

Exploring Equilibrium of A Markov Chain Model of Migration and Social Utility

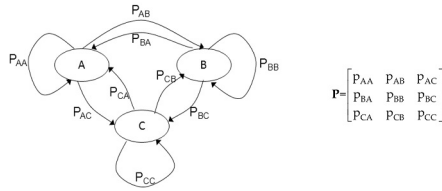
Charley Ferrari, Paul Garaud

December 20, 2015

Markov Chains

Our paper models the markov chain population model suggested by Pan and Nagurney in their paper *Using Markov Chains to Model Human Migration in a Network Equilibrium Framework*, available for download [here](#).

Pan and Nagurney model the question of optimal migration patterns using a markov chain. Each location represents a “state” that the markov chain can take, and transition matrices can be calculated to calculate the behavior of the chain. Below is an example of what the model would look like with three locations.



Transition matrices are calculated by looking at the utility of different locations as functions of population, and the cost of moving, provided as a function of flow. The cost being represented as a function of flow is what allows the markov chain to operate in stages, and not just immediately reach an equilibrium.

Pan and Nagurney introduce a two region example of the model. In their example, the utility functions are defined below:

$$u_1(p) = -p_1 + 8$$

$$u_2(p) = -p_2 + 14$$

And cost is defined by flow:

$$c_{ij} = 2f_{ij}$$

Given an initial population distribution, at each step we can solve for the flow matrix, $F = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix}$.

People will move from locations of lower utility to locations of higher utility. The flow at each step in the markov chain will be determined by solving the below equation:

$$u_i(p) + c_{ij}(f) = u_j(p)$$

The algebraic solution for the first few steps can be seen in the paper:

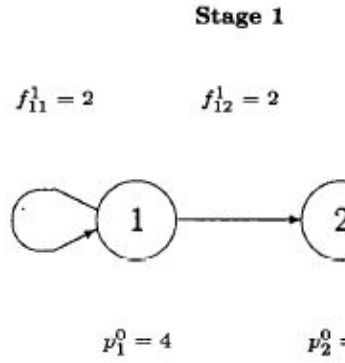


Figure 1. A_1

See Figure 1 where a transition matrix A_1 is associated with the transition matrices A_1 and A_2 .

$$u_1(p) = -p_1 + 8, \quad u_2(p) = -p_2 + 14, \quad c_{12}(f) = 2f_{12}, \quad c_{21}(f) = 2f_{21}.$$

$$A_1 = \begin{bmatrix} 2/4 & 0 \\ 0 & 0 \end{bmatrix}$$

We created the below function to calculate the flow at each step:

```
flowcalcpn <- function(p){
  if(u2pan(p) > u1pan(p)){
    f1 <- (p[1] - p[2] + 6)/4
    f <- c(f1,0)
  }else if(u1pan(p) > u2pan(p)){
    f2 <- (p[2] - p[1] - 6)/4
    f <- c(0,f2)
  }else{f <- c(0,0)}

  return(f)
}
```

This function uses the predefined functions `u1pan` and `u2pan` for solving at each step, but also includes the algebraic solution:

$$f_{12} = \frac{p_1 - p_2 + 6}{4}$$

or

$$f_{21} = \frac{p_2 - p_1 - 6}{4}$$

This is arrived at by solving directly for $u_i(p) + c_{ij}(f) = u_j(p)$. Considering a step towards a population distribution p_n from an initial population distribution p_{n-1} , and assuming our lower utility region is region 1 (so we have a flow from region 1 to region 2), we have:

$$p_{1,n} = p_{1,(n-1)} - f_{12}$$

And similar solutions, depending on the direction of our flow and which location we're looking at. p_n is a function of p_{n-1} and flow.

Starting at an initial population, here is an example of a table stepping through this chain:

```
p0 <- c(4,2)

u0 <- c(u1pan(p0), u2pan(p0))

pdatapan <- data.frame(p1 = 4, p2 = 2, u1 = u0[1], u2 = u0[2],
                      totu = p0 %*% u0 )

iter <- 1:5

for(i in iter){
  p <- c(tail(pdatapan$p1,1),tail(pdatapan$p2,1))
  f <- flowcalcpan(p)
  p[1] <- p[1]-f[1]+f[2]
  p[2] <- p[2]+f[1]-f[2]
  u <- c(u1pan(p),u2pan(p))
  totu <- p %*% u

  pdatapan <- rbind(pdatapan, c(p[1], p[2], u, totu))
}

kable(pdatapan, format = "markdown")
```

p1	p2	u1	u2	totu
4.000	2.000	4.000	12.000	40.00000
2.000	4.000	6.000	10.000	52.00000
1.000	5.000	7.000	9.000	52.00000
0.500	5.500	7.500	8.500	50.50000
0.250	5.750	7.750	8.250	49.37500
0.125	5.875	7.875	8.125	48.71875

Social Utility

We added a last column to the table above: totu. This is meant to be the total utility of the society. This is simply the sum of the utilities of each individual, or, given utility and population vectors, $p \times u^T$. Looking at this column, we can see that people acting in their own self interest do not guarantee a socially optimal distribution.

Solving for optimal social utility is a linear programming problem. We are trying to maximize u_{tot} :

$$u_{tot} = p \times u$$

Subject to the restrictions given by our total population across n regions:

$$\sum_{i=1}^n p_i = p_{tot}$$

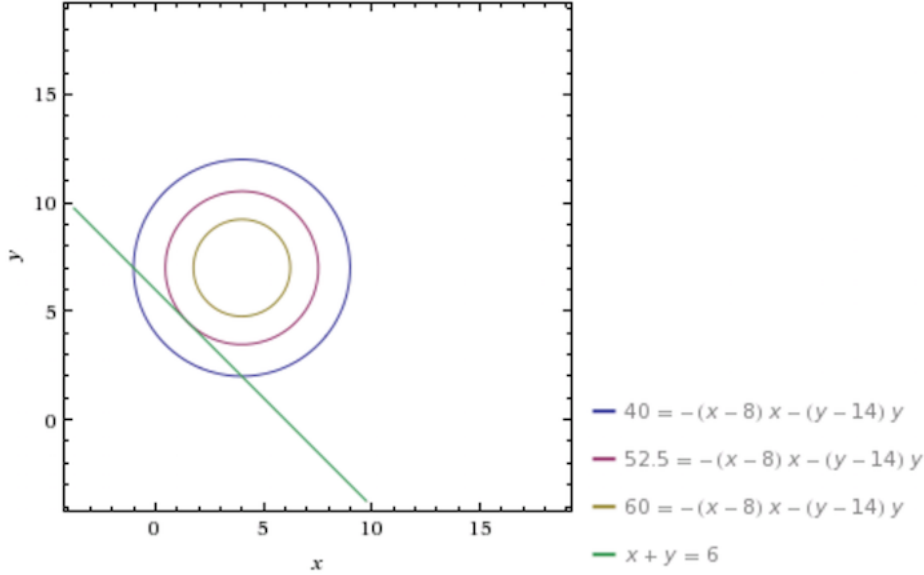
For the two-population example given above, we have:

$$u_{tot} = p_1(8 - p_1) + p_2(14 - p_2)$$

And

$$p_{tot} = p_1 + p_2 = 6$$

We can analyze this problem graphically. Below are a few graphs of different values of u_{tot} . As we have them defined, these are circular functions, and the total population restraint is a straight line in the p_1 p_2 plane.



Algebraically, you can simply solve for p_2 , and end up with u_{tot} as a function of p_1 :

$$u_{tot} = p_1(8 - p_1) + (6 - p_1)(8 + p_1)$$

Which leads to the quadratic equation:

$$u_{tot} = -2p_1^2 + 6p + 48$$

This can pretty simply be solved with a maximum of $u_{tot} = 52.5$ at $p_1 = 1.5$ and $p_2 = 4.5$.

We can expand this over multiple locations, and end up with multidimensional problems. If we add a third location, set $p_{tot} = p_1 + p_2 + p_3 = 40$, and set $u_3 = 18 - p_3$, we end up solving the below problem:

$$u_{tot} = p_1(8 - p_1) + p_2(14 - p_2) + p_3(18 - p_3)$$

Subject to:

$$p_1 + p_2 + p_3 = 40$$

This is a more advanced problem, our population limit is defined as a plane rather than a line. Algebraically solving for p_3 we can at least simplify this problem into maximizing a three-dimensional function:

$$p_3 = 40 - p_1 - p_2$$

$$u_{tot} = -2p_1^2 - 2p_2^2 - 2p_1p_2 + 70p_1 + 76p_2 - 880$$

A function like this would require more advanced methods to maximize, like Gradient ascent. Lets build a function to do this. First, lets build a function to compute the total utility given p_1 and p_2

```
computeFunc <- function(p1,p2){
  return(-2*p1^2 - 2*p2^2 - 2*p1*p2 + 70*p1 + 76*p2 - 880)
}
```

To run gradient ascent, we'll need to calculate the gradient of this function of p_1 and p_2 . This is simply the two partial derivatives: $\frac{du_{tot}}{dp_1}$ and $\frac{du_{tot}}{dp_2}$. Computing these, we have:

$$\frac{du_{tot}}{dp_1} = -4p_1 - 2p_2 + 70$$

and

$$\frac{du_{tot}}{dp_2} = -4p_2 - 2p_1 + 76$$

To complete the gradient ascent, lets create a function that records a number of iterations into a data frame. Starting at a $p_{1,0}$ and $p_{2,0}$, collectively p_0 , we'll calculate our steps as:

$$p_1 = p_0 + \alpha \cdot \nabla u_{tot}(p_0)$$

```
gradientAscent <- function(p1,p2,alpha,numiters){

  grad <- data.frame(p1 = p1, p2 = p2, utot = computeFunc(p1,p2))

  for(i in 1:(numiters-1)){
    newp1 <- p1 + alpha*(-4*p1 - 2*p2 + 70)
    newp2 <- p2 + alpha*(-4*p2 - 2*p1 + 76)
    p1 <- newp1
    p2 <- newp2
    grad <- rbind(grad,c(p1,p2,computeFunc(p1,p2)))
  }

  return(grad)
}
```

Doing 50 iterations, starting with initial guesses $p_1 = 2$ and $p_2 = 2$, and using $\alpha = 0.2$, this leaves us with the results:

p1	p2	utot
2.00000	2.00000	-612.00000
13.60000	14.80000	-13.76000
10.80000	12.72000	11.09120
11.07200	13.42400	12.41702
10.84480	13.45600	12.58950
10.78656	13.55328	12.63939

...

	p1	p2	utot
45	10.66667	13.66667	12.66667
46	10.66667	13.66667	12.66667
47	10.66667	13.66667	12.66667
48	10.66667	13.66667	12.66667
49	10.66667	13.66667	12.66667
50	10.66667	13.66667	12.66667

We successfully converge to the correct maximum. $p_1 = 10.67$, $p_2 = 13.67$, and the maximized social utility is $u_{tot} = 12.67$

Achieving Maximum Social Utility

Now, lets look at the question of how we can acheive maximum social utility. As we've shown above, maximum social utility isn't guarenteed with each individual acting in their own self interest. One way a more socially optimal outcome could be incentivized is through taxation.

One difference to note is at the moment we're talking about socially optimal outcomes, not equitable outcomes. Consider a situation where one individual can make a decision that increases their utility by n , but decreases another individual's (or multiple individual's) utility equally by n . Measuring the relative utilities of everyone would allow us to find out which outcome is more equitable, but regardless of equitability both of these outcomes are equally optimal socially.

Now, consider a situation where one individual can make a decision that increases their utility by n , but decreases another individual's utility by some value greater than n . In this case, we're moving from a socially optimal point to a less socially optimal point. The gain in this individuals utility is less than the loss in utility felt by other people. It's in this individual's self interest to move to a less socially optimal outcome.

Incentives can be used to keep this person from making this decision. Because this individual's decision may decrease social utility by a value greater than n , a tax of n given to this individual could incentivize them to not make a socially un-optimal decision.

Lets look at our migration problem through this lens:

p1	p2	u1	u2	totu
4.000	2.000	4.000	12.000	40.00000
2.000	4.000	6.000	10.000	52.00000
1.000	5.000	7.000	9.000	52.00000
0.500	5.500	7.500	8.500	50.50000
0.250	5.750	7.750	8.250	49.37500
0.125	5.875	7.875	8.125	48.71875

Using our two region example, lets assume we're using a tax of t . Looking at the situation where $p_1 = 1$ and $p_2 = 2$, the game can be modeled using the payoff matrix below:

(p1, p2)		Population in 2	
		Tax	No-Tax
Person in 1	Move	(8,8)	(8,8)
	Stay	(6 + 5t, 10-t)	(6,10)

This is somewhat like the prisoner's dilemma. If $5t < 2$, moving is a strictly dominant strategy for player 1, so they will definitely move. Given this, player 2 gets the same payoff whether or not they tax, so we'll assume they have a 50% chance of taxing. If $5t > 2$, we will eventually end up in the same place.

Consider the situation where player 1 moves and player 2 taxes. Player 1 would be better off if they decided to stay. However, if player 1 decided to stay, player 2 would vote to not tax. Given that, it would make sense for player 1 to move. This behavior suggests there might be a mixed strategy equilibrium.

Lets assume Player 1 will move with probability p_m , Player 2 will tax with probability p_t , and look at the decision facing player 2:

Tax:

$$8p_m + (1 - p_m)(10 - t) = 10 - 2p_m - t + p_mt$$

No Tax:

$$8p_m + 10(1 - p_m) = 10 - 2p_m$$

The p_m that would induce player 2 to mix between these two strategies would occur when these two payoffs are equal. So:

$$10 - 2p_m - t + p_mt = 10 - 2p_m$$

Solving for the above equation, we get $p_m = 1$. In a mixed strategy equilibrium, player 1 will move no matter what the tax is.

Looking at this sort of population model can give us a lot of different views of this problem. The model as presented by Pan and Nagurney shows the network equilibrium of the problem faced by individuals faced with these utilities. Taking social utility into consideration allows different views of the problem, which lent themselves to be modeled using a wide variety of techniques.