Lecture 6: Cellular Automata Dynamics

Complex Systems 530
2/11/20

How to explore the space of CA behaviors?

- For simple models, we can examine the **phase space**
- Phase space is the space (in this case a network) of all possible states of the model

CA phase space

- How many different state configurations can we have?
    - D = number of dimensions (1, 2, 3, etc.)
    - L = length in each dimension (number of cells)
    - r = neighborhood radius (how many cells out to consider)
    - k = number of states (binary, more?)

How many different configurations can we have?

- Total cells in the space: L^D
- Each cell can be in one of k states
- Total possible configurations for the system: k^{L^D}
- E.g., a 2D 10x10 binary CA has $2^{10^2} = 1,048,576$ possible configurations

CA rule space

- How many different rules (CAs) can we have?

- Total cells in neighborhood (including self):

$$(2r+1)^D$$

- Total possible configurations for a single neighborhood (termed situations):

$$k^{(2r+1)^D}$$

- For each situation we map to a resulting state, so total possible rules (CAs) is:

$$k^{k^{(2r+1)^D}}$$

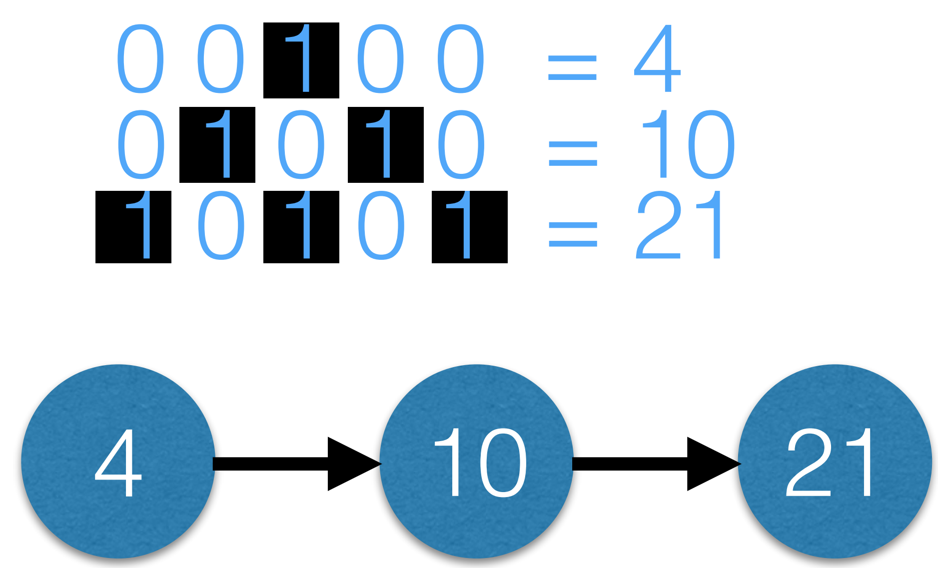
Very big!

Phase space

- Phase space is the space of all possible states of the model—for CA this is discrete, and finite if we have a finite domain
- We can map how one configuration of the model moves to another—forms a network
- Phase space comes from the analogous idea for continuous dynamical systems—there we have a continuous flow from one state to another, for CA we have a directed network

Phase space

- How to map the network of transitions between states?
- We can translate a configuration of space into a binary number, and use this to label each space
- Connect edges from each configuration to the next as we step through time



Phase Space

- We can use the network structure to understand the dynamics of CAs
- Gets tricky for larger grid spaces—many more nodes in the network
- Many of the usual approaches for understanding networks can be used to examine dynamics (cycles, connectedness, etc.)
- Similar to state transition diagram/matrix for Markov models

Phase Space Example

- Binary 1D CA, neighborhood radius 2
- 9 cells in ring arrangement (wrapped boundary)
- ‘Majority rule’
- Total possible configurations = 2^9 = 512

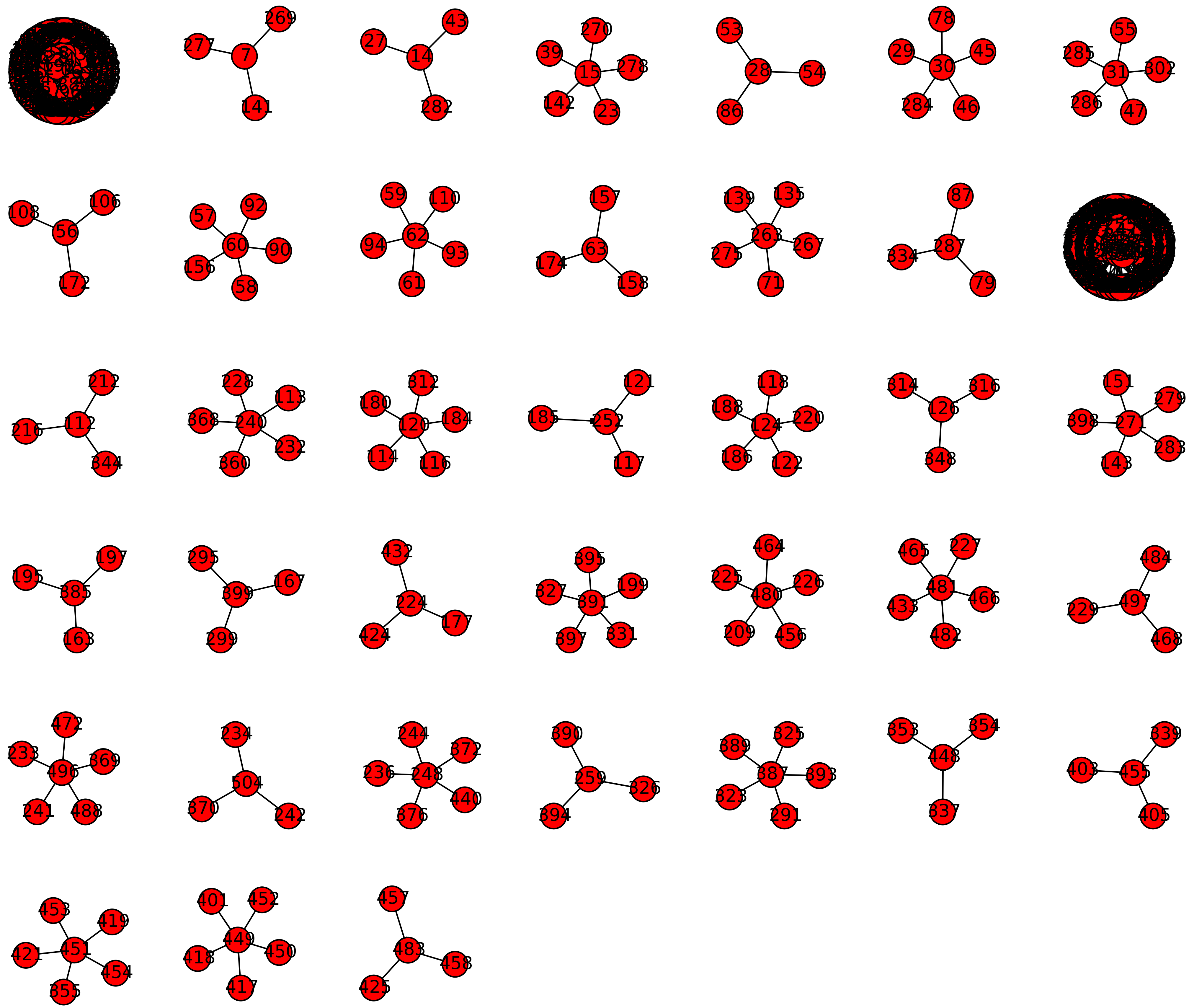


Figure 12.1: Graph-based phase space of the 1-D binary CA model with the majority rule ($r = 2$, $L = 9$) drawn with Code 12.3.

Phase Space Example

- Many different basins of attraction, i.e. network components

- 2 larger basins of attraction—explore with PyCX code

- What is structure overall? What does it look like the majority rule model will do?

- Explore together

Phase Space

- For larger grid sizes, can be much more complicated, networks can become hairball-like
- Some dynamic patterns run for a long time before stabilizing, e.g. the ‘rabbit’ in Game of Life takes 17,331 steps to stabilize (a very long path in the phase space network)

Phase space exploration

- Code phase space for several 1D CA using example code
- Explore together
- Look for:
  - Attracting subsets, cycles, gardens of eden
  - What do these correspond to dynamically?

Mean-field approximation

- As CA get more complicated, direct examination of phase space becomes more challenging
- Mean field approximations give one way to understand the dynamics in a very(!) rough way
- Mean field approximation describes the overall average state of the system over time (i.e. how many on/off cells on average)
- Much lower dimension—but also loses most of what makes CA interesting?

Mean-field approximation

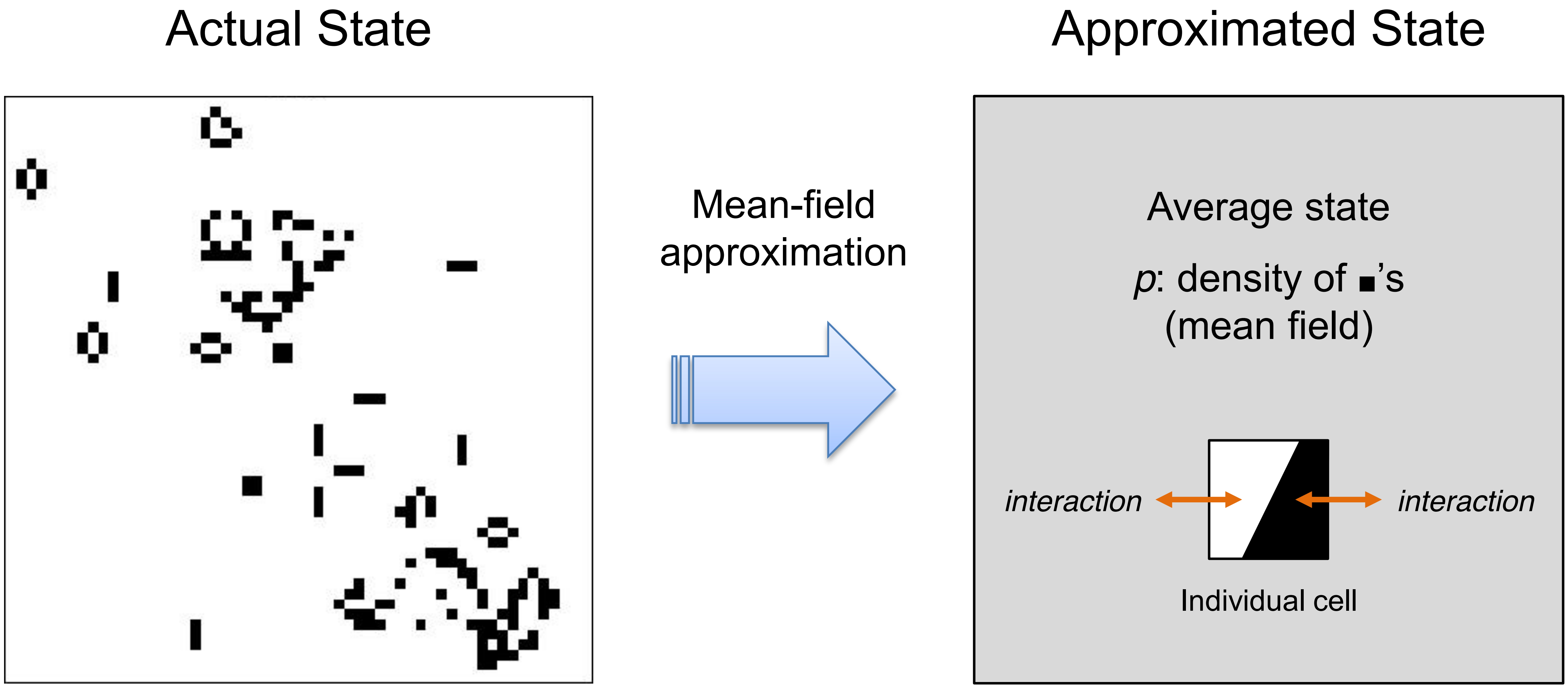


Figure 12.2: Basic idea of the mean-field approximation.

Mean-field approximation

- Consider a 2D binary CA with majority rule
- Let p_t be the density of 1's (on state) in the grid at time t
- We can treat the system probabilistically—work out the probability that a cell would transition on/off given the rules, with no particular knowledge of the exact actual configuration of any given cell

Mean-field approximation

Table 12.1: Possible scenarios of state transitions for binary CA with the majority rule.

Current state	Neighbors' states	Next state	Probability of this transition
0	Four 1's or fewer	0	$(1-p)\sum_{k=0}^{4}\binom{8}{k}p^k(1-p)^{(8-k)}$
0	Five 1's or more	1	$(1-p)\sum_{k=5}^{8}\binom{8}{k}p^k(1-p)^{(8-k)}$
1	Three 1's or fewer	0	$p\sum_{k=0}^{3}\binom{8}{k}p^k(1-p)^{(8-k)}$
1	Four 1's or more	1	$p\sum_{k=4}^{8}\binom{8}{k}p^k(1-p)^{(8-k)}$

- p(state) x p(neighbors' states)

Sayama p. 217 (Chp. 12)

Mean-field approximation

Table 12.1: Possible scenarios of state transitions for binary CA with the majority rule.

Current state	Neighbors' states	Next state	Probability of this transition
0	Four 1's or fewer	0	$(1-p)\sum_{k=0}^{4}\binom{8}{k}p^k(1-p)^{(8-k)}$
0	Five 1's or more	1	$(1-p)\sum_{k=5}^{8}\binom{8}{k}p^k(1-p)^{(8-k)}$
1	Three 1's or fewer	0	$p\sum_{k=0}^{3}\binom{8}{k}p^k(1-p)^{(8-k)}$
1	Four 1's or more	1	$p\sum_{k=4}^{8}\binom{8}{k}p^k(1-p)^{(8-k)}$

- p_{t+1} = p(state is a 1 at next time step)

Sayama p. 217 (Chp. 12)

Mean-field approximation

$$
\begin{aligned}
p_{t+1} &= (1-p)\sum_{k=5}^{8}\binom{8}{k}p^k(1-p)^{(8-k)} + p\sum_{k=4}^{8}\binom{8}{k}p^k(1-p)^{(8-k)} \\
&= \sum_{k=5}^{8}\binom{8}{k}p^k(1-p)^{(8-k)} + p\binom{8}{4}p^4(1-p)^4 \\
&= \binom{8}{5}p^5(1-p)^3 + \binom{8}{6}p^6(1-p)^2 + \binom{8}{7}p^7(1-p) + \binom{8}{8}p^8 + 70p^5(1-p)^4 \\
&= 70p^9 - 315p^8 + 540p^7 - 420p^6 + 126p^5
\end{aligned}
$$

Mean-field approximation

- Gives us a simple, 1-dimensional difference equation that we can use to track the overall probability/density of 1's vs. 0's in the system
- Can determine p_0 from initial conditions and then simulate forward

Cobweb plot

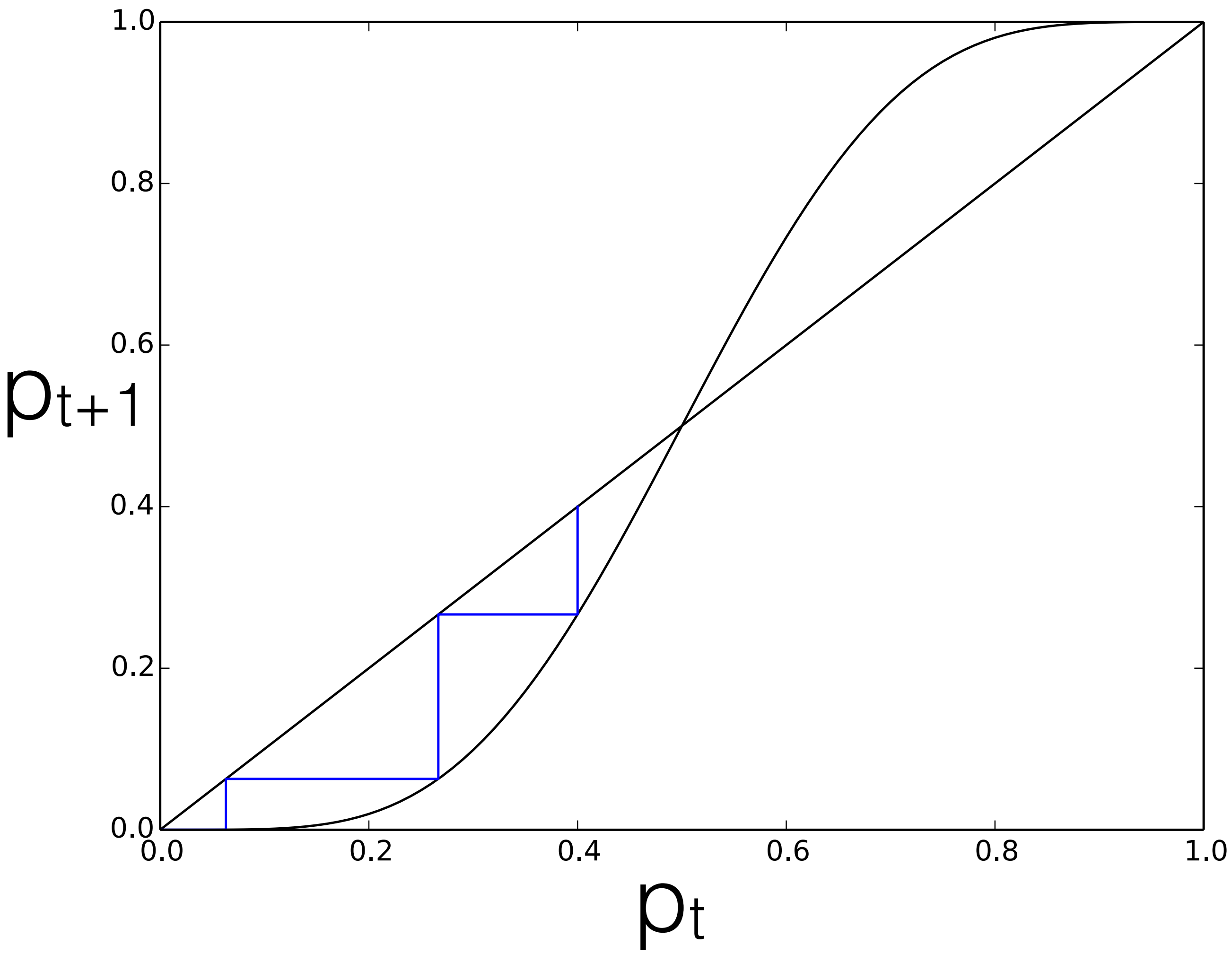


Figure 12.3: Cobweb plot of Eq. (12.10).

- Plots current value vs next value
- Straight line of y = x
- Model function plotted as the curve,

$$p_{t+1} = 70p^9 - 315p^8 + 540p^7 - 420p^6 + 126p^5$$

- Where these two intersect, we have an equilibrium point!

Cobweb plot

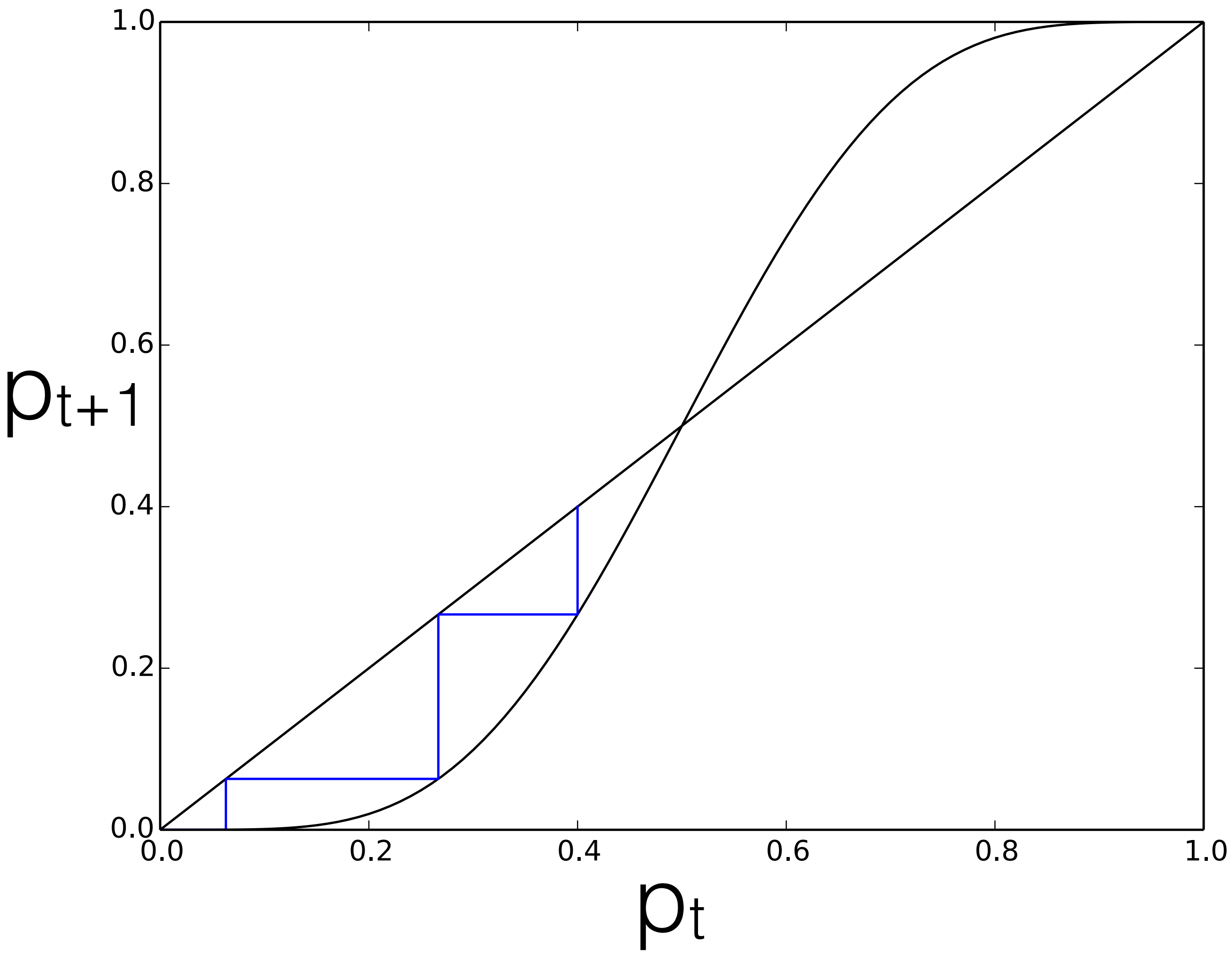


Figure 12.3: Cobweb plot of Eq. (12.10).

- In this case, the cobweb plot shows 3 equilibria
  - All 0 - stable
  - All 1 - stable
  - Half-and-half - unstable
- How true is this to the real CA? Why?

Mean-field approximation

- Does not account for spatial features of the system!
  - It will necessarily be very approximate and represent only the “average” behavior of the system assuming all cells experience a homogeneous ‘neighborhood’
- Is this a good approximation for most CA?
- See also the renormalization group approach for percolation (Sayama Chapter 12)

Extensions to CA

- **Stochastic (probabilistic) CA** - state transitions happen with some probability based on neighboring states (cf. Markov chains)
- **Multi-layer CA** - state values as vectors, e.g. may capture multiple properties or attributes of the agent, or different agents living on the same cell
- **Asynchronous CA** - updates non-simultaneously (e.g. random, ordered, state-triggered)

A note about spaceships & other structures

- Many spaceships and other stable patterns in CA
- An interesting question of whether these are “real”?
- The CA is made of cells, they do all the operations of the model
- The patterns we observe are not actual objects—just persistent patterns that we name treat as separate entities

A note about spaceships and other structures

- Although, this can be said of a lot of things? (E.g. storms, maybe even people?)

- Doesn’t necessarily make the objects in CAs less real because they are composed of cells

For next time…

- Reading
  - Sayama Chapter 12
  - Think Complexity Chapter 7