Boolean Network Models

2/5/03

History

- Kaufmann, 1970s
- Studied organization and dynamics properties of (N,k) Boolean Networks
- Found out that highly connected networks behave differently than lowly connected ones
- Similarity with biological systems: they are usually lowly connected
- We study Boolean Networks as a model that yields interesting complexity of organization and leave out the philosophical context

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 $B=\{b_i \mid i=1,\ldots,k\}$, $b_i : \{0,1\}^k \rightarrow \{0,1\}$**.**

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.

General Properties of BN:

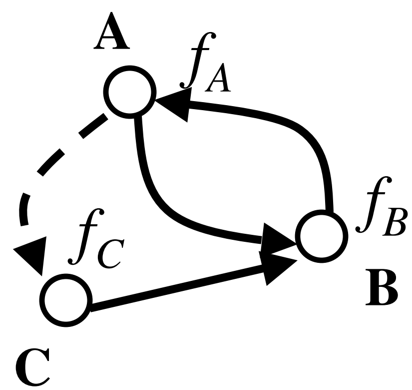
- Fixed Topology (doesn’t change with time)
- Dynamic
- Synchronous
- Node States: Deterministic, discrete (binary)
- Gate Function: Boolean
- Flow: Information

What kind of properties do they exhibit?

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.)

Wiring Diagrams and Truth Tables

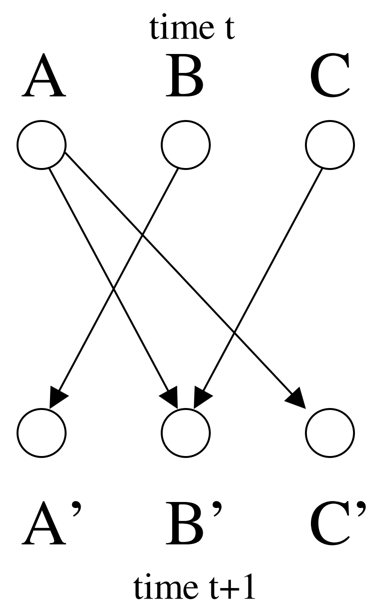


$f_A(B) = B$

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

$f_C(A) = \text{not } A$

Boolean Network



Wiring Diagram

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

Truth Table

Reverse Engineering of BNs

- Fitting the data: given observations of the states of the BN, find the truth table
- In general, many networks will be found
- Available algorithms:
  - Akutsu et al.
  - Liang et al. (REVEAL)

Fitting the Data

- The black box model: m (input, output) pairs
- Each pair is the observations of the states of a system before and after a transitions
- The states are 0,1

Formal Problem

- An <u>example</u> is a pair of observations (I_j, O_j).
- A node is <u>consistent with an example</u>, if there is a Boolean function such that $O_j = f(I_j)$
- A **BN** is <u>consistent with (I_j, O_j)</u> if all nodes are consistent with that example. Similarly, a **BN** is consistent with $\boldsymbol{EX} = \{(I_1, O_1), \ldots, (I_m, O_m)\}$ if it is consistent with each example
- <u>Problem</u>: **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**.

Algorithm (Akutsu et al, 1999)

The following algorithm is for the case of $k=2$, for illustration purposes. It can easily be extended to cases where $k>2$

- For each node v_i
  - For each pair of nodes v_k and v_h and
    - For each Boolean function $\boldsymbol{f}$ of 2 variables (16 poss.)
      - Check if $\boldsymbol{O_j(v_i)=f(I_j(v_k),I_j(v_h))}$ holds for all j.

Analysis of the Algorithm

- Correctness: Exhaustive
- Time: Examine all Boolean functions of 2 inputs, for all node triplets, and all examples

$$O(2 \cdot 2^{2^2} \cdot n^3 \cdot m)$$

- For k inputs (k in front is the 2 above, time to access the k input observations)

$$O(k \cdot 2^{2^k} \cdot n^{k+1} \cdot m)$$

- This is polynomial in n, if k is constant.

Better Algorithms?

- If indegree is fixed to at most k,
  - the best known deterministic algorithms run in $O(mn^k)$ time
  - Randomized: $O(m^{w-2}n^k+mn^{k+w-3})$, where w is the exponent in matrix multiplication, currently $w<2.376$ (Akutsu et al., 2000)
- If indegree is close to n, the problem is NP-complete (Akutsu et al., 2000)

Data Requirement

- How many examples (I,O) do we need to reconstruct a Boolean Network?
- If indegree unbounded 2^n
- If indegree<k, information theoretic aruments yield the following bounds:
  – Upper bound $O(2^{2k} \cdot (2k + \alpha) \cdot \log n)$
  – Lower bound $\Omega(2^k + K \log n)$
- Experiments show that the constant in front of the log n is somewhere in between, i.e. $k2^k$

Boolean Network Dynamics

Nodes, States, Transitions,…

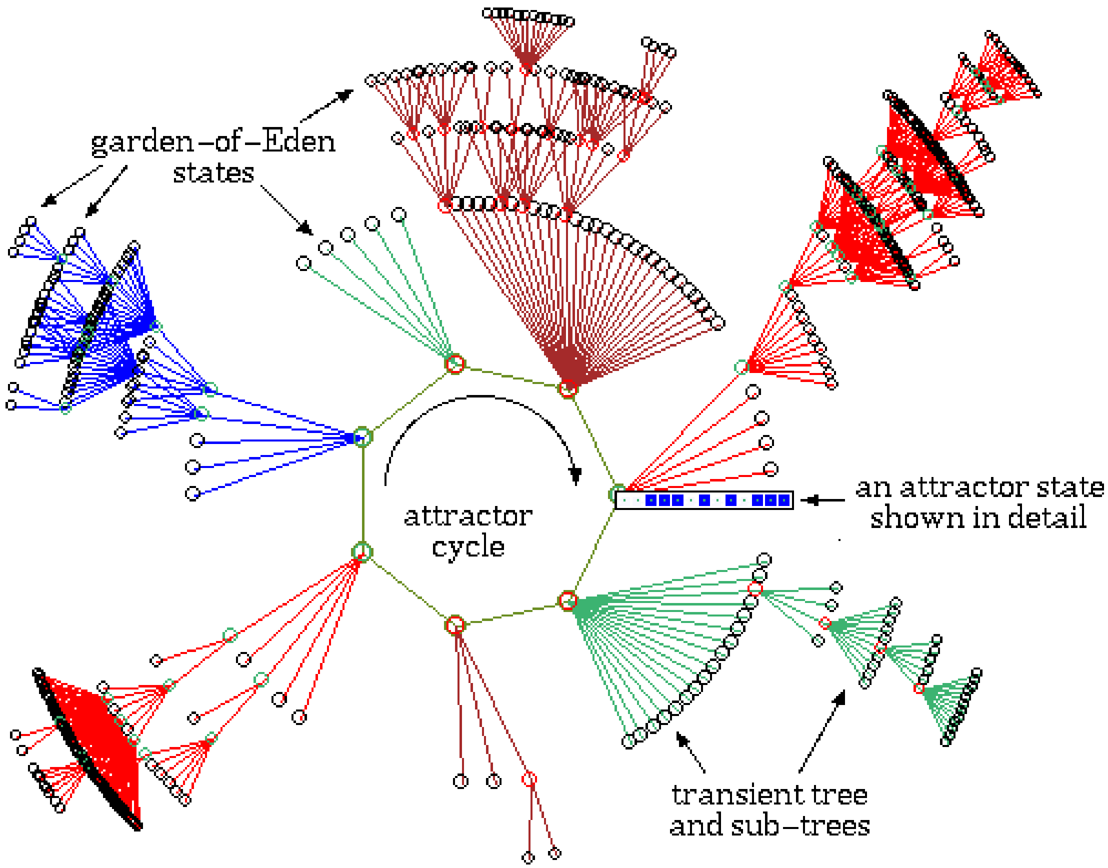
- States: Values of all variables at given time
- Values updated synchronously
- Input → Output

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 → 000 → 001 → 001 . . .)

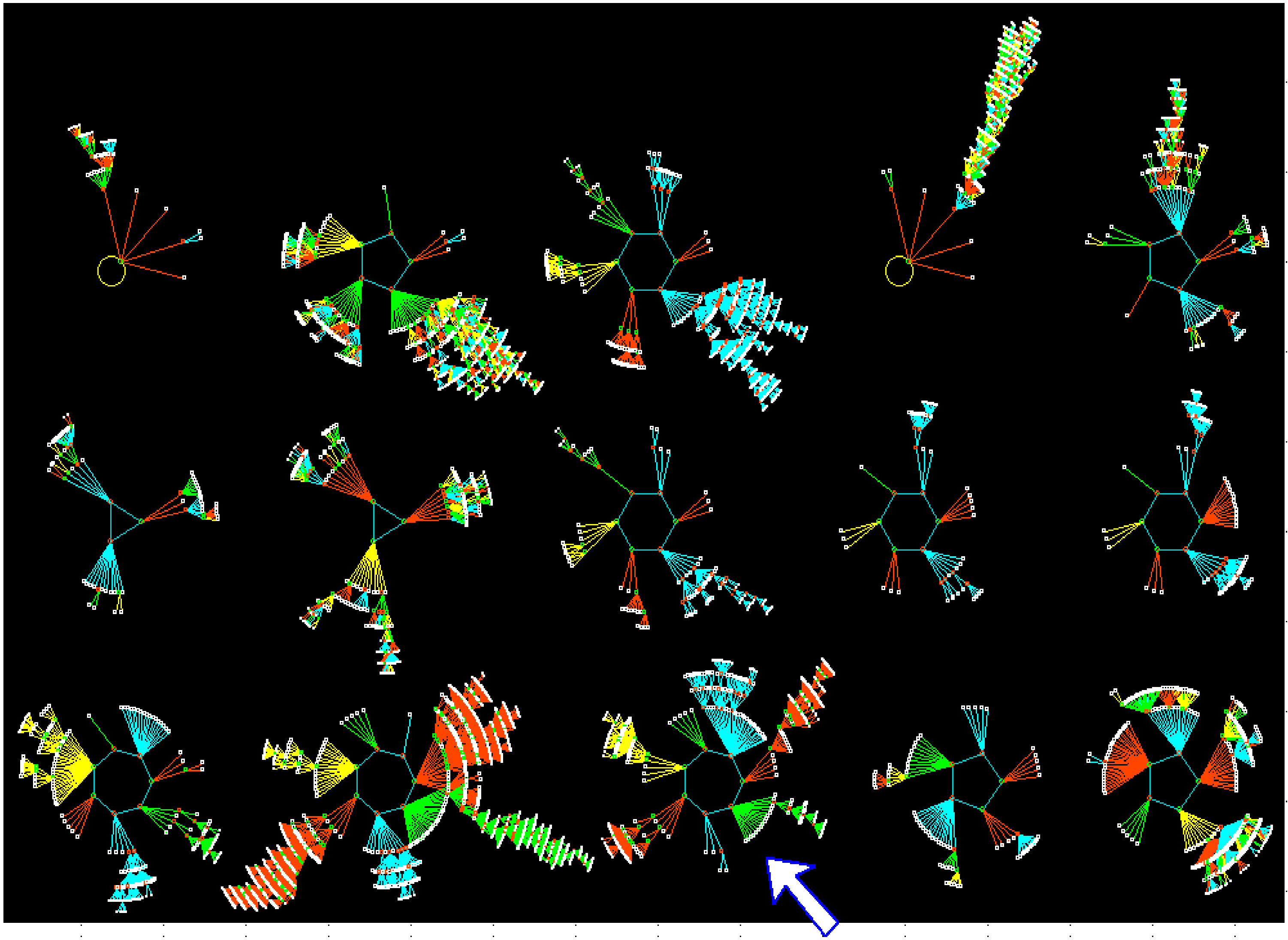
BN Dynamics

- 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



Previous basin of attraction is one of 15 possible ones for n=13 and k=3. Total of 8192 states, and attractors with periods ranging from 1 to 7 (Pictures come from DDLab Galery, Wuensche, Santa Fe Institute)

Why Are BNs Good for Biology?

- Complex behavior (synergistic behavior)
  - Attractor steady states which can be interpreted as memory for the cell
  - Stability and reproducibility
  - Robustness
- The range of behaviors of the system is completely known and analyzable (for smaller networks) and is much smaller than the range of the individual variables
- Organizational properties:
  - high connectivity ($k>5$) yields chaotic behavior
  - Low connectivity ($k=2$) attractor number and median attractor length are *O(Sqrt(n))*
- Simple to implement and use

BN and Biology

Microarrays quantify transcription on a large scale.

The idea is to infer a regulation network based solely on transcription data.

Discretized gene expressions can be used as descriptors of the states of a BN. The wiring and the Boolean functions are reverse engineered from the microarray data.

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.*

Limitations

- BNs are Boolean! Very discrete
- Updates are synchronous
- Only small nets can be reverse engineered with current state-of-the-art algorithms

Summary

- BN are the simplest models that offer plausible real network complexity
- Can be reverse engineered from a small number of experiments $O(\log n)$ if the connectivity is bounded by a constant. 2^n experiments needed if connectivity is high
- Algorithms for reverse engineering are polynomial in the degree of connectivity

References

- Akutsu et al., *Identification of Genetic Networks From a Small Number of Gene Expression Patterns Under the Boolean Network Model*, Pacific Symposium on Biocomputing, 1999
- Akutsu et al., *Algorithms for Identifying Boolean Networks and Related Biological Networks Based on Matrix Multiplication and Fingerprint Function*, RECOMB 00, 2000
- Liang et al., *REVEAL, A General Reverse Engineering Algorithm for Inference of Genetic Network Architectures*, Pacific Symposium on Biocomputing, 1998
- Wuensche, *Genomic Regulation Modeled as a Network With Basins of Attraction*, Pacific Symposium on Biocomputing, 1998
- D'Haeseleer et al., Tutorial on Gene Expression, Data Analysis, and Modeling, PSB, 1999