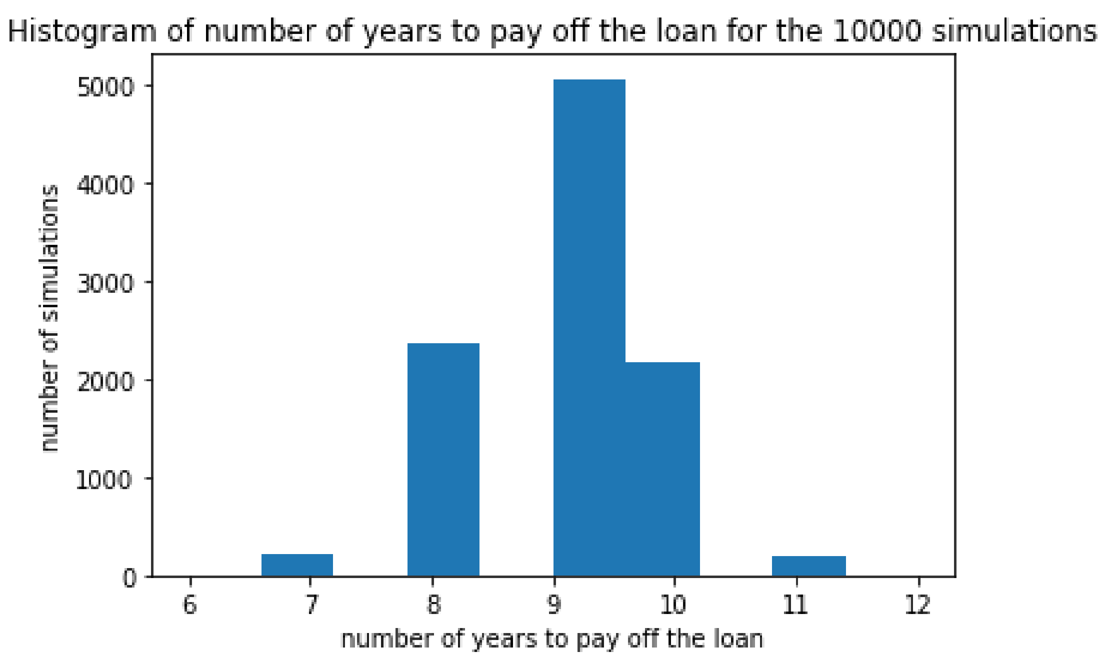
```
In [41]: pay_debt = 0.1*np.exp(ln_inc)

         pay_years = np.argmax(np.cumsum(pay_debt,axis=1)>95000, axis=1)

         plt.hist(pay_years)
         plt.xlabel('number of years to pay off the loan')
         plt.ylabel('number of simulations')
         plt.title('Histogram of number of years to pay off the loan for the 10000 simulations')

         print((pay_years<=10).sum()/10000)
```

```
0.7624
```

In 76.24% of the simulations I am able to pay off the loan in 10 years. ## (d)

```
In [43]: epsilon = s = np.random.normal(0, 0.17, (10000,40))

         inc_0 = np.full(10000, 90000)

         ln_inc_0 = np.log(inc_0)

         ln_inc = np.zeros((10000, 40))

         ln_inc[:,0] = ln_inc_0 + epsilon[:,0]

         for i in range(1,40):
             ln_inc[:,i] = (1-0.4)*(ln_inc_0+0.025*i) + 0.4*ln_inc[:,i-1] + epsilon[:,i]

         pay_debt = 0.1*np.exp(ln_inc)

         pay_years = np.argmax(np.cumsum(pay_debt,axis=1)>95000, axis=1)

         plt.hist(pay_years)
         plt.xlabel('number of years to pay off the loan')
         plt.ylabel('number of simulations')
         plt.title('Histogram of number of years to pay off the loan for the 10000 simulations')

         print((pay_years<=10).sum()/10000)
```

```
0.9804
```



In 98.04% of the simulations I am able to pay off the loan in 10 years.