Mathematical biology
From individual cell behavior
to biological growth and form

Lecture 4: Cellular automata

Roeland Merks
Mathematical Institute, Leiden University
Institute of Biology, Leiden University
http://biomodel.project.cwi.nl
merksrmh@math.leidenuniv.nl
From single cell behavior to tissue scale phenomena

• “Simple” problem:
  – Growth of avascular tumors *in vitro*

Proliferative cells

Necrotic core

Folkmann, J. Exp. Med. 1973
Tumor growth dynamics

• Unregulated cell division: $\frac{dN}{dt} = rN$
  
  $N(t) = N_0 e^{rt}$ : exponential growth?

• Typical growth curve of tumor: not exponential
  
  - Radius $R(t)$ for exponential growth: $R(t) = \sqrt{\frac{N_0}{\pi}} e^{\frac{1}{2}rt}$

  - In vitro tumors: $R(t) \sim t$
Why don’t tumors grow exponentially?

• Unregulated cell division limited by food supply?
  
  – logistic growth? $\frac{dN}{dt} = rN(1 - \frac{N}{K})$

  – $K = \text{“carrying capacity”} - \text{growth to maximum population size}$

• So, tumors do not follow logistic growth
Why space matters: Eden Growth

- **Spatial** model of colony growth (Eden, 1961)
- Stochastic cellular automata model
- Cells divide into adjacent site if space is available
Eden Growth
Eden Growth
Eden Growth is not exponential!
Eden growth: radius grows linearly

\[ R(t) \approx \sqrt{\frac{N(t)}{\pi}} \]

FIG. 2. Time evolution of the tumors mean radii showing a linear behavior [tumor A (up triangles), tumor B (circles), tumor C (squares), and tumor D (down triangles)]. In the inset we can see the fit for early times.
Eden Growth

• Eden growth produces “rough” (fractal) boundaries
  – Faster invasion and more access to nutrients than smooth boundaries

• In vitro “glioma” also has fractal boundary
Eden growth = cellular automata

- Coupled lattice of finite-state automata
- Finite state automaton

Source: wikipedia
Finite-state automaton

- Formally: \((\Sigma, S, s_0, \delta, F)\)
- \(\Sigma\) is the input alphabet (a finite, non-empty set of symbols).
- \(S\) is a finite, non-empty set of states.
- \(s_0\) is an initial state, an element of \(S\).
- \(\delta\) is the state-transition function: \(\delta : S \times \Sigma \rightarrow S\)
- \(F\) is the set of final states, a (possibly empty) subset of \(S\)
Example: even number of 0’s?

\[ \Sigma = \{0,1\} \]
\[ S = \{S_1, S_2\} \]
\[ s_0 = S_1 \]
\[ \delta = \ldots \]
\[ F = S_1 \]
Deterministic cellular automata

- Lattice of FSAs
- Adjacent sites ("cells") are coupled to one another
- $\Sigma = S$, updates a fixed clock ticks
- *Neighborhood*: what lattice sites are adjacent?
  - Von Neumann
  - Moore
Typical CA rules

- $\Sigma = S = \{0, 1\}$
- **Von Neumann Neighborhood**
  - $\delta : 0 \rightarrow 1$, if $\geq 1$ neighbor is 1
  - $\delta : 0 \rightarrow 0$, if 0 neighbors are 0
  - $\delta : 1 \rightarrow 1$
Typical CA rules
Conway’s Game of Life

• $\Sigma = S = \{0, 1\}$

• Moore Neighborhood

  $\delta : 0 \rightarrow 1$, if 3 neighbors are 1
  $\delta : 1 \rightarrow 1$, if 2 or 3 neighbors are 1
  $\delta : 1 \rightarrow 0$, if $\geq 4$ neighbors are 1
  $\delta : 1 \rightarrow 0$, if $< 2$ neighbors are 1
  $\delta : \{0, 1\} \rightarrow 0$, otherwise

• “Game of Life”
Conway’s Game of Life
Typical CA rules
Typical CA rules
Typical CA rules
Typical CA rules
Conway’s Game of Life
Classification of cellular automata
Wolfram, 1981

- 1D cellular automata, left and right neighbors

- How many possible rules?
  - e.g., \{0,0,0\}→0, \{0,1,0\}→1, \{1,1,0\}→0
Rules
“Wolfram classes”

- Class 1: Converges onto uniform state
- Class 2: Converges onto repetitive or stable state
- Class 3: Converges onto “random” state
- Class 4: ....
Universal computation in class 4

- Localized patterns in background pattern
  (00010011011111)
Patterns interact with one another

- Particle interactions can form logical gates

Just passing through... Interacting.. forms third structure
Universal computing in class IV cellular automata: ‘tag system’

• Possible to construct ‘cyclic tag system’ -> universal computation (Cook 1985 / 2005)
• $n$-tag system: remove $n$ symbols from front
• Add symbol to end based on ‘production rules’
• Halt if symbol ‘$H$’ is encountered

2-tag system
Alphabet: \{a,b,c,H\}
Production rules:
  a --> ccbah
  b --> cca
  c --> cc

Computation
Initial word: baa
  acca
  caccbaH
  ccbaHcc
  baHcccc
  Hccccca (halt).
‘cyclic tag system’

- Symbols: \{0,1\}
- Delete first symbol:
  - if ‘1’ add next “production” to end
  - If ‘0’ add nothing

- Can make ‘cyclic tag system’ from particles in class IV CA.
- Cyclic tag system is capable of universal computation!
2D CA: Voting Rule

- Majority rule:
  - \( \Sigma = S = \{0,1\} \), Moore Neighborhood
  - \( \delta : \{0,1\} \rightarrow 1 \), if majority of nbs is 1
  - \( \delta : \{0,1\} \rightarrow 0 \), if majority of nbs is 0
  - \( \delta : x \rightarrow x \), otherwise

- Initial condition: \( P(0)=0.5, P(1)=0.5 \)
- Outcome?
2D CA: Voting Rule

$t=0$

$t=1$

$t=2$
2D CA: Voting Rule
2D CA: Voting Rule

$t=10$

$t=20$

$t=30$
Probabilistic cellular automata

- Synchronous updating
- Transition rules are probabilistic
- Example:
  - Von Neumann Neighborhood

\[ \Sigma = S = \{0,1\} \]

- \( \delta : 0 \rightarrow 1, \text{ if } \geq 1 \text{ neighbor is } 1, \text{ with } P = p, \text{ else } 0 \rightarrow 0 \)
- \( \delta : 0 \rightarrow 0, \text{ if } 0 \text{ neighbors are } 0 \)
- \( \delta : 1 \rightarrow 1 \)
Probabilistic voting rule

• Update probability:
  – \( P(\text{update}) = p \)
  – \( P(\text{flip}) = 1 - p \)

• Any changes to the system?

  \[ p = 0.999 \]
Probabilistic voting rule

• Update probability:
  – $P(\text{update}) = \rho$
  – $P(\text{flip}) = 1 - \rho$

• $\rho = 0.99$
Probabilistic voting rule

• Update probability
  – $P(\text{update}) = p$
  – $P(\text{flip}) = 1 - p$
• $p = 0.9$
Probabilistic voting rule

• Update probability:
  – $P(\text{update}) = p$
  – $P(\text{flip}) = 1 - p$

• $p = 0.7$?
Probabilistic voting rule

• Update probability:
  – $P(update)=p$
  – $P(flip)=1-p$

• $p=0.5$?
Probabilistic voting rule

• Update probability:
  – $P(\text{update}) = p$
  – $P(\text{flip}) = 1 - p$

• $p = 0.1$ ?
Asynchronous updating

- Update probability:
  - \( P(\text{update}) = 1 \)
  - \( P(\text{flip}) = 0 \)
- Update one site at a time
Synchronicity vs. asynchronicity

$p=0.9$
Synchronicity vs. asynchronicity

$p=0.5$