Boolean Network Models

Sarah Maddox

January 22, 2018

Adapted from http://web.cs.ucdavis.edu/~filkov/classes/289a-W03/l10.pdf

Boolean Networks

Boolean network: a graph G(V,E), annotated with a set of states $X=\{x_i \mid i=1,\ldots,n\}$, together with a set of Boolean functions $B=\{b_i \mid i=1,\ldots,k\}$, $b_i : \{0,1\}^k \rightarrow \{0,1\}$. What??

Gate: Each node, v_i, has associated to it a function , with inputs the states of the nodes connected to v_i.

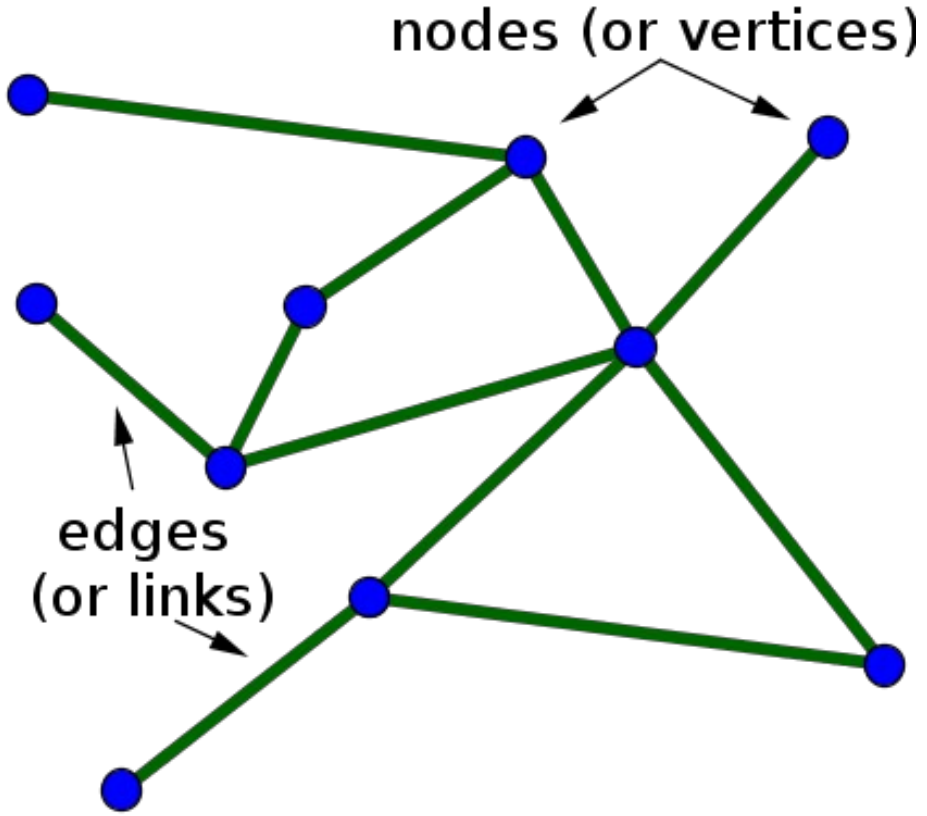
Dynamics: The state of node v_i at time t is denoted as $x_i(t)$. Then, the state of that node at time t+1 is given by:

$$x_i(t+1) = b_i(x_{i1}, x_{i2}, \ldots, x_{ik})$$

where x_{ij} are the states of the nodes connected to v_i.

Brief introduction to graphs

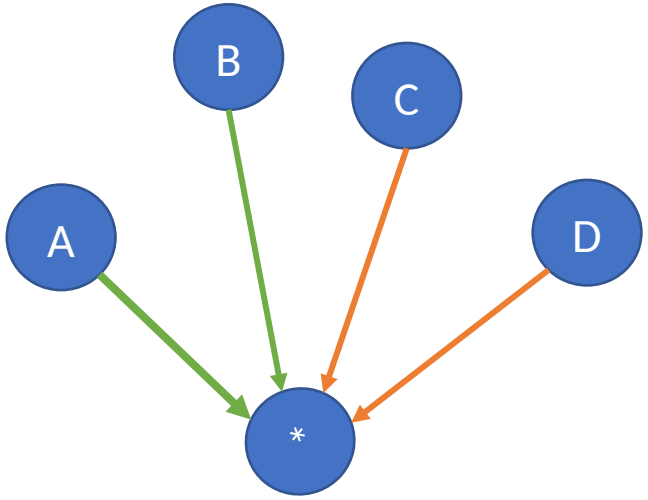
- Nodes (in our case, genes/proteins) connected by edges (in our case, rules of interaction)
- Nodes have associated Boolean rules in a **Boolean network** based on input nodes
- Our networks will be **directed**



Boolean Functions

- True, False: 1,0
- Boolean Variables: x can be true or false
- Logical Operators: and, or, not
- Boolean Functions: k input Boolean variables, connected by logical operators, 1 output Boolean value
- Total number, B, of Boolean functions of k variables: 2^{2^k} (k =1, B=4; k=2, B=16; etc.)

Sample rule for * activation:

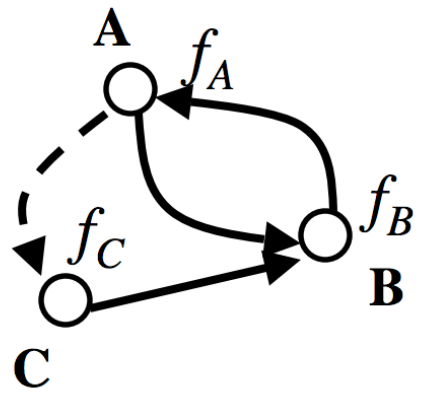


A and B are activating
C and D are inactivating (NOT)

Possible Rules:
(A and B) and (not C or not D)
A or B or not C or not D
(A or B) and not C and not D
…

Wiring Diagrams and Truth Tables

Rules are functions of INPUT nodes

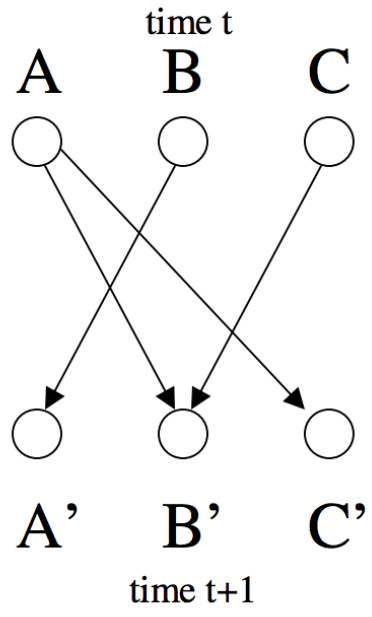


Rule for A $f_A(B) = B$

Rule for B $f_B(A, C) = A \text{ and } C$

Rule for C $f_C(A) = \text{not } A$

Boolean Network



Wiring Diagram

State	INPUT			OUTPUT		
	A	B	C	A'	B'	C'
1	0	0	0			
2	0	0	1			
3	0	1	0			
4	0	1	1			
5	1	0	0			
6	1	0	1			
7	1	1	0			
8	1	1	1			

Truth Table

How do we use Boolean networks to **figure out the rules of interaction**?

- Reverse engineering
  - Fitting the data: given observations of the states of the BN, find the truth table
  - States are {0, 1} for each node
  - Black box model: several (input, output) pairs: observations of the states before and after a transition
- In general, many possible networks may be found that fit the data
- How to decide?
  - Different algorithms

  available to choose best rules

For each node v_i

– For each pair of nodes v_k and v_h and

- For each Boolean function f of 2 variables (16 poss.)

– Check if $O_j(v_i)=f(I_j(v_k),I_j(v_h))$ holds for all j.

Formal Problem

- An example is a pair of observations (I_j, O_j).
- A node is consistent with an example, if there is a Boolean function such that $O_j = f(I_j)$
- A **BN** is consistent with (I_j, O_j) if all nodes are consistent with that example. Similarly, a **BN** is consistent with $EX = \{(I_1, O_1), \ldots, (I_m, O_m)\}$ if it is consistent with each example
- Problem: **Given EX, n the number of nodes in the network, and k (constant) the max indegree of a node, find a BN consistent with the data.**

Nodes, States, Transitions,…

- States: Values of all variables at given time
- Values updated synchronously
- Input → Output

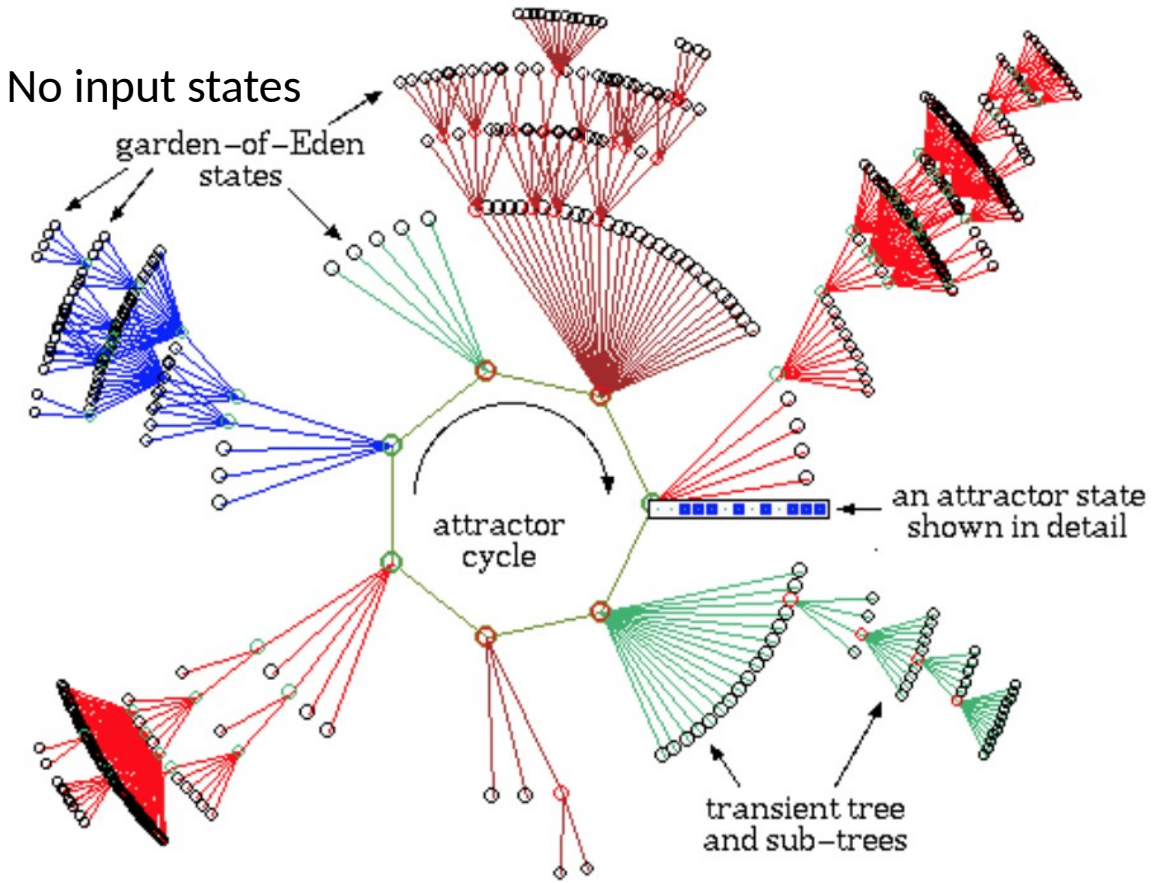
State	INPUT			OUTPUT		
	A	B	C	A’	B’	C’
1	0	0	0	0	0	1
2	0	0	1	0	0	1
3	0	1	0	1	0	1
4	0	1	1	1	0	1
5	1	0	0	0	0	0
6	1	0	1	0	1	0
7	1	1	0	1	0	0
8	1	1	1	1	1	0

- Ex. (100 → 

Get stuck in this state = attractor

BN Dynamics

- Trajectories: series of state transitions
- Attractors: repeating trajectories
- Basin of Attraction: all states leading to an attractor



One attractor basin for a BN n=13, k=3. The cycle is of size 7

Wuensch, PSB 1998

Why Boolean Networks?

- Can describe complex behavior
  - Steady states of network are biologically relevant
  - Stability and reproducibility
- Range of behaviors of system can be completely known and analyzed (for smaller networks; larger networks are computationally limiting)
  - Assume the things in the network are either off or on (0 or 1)-– no in between behaviors
- Simple to implement and use
  - Cell collective, Cytoscape

Boolean Network Transitions and Biology

- Example: Microarrays (gene expression data) quantify transcription on a large scale
- Infer network of regulation based on transcription data alone (is this realistic? Possible?)
- Discretized gene expression (or fuzzy logic!!) can be used to build rules about states of network
- Wiring and Boolean functions are **both** reverse engineered from the microarray data
  - How can we figure out the wiring? The direction of nodes?

BN and Biology, Cont'd.

From mRNA measures to a Regulation Network:

1 *Continuous gene expression values are discretized as being 0 or 1 (on, off), (each microarray is a binary vector of the states of the genes);*

2 *Successive measurements (arrays) represent successive states of the network i.e. X(t)->X(t+1)->X(t+2)…*

3 *A BN is reverse engineered from the input/output pairs: (X(t),X(t+1)), (X(t+1),X(t+2)), etc.*

Summary

- Boolean networks are the simplest models (discrete, simultaneous, deterministic updates)
- Still plausibly complex and describe important features of real world networks
- Can be reverse engineered from a relatively small amount of data
- Limitations
  - Boolean, obviously → potential solution = fuzzy logic?
  - Simplest version doesn’t include stochasticity
  - Only small networks are computationally feasible