Cellular Automata I

Modelling Complex Systems

Some of this lecture is adapted from: Lectures of Irene Marcovici and previous slides of David Sumpter.

What is a cellular automata (CA)?

A CA consists of an array of cells each with an integer "state".

On each time step a local update rule is applied to the cells. The update rule defines how the particular cell will update its state as a function of its neighbours state.

The CA is run over time and the evolution of the state is observed.



- white = 0, black = 1
- 111 -> 0 110 -> 0 101 -> 0 100 -> 1
 - 011 —> 1 010 —> 1
 - 001 -> 1
 - 000 -> 0



- 2^8 = 256 rules in total
- rule 0
 rule
 255

Classified based on patterns

- Class 1: Fixed; all cells converge to a constant 0 or 1 set
 - Class 2: **Periodic**; repeats the same pattern, like a loop
 - Class 3: Chaotic; pseudo-random

Class 4: **Complex local structures**; exhibits behaviours of both class 2 and class 3; with long lived hard to classify structure

Feels that we understand class 1 - 3, but not 4.

Class 1: <u>Fixed</u>; e.g., rule 8 (00001000) Class 2: Periodic; e.g., rule 50 (00110010) Class 3: Chaotic; e.g., rule 30 (00011110) Class 4: Complex local structures; e.g., rule 110 (01101110)

Class 1: <u>Fixed</u>; e.g., rule 8 (00001000) Class 2: Periodic; e.g., rule 50 (00110010) Class 3: Chaotic; e.g., rule 30 (00011110) Class 4: Complex local structures; e.g., rule 110 (01101110)



More complex cA

CA can be extended:

- 1. More states for single grid
- 2. Longer range interactions
- 3. Two or more dimensions
- 4. Hexagonal or other grids

Formal definition

- We start from a **configuration**.
- All the cells update simultaneously their colour, and choose their new colour in function of the colours they observe in a finite neighbourhood.

If all cells apply simultaneously the same local rule, the update dynamics is called a **cellular automaton**.

Formal definition: cellular automata on inf line

Let \mathcal{A} be a finite set of symbols, called the **alphabet**.

We denote by $\mathcal{A}^{\mathbb{Z}}$ the set of **configurations**. An element of $\mathcal{A}^{\mathbb{Z}}$ is a sequence $(x_k)_{k \in \mathbb{Z}}$, with $x_k \in \mathcal{A}$ for $k \in \mathbb{Z}$.

Definition

 $\mathcal{A} = \{\Box, \blacksquare\}, r = 2$

A map $F : \mathcal{A}^{\mathbb{Z}} \to \mathcal{A}^{\mathbb{Z}}$ is a **cellular automaton** if there exists a **radius** $r \ge 0$ and a **local function** $f : \mathcal{A}^{2r+1} \to \mathcal{A}$ such that:

$$F(x)_k = f(x_{k-r},\ldots,x_{k+r-1},x_{k+r}).$$



Langton's ant

Squares on a plane are coloured either black or white. Starting configuration - identify one square as the "ant" pointing up. The ant can travel in any of the four cardinal directions at each step it takes. The "ant" moves according to the rules below:

- At a white square, turn 90° clockwise, flip the color of the square, move forward one unit
- At a black square, turn 90° counter-clockwise, flip the color of the square, move forward one unit

Langton's ant



Animation: <u>https://www.youtube.com/watch?v=F8-c2bawttU</u>

Formal definition: cellular automata on inf line

Let \mathcal{A} be a finite set of symbols, called the **alphabet**.

We denote by $\mathcal{A}^{\mathbb{Z}}$ the set of **configurations**. An element of $\mathcal{A}^{\mathbb{Z}}$ is a sequence $(x_k)_{k \in \mathbb{Z}}$, with $x_k \in \mathcal{A}$ for $k \in \mathbb{Z}$.

Definition

A map $F : \mathcal{A}^{\mathbb{Z}} \to \mathcal{A}^{\mathbb{Z}}$ is a **cellular automaton** if there exists a **radius** $r \ge 0$ and a **local function** $f : \mathcal{A}^{2r+1} \to \mathcal{A}$ such that:

$$F(x)_k = f(x_{k-r},\ldots,x_{k+r-1},x_{k+r}).$$





Cellular Automata is an eroder if:

- from a configuration with a finite number of blue cells, it reaches the all-white configuration,
- from a configuration with a finite number of white cells, it reaches the all-blue configuration.

Goal: being able to **correct** some finite "mistakes" occuring on a monochromatic configuration.



Try to design a CA which is an eroder in 2D:

IDEA 1: take majority of current state and the up, down, left, right neighbours.



Try to design a CA which is an eroder in 2D:

IDEA 1: take majority of current state and the up, down, left, right neighbours.



This is a fixed point!



Try to design a CA which is an eroder in 2D:

IDEA 2: take majority of current state and the 8 neighbouring cells (including corners).



This erodes :)



Try to design a CA which is an eroder in 2D:

IDEA 2: take majority of current state and the 8 neighbouring cells (including corners).



But this is a fixed point!



Try to design a CA which is an eroder in 2D:

IDEA 3: take majority state of current state and the up, right neighbours.



This cellular automata is an eroder.

- from a configuration with a finite number of blue cells, it reaches the all-white configuration,
- from a configuration with a finite number of white cells, it reaches the all-blue configuration.

Probability Theory Cellular Automata is a classifier if:







p=0.8

Probability Theory Cellular Automata is a classifier if:



On infinite grid. For fixed p<1/2 - with probability 1 evolves to all 0 state. For fixed p>1/2 - with probability 1 evolves to all 1 state

Cellular Automata is a classifier if:

On infinite grid -For fixed p<1/2 - with probability 1 evolves to all 0 state. For fixed p>1/2 - with probability 1 evolves to all 1 state

Do there exist classifiers on 2D grids? (Yes!)

Theorem (Busic - Fates -Mairesse - Marcovici 2013) On infinite 2D grid. The majority CA on cell itself and up and right neighbours is a classifier.



Do there exist classifiers on 1D grids (i.e. infinite line)? (Unknown!) Theorem (Taati 2015)

On infinite 1D grid. There exists a cellular automata: For fixed p<0.0017 - with probability 1 evolves to all 0 state. For fixed p>1-0.0017 - with probability 1 evolves to all 1 state

Open Question: Does there exist a CA which is a classifier in 1D?

- World: 2D orthogonal grid of square cells
- **States**: Dead (0, white) or Alive (1, black)
 - Reproduction: 0 -> 1, if #(Alive neighbours) = 3
 - Surviving: 1 -> 1, if #(Alive neighbours) = 2 or
 3
 - Underpopulation: 1 -> 0, if #(Alive neighbours) < 2
 - Overpopulation: 1 -> 0, if #(Alive neighbours) > 3
 - Otherwise, no change

- Reproduction:
- 0 -> 1, if #(Alive neighbours) = 3
- Surviving:
- 1 -> 1, if #(Alive neighbours) = 2 or 3
- Underpopulation:
- 1 -> 0, if #(Alive neighbours) < 2
- Overpopulation:
- 1 -> 0, if #(Alive neighbours) > 3



Example: Glider







More examples:



https://en.wikipedia.org/wiki/Conway's_Game_o

Large-scale structures

https://vimeo.com/5428232

Computational gates (for interest!)

https://www.youtube.com/watch?v=vGWGeund3eA

Bellos (2015) Alex Through The Looking Glass

Randomness in Update Rules

As well as random initial conditions (e.g. random positions, or each cell beginning in a random state independently), we may have random update rules.

Cellular Automata to model bi-directional pedestrian movement

Configuration: NxN grid with wrap around or side walls. States: empty, up-pedestrian, down-pedestrian

Update Rules (Many parameters!)

- If the square in front of the pedestrian is unoccupied, the pedestrian will always continue forward
- If the square is occupied, there are a number of possibilities:
 - If the left and right squares are both occupied, the pedestrian will take a step back with probability p_b and wait with probability p_w
 - If only the left or the right square is occupied the pedestrian will move to the unoccupied square with probability p_l or p_l , and wait with probability p_w
 - If neither the left or right square is occupied the pedestrian will move to the left or the right square with probability p_l and p_l respectively, and wait with probability p_w

Fang, WF & Yang, LZ & Fan, WC. (2003). Simulation of bi-direction pedestrian movement using a cellular automata model. Physica A: Statistical Mechanics and its Applications. 321. 633-640. Philip Alarik, Graf Brolund

Philip Alarik, Martin Engquist, Alice Graf Brolund

Cellular Automata to model pedestrians

Update Rules (Many parameters!)

- If the square in front of the pedestrian is unoccupied, the pedestrian will always continue forward
- If the square is occupied, there are a number of possibilities:
 - If the left and right squares are both occupied, the pedestrian will take a step back with probability p_b and wait with probability p_w
 - If only the left or the right square is occupied the pedestrian will move to the unoccupied square with probability p_l or p_l , and wait with probability p_w
 - If neither the left or right square is occupied the pedestrian will move to the left or the right square with probability p_l and p_l respectively, and wait with probability p_w

Fang, WF & Yang, LZ & Fan, WC. (2003). Simulation of bi-direction pedestrian movement using a cellular automata model. Physica A: Statistical Mechanics and its Applications. 321. 633-640.

Philip Alarik, Martin Engquist, Alice Graf Brolund

Cellular Automata to model pedestrians Update Rules (Many parameters!)



Fang, WF & Yang, LZ & Fan, WC. (2003). Simulation of bi-direction pedestrian movement using a cellular automata model. Physica A: Statistical Mechanics and its Applications. 321. 633-640.

Philip Alarik, Martin Engquist, Alice Graf Brolund



For Fig a, b, c, d $p_b=0.5$, For Fig e $p_b=0.8$.

Fang, WF & Yang, LZ & Fan, WC. (2003). Simulation of bi-direction pedestrian movement using a cellular automata model. Physica A: Statistical Mechanics and its Applications. 321. 633-640.

Philip Alarik, Martin Engquist, Alice Graf Brolund