Markov Chains

Raina Zhang

Ergodic Markov Chains

The tested Matrix is as follows:

```
[0.1  0.4  0.5]
[0.6  0.2  0.2]
[0.2  0    0.8]
```

With the Eigenvector method, where it calculates the eigenvector with eigenvalue = 1, the result is as follows:

π* = [[[0.22222222],[0.11111111],[0.66666667]

Verifying this result using matrix multiplication, multiplying by itself for 6 iterations, for a total of M^64, the result is the following Matrix:

```
[0.22222222 0.22222222 0.22222222]
[0.11111111 0.11111111 0.11111111]
[0.66666667 0.66666667 0.66666667]
```

Where π* is any one of the columns, resulting in:

π* = [[[0.22222222],[0.11111111],[0.66666667]]

The simulation runs 1000 times before starting to record the results, which it then records and calculates the time average for iterations 1000-2000. This process then repeats 10 times to find the time average for 10 iterations, which then averages to find the ensemble average for the closest result. The results for 3 runs of the simulation are as follows:

π* = [0.2232 0.1144 0.6624]
π* = [0.2238 0.1121 0.6641]
π* = [0.2262 0.1173 0.6565]

Increasing the number of iterations before starting to record for the time average and increasing the number of times the simulation is repeated before taking the ensemble average will give a closer result.

The result for the matrix multiplication method is exactly the result given from the eigenvector. The results from the simulation are accurate enough, with deviation mostly happening after the first two decimal values. With more iterations for the simulation, we can expect more accurate results.

Absorbing Markov Chains:

The tested Matrix is as follows:

```
[1    0    0    0  ]
[0.1  0.2  0.3  0.4]
[0.4  0.1  0.3  0.2]
[0    0    0    1  ]
```

With the Chapman-Kolmogorov Equation solution, we end up with 2 vectors as follows:

```
v1 = [0.35849057 0.62264151]
v2 = [0.64150943 0.37735849]
```

v1 is the probability vector that the chain will converge to 1, with 0.35849057 being the probability that it converges to 1 if the start state is 2, and 0.62264151 for when the start state is 3. If given the starting vector π(0), multiplying the values of the starting vector with these values will give the results for that particular starting vector. For example, if the starting vector was:

```
π(0) = [0, 0.3, 0.7, 0]
```

The final probability vector in regards to that starting vector will be:

```
Pr(1) = (0.3 * 0.35849057) + (0.7 * 0.62264151)
```

v2 is the probability vector that the chain will converge to 4, with the first value being if the start state is 2, and second being if the start state is 3. The same about the starting vector being multiplied in is true here.

The simulation simulates if the start state is always 2, and if the start state is always 3. The starting vectors are as follows:

1. $\pi(0) = [0, 1, 0, 0]$
2. $\pi(0) = [0, 0, 1, 0]$

The resulting vector is slightly different from the vectors obtained from the CK-equations, as the vectors from the previous solution is the probability that it will converge to the absorbing state for each non-absorbing state, whereas the vector for this simulation is the probability to converge to either absorbing state when the starting state is a non-absorbing state. In other words, the CK-equations give the probability that it will converge to state 1 in one vector, and the probability that it will converge to state 4 in the other, where the simulation gives the probability that it will converge to either state 1 or state 4 given that the starting state is 2 or 3.

The simulation takes the ensemble average of 20000 test cases. The results for 3 runs of the simulation are as follows:

1. Starting at state 2:[0.36055, 0.63945]
   Starting at state 3:[0.62205, 0.37795]

2. Starting at state 2:[0.36065, 0.63935]
   Starting at state 3:[0.62115, 0.37885]

3. Starting at state 2:[0.35555, 0.64445]
   Starting at state 3:[0.62405, 0.37595]