09 Segregation Models
Cities are Complex Systems

What happens?

- Traffic
- Pollution
- Epidemics
- Inequalities
  - Housing
  - Economic
  - Racial
Cities are Complex Systems
Can we model them?

Agent-Based Models (ABMs)

- **Pro:**
  - Detailed representation
  - Explainable
  - What-if tool

- **Cons:**
  - Curse of dimensionality
  - No predictions
  - Only simulative

Artificial Intelligence Models

- **Pro:**
  - Accurate algorithms
  - Latent Knowledge
  - Data-driven

- **Cons:**
  - Hard to interpret
  - Hard to control
  - Performance based
Segregation

The act by which a (natural or legal) person separates other persons on the basis of one of the enumerated grounds without an objective and reasonable justification.

- European Commission against Racism and Intolerance
Segregation
C’mon it’s 2022... (I)
Segregation
C’mon it’s 2022... (II)

(a) Black population ≥ 7% in 2000
White population ≥ 84% in 2000

(b) Black population ≥ 7% in 2016
White population ≥ 84% in 2016
Segregation

Did you mean Schelling?

DYNAMIC MODELS OF SEGREGATION†

THOMAS C. SCHELLING

Harvard University

Some segregation results from the practices of organizations, some from specialized communication systems, some from correlation with a variable that is non-random; and some results from the interplay of individual choices. This is an abstract study of the interactive dynamics of discriminatory individual choices. One model is a simulation in which individual members of two recognizable groups distribute themselves in neighborhoods defined by reference to their own locations. A second model is analytic and deals with compartmented space. A final section applies the analytics to ‘neighborhood tipping.’ The systemic effects are found to be overwhelming: there is no simple correspondence of individual incentive to collective results. Exaggerated separation and patterning result from the dynamics of movement. Inferences about individual motives can usually not be drawn from aggregate patterns. Some unexpected phenomena, like density and vacancy, are generated. A general theory of ‘tipping’ begins to emerge.
Schelling Model
An Overview (I)

The Model

- **First** ABM of the history
- City as a **chessboard**
- **Two races** of agents (b/w)
  - 1 householder occupy 1 houseunits (cell)
- **Initially** agents at **random**
Schelling Model

An Overview (II)

The Dynamics

- Each agent has 8 direct neighbours.
- At each time step, dissatisfied agents relocate.
- Agents desire a fraction ($B_a$) of their neighbors ($B$) to be like them for being satisfied.
  - the higher $B_a$, the higher the intolerance.
- If $B < B_a$, the agent relocate to a free cell where it is satisfied.
Schelling Model

Let’s start... From the end!

Discovery #1
- Even if all agents tolerate up to $\frac{2}{3}$ different neighbours
  - City becomes segregated

Discovery #2
- Segregation is **sudden**
  - Small number of **step** is needed

Discovery #3
- At the end city is **more** segregated than necessary
  - Overwhelming effect

Tolerance ≠ Inclusion
Simulating Schelling


https://ncase.me/polygons/
Schelling Model

Simulations (I)

Question #1
Which is (±) the number of steps needed for segregating a 50x50 city? (Schelling default values)

Answer #1
12
Question #2
What happens to the convergence time if we change city dimensions?

Answer #2
Nothing
Schelling Model

Simulations (III)

**Question #3**
Which final segregation level do you expect with a intolerance level of 1/3? (Classical Schelling)

**Answer #3**
>50%
Schelling Model

Simulations (IV)

**Question #4**
What happens to the convergence **time** if we increase the **intolerance**? And to the final segregation level?

**Answer #4**
Increase, especially the second
Schelling’s take-home messages

● A crowd of tolerants is not a tolerant crowd

● Tipping point can, theoretically, exist
  ○ Less racist agents will follow the herd too:
    ■ If everyone of my race is leaving... I’m leaving

● “People get separated along many lines, and in many ways”
Schelling Model

Classical Variants

- Model change
  - Agent **vision** and behavior\(^1\)
  - Environment setting\(^2\)
  - Population proportions
- Role of
  - Venues\(^3\)
    - Mitigate segregation
  - Vision\(^4\)
    - Neighborhood

---

Schelling Model

A physics perspective

- Segregation is a **phase transition**\(^1\)
- It happens like a clustering of **droplets**\(^2\)
  - Same law of **Ising Model**\(^3\)
- **Tipping** point is hard to find\(^4\)
- **Attractors** exist\(^5\)

---


Schelling Model
A networks perspective

- Different **network** structure of the city are not crucial¹:
  - Several **indexes** developed²,³
- The more the **heavy** infrastructures in the **road** network
  - The more the segregation levels⁴

---

³ Eichenique, F. and Fryer Jr, R. G. " A measure of segregation based on social interactions ". 2007
⁴ G. Toth, J. Wachs, R. Di Clemente, A. Jakobi, B. Sagvari, J. Kertesz, and B. Lengyel, "Inequality is rising where social network segregation interacts with urban topology," 2021
Schelling Model

Mobility and AI

- Place and Social exploration explain experienced income segregation\(^1\)
  - Tested on large American cities
- Assessing the quality of a political partition studying the strength of the border\(^2\)
- First use of Reinforcement Learning\(^3\)

Modelling Agent-Based models

● Hard:
  ○ Need to account for multilinearity, time management, simultaneous choices etc.

● Need for a common framework
  ○ Several have been used in the past (NetLogo, JADE etc.)

● ... and for Data Science
  ○ MESA
MESA
A Python Framework

- Common **programming** language suite for ABM
- Two main classes:
  - **Model**
    - Stores variables and methods concerning the environment and the policies
  - **Agent**
    - Defines agents' behaviour. Each agent has a unique ID.
    - Each agent is linked to a model (accessible with `Agent.model`)
- Automatize handling of the **schedule** of an underlying **concurrent** structure
First toy ABM

- The best way to explain an ABM is to make an example of it
- ...so let’s see an example of a toy ABM
  - There are N agents
    - All of them start with 1 unit of money
    - Placed at random on a grid (can share a cell)
  - At every step an agent:
    - Moves at random within its neighborhood
    - Gives 1 unit of money (if they have it) to some other agent in the same cell
def compute_gini(model):
    agent_wealths = [agent.wealth for agent in
                     model.schedule.agents]
    x = sorted(agent_wealths)
    N = model.num_agents
    B = sum(xi * (N - i) for i, xi in enumerate(x)) / (N * sum(x))
    return 1 + (1 / N) - 2 * B

class MoneyModel(mesa.Model):
    def __init__(self, N, width, height):
        self.num_agents = N
        self.grid = mesa.space.MultiGrid(width, height, True)
        self.schedule = mesa.time.RandomActivation(self)
        self.running = True

        # Create agents
        for i in range(self.num_agents):
            a = MoneyAgent(i, self)
            self.schedule.add(a)
            # Add the agent to a random grid cell
            x = self.random.randrange(self.grid.width)
            y = self.random.randrange(self.grid.height)
            self.grid.place_agent(a, (x, y))

    self.datacollector = mesa.DataCollector(
        model_reporters={"Gini": compute_gini},
        agent_reporters={"Wealth": "wealth"})

    def step(self):
        self.datacollector.collect(self)
        self.schedule.step()
- Our MoneyAgent extend the MESA’s Agent class. The starting wealth value is 1.

- Auxiliary function that define my movement. An agent retrieve its neighbors CELLS calling `model.grid.get_neighborhood` and select one position at random. Then it uses `model.grid.move_agent` for moving.

- Function for giving money. An agent access the list of co-located agents (`model.grid.get_cell_list_contents`). If it has other agent in its cell it pick one at random and give a unit of money

- Define behaviour at each step. I move and then I call the function for giving money. CALLED BY `model.step()` (which in turn calls `model.schedule.step()` that respects the policy)
model = MoneyModel(50, 10, 10)
for i in range(100):
    model.step()

gini = model.datacollector.get_model_vars_dataframe()
gini.plot()

agent_wealth = model.datacollector.get_agent_vars_dataframe()
agent_wealth.head()
Running: Batch Run

```python
params = {"width": 10, "height": 10, "N": range(10, 500, 10)}
results = mesa.batch_run(
    model_cls = MoneyModel,
    parameters=params,
    iterations=5,
    max_steps=100,  # Halt condition
    number_processes=1,  # Multithreading for speed-up
    data_collection_period=1,  # Step of collection (1 means after each step)
    display_progress=True)
```

- How many run of the model?
  - `range(10, 500, 10) = 49 elements`
  - 2 fixed parameters (w and h)
  - 5 iterations
  - ... $49 \times 5 = 245$

```python
import pandas as pd

results_df = pd.DataFrame(results)
print(results_df.keys())
```

- Results can be stored in a df that contains the state of each run, agents etc.
  - Avg. 250 agents per simulation
  - 245 runs
    - Max 100 step per run
  - The df can be long $245 \times 250 \times 101 = 6186250$
Material


- [tutorial] Introductory Tutorial to Mesa (with the MoneyModel example)
Homeworks to be delivered by Friday, November 25th 2022
Homework 9.1

Literature propose 4 metrics for quantifying the segregation levels from data: (i) **exposure**: the extent to which different populations share the same residential areas; (ii) the **evenness** (and clustering): to which extent populations are evenly spread in the metropolitan area; (iii) **concentration**: to which extent populations concentrate in the areal units they occupy; and (iv) **centralization**: to which extent populations concentrate in the center of the city.

- Propose 4 simple mathematical formulas for calculate these metrics
- Implement them in Python/Mesa
- Create 4 plots that shows the trend of these 4 metrics during the Schelling dynamics (seen in class 11/11/2022)
Homework 9.2

In Schelling model, agents move at random in a place they are happy. What if agents move according to a law similar to the Gravity model? I.e. what if the probability of going in a cell is inversely proportional to the distance between start and origin cell?

- Modify the proposed Schelling model implementation so as to take into account the distance factor.
- Compare the time of convergence of the Classical Schelling with the Distance Schelling: is a model faster than the other? Try to propose an explanation.
- Submit a (well-commented) notebook