

Reconstructing the rules of 1D cellular automata using closure systems ^{*}

José L. Balcázar, Gemma C. Garriga, and Pablo Díaz-López

LARCA Research Group
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Campus Nord, Jordi Salgado 1-3
08034 Barcelona, Spain
{balqui,gcasas,pdiaz}@lsi.upc.edu

Abstract. We consider the problem of identifying the rules conforming the local map of a cellular automaton; we explore the capabilities of a closure-based algorithm for this task. The algorithm has been previously proven to identify an optimal Horn-like formula true for the data, in a very precise mathematical sense. A key property of the algorithm is its ability to handle a sequential structure on the data and lift it to the Horn-like rules, thus making it apt to compare the rules it obtains with the ones that originated the data. The outcomes of the experimentation are described.

Keywords: cellular automaton, propositional Horn logic, association rules.

1 Introduction

Closure systems form a very basic mathematical concept, related to many applications. In the field of Formal Concept Analysis, closure systems have been used widely to represent knowledge and to infer rules from data; and some extensions of this work, such as the notion of confidence-based association rules, became recently a cornerstone of the field of Data Mining. In the recent years, the authors have developed appropriate notions of closure systems for the analysis of structured data, notably in the form of sequential itemsets. Up to now, such systems, and similar ones, have been used to model various sorts of data found in diverse application domains; most notably, web-based and user interaction data, and other technical information. We have also constructed, under the DELIS Integrated Project of the Complex Systems Initiative, an implementation in the form of a research toolkit, the ISSA system, that includes algorithms for the analysis of sequential closure systems, one of which is our generalization of deterministic association rules to sequential data. Here we describe the application of this sort of analysis to cellular automata.

Cellular automata are computational systems based on sets of simple rules, introduced back in the 1940's. They operate on a space divided into cells, organized into a regular structure (usually, low-dimensional rectangular or hexagonal grids or tori) with a clear, uniform notion of neighborhood of each cell. Cells may change state among a (usually small) number of states, according to the so-called *local map*, the set of rules that govern the evolution of the system along discrete time steps. Each rule in the local map specifies the change of state

^{*} This work is partially supported by the European Commission - Fet Open project DELIS IST-001907 Dynamically Evolving Large Scale Information Systems and by MCYT TIC 2002-04019-C03-01 (MOISES).

(or absence of change) of a cell on the basis of the configuration of states of cells in the neighborhood of the cell itself.

For instance, in an one-dimensional grid, the state of a cell at a given time will depend on the states of a number of cells in the previous moment: the cell itself, the k cells to its right, and the k cells to its left. These $2k + 1$ cells constitute the neighborhood. These 1D cellular automata are frequently employed to create “textures” on a 2D space or image, by choosing one row of pixels (usually the topmost or the leftmost one) and drawing successive rows of pixels according to the evolution of the cellular automaton; some famous geometries and many interesting evolutions can be obtained in this way, including models of physical phenomena, such as heat-flow and turbulence, as well as computer-generated 2D textures, some of which reproduce extremely well human perceptions of real textures. Let us just mention as a simple example the Pascal triangle of (the parity of the value of) binomial coefficients, showed in figure 1. It is readily verified that this picture is obtained as the evolution of a very simple cellular automaton whose cells can be in two states (ON and OFF, say), and where the local map specifies that a cell is ON exactly if, in the previous time step, exactly one of the two immediate neighbors was ON. The start configuration has exactly one cell ON.



Fig. 1. Pascal’s triangle of binomial coefficients obtained with a cellular automaton

Using the AND boolean function instead of the parity for combining the values of the two neighbors gives Wolfram’s well-known AND automaton; slightly more involved rules are able to create extremely sophisticated behavior, including self-reproduction. Even insisting on limited 2D grids with just two states per cell and a neighborhood relation that considers, for each cell, the eight immediate cells surrounding it, the very simple and famous rules of Conway’s LIFE already give rise to a fascinating system of which a major fact has been proved: it can simulate arbitrary computations and has full Turing computational power. And, with a similar neighborhood and the apparently small extension of cells with 3 states, the amazing evolutions of Brian’s Brain are well worth admiration. The Modern Cellular Automata web page provides plenty of material to read, learn, or just enjoy with the visualization of cellular automata at work.

Here, we focus our attention on the problem of learning the set of hidden rules that run the evolution of a cellular automaton. Starting from the sequence of evolutions through the time and considering that the set of rules that generated such evolution is not known, we would like to discover an approximation of those rules. There are some previous works along this sort of analysis. Several of them attempt at modeling textures generated by a sweeping 1D automaton, by identifying “coherent structures” along the spatiotemporal distributions provided by the evolving system, constructing filtering systems for detecting specific phenomena in these evolutions (see [1] and the references there).

We rather take a somewhat different standpoint that the problem of synthesizing the unknown local map rules from the behavior of the system can be seen as a learning problem

that can be addressed with the techniques of knowledge discovery. Previous works (some of them very recent, which shows the timeliness of our study) include a few cases of statistical analysis (such as fitting hidden Markov models via expectation-maximization [2] or applying a Minimum Description Length approach to approximate a stochastic form of cellular automata with probabilistic local maps [3]), some proposals based on genetic algorithms (such as [4] and [5]) and, closer to our case, the use of Data Mining techniques for rule induction [6]. In this paper, we will study the sequence of evolutions of a cellular automaton by means of a recent variant of association rules that is particularly well-tailored to the study of evolving systems, namely, those obtained from a notion of closure operator on sequentially structured data proposed recently by the present authors [7].

Specifically, we proved there that a novel closure system defined in [8], appropriately employed, can extract from sequential data a family of implicational rules that can be mathematically characterized in terms of Horn logic for a propositionalization of the sequential setting, as the empirical Horn approximation of the data, that is, the set of Horn rules that minimally describe the given set of evolutions. Our question now is to what extent these rules can uncover the hidden function that governs the evolution of a cellular automaton. Given that the ISSA system alluded to above implements the corresponding algorithmics so that we are indeed able to operate on real data through these conceptual mechanisms, we are in a position to gather some experimental evidence of the strengths and weaknesses of this data analysis method for the task of reconstructing rules of cellular automata.

The setting of our work is as follows: given is a (large enough) cellular automata, constructed by ourselves so that the rules that govern it are known; the local evolution of a small neighborhood is extracted for each cell, and the data obtained in this form is fed into our ISSA system. Properly handled, the outcome is a set of rules. Then we compare these rules to the ones that actually were used to construct the cellular automaton. Success is defined when the two sets of rules are logically equivalent; but there is absolutely no guarantee of such success, since ISSA was conceived for the analysis of extremely different data.

2 Horn rules, closure systems, and sequences

Assume a standard propositional logic language with a finite set of propositional variables. A literal is either a propositional variable, called a positive literal, or its negation, called a negative literal. A clause is a disjunction of literals. A clause is *Horn* if and only if it contains at most one positive literal. A Horn formula is a disjunction of Horn clauses. A *model* is a complete truth assignment, i.e. a mapping from the variables to $\{0, 1\}$. A set of models is Horn if there is a Horn formula which axiomatizes it, in the sense that it is satisfied precisely by the models in the set.

Then, the minimal Horn set of models including a given set is known as the empirical Horn approximation. A Horn formula axiomatizing it can be constructed as the conjunction of all the Horn formulas that are true of all the models of the original given set.

A Horn formula defines a closure system on the variables. The closure of a set of variables is formed by all those variables that are consequences of those in the set through the implications indicated in the Horn formula. Dually, from a closure system, there are a number of ways of obtaining implications that actually have been proved to correspond to a Horn formula axiomatizing the empirical Horn approximation.

2.1 Identifying Horn rules from a set of sequences

In this section we revisit the results presented in [7]. There, a notion of deterministic association rules is defined from the Galois connection framework of [8]. The set of all the rules obtained from this process turns out to define exactly the natural extension of the notion of empirical Horn approximation to a set of sequences.

More specifically, we are given an input set of sequences $S = \{s_1, \dots, s_n\}$, where each sequence in this set S is defined to be an ordered list of sets of variables $s_i = \langle (V_1)(V_2) \dots (V_m) \rangle$; with this notation we mean that the set of variables V_i occurs before the set V_j for $i < j$.

This set S can be transformed into a lattice of closed sets of sequences (see [8, 7]); and from there, the work in [7] derives a proper notion of generator for each closed set. The idea is that each generator will correspond to the antecedent of a rule, and its closure to the implied the consequent. Based on this formalization, it is possible to derive the notion of association rules that deterministically hold for all the sequences $s_i \in S$. These association rules have the form $S' \rightarrow s$, where S' is a set of sequences (i.e. the generators), and s is a single sequence being the consequent.

The main property of these rules is that they hold in all the input sequences s_i , that is, for each $s_i \in S$ containing all the sequences S' , it holds that $s \subseteq s_i$ as well. It can be proven that these proposed rules can be formally justified by a purely logical characterization, namely, a natural notion of empirical Horn approximation for ordered data. The algorithmic procedure to come up with such Horn implications is discussed in [7].

3 Horn rules in cellular automata

Whereas the natural application of Horn rules for sequential data is in the data mining realm of the analysis of ordered transactional data, here we tackle the somewhat more challenging problem of using it for the analysis of a very different form of information: the evolutions of cellular automata. Below we discuss in some more depth the reasons why our method could encounter difficulties in the analysis we develop. We use the ISSA implementation of the method of analysis described in the previous section. This implementation offers the additional possibility of discarding those Horn rules whose frequency of apparition in the data is below a user-tunable threshold. Several values for this threshold have been used in the experiments.

3.1 Details of the setting

Our results in this paper are still rather preliminary. We have taken the following initial working assumptions:

- we limit ourselves to two-state 1D automata where the state of each cell depends only on the states of the two neighbors in the previous generation, which is a simplification that still leaves in the picture well-known complex systems such as Wolfram’s AND automaton and also the parity automaton that is able to trace the self-similarity structure of the Pascal triangle;
- we assume the initial generation to be a random initialization (which is always the same for each size to guarantee reproducibility);

- each step is encoded by the previous and current states of the left neighbor (l or L), of the right neighbor (r or R) and of the cell itself (c or C), so that each piece of data that the algorithms receive have a form similar to $[(l,c,R),(l,C,r)]$, meaning that at some evolution of the system, at one particular spot, the cell goes from state False (c) to True (C), and at the same time the left neighbor was and remains in state False (l), and the right neighbor changes from True (R) to False (r).

Under these circumstances, we analyze the data repeatedly adjusting a number of parameters:

- the number of cells,
- the boolean function that updates the state,
- the number of generations, and
- the threshold of significancy, the internal parameter of ISSA mentioned above, whose operation indicates the system that a configuration which appears with an empirical probability below the threshold should be disregarded.

3.2 Results

There are a number of reasons why ISSA must be expected to have difficulties in finding the successful rules. First, it is not informed that the cell data of the second 3-tuple depends on the left and right data on the first, nor that all the other states depend on information that is not available to it. Second, although actually the next piece of data it receives corresponds to the right neighbor configuration and pieces of data are correlated, the algorithm is not informed of this fact.

Third, and more seriously, ISSA works in a purely propositional form so that the rules it is able to output are restricted to Horn clauses over partial orderings labeled by the literals l, c, r, L, C, R. Thus, in principle, ISSA does not have explicitly enough expressive power to say that the state of the cell becomes, say, the parity of the states of the neighbors; as we see in a moment, ISSA provided us with the surprise of finding by itself a way of expressing the necessary correlations.

Fourth, and finally, given that the initial configuration is random, there is no guarantee that all the potential instances of the rule employed (that is, all possible combinations of states of the left and right neighbor) appear in the evolution of the system with a frequency above the significancy threshold.

Of all these difficulties, it turned out that the analysis power of ISSA did not seem to be affected by the first two; and that it gave us a way of encoding, in a form developed by the program itself, the rules of the automaton, under the appropriate values of the parameters. The fourth one, though, was the most relevant one: if the significancy threshold is set too high, the description of the rules easily misses cases that were not frequent enough in the specific evolution under analysis, whereas when it is lowered too much, we start finding correlations that do not happen often enough to make sure that counterexamples are found, and thus we get as output rules that are, essentially, statistical noise.

As for the effect of the other parameters, they are as expected: more cells give better results (that is, better chances of success with respect to the random initialization), whereas more generations immediately lead the system astray into repeated failures. We found similar results for all the boolean functions (modulo symmetries due to permutation of left and right or due to negation).

As examples of how ISSA describes the rules behind the automaton, we indicate its results on a large automaton working under a disjunctive rule (that is, the state of a cell is the OR of the two states of the neighbors in the previous generation):

- if 'l c' then 'r c', meaning that if the cell is False and in the previous generation its left neighbor is False, then in the previous generation the right neighbor was False as well
- dually, if 'r c' then 'l c'
- if 'l C' then 'R C', meaning that if the cell is True and in the previous generation its left neighbor is False, then in the previous generation the right neighbor was True
- dually, if 'r C' then 'L C'

Similarly, for the AND function, we get rules such as: if 'L c' then 'r c', meaning that if in the previous generation the left neighbor was True, and the cell is False, then in the previous generation the right neighbor was False as well.

4 Conclusions

Under appropriate parameter settings, ISSA is able to extract reasonable rules from simple 1D cellular automata evolutions, even though it was originally designed for searching for very different correlations in very different datasets. Forthcoming work would consist in experimenting with more complex cellular automata (larger neighborhoods, 2D systems) just enough to glean intuitions, and then try and extend some of the ISSA features into enough first-order logic to transcend the propositional limitation, which is now by far the most restrictive facet.

It must be mentioned as well that these experiments have given us progress not only about cellular automata, but also in understanding the deep consequences of our proposals for the analysis of sequential data based on closure systems. Some of the details of the ISSA implementation were actually motivated by the experimentation we ran on cellular automata data.

References

1. Shalizi C.R., Haslinger R., Rouquier J.B., Klinkner K.L., Moore C. Automatic filters for the detection of coherent structure in spatiotemporal systems. arXiv:nlin.CG/0508001.v1. 2005.
2. Turin W. Fitting probabilistic automata via the EM algorithm. *Communications in Statistics – Stochastic Models* 12, 405–424. 1996,
3. Craiu R.V., Lee T.C.M. Pattern generation using likelihood inference for cellular automata. *IEEE Transactions on Image Processing* (to appear). 2005.
4. Richards F.C., Meyer T.P. and Packard N.H. Extracting cellular automaton rules directly from experimental data. *Phys. D*, 45:1-3, 189–202. 1990.
5. Yang Y.X., Billings S.A. Neighborhood detection and rule selection from cellular automata patterns. *IEEE Transactions on Systems*, 30:6, 840–847. 2000.
6. Li X. and Gar-On Yeh A. Data mining of cellular automata's transition rules. *International Journal of Geographical Information Science*, 18:8, 723–744. 2004.
7. Balcázar J.L., Garriga G.C. On Horn Axiomatizations for Sequential Data. *Proceedings of the 10th Int. Conference on Database Theory*, 215–229. 2005.
8. Garriga G.C. Towards a Formal Framework for Mining General Patterns from Structured Data. *Proc. of the 2nd. Multi-relational Datamining*, in *KDD Int. Conf.* 2003.