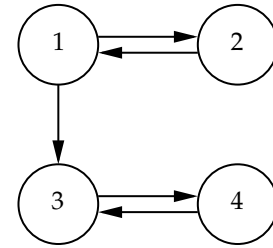


Influence of update schemes on boolean networks

Tom Zuidberg
455969

Abstract—In this paper we will introduce boolean networks (BN) and their relevance to gene regulatory networks (GRN). We will have a closer look on update schemes. Specifically, synchronous, sequential and asynchronous update schemes and their effect on the behavior of BN and GRN respectively.

will be change once introduction and conclusion are done



$$\begin{aligned} f_1 &= \neg x_2 \\ f_2 &= x_1 \\ f_3 &= x_1 \oplus x_4 \\ f_4 &= x_3 \end{aligned}$$

Fig. 1. Example of a boolean network with four nodes. Each vertex indicates that a node is part of the update function of the other node. For instance x_1 is part of f_2 and f_3 . Note that $\neg$ means the logical NOT and $\oplus$ means the logical XOR function.

1 INTRODUCTION

Boolean networks were first introduced by Stuart Kauffman in the late 1960s as a simplified model of biochemical systems [5]. In this framework, genes or regulatory components are represented as binary variables whose states evolve according to logical functions encoding regulatory interactions. Despite their simplicity, boolean networks have proven useful in capturing qualitative dynamical behavior in complex biochemical systems and have since become a standard tool of modeling in this field.

A key abstraction in Boolean network modeling is the representation of dynamics via a state graph, where nodes correspond to network states and directed edges represent transitions induced by update functions. The long term behavior of the network is typically described in terms of attractors in this graph, which are often interpreted as stable cellular or functional states [5].

When it comes to modeling a system by a boolean network, the choice of the update scheme takes an important role, as it defines how and when individual parts of the network change. Some of the most commonly used schemes include the synchronous scheme, where all nodes update synchronously; asynchronous scheme, where only individual nodes are updated; sequential and probabilistic which introduces randomness into the system.

The choice of the update scheme will lead to a different behavior, such as other attractors and cycles, a change in reachability and different robustness with regards to perturbations. In the following, we will introduce different update schemes and their respective effects on boolean networks illustrated on examples and find the best use case for each scheme.

use more references!

2 BOOLEAN NETWORKS

A boolean network consists of nodes $x_i(t)$ that have a boolean state, either 0 or 1, at a point in time t . Each node has a corresponding update function

$$x_i(t+1) = f_i(x_1(t), x_2(t), \dots, x_n(t))$$

expressing the new state of $x_i(t+1)$. The state of the boolean network can be describe as a boolean number $x(t) = x_1x_2 \dots x_n$ where each node is replaced with the corresponding state 0 or 1.

Consider the boolean network shown in fig. 1. For example, if the state is $x(t) = 0011$ at time t , meaning that x_1, x_2 are 0 and x_3, x_4 are 1. After applying the synchronous update, the next state $x(t+1)$ is 1011. Updating the network multiple times creates a trajectory, which is the sequence of the states starting from the initial state. With our example, the trajectory is $0011 \rightarrow 1011 \rightarrow 1101 \rightarrow 0100 \rightarrow \dots$. Each trajectory in a reasonably small boolean network eventually reaches one of two scenarios. Either it falls into a state, called attractor, that doesn't change when updated, or it falls into a cycle of states.

A state graph shows all possible states of the network with the corresponding trajectory. For example, the state

$x(t)$	$x(t+1)$
0000	1000
0001	1010
0010	1001
0011	1011
0100	0000
0101	0010
0110	0001
0111	0011
1000	1110
1001	1100
1010	1111
1011	1101
1100	0110
1101	0100
1110	0111
1111	0101

Fig. 2. State transitions for all states of the boolean network from fig. 1 using synchronous update scheme once.

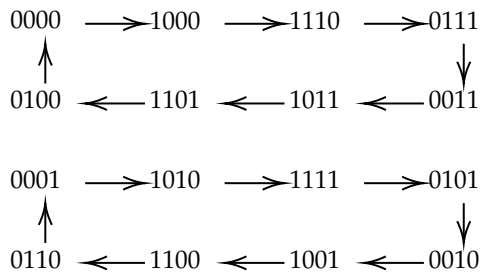


Fig. 3. State graph of the boolean network shown in fig. 1 using synchronous update scheme.

graph in fig. 3 shows the trajectories from the boolean network in fig. 1. These state graphs help visualizing the attractors and cycles, e.g. 2 cycles of length 8.

In boolean networks, robustness refers to stability of the behavior with regards to perturbations, e.g. a single node state change. A boolean network with poor robustness means that a simple flip of a node can change the resulting attractor, respectively cycle for that trajectory. In regards to our example from fig. 3, if either x_3 or x_4 (not both at once) is flipped, the cycle switches. A strong robustness however would mean that even with such perturbations, the resulting attractor or cycle doesn't change.

currently feels to short. once introduction is implemented, create a better transition/section 2 introduction. Also introduce robustness!

combine figure 1, 2, 3 into a single figure* (page-width figure), or split the table from figure 2 into to parts of 8 states

3 UPDATE SCHEMES

Explain different update schemes including characteristics for behavior especially chaotic behavior. These will mostly focus on boolean networks only. Maybe mention of use-cases for each update scheme.

update this text.. (or remove it)

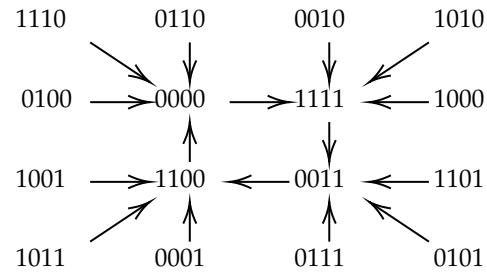


Fig. 4. State graph of the boolean network shown in fig. 1 using the sequential update with the sequence (x_1, x_2, x_3, x_4)

3.1 Synchronous scheme

In section 2, we have used the synchronous update scheme to introduce the general dynamics of boolean networks. This scheme assumes that each update function takes the exact same amount of time and executes at the same time, which means that regardless of the outcome of the first function, any subsequent function will have the same result. Updating the network with this method is quite simple, always leads to a deterministic output and doesn't require a lot of computing power in order to simulate larger networks, as each state of the network results in exactly one updated state. In regards to gene regulatory networks however, this is an oversimplification as normal GRNs never execute fully synchronously. Furthermore, the change of one node has no effect on any other function, whereas larger biochemical systems often show dependencies between functions.

add more context and information about grn and their effects, split text into logical paragraphs

3.2 Sequential scheme

Unlike the synchronous scheme, the sequential scheme allows for side effects when updating the network by having an order in which nodes are updated at each step. When using the sequential update scheme, the update cycle evaluates each after another in a fixed order. For example take the sequence (x_1, x_2, x_3, x_4) . Updating the state 0000 means first using f_1 which results in an intermediate state 1000, then using f_2 , then f_3 , and lastly f_4 . This will lead to $0000 \rightarrow 1111$, a different state as when using the synchronous update scheme. In fact, the whole state graph changes depending on the update scheme used, namely every initial state will fall into a single 4-state cycle being $0000 \rightarrow 1111 \rightarrow 0011 \rightarrow 1100 \rightarrow 0000$, see fig. 4.

While the sequential update scheme seems to be different from synchronous, it is actually possible to derive a set of functions for a synchronous update scheme that have the same outcome as the current set for a particular sequential scheme. This can be achieved by reevaluating each function in the order of the sequence and replacing x_i with f'_i if f'_i has been computed previously where f'_i represents the new function for synchronous use. In case of the boolean network from fig. 1 with sequence (x_1, x_2, x_3, x_4) , the resulting functions are

$$f'_1 = \neg x_2,$$

$$f'_2 = f'_1 = \neg x_2,$$

$$f'_3 = f'_1 \oplus x_4 = \neg x_2 \oplus x_4,$$

$$f'_4 = f'_3 = \neg x_2 \oplus x_4.$$

This means that sequential schemes share the same characteristics as the synchronous scheme. Furthermore, if the side-effects from the sequential scheme are desired, the synchronous function set would become larger and more complex.

An extension of the sequential scheme is the block-sequential scheme. This allows to divide the nodes into groups, called blocks, that run synchronously, while the blocks execute in sequence. This allows for further fine-tuning of the behavior and can simplify the modeling of realistic systems. The block-sequential scheme can also be converted to a synchronous scheme, thus producing the same behavior as previously described.

3.3 Probabilistic scheme

The probabilistic scheme is an extension of synchronous, sequential and non-random asynchronous scheme as it introduces randomness into the system. Instead of having one function per node, a node has multiple functions with individual probabilities. Upon updating a node, one of these functions is chosen randomly with regards to the probability distribution. This allows the introduction of noise and uncertainty into the system.

An example of how this can drastically change the outcome can be visualized when adding a second function to the node x_4 from the example in fig. 1. Let $f_{4switch} = \neg f_4 = \neg x_3$ with an equal probability distribution. This simple addition to the boolean network causes that on average, every second update the state switches from one cycle to the other. Thus there are no longer two individual cycles. Any initial state will eventually reach all possible states and will never fall into a repeating pattern.

refer to [1] and potentially add figure 2 from that paper

3.4 Asynchronous scheme

While synchronous, sequential and probabilistic schemes update all the nodes in one time step, the asynchronous scheme only updates one node per time step. The order in which nodes are updated depend on the chosen asynchronous scheme.

- Asynchronous deterministic is similar to sequential in that it has a sequence of nodes in which the nodes are updated. However, nodes can appear an arbitrary amount of times in one sequence, allowing for a more dynamical behavior modeling compared to sequential scheme.
- Asynchronous random is pretty straight forward. On each update cycle a random node is chosen to be updated.

During early stages of research of boolean networks, it was believed that asynchronous update schemes would lead to a more realistic modeling of biochemical systems [1]. However, as studies have shown, asynchronous update schemes and synchronous update schemes can lead to the

same meaningful behavior for biochemical systems. Furthermore, assuming that one update process takes a few seconds to minutes, updating a large system can take days to have every node updated once, if not longer[1].

create introduction, conclusion/transition to the next scheme for each scheme

4 RELEVANCE FOR GENE REGULATORY NETWORKS

Tie the update schemes and their different outcomes or behavior to GRN.

Emphasizing the drawbacks of asynchronous models when applied to GRN e.g. it takes way to long to update a GRN using asynchronous deterministic for it to have an effect; assuming that one update takes a few minutes, when the whole process can take days to complete.[1]

5 CONCLUSION

While initially believed that different update schemes lead to different outputs, it turns out that in general, most schemes are equal, especially synchronous, sequential, block-sequential and asynchronous deterministic. It only requires a different function set, in order to move from one update scheme to the next one. Only the probabilistic scheme and asynchronous random are non deterministic and cannot truly be changed to a different update scheme with just a change of functions.

Choosing an update scheme depends mostly on which effect is required and most simple to model using a certain scheme. No side effects? Choose synchronous. With side effects? Choose sequential or asynchronous. Have a system with noise? Give probabilistic a try or maybe even asynchronous random.

References: [1][2][3][4][5]

currently not really using references. i need to go over everything again and add the references at the right spots and even mention them explicitly in sentences for further context.

REFERENCES

- [1] Julian D Schwab, Silke D Kühlwein, Nensi Ikononi, Michael Kühl, and Hans A Kestler. Concepts in boolean network modeling: What do they all mean? *Computational and structural biotechnology journal*, 18:571–582, 2020.
- [2] Julio Aracena, Eric Goles, Andrés Moreira, and Luis Salinas. On the robustness of update schedules in boolean networks. *Biosystems*, 97(1):1–8, 2009.
- [3] Stefan Bornholdt. Boolean network models of cellular regulation: prospects and limitations. *Journal of the Royal Society interface*, 5(suppl_1):S85–S94, 2008.
- [4] Eric Goles and Mathilde Noual. Block-sequential update schedules and boolean automata circuits. *Discrete Mathematics & Theoretical Computer Science*, (Proceedings), 2010.
- [5] Tomas Helikar, Naomi Kochi, John Konvalina, and Jim A Rogers. Boolean modeling of biochemical networks. *The Open Bioinformatics Journal*, 5(1), 2011.