A First-Order Axiomatization for Transition Learning with Rich Constraints

Jáchym Barvínek and Filip Železný

Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Computer Science
{barvijac,zelezny}@fel.cvut.cz

Abstract. We address the task of learning a dynamic Boolean network model from data about its state transitions, and constraints regarding the known attractors of the system. To this end, we propose a learning strategy where such prior knowledge is compiled into a first-order theory along with axioms describing the Boolean transitions functions and the language bias for their representation. The learning task is then posed as a model-finding task, in that the Boolean network is obtained as a model of the input first-order theory. With this framework, we support experimentally the hypothesis that attractor constraints reduce the number of state transition examples needed to identify the target model.

1 Introduction

Recent explorations of molecular and genetic networks have stimulated research into the dynamics of Boolean network models [6]. Dynamic Boolean models have been proposed for numerous biological processes including cell cycle regulation or cell differentiation; see e.g. [3] for references to the primary sources. Some classical techniques of dynamic systems theory such as state-space analysis have been reformulated for the special Boolean domain [4, 1].

A central task in the biological context is to learn a Boolean network from observations of its state-to-state transitions. This was formalized by [5] as learning a logic program from pairs of successive logical interpretations. While [5] frames the task in the first-order setting, the learning problem essentially amounts to propositional classification. In particular, learning examples are truth-value assignments to a finite set of propositional variables at times k and $k + 1$, for a finite set of time instants k. The goal is to learn a formula for each variable that determines its truth-value at time $k + 1$ as a function of the truth values of all variables at time k. It is assumed that the functions are time-invariant, i.e. they do not depend on k explicitely.

Collecting a data set of state transitions is highly challenging from the biological viewpoint. Although certain kinds of events captured by Boolean variables are quite amenable to measurement (e.g. the expression of a gene), other important events are much more difficult to detect (e.g. protein phosphorylation). It is thus likely that such data will be scarce in real-world biological applications. On the other hand, some global properties of the real system's dynamics can be observed easily, yet they have not been so far considered a possible input for learning. For example we may observe that the cell always ends up in apoptosis if evolving from one of a certain set of initial conditions, or as a result of a certain class of perturbations. Such observed properties translate readily into inductive constraints positing that the model to be learned has a specific attractor state. There are likely many other kinds of global dynamic properties that may form useful constraints, e.g. bounds on the maximum number of different states visited, etc.

Our specific goal for this initial exploration is to test the hypothesis that by exploiting prior knowledge about attractors we may reduce significantly the number of transition examples needed to identify the target model. To test this, we need a learning framework capable to interpret constraints such as *the model has exactly one periodic attractor of proper length 2.* Such a framework is our secondary contribution. In particular, we design a framework in which the learned Boolean network can be obtained as a finite model of a first-order logic theory with equality. The theory consists of common axioms defining the semantics of propositional programs, problem-specific sentences expressing the global attractor constraints, and finally the known ground transitions.

2 Problem Setting

We consider propositional variables $V = \{p_1, \dots p_n\}$. We are given a finite set of assignments (intepretations) $\mathcal{I} = \{I_{k_1} \dots, I_{k_m}\}$ associated with discrete times $k_i \in N$, where each $I_k : V \to \{0, 1\}$. The model we seek consists of n propositional formulas ϕ_i formed with variables in V. The formulas define (synchronous) state transitions, that is to say, p_i is true at time $k + 1$ iff ϕ_i is satisfied under the truth values of all variables at time k. The model thus *fits* $\mathcal{I}$ iff for any $I_k, I_{k+1} \in \mathcal{I}$, $I_{k+1}(p_i) = 1$ iff $I_k \models \phi_i$. Denoting $p_i(k) \equiv I_k(p_i)$, we can express the transitions as $p_i(k + 1) = \phi_i(p_1(k) \dots p_n(k))$, treating ϕ_i as a Boolean function.

In the rest of the paper we assume that the formulas ϕ_i are literal conjunctions, and we adhere to the illustrative example 2 from [5], which is as follows

$$p(k+1) = q(k) \tag{1}$$

$$q(k+1) = p(k) \wedge r(k) \tag{2}$$

$$r(k+1) = \neg p(k) \tag{3}$$

This Boolean network has two attractors. One is the stable state where only r is true, and the other alternates between the state where only q is true and one with only p and r true.

In the sequel, we explore how the model (i.e., the three formulas above) can be identified from $\mathcal{I}$ and from prior constrains regarding the model's attractors. The strategy adopted is that the latter two, along with axioms regarding the syntax and semantics of propositional conjunctions are combined in a first-order theory T. The Boolean networks conforming to the said inputs are obtained as finite models of T. We next present two different approaches to such axiomatization.

3 Relational Axiomatization

Here, the vocabulary for forming T includes a constant for each propositional variable in V, so for the running example, the constants are $\mathsf{p}, \mathsf{q}, \mathsf{r}$. Further, we have a finite number of constants $\mathsf{i}_1, \mathsf{i}_2, \ldots$ corresponding to interpretations $I_1, I_1, \ldots$ which we are also going to call *states*. With the running example, we have up to 8 such constants, depending on the set of states present as examples in a particular experimental setting. Finally, we have a function symbol s representing the state *succession* function. E.g. $\mathsf{s}(\mathsf{i}_1) = \mathsf{i}_2$ stipulates that state I_2 directly follows state I_1. As for predicates, we include $\mathrm{pos}/2$, $\mathrm{neg}/2$, and $\mathrm{true}/2$ whose meaning we explain through the running example. The meaning of $\mathrm{pos}/2$ is that if for instance $\mathrm{pos}(\mathsf{q}, \mathsf{p})$ holds, then q appears as a positive literal in the conjunction for p such as in (1). Similarly, $\mathrm{neg}/2$ indicates a negative occurrence. Lastly, $\mathrm{true}(\mathsf{i}_3, \mathsf{q})$ means that q is true in (assigned 1 by) interpretation I_3.

The first axiom in T defines the semantics of conjunctive transitions. In particular, it expresses that if and only if a variable is true in the successor of a state S, then all variables with positive (negative) occurrence in the conjunction must be true (false) in S.

$$\forall S, V\ \mathrm{true}(\mathsf{s}(S), V) \leftrightarrow \\ \forall W\ (\mathrm{pos}(W, V) \rightarrow \mathrm{true}(S, W)) \wedge (\mathrm{neg}(W, V) \rightarrow \neg \mathrm{true}(S, W)) \quad (4)$$

The second axiom encodes the language bias for the propositional models by restricting the pos and neg relations to the constants $\mathsf{p}, \mathsf{q}, \mathsf{r}$.

$$\forall V, W\ (\mathrm{pos}(V, W) \vee \mathrm{neg}(V, W)) \rightarrow \\ (W = p \vee W = q \vee W = r) \wedge (V = p \vee V = q \vee V = r) \quad (5)$$

The remaining formulas in T describe the particular learning instance. Each supplied interpretation I_k is encoded as a set of ground facts; e.g. $\mathrm{true}(\mathsf{i}_1, \mathsf{p})$ and $\neg\mathrm{true}(\mathsf{i}_1, \mathsf{q})$ express respectively that p (q) is (not) true in I_1. Further facts in the form $\mathsf{s}(\mathsf{i}_2) = \mathsf{s}_3$ postulate the known successions of states. The last group of formulas describe the known facts about attractors. For example the formula

$$\exists S\ \mathsf{s}(S) = S. \quad (6)$$

stipulates, under the presence of (4), that the model has an attractor state (a state transiting to itself), whereas

$$\exists S\ \neg(\mathsf{s}(S) = S) \wedge \mathsf{s}(\mathsf{s}(S)) = S. \quad (7)$$

posits that it has a periodic attractor of proper length 2 (i.e. two alternating states).

In general, T has multiple finite models M, $M \models T$, which all represent a correct solution to the learning problem. M *identifies* the target system (1-3) iff it assigns the following relations to the predicates $\mathrm{pos}/2$ and $\mathrm{neg}/2$:

$\mathrm{pos}(\downarrow, \rightarrow)$	$M(\mathsf{p})$	$M(\mathsf{q})$	$M(\mathsf{r})$
$M(\mathsf{p})$	0	1	0
$M(\mathsf{q})$	1	0	0
$M(\mathsf{r})$	0	1	0

$\mathrm{neg}(\downarrow, \rightarrow)$	$M(\mathsf{p})$	$M(\mathsf{q})$	$M(\mathsf{r})$
$M(\mathsf{p})$	0	0	1
$M(\mathsf{q})$	0	0	0
$M(\mathsf{r})$	0	0	0

Here, $M(\mathsf{p})$ stands for the domain element that p is mapped to by M, and analogically so for q and r. The tables do not show the Cartesian products for other domain elements as they are all 0 due to axiom (5).

4 Functional Axiomatization

Our alternative axiomatization uses no predicates except for the special equality predicate. We introduce two constants t, f representing the *true* and *false* Boolean values, and the following axiom making sure the two are different

$$\neg(\mathsf{f} = \mathsf{t}) \tag{8}$$

Conjunctions are modeled through the function $\mathsf{conj}/9$ satisfying the next axiom

$$\begin{aligned} \forall P_p, P_q, P_r, N_p, N_q, N_r, I_p, I_q, I_r\ \mathsf{conj}(P_p, P_q, P_r, N_p, N_q, N_r, I_p, I_q, I_r) = t \\ \leftrightarrow ((P_p = \mathsf{t} \rightarrow I_p = \mathsf{t}) \wedge (N_p = \mathsf{t} \rightarrow I_p = \mathsf{f}) \wedge \\ (P_q = \mathsf{t} \rightarrow I_q = \mathsf{t}) \wedge (N_q = \mathsf{t} \rightarrow I_q = \mathsf{f}) \wedge \\ (P_r = \mathsf{t} \rightarrow I_r = \mathsf{t}) \wedge (N_r = \mathsf{t} \rightarrow I_r = \mathsf{f})) \end{aligned} \tag{9}$$

where the P (N, respectively) arguments indicate the presence of the subscripted variable in a positive (negative) literal in the conjunction, and the I arguments correspond to the truth-values of the subscripted variables. So for example, $\mathsf{conj}(\mathsf{t}, \mathsf{f}, \mathsf{t}, \mathsf{f}, \mathsf{f}, \mathsf{f}, \mathsf{t}, \mathsf{f}, \mathsf{t}) = \mathsf{t}$ is a consequence of axiom (9).

To model the transitions (1-3), we introduce three functions $\mathsf{p}/3, \mathsf{q}/3, \mathsf{r}/3$ and postulate that they must coincide with some conjunction

$$\begin{aligned} &\exists P_p, P_q, P_r, N_p, N_q, N_r \quad \forall I_p, I_q, I_r \\ &\mathsf{p}(I_p, I_q, I_r) = \mathsf{conj}(P_p, P_q, P_r, N_p, N_q, N_r, I_p, I_q, I_r) \end{aligned} \tag{10}$$

The remaining axioms are again specific for a learning instance. The known state transitions are encoded through ground facts such as

$$\mathsf{p}(\mathsf{t}, \mathsf{f}, \mathsf{t}) = \mathsf{f}$$

where the arguments stand for the truth values of p, q, r in the source state and the function value corresponds to the truth-value of the indicated variable in the successive state.

The following constraint expresses that there is an attractor state transiting to itself

$$\exists I_p, I_q, I_r\ \mathsf{p}(I_p, I_q, I_r) = I_p \wedge \mathsf{q}(I_p, I_q, I_r) = I_q \wedge \mathsf{r}(I_p, I_q, I_r) = I_r$$

To encode the existence of higher-order attractors, we proceed in a spirit analogical to the extension of (6) towards (7).

Any model found for the theory consisting of the above formulas determines the transition rules by a full instantiation of the $\mathsf{p}/3, \mathsf{q}/3, \mathsf{r}/3$ functions. The propositional formulas defining the function are determined from the Skolem constants assigned by the model to the existentially quantified variables in (10).

5 Experiments

All experiments consists of finding finite models of theories T using the model finder Paradox [2], with the aim to identify the Boolean model (1-3).

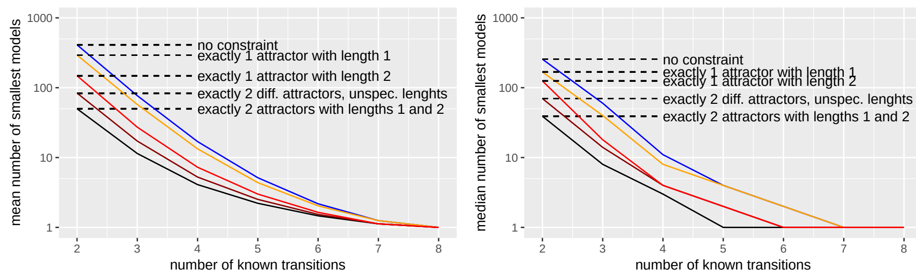

Fig. 1. Mean (left) and median (right) numbers of models (in logarithmic scale) found for theories containing different numbers of observed transitions and different attractor constraints described in the legend. Lower is better.

All experiments with the relational axiomatization include the common axioms (4-5) in T. Similarly, all the functional axiomatization experiments use (8-10). For each axiomatization, we considered five classes of theories, differing in the attractor constraint added to T. The five different constraints are listed as legends in the plots in Fig. 1. For each of the 10 theory classes, we produced a number of theories differing in the number of transition examples encoded in the theory. In particular, we produced a theory for each subset of size at least 2 of the Boolean state space, which has $2^3 = 8$ elements.

For each experiment (i.e. each theory T) multiple finite models are found in general. All resulting models of the functional axiomatization have the same domain size 2. The relational axiomatization produces models of different domain sizes and we discard all models larger than the smallest one.

The resulting model set always contains the model corresponding to (1-3). The *success measure* of the identification task is thus inversely proportional to the total number of models found (this corresponds to the chance of picking the right model from the resulting model set). We report[1] the mean and median of the number of models in Fig. 1. It is evident that a strong attractor constraint reduces the model uncertainty significantly; for the smallest considered number 2 of observed transitions, this can be almost by an order of magnitude in comparison to using no attractor constraint.

To interpret the results comparatively, note that the learning approach from [5] needs all 8 observed transitions to identify the model. Complete sets of transitions are also required by other popular approaches to Boolean network learning [1].

6 Discussion and Conclusion

Although only with a simple example, we have found empirical support for the hypothesis that by exploiting for prior knowledge on the character of attractors of a Boolean network, such a network can be identified with a smaller number of state-transition examples.

[1] These statistics pertain to the functional approach; the relational one exhibited almost identical trends.

To account for such attractor constraints in the learning process, we formulated a first-order framework for learning propositional formulas. Here, the learned network model is encoded in a finite model of a first-order theory, which specifies the learning instance. Interestingly, this setting contrasts with the well known ILP setting of learning from interpretations, where models represent input data and a first-order theory is searched.

This present framework is strong and allows more complex learning scenarios beyond those exemplified in the preceding sections. Virtually any part of the input theory can be missing (parts of known state transitions, entire transitions, knowledge on state successions, etc.) with the only consequence that the theory will allow a greater number of models. Conversely, any partial knowledge on the structure of the unknown Boolean network (the presence of a certain literal in a rule, entire rule, etc.) can be easily encoded in the input theory. Similar flexibility pertains to the language bias: while our axioms postulated conjunctions, they can be straightforwardly extended to DNF's or other syntactical structures. Also, more interesting and complex constraints regarding attractors can be specified, which we do not demonstrate here due to limited space.

For this generality, we will likely pay with limited scalability. Although the reported runtimes for the running example used in this study was typically 0s with the Paradox model finder, the perspectives for larger tasks are rather dire, since the latter algorithm is based on a reduction to SAT preventing the exploitation of domain-specific heuristics. We thus plan to develop specialized algorithms for Boolean network learning, yet still able to interpret at least the rich attractor constraints.

On the other hand, the first-order axiomatization approach presented here could scale up to a slightly different flavor of Boolean network learning tasks, in which one would not aim at identifying a single ground Boolean model. In particular, using the axiomatization approach we could still reason through *lifted inference* about the properties of the Boolean network models conforming to prior knowledge (transitions, attractors) without the need to construct specific ground models. Indeed, such reasoning could be implemented by deriving consequences from the input theory through proof finding.

References

1. Cheng, D., Qi, H.: State–space analysis of boolean networks. IEEE Transactions on Neural Networks 21 (2010)
2. Claessen, K., Sörensson, N.: New techniques that improve MACE-style finite model finding. In: Proceedings of the CADE-19 Workshop: Model Computation-Principles, Algorithms, Applications (2003)
3. Dubrova, E., Teslenko, M.: A SAT-based algorithm for finding attractors in synchronous boolean networks. IEEE/ACM Trans. Comput. Biol. Bioinformatics 8 (2011)
4. Hochma, G., Margaliot, M., Fornasini, E., Valcher, M.E.: Symbolic dynamics of boolean control networks. Automatica 49 (2013)
5. Inoue, K., Ribeiro, T., Sakama, C.: Learning from interpretation transition. Machine Learning 94 (2014)
6. Novere, N.L.: Quantitative and logic modelling of molecular and gene networks. Nature Reviews Genetics 16 (2015)