Discrete Approximations of Continuous Models

Winfried Just
Department of Mathematics, Ohio University

October 10, 2013
Seminari de Matemàtica Aplicada, Universitat de Girona
Girona, Catalunya

The general problem

The general problem

Suppose we have a natural system N and some variables $\vec{v}(t)$ that change over time. We can consider mathematical models M_0, M_1 that predict the dynamics of the variables $\vec{v}$.

The general problem

Suppose we have a natural system N and some variables $\vec{v}(t)$ that change over time. We can consider mathematical models M_0, M_1 that predict the dynamics of the variables $\vec{v}$.

When can we be assured that models M_0, M_1 make equivalent predictions about the variables $\vec{v}$?

The general problem

Suppose we have a natural system N and some variables $\vec{v}(t)$ that change over time. We can consider mathematical models M_0, M_1 that predict the dynamics of the variables $\vec{v}$.

When can we be assured that models M_0, M_1 make equivalent predictions about the variables $\vec{v}$?

With some appropriated definitions, this questions can be turned into a mathematical problem; whereas the question whether the model makes true predictions about $\vec{v}$ is empirical and goes beyond mathematics.

Some caveats

Some caveats

- The models M_0, M_1 may contain additional variables beyond those that model $\vec{v}$.

Some caveats

- The models M_0, M_1 may contain additional variables beyond those that model $\vec{v}$.
- The models M_0, M_1 may be of different types: ODE, PDE systems, discrete-time systems with a continuous or discrete state space (e.g. Boolean), or even stochastic processes of various kinds.

Some caveats

- The models M_0, M_1 may contain additional variables beyond those that model $\vec{v}$.
- The models M_0, M_1 may be of different types: ODE, PDE systems, discrete-time systems with a continuous or discrete state space (e.g. Boolean), or even stochastic processes of various kinds.
- The meaning of equivalent predictions is far from obvious when M_0, M_1 are of different types. In general, the precise definition will depend on the particular aspects of the dynamics that M_0, M_1 are supposed to model.

DE models of gene regulation

What is going on in a cell biochemically at any given time is determined by which genes are being expressed at the time. The possible expression patterns form the states of the gene regulatory network.

What is going on in a cell biochemically at any given time is determined by which genes are being expressed at the time. The possible expression patterns form the states of the gene regulatory network.

How can we study this network?

DE models of gene regulation

What is going on in a cell biochemically at any given time is determined by which genes are being expressed at the time. The possible expression patterns form the states of the gene regulatory network.

How can we study this network?

One approach would be to conceptualize a state of a gene regulatory network as a vector of concentrations of messenger RNA (mRNA). This would give a system of ODEs (or PDEs if location in the cell is also taken into account).

What is going on in a cell biochemically at any given time is determined by which genes are being expressed at the time. The possible expression patterns form the states of the gene regulatory network.

How can we study this network?

One approach would be to conceptualize a state of a gene regulatory network as a vector of concentrations of messenger RNA (mRNA). This would give a system of ODEs (or PDEs if location in the cell is also taken into account).

Let M_1 denote the resulting model.

Boolean models of gene regulation

It is difficult to measure actual mRNA concentrations with reasonable accuracy. But it is easy to take fuzzy snapshots of mRNA levels at different times even for all genes of an organism simultaneously using microarrays. These snapshots reveal only whether the expression level of a gene is high or low (sort of). One is thus tempted to construct a model M_0 of gene regulation that is a Boolean system, where

Boolean models of gene regulation

It is difficult to measure actual mRNA concentrations with reasonable accuracy. But it is easy to take fuzzy snapshots of mRNA levels at different times even for all genes of an organism simultaneously using microarrays. These snapshots reveal only whether the expression level of a gene is high or low (sort of). One is thus tempted to construct a model M_0 of gene regulation that is a Boolean system, where

- Variable i represents the mRNA level of gene i.

Boolean models of gene regulation

It is difficult to measure actual mRNA concentrations with reasonable accuracy. But it is easy to take fuzzy snapshots of mRNA levels at different times even for all genes of an organism simultaneously using microarrays. These snapshots reveal only whether the expression level of a gene is high or low (sort of). One is thus tempted to construct a model M_0 of gene regulation that is a Boolean system, where

- Variable i represents the mRNA level of gene i.
- Expression levels take only values 0 (low) and 1 (high).

Boolean models of gene regulation

It is difficult to measure actual mRNA concentrations with reasonable accuracy. But it is easy to take fuzzy snapshots of mRNA levels at different times even for all genes of an organism simultaneously using microarrays. These snapshots reveal only whether the expression level of a gene is high or low (sort of). One is thus tempted to construct a model M_0 of gene regulation that is a Boolean system, where

- Variable i represents the mRNA level of gene i.
- Expression levels take only values 0 (low) and 1 (high).
- Time proceeds in discrete steps.

Boolean models of gene regulation

It is difficult to measure actual mRNA concentrations with reasonable accuracy. But it is easy to take fuzzy snapshots of mRNA levels at different times even for all genes of an organism simultaneously using microarrays. These snapshots reveal only whether the expression level of a gene is high or low (sort of). One is thus tempted to construct a model M_0 of gene regulation that is a Boolean system, where

- Variable i represents the mRNA level of gene i.
- Expression levels take only values 0 (low) and 1 (high).
- Time proceeds in discrete steps.
- At each time step, all variables are updated simultaneously.

Boolean models of gene regulation

It is difficult to measure actual mRNA concentrations with reasonable accuracy. But it is easy to take fuzzy snapshots of mRNA levels at different times even for all genes of an organism simultaneously using microarrays. These snapshots reveal only whether the expression level of a gene is high or low (sort of). One is thus tempted to construct a model M_0 of gene regulation that is a Boolean system, where

- Variable i represents the mRNA level of gene i.
- Expression levels take only values 0 (low) and 1 (high).
- Time proceeds in discrete steps.
- At each time step, all variables are updated simultaneously.

All these assumptions are biologically unrealistic.

What kind of mathematical object is M_0?

- The state space is $\{0, 1\}^n$, where n is the number of genes.

What kind of mathematical object is M_0?

- The state space is $\{0, 1\}^n$, where n is the number of genes.
- Time τ takes values in the set $\mathbb{N}$ of nonnegative integers.

What kind of mathematical object is M_0?

- The state space is $\{0,1\}^n$, where n is the number of genes.
- Time τ takes values in the set $\mathbb{N}$ of nonnegative integers.
- There exists a function $f : \{0,1\}^n \to \{0,1\}^n$ called the updating function such that
  $\vec{s}(\tau + 1) = f(\vec{s}(\tau))$.

What kind of mathematical object is M_0?

- The state space is $\{0,1\}^n$, where n is the number of genes.
- Time τ takes values in the set $\mathbb{N}$ of nonnegative integers.
- There exists a function $f : \{0,1\}^n \to \{0,1\}^n$ called the updating function such that $\vec{s}(\tau+1) = f(\vec{s}(\tau))$.
- The i-th component $f_i : \{0,1\}^n \to \{0,1\}^n$ of f is called the regulatory function of gene number i.

Note that M_0 is uniquely determined by f.

Some empirical observations

- Boolean models often appear to work pretty well for gene regulation.

Some empirical observations

- Boolean models often appear to work pretty well for gene regulation.
- The processes of of transcription (production of mRNA), translation (of mRNA into proteins) and diffusion of mRNA proceed at different time scales.

Some empirical observations

- Boolean models often appear to work pretty well for gene regulation.
- The processes of of transcription (production of mRNA), translation (of mRNA into proteins) and diffusion of mRNA proceed at different time scales.
- Gene regulatory and other biological networks appear to involve more intermediaries than seems strictly necessary.

Some empirical observations

- Boolean models often appear to work pretty well for gene regulation.
- The processes of of transcription (production of mRNA), translation (of mRNA into proteins) and diffusion of mRNA proceed at different time scales.
- Gene regulatory and other biological networks appear to involve more intermediaries than seems strictly necessary.

Why do Boolean systems work so well as models of gene regulatory networks?

Some empirical observations

- Boolean models often appear to work pretty well for gene regulation.
- The processes of of transcription (production of mRNA), translation (of mRNA into proteins) and diffusion of mRNA proceed at different time scales.
- Gene regulatory and other biological networks appear to involve more intermediaries than seems strictly necessary.

Why do Boolean systems work so well as models of gene regulatory networks?
What is the role of separation of timescales and intermediaries in this phenomenon?

Some empirical observations

- Boolean models often appear to work pretty well for gene regulation.
- The processes of of transcription (production of mRNA), translation (of mRNA into proteins) and diffusion of mRNA proceed at different time scales.
- Gene regulatory and other biological networks appear to involve more intermediaries than seems strictly necessary.

Why do Boolean systems work so well as models of gene regulatory networks?
What is the role of separation of timescales and intermediaries in this phenomenon?
Under what conditions is M_0 guaranteed to be a good approximation of M_1?

Some empirical observations

- Boolean models often appear to work pretty well for gene regulation.
- The processes of of transcription (production of mRNA), translation (of mRNA into proteins) and diffusion of mRNA proceed at different time scales.
- Gene regulatory and other biological networks appear to involve more intermediaries than seems strictly necessary.

Why do Boolean systems work so well as models of gene regulatory networks?
What is the role of separation of timescales and intermediaries in this phenomenon?
Under what conditions is M_0 guaranteed to be a good approximation of M_1?
In other words, which conditions guarantee that a DE model M_1 will exhibit switchlike behavior?

What is an approximation in this context? Comparing apples with oranges

The mRNA concentrations $\vec{v}$ of N will be represented by vectors $\vec{x}$ of reals in M_1 and Boolean vectors $\vec{s}$ in M_0.

What is an approximation in this context? Comparing apples with oranges

The mRNA concentrations $\vec{v}$ of N will be represented by vectors $\vec{x}$ of reals in M_1 and Boolean vectors $\vec{s}$ in M_0.

We can fix a discretization S that maps real vectors $\vec{x}$ to Boolean vectors $\vec{s} = S(\vec{x})$.

What is an approximation in this context? Comparing apples with oranges

The mRNA concentrations $\vec{v}$ of N will be represented by vectors $\vec{x}$ of reals in M_1 and Boolean vectors $\vec{s}$ in M_0.

We can fix a discretization S that maps real vectors $\vec{x}$ to Boolean vectors $\vec{s} = S(\vec{x})$.

Then we can consider M_0 a good approximation of M_1 if for each ODE trajectory $\vec{x}(t)$ that starts from initial condition $\vec{x}(0) \in U$ for some large enough U the corresponding discretized trajectory $S(\vec{x}(t))$ is consistent with the Boolean trajectory $\vec{s}(\tau)$, where $\vec{s}(0) = S(\vec{x}(0))$, that is, if the updating function f of M_1 correctly predicts, at all future times, which discretized state will be entered next by the DE trajectory.

What is an approximation in this context? Comparing apples with oranges

The mRNA concentrations $\vec{v}$ of N will be represented by vectors $\vec{x}$ of reals in M_1 and Boolean vectors $\vec{s}$ in M_0.

We can fix a discretization S that maps real vectors $\vec{x}$ to Boolean vectors $\vec{s} = S(\vec{x})$.

Then we can consider M_0 a good approximation of M_1 if for each ODE trajectory $\vec{x}(t)$ that starts from initial condition $\vec{x}(0) \in U$ for some large enough U the corresponding discretized trajectory $S(\vec{x}(t))$ is consistent with the Boolean trajectory $\vec{s}(\tau)$, where $\vec{s}(0) = S(\vec{x}(0))$, that is, if the updating function f of M_1 correctly predicts, at all future times, which discretized state will be entered next by the DE trajectory.

All of numerical analysis is essentially based on this type of construction, except that there the concern is with not exceeding the error tolerance over a finite time interval.

What is the next state?

Time t in the DE model does not admit any kind of next state.
We need to define what this means.

What is the next state?

Time t in the DE model does not admit any kind of next state. We need to define what this means.

Consider a time t_{switch} when $\vec{s}(\tau) = \lim_{t \to t^-_{switch}} S(\vec{x}(t)) \neq \lim_{t \to t^+_{switch}} S(\vec{x}(t)) = \vec{s}(\tau + 1)$.

- If $\vec{s}(\tau + 1) = f(\vec{s}(\tau))$ in all such situations, we have strong consistency.

What is the next state?

Time t in the DE model does not admit any kind of next state. We need to define what this means.

Consider a time t_{switch} when $\vec{s}(\tau) = \lim_{t \to t_{switch}^-} S(\vec{x}(t)) \neq \lim_{t \to t_{switch}^+} S(\vec{x}(t)) = \vec{s}(\tau + 1)$.

- If $\vec{s}(\tau + 1) = f(\vec{s}(\tau))$ in all such situations, we have strong consistency.
- If $s_i(\tau + 1) = f_i(\vec{s}(\tau))$ for the variable number i whose Boolean state actually changes and $s_j(\tau + 1) = s_j(\tau)$ for $j \neq i$, we have consistency.

What is the next state?

Time t in the DE model does not admit any kind of next state. We need to define what this means.

Consider a time t_{switch} when
$\vec{s}(\tau) = \lim_{t \to t_{switch}^-} S(\vec{x}(t)) \neq \lim_{t \to t_{switch}^+} S(\vec{x}(t)) = \vec{s}(\tau + 1)$.

- If $\vec{s}(\tau + 1) = f(\vec{s}(\tau))$ in all such situations, we have strong consistency.
- If $s_i(\tau + 1) = f_i(\vec{s}(\tau))$ for the variable number i whose Boolean state actually changes and $s_j(\tau + 1) = s_j(\tau)$ for $j \neq i$, we have consistency.
- Other notions of consistency are meaningful if we can treat changes in the Boolean state of several variables that occur in very short intervals as simultaneous.

A problem from mathematical neuroscience

Recordings from certain neuronal tissues (of real organisms) reveal the following pattern: Time seems to be partitioned into episodes with surprisingly sharp boundaries. During one episode, a group of neurons fires, while other neurons are at rest. In the next episode, a different group of neurons fires. Group membership may vary from episode to episode, a phenomenon called dynamic clustering.

A problem from mathematical neuroscience

Recordings from certain neuronal tissues (of real organisms) reveal the following pattern: Time seems to be partitioned into episodes with surprisingly sharp boundaries. During one episode, a group of neurons fires, while other neurons are at rest. In the next episode, a different group of neurons fires. Group membership may vary from episode to episode, a phenomenon called dynamic clustering.

Why? Can we mathematically explain this phenomenon?

An ODE Model of Neuronal Networks

by Terman D, Ahn S, Wang X, Just W, Physica D. 2008

Each excitatory (E-) cell satisfies

$$\begin{aligned}
\frac{dv_i}{dt} &= f(v_i, w_i) - g_{EI} \sum s_j^I (v_i - v_{syn}^I) \\
\frac{dw_i}{dt} &= \epsilon g(v_i, w_i) \\
\frac{ds_i}{dt} &= \alpha(1 - s_i) H(v_i - \theta_E) - \beta s_i.
\end{aligned}$$

Each inhibitory (I-) cell satisfies

$$\begin{aligned}
\frac{dv_i^I}{dt} &= f(v_i^I, w_i^I) - g_{IE} \sum s_j (v_i^I - v_{syn}^E) - g_{II} \sum s_j^I (v_i^I - v_{syn}^I) \\
\frac{dw_i^I}{dt} &= \epsilon g(v_i^I, w_i^I) \\
\frac{dx_i^I}{dt} &= \epsilon \alpha_x (1 - x_i^I) H(v_i^I - \theta_I) - \epsilon \beta_x x_i^I \\
\frac{ds_i^I}{dt} &= \alpha_I (1 - s_i^I) H(x_i^I - \theta_x) - \beta_I s_i^I.
\end{aligned}$$

Can we analyze the dynamics of this model?

- The architecture involves a layer of excitatory neurons and a layer of inhibitory neurons that mediate the firing of the excitatory neurons. The whole layer acts as a pacemaker.

Can we analyze the dynamics of this model?

- The architecture involves a layer of excitatory neurons and a layer of inhibitory neurons that mediate the firing of the excitatory neurons. The whole layer acts as a pacemaker.
- Individual neurons are usually modeled by the the Hodgkin-Huxley Equations, which are nonlinear ODEs involving multiple time scales.

Can we analyze the dynamics of this model?

- The architecture involves a layer of excitatory neurons and a layer of inhibitory neurons that mediate the firing of the excitatory neurons. The whole layer acts as a pacemaker.
- Individual neurons are usually modeled by the the Hodgkin-Huxley Equations, which are nonlinear ODEs involving multiple time scales.
- These are difficult to analyze mathematically even for single neurons, let alone for large networks.

Can we analyze the dynamics of this model?

- The architecture involves a layer of excitatory neurons and a layer of inhibitory neurons that mediate the firing of the excitatory neurons. The whole layer acts as a pacemaker.
- Individual neurons are usually modeled by the the Hodgkin-Huxley Equations, which are nonlinear ODEs involving multiple time scales.
- These are difficult to analyze mathematically even for single neurons, let alone for large networks.

Can we analyze the dynamics of this model?

- The architecture involves a layer of excitatory neurons and a layer of inhibitory neurons that mediate the firing of the excitatory neurons. The whole layer acts as a pacemaker.
- Individual neurons are usually modeled by the the Hodgkin-Huxley Equations, which are nonlinear ODEs involving multiple time scales.
- These are difficult to analyze mathematically even for single neurons, let alone for large networks.

Fortune cookie: Doing the impossible is kind of fun.

Can we analyze the dynamics of this model?

- The architecture involves a layer of excitatory neurons and a layer of inhibitory neurons that mediate the firing of the excitatory neurons. The whole layer acts as a pacemaker.
- Individual neurons are usually modeled by the the Hodgkin-Huxley Equations, which are nonlinear ODEs involving multiple time scales.
- These are difficult to analyze mathematically even for single neurons, let alone for large networks.

Fortune cookie: Doing the impossible is kind of fun.

Let us call the model that we just described M_1.

The following is true in at least some neuronal networks.

- Neurons fire or are at rest.
- After a neuron has fired, it has to go through a certain refractory period when it cannot fire.
- A neuron will fire when it has reached the end of its refractory period and when it receives firing input from a specified minimal number of other neurons.

Let us build a simple model M_0 of neuronal networks based on these facts.

A Discrete Dynamical System Model

A directed graph $D = (V_D, A_D)$ and integers n (size of the network), p_i (refractory period), th_i (firing threshold).

A Discrete Dynamical System Model

A directed graph $D = (V_D, A_D)$ and integers n (size of the network), p_i (refractory period), th_i (firing threshold).

A state $\vec{s}(\tau)$ at the discrete time τ is a vector:
$\vec{s}(\tau) = (s_1(\tau), \ldots, s_n(\tau))$ where $s_i(\tau) \in \{0, 1, \ldots, p_i\}$ for each i.
The state $s_i(\tau) = 0$ means neuron i fires at time τ.

A Discrete Dynamical System Model

A directed graph $D = (V_D, A_D)$ and integers n (size of the network), p_i (refractory period), th_i (firing threshold).

A state $\vec{s}(\tau)$ at the discrete time τ is a vector:
$\vec{s}(\tau) = (s_1(\tau), \ldots, s_n(\tau))$ where $s_i(\tau) \in \{0, 1, \ldots, p_i\}$ for each i.
The state $s_i(\tau) = 0$ means neuron i fires at time τ.

Dynamics on the discrete network:

- If $s_i(\tau) < p_i$, then $s_i(\tau + 1) = s_i(\tau) + 1$.

A Discrete Dynamical System Model

A directed graph $D = (V_D, A_D)$ and integers n (size of the network), p_i (refractory period), th_i (firing threshold).

A state $\vec{s}(\tau)$ at the discrete time τ is a vector:
$\vec{s}(\tau) = (s_1(\tau), \ldots, s_n(\tau))$ where $s_i(\tau) \in \{0, 1, \ldots, p_i\}$ for each i.
The state $s_i(\tau) = 0$ means neuron i fires at time τ.

Dynamics on the discrete network:

- If $s_i(\tau) < p_i$, then $s_i(\tau + 1) = s_i(\tau) + 1$.
- If $s_i(\tau) = p_i$, and there exists at least th_i neurons j with $s_j(k) = 0$ and $< j, i > \ \in A_D$, then $s_i(\tau + 1) = 0$.

A Discrete Dynamical System Model

A directed graph $D = (V_D, A_D)$ and integers n (size of the network), p_i (refractory period), th_i (firing threshold).

A state $\vec{s}(\tau)$ at the discrete time τ is a vector:
$\vec{s}(\tau) = (s_1(\tau), \dots, s_n(\tau))$ where $s_i(\tau) \in \{0, 1, \dots, p_i\}$ for each i.
The state $s_i(\tau) = 0$ means neuron i fires at time τ.

Dynamics on the discrete network:

- If $s_i(\tau) < p_i$, then $s_i(\tau + 1) = s_i(\tau) + 1$.
- If $s_i(\tau) = p_i$, and there exists at least th_i neurons j with $s_j(k) = 0$ and $< j, i > \ \in A_D$, then $s_i(\tau + 1) = 0$.
- If $s_i(\tau) = p_i$ and there do not exist th_i neurons j with $s_j(\tau) = 0$ and $< j, i > \ \in A_D$, then $s_i(\tau + 1) = p_i$.

A Discrete Dynamical System Model

A directed graph $D = (V_D, A_D)$ and integers n (size of the network), p_i (refractory period), th_i (firing threshold).

A state $\vec{s}(\tau)$ at the discrete time τ is a vector:
$\vec{s}(\tau) = (s_1(\tau), \dots, s_n(\tau))$ where $s_i(\tau) \in \{0, 1, \dots, p_i\}$ for each i.
The state $s_i(\tau) = 0$ means neuron i fires at time τ.

Dynamics on the discrete network:

- If $s_i(\tau) < p_i$, then $s_i(\tau + 1) = s_i(\tau) + 1$.
- If $s_i(\tau) = p_i$, and there exists at least th_i neurons j with $s_j(k) = 0$ and $< j, i > \ \in A_D$, then $s_i(\tau + 1) = 0$.
- If $s_i(\tau) = p_i$ and there do not exist th_i neurons j with $s_j(\tau) = 0$ and $< j, i > \ \in A_D$, then $s_i(\tau + 1) = p_i$.

If $p_i = 1$ for all i then this is a Boolean system.

Theorem

For each ODE model M_1 *of neuronal networks as described above, if the intrinsic and synaptic properties of the cells are chosen appropriately, the dynamics of* M_1 *will exhibit dynamic clustering.*

Reducing Neuronal Networks to Discrete Dynamics,

by Terman D, Ahn S, Wang X, Just W, Physica D. 2008

Theorem

For each ODE model M_1 of neuronal networks as described above, if the intrinsic and synaptic properties of the cells are chosen appropriately, the dynamics of M_1 will exhibit dynamic clustering. Moreover, there exists a discrete model M_0 as on the previous slide that correctly predicts, for a large region U of the state space of M_1 and all times t which neurons will fire during which episodes.

Theorem

For each ODE model M_1 of neuronal networks as described above, if the intrinsic and synaptic properties of the cells are chosen appropriately, the dynamics of M_1 will exhibit dynamic clustering. Moreover, there exists a discrete model M_0 as on the previous slide that correctly predicts, for a large region U of the state space of M_1 and all times t which neurons will fire during which episodes.

The corresponding models M_0 are much more tractable than the models M_1. In particular, they permit us to study the dependence of the dynamics on the network connectivity.

Some drawbacks of this result

- The right-hand sides of the ODEs for M_1 contain some discontinuities (some use the Heaviside function).

Some drawbacks of this result

- The right-hand sides of the ODEs for M_1 contain some discontinuities (some use the Heaviside function).
- The required architecture of the network is very special.

Some drawbacks of this result

- The right-hand sides of the ODEs for M_1 contain some discontinuities (some use the Heaviside function).
- The required architecture of the network is very special.
- The ODE models are very complicated and difficult to analyze.

Some drawbacks of this result

- The right-hand sides of the ODEs for M_1 contain some discontinuities (some use the Heaviside function).
- The required architecture of the network is very special.
- The ODE models are very complicated and difficult to analyze.

We would really like to know what kind of architectures in general favor or imply consistency of a ODE system with a Boolean system. In particular, we want to understand the role of separation of timescales and of intermediary variables.

Recall the following notions

Let t denote time in the ODE model M_1, let τ denote time in the Boolean approximation M_0. Consider a time t_{switch} when $\vec{s}(\tau) = \lim_{t \to t^-_{switch}} S(\vec{x}(t)) \neq \lim_{t \to t^+_{switch}} S(\vec{x}(t)) = \vec{s}(\tau + 1)$.

- If $\vec{s}(\tau + 1) = f(\vec{s}(\tau))$ in all such situations, we have strong consistency.
- If $s_i(\tau + 1) = f_i(\vec{s}(\tau))$ for the variable number i whose Boolean state actually changes and $s_j(\tau + 1) = s_j(\tau)$ for $j \neq i$, we have consistency.
- Other notions of consistency are meaningful if we can treat changes in the Boolean state of several variables that occur in very short intervals as simultaneous.

Related results in the literature

- Leon Glass and his followers have produced a large body of work on consistency (but not in general strong consistency) between so-called piecewise linear ODE models of gene regulatory networks and their Boolean approximations. Again, the right-hand sides of the ODEs in these models have discontinuities.
- E. Gehrmann, B. Drossel, Boolean versus continuous dynamics on simple two-gene modules, *Phys. Rev. E* **82** (2010) 046120 prove strong consistency for one simple example of ODE and Boolean networks. The right-hand sides of the ODE model in this example are Lipschitz-continuous.

Our plan

Construct classes $\mathbb{D}$ of toy ODE models such that

Our plan

Construct classes $\mathbb{D}$ of toy ODE models such that

- The ODEs for individual variables are easy to understand.

Our plan

Construct classes $\mathbb{D}$ of toy ODE models such that

- The ODEs for individual variables are easy to understand.
- The right-hand sides of the ODEs are Lipschitz-continuous.

Our plan

Construct classes $\mathbb{D}$ of toy ODE models such that

- The ODEs for individual variables are easy to understand.
- The right-hand sides of the ODEs are Lipschitz-continuous.
- The class $\mathbb{D}$ is universal in the sense that each Boolean system M_0 can be treated as a (not necessarily consistent) natural approximation of a system $M_1 \in \mathbb{D}$.

Our plan

Construct classes $\mathbb{D}$ of toy ODE models such that

- The ODEs for individual variables are easy to understand.
- The right-hand sides of the ODEs are Lipschitz-continuous.
- The class $\mathbb{D}$ is universal in the sense that each Boolean system M_0 can be treated as a (not necessarily consistent) natural approximation of a system $M_1 \in \mathbb{D}$.
- The class $\mathbb{D}$ permits explorations of how network connectivity (in particular: intermediary variables) and separation of time scales influence the presence or absence of consistency between M_1 and M_0.

Our plan

Construct classes $\mathbb{D}$ of toy ODE models such that

- The ODEs for individual variables are easy to understand.
- The right-hand sides of the ODEs are Lipschitz-continuous.
- The class $\mathbb{D}$ is universal in the sense that each Boolean system M_0 can be treated as a (not necessarily consistent) natural approximation of a system $M_1 \in \mathbb{D}$.
- The class $\mathbb{D}$ permits explorations of how network connectivity (in particular: intermediary variables) and separation of time scales influence the presence or absence of consistency between M_1 and M_0.

W. Just, M. Korb, B. Elbert, and T. Young; Two classes of ODE models with switch-like behavior, *Physica D* **264** (2013) 35–48.

In order to achieve universality of our class $\mathbb{D}$ we need to translate a Boolean system with updating function f into ODE systems $D(f, \vec{\gamma})$, where $\vec{\gamma}$ is a vector of parameters.

Conversion of Boolean into ODE systems

In order to achieve universality of our class $\mathbb{D}$ we need to translate a Boolean system with updating function f into ODE systems $D(f, \vec{\gamma})$, where $\vec{\gamma}$ is a vector of parameters.

For this we need a conversion
$Q = Q^f = (Q_1, \dots, Q_n) : \mathbb{R}^n \to [0, 1]^n$.

Conversion of Boolean into ODE systems

In order to achieve universality of our class $\mathbb{D}$ we need to translate a Boolean system with updating function f into ODE systems $D(f, \vec{\gamma})$, where $\vec{\gamma}$ is a vector of parameters.

For this we need a conversion
$Q = Q^f = (Q_1, \ldots, Q_n) : \mathbb{R}^n \to [0, 1]^n$.

Conversions are Lipschitz continuous,

Conversion of Boolean into ODE systems

In order to achieve universality of our class $\mathbb{D}$ we need to translate a Boolean system with updating function f into ODE systems $D(f, \vec{\gamma})$, where $\vec{\gamma}$ is a vector of parameters.

For this we need a conversion
$Q = Q^f = (Q_1, \dots, Q_n) : \mathbb{R}^n \to [0, 1]^n$.

Conversions are Lipschitz continuous,

constant on $(-\infty, -1] \times \{\vec{y}\}$, and on $[1, \infty) \times \{\vec{y}\}$, where $\{\vec{y}\} \in \mathbb{R}^{n \setminus \{i\}}$,

Conversion of Boolean into ODE systems

In order to achieve universality of our class $\mathbb{D}$ we need to translate a Boolean system with updating function f into ODE systems $D(f, \vec{\gamma})$, where $\vec{\gamma}$ is a vector of parameters.

For this we need a conversion
$Q = Q^f = (Q_1, \dots, Q_n) : \mathbb{R}^n \to [0, 1]^n$.

Conversions are Lipschitz continuous,

constant on $(-\infty, -1] \times \{\vec{y}\}$, and on $[1, \infty) \times \{\vec{y}\}$, where $\{\vec{y}\} \in \mathbb{R}^{n \setminus \{i\}}$,

$Q(\tilde{s}) = f(\tilde{s})$, where for a Boolean vector s we have $\tilde{s}_i = 2s_i - 1$.

Conversion of Boolean into ODE systems

In order to achieve universality of our class $\mathbb{D}$ we need to translate a Boolean system with updating function f into ODE systems $D(f, \vec{\gamma})$, where $\vec{\gamma}$ is a vector of parameters.

For this we need a conversion
$Q = Q^f = (Q_1, \ldots, Q_n) : \mathbb{R}^n \to [0,1]^n$.

Conversions are Lipschitz continuous,

constant on $(-\infty, -1] \times \{\vec{y}\}$, and on $[1, \infty) \times \{\vec{y}\}$, where $\{\vec{y}\} \in \mathbb{R}^{n \setminus \{i\}}$,

$Q(\tilde{s}) = f(\tilde{s})$, where for a Boolean vector s we have $\tilde{s}_i = 2s_i - 1$.

The paper discusses several natural classes of conversion schemes.

The ODEs

Let $g(x) = 3x - x^3 - 3$.

The ODEs

Let $g(x) = 3x - x^3 - 3$.

The ODE for variable i will take the form

$\dot{x}_i = \gamma_i(g(x_i) + 6Q_i(\vec{x}))$,

The ODEs

Let $g(x) = 3x - x^3 - 3$.

The ODE for variable i will take the form

$\dot{x}_i = \gamma_i(g(x_i) + 6Q_i(\vec{x}))$,

where Q_i is the i^{th} coordinate of a given conversion $Q = Q^f$,

The ODEs

Let $g(x) = 3x - x^3 - 3$.

The ODE for variable i will take the form

$$\dot{x}_i = \gamma_i(g(x_i) + 6Q_i(\vec{x})),$$

where Q_i is the i^{th} coordinate of a given conversion $Q = Q^f$, and γ_i is a positive constant that allows us to implement separation of time scales.

Properties of the ODEs

$\dot{x}_i = \gamma_i(g(x_i) + 6Q_i(\vec{x}))$, where $g(x) = 3x - x^3 - 3$. Think if Q_i as a constant.

Properties of the ODEs

$\dot{x}_i = \gamma_i(g(x_i) + 6Q_i(\vec{x}))$, where $g(x) = 3x - x^3 - 3$. Think if Q_i as a constant.

- If $Q_i < 1/6$, there exists a unique globally stable equilibrium $x_i^* < -1$.

Properties of the ODEs

$\dot{x}_i = \gamma_i(g(x_i) + 6Q_i(\vec{x}))$, where $g(x) = 3x - x^3 - 3$. Think if Q_i as a constant.

- If $Q_i < 1/6$, there exists a unique globally stable equilibrium $x_i^* < -1$.
- If $Q_i > 5/6$, there exists a unique globally stable equilibrium $x_i^{**} > 1$.

Properties of the ODEs

$\dot{x}_i = \gamma_i(g(x_i) + 6Q_i(\vec{x}))$, where $g(x) = 3x - x^3 - 3$. Think if Q_i as a constant.

- If $Q_i < 1/6$, there exists a unique globally stable equilibrium $x_i^* < -1$.
- If $Q_i > 5/6$, there exists a unique globally stable equilibrium $x_i^{**} > 1$.
- If $Q_i \in (1/6, 5/6)$, there exist three equilibria, locally stable ones $x_i^* < -1$ and $x_i^{**} > 1$, as well as an unstable one $-1 < x^\circ < 1$.

Properties of the ODEs

$\dot{x}_i = \gamma_i(g(x_i) + 6Q_i(\vec{x}))$, where $g(x) = 3x - x^3 - 3$. Think if Q_i as a constant.

- If $Q_i < 1/6$, there exists a unique globally stable equilibrium $x_i^* < -1$.
- If $Q_i > 5/6$, there exists a unique globally stable equilibrium $x_i^{**} > 1$.
- If $Q_i \in (1/6, 5/6)$, there exist three equilibria, locally stable ones $x_i^* < -1$ and $x_i^{**} > 1$, as well as an unstable one $-1 < x^\circ < 1$.
- $[-2.1038, 2.1038]^{2n}$ is forward invariant and can be considered the state space of our ODE systems.

An example

Let Q_i be a nondecreasing function that takes the value 0 whenever $x_j \in (-\infty, -1]$ and takes the value 1 whenever $x_j \in [1, \infty)$. We can think of Q_i as the i^{th} coordinate of a conversion of a Boolean function f with $f_i(\vec{s}) = s_j$.

An example

Let Q_i be a nondecreasing function that takes the value 0 whenever $x_j \in (-\infty, -1]$ and takes the value 1 whenever $x_j \in [1, \infty)$. We can think of Q_i as the i^{th} coordinate of a conversion of a Boolean function f with $f_i(\vec{s}) = s_j$. Consider

$\dot{x}_i = \gamma_i(g(x_i) + 6Q_i(\vec{x}))$, where $g(x) = 3x - x^3 - 3$ and an initial state where $x_i(0) \in (-\infty, -1]$ while $x_j(t) \in [1, \infty)$ for all times t.

An example

Let Q_i be a nondecreasing function that takes the value 0 whenever $x_j \in (-\infty, -1]$ and takes the value 1 whenever $x_j \in [1, \infty)$. We can think of Q_i as the i^{th} coordinate of a conversion of a Boolean function f with $f_i(\vec{s}) = s_j$. Consider

$\dot{x}_i = \gamma_i(g(x_i) + 6Q_i(\vec{x}))$, where $g(x) = 3x - x^3 - 3$ and an initial state where $x_i(0) \in (-\infty, -1]$ while $x_j(t) \in [1, \infty)$ for all times t.

Then $Q_i > 5/6$, so there exists a unique globally stable equilibrium $x_i^{**} > 1$.

An example

Let Q_i be a nondecreasing function that takes the value 0 whenever $x_j \in (-\infty, -1]$ and takes the value 1 whenever $x_j \in [1, \infty)$. We can think of Q_i as the i^{th} coordinate of a conversion of a Boolean function f with $f_i(\vec{s}) = s_j$. Consider

$\dot{x}_i = \gamma_i(g(x_i) + 6Q_i(\vec{x}))$, where $g(x) = 3x - x^3 - 3$ and an initial state where $x_i(0) \in (-\infty, -1]$ while $x_j(t) \in [1, \infty)$ for all times t.

Then $Q_i > 5/6$, so there exists a unique globally stable equilibrium $x_i^{**} > 1$.
Variable i will move into the interval $[1, \infty)$ and thus will eventually change its Boolean state to 1, as predicted by the Boolean updating function.

An example

Let Q_i be a nondecreasing function that takes the value 0 whenever $x_j \in (-\infty, -1]$ and takes the value 1 whenever $x_j \in [1, \infty)$. We can think of Q_i as the i^{th} coordinate of a conversion of a Boolean function f with $f_i(\vec{s}) = s_j$. Consider

$\dot{x}_i = \gamma_i(g(x_i) + 6Q_i(\vec{x}))$, where $g(x) = 3x - x^3 - 3$ and an initial state where $x_i(0) \in (-\infty, -1]$ while $x_j(t) \in [1, \infty)$ for all times t.

Then $Q_i > 5/6$, so there exists a unique globally stable equilibrium $x_i^{**} > 1$.
Variable i will move into the interval $[1, \infty)$ and thus will eventually change its Boolean state to 1, as predicted by the Boolean updating function.

This example allows us to incorporate equations into our system that essentially copy the Boolean value of some variable j to variable i, at whatever time scale we choose.

Signature variables and signaling variables

Think of n variables of a natural system N whose dynamics we are interested in. We will represent them in our ODE systems as variables $x_1, \ldots, x_n$, called signature variables. We are interested in achieving (strong) consistency with the dynamics of their Boolean counterparts $s_1, \ldots, s_n$ as governed by a Boolean updating function f.

Signature variables and signaling variables

Think of n variables of a natural system N whose dynamics we are interested in. We will represent them in our ODE systems as variables $x_1, \dots, x_n$, called signature variables. We are interested in achieving (strong) consistency with the dynamics of their Boolean counterparts $s_1, \dots, s_n$ as governed by a Boolean updating function f.

To achieve this goal, we will need additional signaling variables $x_{n+1}, \dots, x_{2n}$ that will not have counterparts in the Boolean approximation.

Signature variables and signaling variables

Think of n variables of a natural system N whose dynamics we are interested in. We will represent them in our ODE systems as variables $x_1, \ldots, x_n$, called signature variables. We are interested in achieving (strong) consistency with the dynamics of their Boolean counterparts $s_1, \ldots, s_n$ as governed by a Boolean updating function f.

To achieve this goal, we will need additional signaling variables $x_{n+1}, \ldots, x_{2n}$ that will not have counterparts in the Boolean approximation.

We construct a system $D(f, \vec{\gamma})$ by choosing the DE for each signature variable x_i as in the example on the previous slide, with $j = x_{n+i}$ (so the value of the signaling variable gets essentially copied to the corresponding signature variable).

Signature variables and signaling variables

Think of n variables of a natural system N whose dynamics we are interested in. We will represent them in our ODE systems as variables $x_1, \ldots, x_n$, called signature variables. We are interested in achieving (strong) consistency with the dynamics of their Boolean counterparts $s_1, \ldots, s_n$ as governed by a Boolean updating function f.

To achieve this goal, we will need additional signaling variables $x_{n+1}, \ldots, x_{2n}$ that will not have counterparts in the Boolean approximation.

We construct a system $D(f, \vec{\gamma})$ by choosing the DE for each signature variable x_i as in the example on the previous slide, with $j = x_{n+i}$ (so the value of the signaling variable gets essentially copied to the corresponding signature variable).

The signaling variable x_{n+i} takes input from the signature variables only, with Q_{n+i} being a conversion of the Boolean regulatory function f_i.

Signaling variables are needed

Consider the one-dimensional Boolean system M_0 given by the updating function
$f(s) = \neg s$.

Signaling variables are needed

Consider the one-dimensional Boolean system M_0 given by the updating function
$f(s) = \neg s$.

Then M_0 cannot be consistent with a one-dimensional ODE system M_1 for any discretization based on a single threshold.

Signaling variables are needed

Consider the one-dimensional Boolean system M_0 given by the updating function
$f(s) = \neg s$.

Then M_0 cannot be consistent with a one-dimensional ODE system M_1 for any discretization based on a single threshold.

But we get strong consistency with $D(f, \vec{\gamma})$ for any $\vec{\gamma}$.

Strong consistency is possible only for some Boolean systems

Consider a Boolean system M_0 with updating function f. We say that f (or M_0) is one-stepping if for every $\vec{s}$ the Boolean vectors $\vec{s}$ and $f(\vec{s})$ differ in at most one coordinate.

Strong consistency is possible only for some Boolean systems

Consider a Boolean system M_0 with updating function f. We say that f (or M_0) is one-stepping if for every $\vec{s}$ the Boolean vectors $\vec{s}$ and $f(\vec{s})$ differ in at most one coordinate.

Lemma

A Boolean system can be strongly consistent with an ODE system M_1 for any reasonable discretization only if M_0 is one-stepping.

Strong consistency is possible only for some Boolean systems

Consider a Boolean system M_0 with updating function f. We say that f (or M_0) is one-stepping if for every $\vec{s}$ the Boolean vectors $\vec{s}$ and $f(\vec{s})$ differ in at most one coordinate.

Lemma

A Boolean system can be strongly consistent with an ODE system M_1 for any reasonable discretization only if M_0 is one-stepping.

Proof: For any discretization with nice enough boundaries, most trajectories of M_1 will cross only one boundary at a time.

A Theorem

Theorem

Let M_0 be a Boolean system with a **one-stepping** *updating function $f : \{0,1\}^n \to \{0,1\}^n$ and let $\vec{\gamma}^- = (\gamma_1, \dots, \gamma_n)$ be a fixed vector of positive reals.*

A Theorem

Theorem

Let M_0 be a Boolean system with a **one-stepping** *updating function $f : \{0,1\}^n \to \{0,1\}^n$ and let $\vec{\gamma}^- = (\gamma_1, \dots, \gamma_n)$ be a fixed vector of positive reals. Then there exist $\mu > 0$ such that for every extension of $\vec{\gamma}^-$ to a $2n$-dimensional vector $\vec{\gamma}$ of positive reals with $\gamma_{i+n} < \mu$ for all i,*

Theorem

Let M_0 *be a Boolean system with a* **one-stepping** *updating function* $f : \{0,1\}^n \to \{0,1\}^n$ *and let* $\vec{\gamma}^- = (\gamma_1, \dots, \gamma_n)$ *be a fixed vector of positive reals. Then there exist* $\mu > 0$ *such that for every extension of* $\vec{\gamma}^-$ *to a 2n-dimensional vector* $\vec{\gamma}$ *of positive reals with* $\gamma_{i+n} < \mu$ *for all* i*, the systems* M_0 *and* $D(f, \vec{\gamma})$ *are* **strongly** *consistent.*

Theorem

Let M_0 be a Boolean system with a **one-stepping** *updating function $f : \{0,1\}^n \to \{0,1\}^n$ and let $\vec{\gamma}^- = (\gamma_1, \dots, \gamma_n)$ be a fixed vector of positive reals. Then there exist $\mu > 0$ such that for every extension of $\vec{\gamma}^-$ to a 2n-dimensional vector $\vec{\gamma}$ of positive reals with $\gamma_{i+n} < \mu$ for all i, the systems M_0 and $D(f, \vec{\gamma})$ are* **strongly** *consistent.*

We have already seen that the assumption that f is one-stepping is necessary in this theorem.

A more general Theorem

We also have a more technical notion of monotone-stepping Boolean functions. All one-stepping Boolean functions are monotone-stepping, but not *vice versa*.

Theorem

Let M_0 be a Boolean system with a **monotone-stepping** *updating function $f : \{0,1\}^n \rightarrow \{0,1\}^n$ and let $\vec{\gamma}^- = (\gamma_1, \ldots, \gamma_n)$ be a fixed vector of positive reals. Then there exist $\mu > 0$ such that for every extension of $\vec{\gamma}^-$ to a $2n$-dimensional vector $\vec{\gamma}$ of positive reals with $\gamma_{i+n} < \mu$ for all i, the systems M_0 and $D(f, \vec{\gamma})$ are consistent.*

Open problems: Possible extensions

- Some additional assumption on f is needed in the last theorem, but the assumption that f is monotone-stepping is too strong. It remains open to find a necessary and sufficient condition on f for which the conclusion of the last theorem holds.

Open problems: Possible extensions

- Some additional assumption on f is needed in the last theorem, but the assumption that f is monotone-stepping is too strong. It remains open to find a necessary and sufficient condition on f for which the conclusion of the last theorem holds.
- The particular form of the DEs in our class $\mathbb{D}$ allowed us to find the proofs of the above theorems, but the argument really seems to require only a particular type of interacting bifurcations. It remains to formulate and prove versions of the theorems in such a more general form.

Directions for Further Research

How about arbitrary functions? The second theorem can be extended to some, but not all Boolean functions. But we know (not yet published) that of M_0 is an arbitrary Boolean system, then M_0 is consistent with some ODE system M_1.

Directions for Further Research

How about arbitrary functions? The second theorem can be extended to some, but not all Boolean functions. But we know (not yet published) that of M_0 is an arbitrary Boolean system, then M_0 is consistent with some ODE system M_1.

This follows from the observation that every Boolean system can be embedded into a one-stepping Boolean system with additional Boolean variables, which allows to construct $M_1 \in \mathbb{D}$, but with a more complicated relationship between M_0 and M_1. In effect, M_1 will have a lot more intermediary variables.

Directions for Further Research

How about arbitrary functions? The second theorem can be extended to some, but not all Boolean functions. But we know (not yet published) that of M_0 is an arbitrary Boolean system, then M_0 is consistent with some ODE system M_1.

This follows from the observation that every Boolean system can be embedded into a one-stepping Boolean system with additional Boolean variables, which allows to construct $M_1 \in \mathbb{D}$, but with a more complicated relationship between M_0 and M_1. In effect, M_1 will have a lot more intermediary variables.

The particular forms of the DEs for the intermediary variables may not matter all that much. We can think of M_1 as a network of intermediary (signaling) and signature variables.

Directions for Further Research

How about arbitrary functions? The second theorem can be extended to some, but not all Boolean functions. But we know (not yet published) that of M_0 is an arbitrary Boolean system, then M_0 is consistent with some ODE system M_1.

This follows from the observation that every Boolean system can be embedded into a one-stepping Boolean system with additional Boolean variables, which allows to construct $M_1 \in \mathbb{D}$, but with a more complicated relationship between M_0 and M_1. In effect, M_1 will have a lot more intermediary variables.

The particular forms of the DEs for the intermediary variables may not matter all that much. We can think of M_1 as a network of intermediary (signaling) and signature variables.

What structural properties of this network favor or imply consistency?