Cellular automata and probabilistic L systems: An example in Ecology

Manuel Alfonseca; Alfonso Ortega; Alberto Suárez

26th November 1999

Universidad Autónoma de Madrid. Departamento de Ingeniería Informática.
$\{manuel.alfonseca, alfonso.ortega, alberto.suarez\}$@ii.uam.es

Cellular automata. A cellular automaton is defined as the six-fold (G, G_0, N, Q, f, T), where G is a matrix of automata. G_0 is the initial state of the grid and is a mapping $G_0 : G \to Q$ an injective function that assigns an initial state to each automaton in the grid. N (neighborhood) is a function that assigns to each automaton in the grid the set of its neighbors. Q is the set of possible states of every automaton in the grid. f is the transition mapping $f : Q \times Q^n \to Q$ where $f(q_0, (q_1, ..., q_n)), n = Q$ is the next state of any automaton in the grid if its current state is q_0 and whose neighborhood's states are $(q_1, ..., q_n)$. $T \subseteq Q$ is the set of final or target states. Every automata in the grid has the same number of neighbors, transition mapping and set of possible and final states.

It is obvious that each finite automaton in the grid is defined $a = (Q^n, Q, f, G_0(a), T)$

Probabilistic Cellular Automata. Cellular automata are probabilistic if each automaton in the grid is a probabilistic finite automata.

In probabilistic cellular automata, the automata on the grid choose thir next state from a set of options, assigning probabilities to each transition while the pure non-deterministic approach only establishes the set of options.

A probabilistic cellular automaton is the six-fold (G, G_0, N, Q, M, T), where G, N, Q, T are defined as in a cellular automaton, and G_0 is the initial state of the grid and is the mapping $G_0 : G \to ([0, 1] \bigcap \Re) \, |\forall x \in$

$G \Rightarrow \sum_{i=1}^{\#Q} \prod_i \left(G_0\left(x\right)\right) = 1$, an injective function that assigns an initial state vector to each automaton in the grid. The state's vector of an automaton shows the probability that the automaton be in each state. The following notations will be used indistinctly in the following pages:

$\prod_i \left(G_0\left(x\right)\right) = \prod_{q_i} \left(G_0\left(x\right)\right)$=probability that the automaton x is in state q_i at the initial moment.

M is the transition matrix, a matrix of probabilities of transition between states, with dimension $\#Q^n \times \#Q \times \#Q$. In order to simplify the notation that M is considered a family of $\#Q^n$ square matrices $\#Q \times \#Q$ (there is a matrix for each particular neighborhood configuration). Each finite probabilistic automaton a in the grid is defined $a = (Q^n, Q, M, G_0(a), T)$

Example: assume an infinite square grid. The concatenation of the row and column indices identifies the automaton at this position in the grid. Von Neumann neighborhood will be used. Automata are binary, that is $Q = \{0, 1\}$. The cellular automaton is
$pca_1 = (G_1, G_0, N_{v_N}, Q, M, T)$ where $G_1 \in M_{Z\times Z}$, is an infinite square matrix of automata around position (0,0). $G_0(x) = 0.5 \forall x \in Q$, i.e. each state is initially equiprobable. $N_{v_N} : F \rightarrow F^4 | \forall (i,j) \in Z \times Z, N_{v_N}\left(G\left[i,j\right]\right) = \left(G\left[i-1,j\right], G\left[i,j+1\right],\right.$
$G\left[i+1,j\right], G\left[i,j-1\right] right)$

$$M = \left\{ \begin{array}{cccc} \begin{bmatrix} 0.1 & 0.9 \\ 0.8 & 0.2 \end{bmatrix} & \begin{bmatrix} 0.3 & 0.7 \\ 0.4 & 0.6 \end{bmatrix} & \begin{bmatrix} 0.5 & 0.5 \\ 0.6 & 0.4 \end{bmatrix} & \begin{bmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{bmatrix} \\ \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix} & \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} & \begin{bmatrix} 0.5 & 0.5 \\ 0.6 & 0.4 \end{bmatrix} & \begin{bmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{bmatrix} \\ \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix} & \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} & \begin{bmatrix} 0.5 & 0.5 \\ 0.6 & 0.4 \end{bmatrix} & \begin{bmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{bmatrix} \\ \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix} & \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} & \begin{bmatrix} 0.5 & 0.5 \\ 0.9 & 0.1 \end{bmatrix} & \begin{bmatrix} 0.2 & 0.8 \\ 0.4 & 0.6 \end{bmatrix} \end{array} \right\}$$

where matrices are disposed from left to right and from top to down. The first matrix is M_{0000} and the last is M_{1111}.

Let us choose an automaton in the grid and name it x.

	1	
4	x	2
	3	

Assume that the five automata have the following probability vectors at a given moment (the indices identify the automata in the previous figure, the first position in the vector is the probability of be 0): $p_1 = (0.2, 0.8), p_2 = (0.6, 0.4), p_3 = (0.3, 0.7), p_4 = (0.9, 0.1), p_x = (0.1, 0.9)$

If the neighborhood configuration of automaton x were, for instance (0,1,0,0), the following matrix operation computes the next state vector for automaton x: $p_x \times M_{0100}$.

The probability of this situation is $p_1[0] \star p_2[1] \star p_3[0] \star p_4[0]$. We have to compute the equivalent probabilities for all possible neighborhood configurations and add the results, thus getting:
$p_x \times \left(\sum_{i,j,k,l \in \{0,1\}} (p_1[i] \star p_2[j] \star p_3[k] \star p_4[l]) \times M_{ijkl} \right)$ which can be expressed by means of the tensor product, where the dot operator represents the element by element matrix product.

$$p_x \times \left(\sum \left(\prod^{\tau}_{m \in \{1,2,3,4\}} p_m \right) \bullet M \right).$$

A configuration of a probabilistic cellular automaton. A configuration C of a probabilistic cellular automaton is a time dependent mapping $C(t) : F \to Q$ that assigns a state to each automaton in the grid.

The probability that a probabilistic cellular automaton (A) is in a given configuration (C) at a given moment (t) will be denoted $p_{t,A}(C)$ where t and A will be omitted whenever they are obvious from the context.

Each automaton in the grid has a state vector that shows the probability for it to be in each possible state. So, the event "the automaton is in configuration C" could be expressed as $\bigcap_{a \in G}$"the automaton is in configuration C(a)".

If v_a is the state vector for automaton a then $\pi_{C(a)}(v_a)$ represents the probability that automaton a is in state C(a):

$p_{t,A}(C) = \prod_{a \in G} \pi_{C(a)}(v_a)$

The sum of these probabilities over the set of all possible configurations must equal 1.

$p_{t,A}(C) = \sum_{C \in set-of-possible-configurations} \left(\prod_{a \in G} \pi_{C(a)}(v_a) \right) = 1$

This expression assumes that the set of possible configurations is ordered.

Bidimensional IL Systems. A bidimensional L System is an L System whose words are matrices of characters instead of linear strings. In order to clarify the notation, the following conventions will be followed:

The context will always be written before the symbol changed by the production rule. The context will be determined by a function c that generates the horizontal and vertical displacements of the context symbols with respect to the current symbol

Formally, a bidimensional $\langle K, 0\rangle$IL System is defined as the fivefold $\langle \sum, P, g, \omega, c\rangle$ where $\sum, P, g, \omega$ are defined in the usual way and $c : [1, k] \bigcap \aleph \rightarrow \{-1, 0, +1\}$.

Example: a bidimensional IL System with von Neumann neighborhood is an extended $\langle 4, 0\rangle$system whose c function is defined as follows: $c(1) = (0, +1)$, $c(2) = (+1, 0)$, , $c(3) = (0, -1)$, $c(1) = (-1, 0)$.

Graphically

	x_1	
x_4	x	x_2
	x_3	

If Moore neighborhood is used we will have an $\langle 8, 0\rangle$ bidimensional IL System with the following c function: $c(1) = (-1, +1)$, $c(2) = (0, +1)$, $c(3) = (+1, +1)$, $c(4) = (+1, 0)$, $c(5) = (+1, -1)$, $c(6) = (0, -1)$, $c(7) = (-1, -1)$, $c(8) = (-1, 0)$.

Graphically

x_1	x_2	x_3
x_8	x	x_4
x_7	x_6	x_5

Probabilistic L Systems We define a probabilistic L System as an L System where each production rule has an associated probability with the restriction that the sum of the probabilities associated to all the rules applicable to a symbol at any time must be 1.

A deterministic L System (DL) can be seen as probabilistic with a probability of 1 associated to every rule.

In a DL System, a derivation is linear. In a probabilistic L System (S), it is a tree, with a probability associated to each branch and the sum of the probabilities associated to all the branches with the same origin being 1. This tree will be called $T_n(S)$, and n is the depth of the tree.

Formally, a probabilistic L System is an L System where the rule set P has been replaced by a set of pairs $(R, p(R))$ where R is a derivation rule and $p(R)$ its probability, with the restriction that if $P' \subset P$is the set of rules applicable to a symbol at a given context, $\sum_{RinP'} = 1$

Example: Assume the following probabilistic $\langle 1 : 1\rangle$ IL System $S_2 = \langle \sum_2, P_2, 0011\rangle$ where

$\sum_2 = \{0, 1, g\}$ (g is the end marker)

$$P_2 = \left\{ \begin{array}{ll} (xsg ::= x, 1) & \forall x \in \sum_2, \forall s \in \{0,1\} \\ (gsx ::= x, 1) & \forall x \in \sum_2, \forall s \in \{0,1\} \\ (x0y ::= x, 0.3) & \forall x, y \in \{0,1\} \\ (x0y ::= y, 0.7) & \forall x, y \in \{0,1\} \\ (x1y ::= y, 0.3) & \forall x, y \in \{0,1\} \\ (x1y ::= x, 0.7) & \forall x, y \in \{0,1\} \end{array} \right\}$$

i.e. the first symbol in the string becomes its right neighbor, the last becomes its left neighbor, intermediate 0s become their left/right context with probability 0.3/0.7 and intermediate 1s do the same with probability 0.7/0.3.

Figure 1 shows the first three derivations of system S_2 indicating the probability of each branch.

Observe that the probability of reaching a node by a given path is the product of the probabilities of the branches that go from the axiom to that node along that path. The sum of the probabilities of all the nodes in the tree at a given derivation level is 1.

If a node at derivation level k may be reached by more then one path, the probability that the word is generated by a derivation of depth k is the sum of the probabilities of all the paths, computed as above. Formally, the probability that word x is generated by system S in n derivations is $p_{n,S}(x) = \sum_{\rho(D_i)=x \wedge D_i \in T_n(S)} \left(\prod_{b_j \in D_i} \left(\prod_{r_k \in b_j} p\left(r_k\right) \right) \right)$ where

D_i is a path in tree $T_n(S)$.

b_j is a branch in path D_i.

r_k is a production rule applied at branch b_j.

$\rho(D_i)$ is the result of derivation D_i.

Each path in the tree is a derivation. The expression could be written alternatively in Lindenmayer notation as follows: $p_{n,S}(x) = \sum_{\rho(D_i)=x \wedge D_i \in T_n(S)} \left(\prod_{(l,k) \in O_i} p\left(p_i\left((l,k)\right)\right) \right)$ where $p\left(p_i\left((l,k)\right)\right)$ is the probability associated to the production rule $p_i\left((l,k)\right)$.

Language generated by a probabilistic L System. Let S be a probabilistic L System. Let n be a natural number ($n \in \aleph$). Let θ be a real number ($\theta \in \Re$).

The language generated by S_θ, is defined

$$L\left(S, \theta\right) = \left\{ x \vert \exists \in \aleph : x \in L_n(S) \wedge p_{n,S}\left(x\right) \geq \theta \right\}$$

In the previous example, the language for threshold 0 is $L\left(S_2, 0\right) = \{0001, 0011, 0101, 0111, 0000, 0010, 1010, 1011, 1111\}$

Step-equivalence between probabilistic L Systems and probabilistic cellular automata. Let A be a probabilistic cellular automaton. Let S be a probabilistic bidimensional IL System.

Definition: S is step-equivalent to A if and only if

$$\forall t \in \aleph, \forall C(\text{configuration of A})\exists\theta \in \Re, w \in L(S) \text{ such that}$$

$$p_{t,L}(w) = p_{t,A}(C)$$

Probabilistic L System associated to a probabilistic cellular automaton.

Theorem: Given a probabilistic cellular automaton $A = (G, G_0, N, M, Q)$ there is an equivalent probabilistic bidimensional IL System that is step-equivalent to the cellular automaton.

Constructive proof: Consider the bidimensional $\langle n : 0\rangle$ IL System $S = \langle \sum, P, g, \omega, c\rangle$ where $\sum = Q \bigcup \{s_i\}$ and s_i will be used to express the axiom of the L System; g is a symbol not in $\sum$; P is the set of the pairs $(rules, probability)$: $\left(\overrightarrow{q}x ::= y, M_{\overrightarrow{q}}[x, y]\right)$ axiom ω is a matrix with the same dimensions as G whose elements are all equal to s_i; c and N refer to the same elements in their matrices. Rules with g use $M_{\overrightarrow{q}'}$ where $\overrightarrow{q}'$ is obtained from $\overrightarrow{q}$ by replacing g by the appropriate boundary symbol in the automaton.

It is easy to see that S is step equivalent to A .

Example:

The cellular automaton. Assume an automaton whose mean-field evolution follow the Lotka-Volterra equations for a predator (species Y, carnivorous) and a prey (species X, herbivorous), with a slight modification that accounts for the saturation of the herbivorous species

$$\begin{aligned} \frac{dN_X(t)}{dt} &= K_1 N_X(t)\left(1 - \frac{N_X(t)}{N_X^{sat}}\right) - K_2 N_Y(t) N_X(t) \\ \frac{dN_Y(t)}{dt} &= -K_3 N_Y(t) + K_4 N_Y(t) N_X(t) \end{aligned}$$

where N_X^{sat} is the saturation level of species X. The saturation term is necessary since the automaton cannot represent the unlimited growth of species X in the absence of individuals of the specie Y. The territory in which the population dynamics takes place is a regular two-dimensional square lattice with periodic boundary conditions, with only nearest neighbors displacements allowed. The solution to the inverse problem of finding the reactive rules that yield a specified set of mean-field equations has been given by Boon et al. in their extensive review on reactive lattice-gas automata. The reactive rules are encoded into a reaction probability matrix, whose entries are the probability of obtaining an outgoing configuration $\mathbf{n}^{out} = \{n_Y^{out}, n_X^{out}\}$ from a given incoming $\mathbf{n}^{in} = \{n_Y^{in}, n_X^{in}\}$. In particular, one possible prescription leading to the previous expressions in the mean-field limit is

$$\begin{aligned}
p(\mathbf{n}^{in} \to \mathbf{n}^{out}) &= h\ K_1\ n_X^{in}\ \delta\left(n_X^{out}, n_X^{in}+1\right)\ \delta\left(n_Y^{out}, n_Y^{in}\right)\ \left(1-\delta\left(n_X^{in}, m\right)\right) + \\
&h\ \left[K_2\ n_X^{in} n_Y^{in} + \frac{K_1}{N_X^{sat}} \frac{m}{m-1} n_X^{in}\left(n_X^{in}-1\right) - K_1\ n_X^{in}\ \delta\left(n_X^{in}, m\right)\right] \delta\left(n_X^{out}, n_X^{in}-1\right) \delta\left(n_Y^{out}, n_Y^{in}\right) + \\
&hK_4 n_X^{in} n_Y^{in} \delta\left(n_X^{out}, n_X^{in}\right) \delta\left(n_Y^{out}, n_Y^{in}+1\right) \left(1-\delta\left(n_Y^{in}, m\right)\right) + \\
&h\left[K_3\ n_Y^{in} - K_4\ n_X^{in} n_Y^{in} \delta\left(n_Y^{in}, m\right)\right] \delta\left(n_X^{out}, n_X^{in}\right) \delta\left(n_Y^{out}, n_Y^{in}-1\right) \quad \text{for } \mathbf{n}^{in} \neq \mathbf{n}^{out} \\
p(\mathbf{n}^{in} \to \mathbf{n}^{in}) &= 1 - \sum_{\mathbf{n}^{out} \neq \mathbf{n}^{in}} p(\mathbf{n}^{in} \to \mathbf{n}^{out}) \qquad (1)
\end{aligned}$$

where $\delta(n, n^{'})$ is a Kronecker delta (an indicator equal to 1 if $n = n^{'}$ and 0 otherwise), the inverse of h represents the reaction time-scale, and m is the maximum number of particles of a single species at a given node, which coincides with the number of channels associated to a node. In the present model $m = 4$, meaning that this is the maximum number of individuals of each species that may occupy a given node. The condition that $p(\mathbf{n}^{in} \to \mathbf{n}^{out})$ be a probability (i.e. a non-negative number in the interval $[0, 1]$) imposes restrictions on the possible values of the reaction constants K_i , h and N_X^{sat} that can be used in the simulations. In particular N_X^{sat}should be smaller or equal to m (this upper limit corresponds to full occupation of an automaton node).

Figure 2 depicts the time-evolution of the automaton for $K_1 = K_2 = K_3 = K_4 = 1$, $h = 0.0313$ (close to the maximum possible value of h). For these values of the reaction constants, there are configurations (in particular those where $n_Y^{in} = m$), for which Eq. (1) yields negative values. These negative values are set to zero. This procedure does not alter the mean-field behavior of the automaton in a significant manner, since the configurations affected appear rather infrequently. In the same plot the dynamics corresponding to the solution of the mean-field equations are also displayed. The mean-field equations in this automaton provide a very good approximation to the evolution of the species node densities, even for the largest possible values of h, although, as the figure shows, the frequency of the damped oscillations predicted by the mean-field approximation is slightly smaller than the frequency of the actual simulated time-series. These small discrepancies can once again be accounted for by the (limited) influence of correlations in the dynamics. The observation that the influence of correlations is small in this automaton is corroborated by the absence of spatial structure of the species populations. In Figure 3 we compare the results of simulations in an automaton with the same characteristics as the previous one, except that the inverse reaction time-scale is $h = 0.00313$. The mean-field approximation provides in this case an excellent description of the global population dynamics in the echosystem.

The step-equivalent IL System. Each automaton in the grid contains several individuals of each species. Let us call l=x(t) and k=y(t) the number of individuals of species x and y at time t. The previous probability prescription can be represented by means of the state diagram in figure 4, where

$$p_1 = \left\{ \begin{array}{cc} K_1 x(t) & x(t) \neq m \\ 0 & x(t) = m \end{array} \right\}, \; p_2 = \left\{ \begin{array}{cc} K_4 x(t) y(t) & y(t) \neq m \\ 0 & y(t) = m \end{array} \right\},$$

$$p_3 = max \left(0, \left\{ \begin{array}{cc} K_3 y(t) & y(t) \neq m \\ K_3 y(t) - K_4 x(t) y(t) & y(t) = m \end{array} \right\} \right),$$

$$p_4 = max \left(0, \left\{ \begin{array}{cc} K_2 x(t) y(t) + \frac{K_1}{N_X^{sat}} \frac{m}{m-1} x(t) \left[x(t) - 1 \right] & x(t) \neq m \\ K_2 x(t) y(t) + \frac{K_1}{N_X^{sat}} \frac{m}{m-1} x(t) \left[x(t) - 1 \right] - K_1 x(t) & x(t) = m \end{array} \right\} \right)$$

This probability distribution is shown in the following table, which represents the rules of the step-equivalent IL system:

$(l,k)_t$	$(l,k)_{t+1}$	$(l+1,k)_{t+1}$	$(l,k+1)_{t+1}$	$(l,k-1)_{t+1}$	$(l-1,k)_{t+1}$
(0,0)	(0,0),1	-	-	-	-
(0,1)	(0,1),1	-	-	-	-
(0,2)	(0,2),1	-	-	-	-
(0,3)	(0,3),1	-	-	-	-
(0,4)	(0,4),1	-	-	-	-
(1,0)	(1,0),1	-	-	-	-
(1,1)	$(1,1), 1 - \Sigma p_i$	$(2,1), hK_1$	$(1,2), hK_4$	$(1,0), hK_3$	$(0,1), hp_4$
(1,2)	$(1,2), 1 - \Sigma p_i$	$(2,1), hK_1$	$(1,3), 2hK_4$	$(1,2), 2hK_3$	$(0,2), hp_4$
(1,3)	$(1,3), 1 - \Sigma p_i$	$(2,3), hK_1$	$(1,4), 3hK_4$	$(1,2), 3hK_3$	$(0,3), hp_4$
(1,4)	$(1,4), 1 - \Sigma p_i$	$(2,4), hK_1$	-	$(1,3), 4h(K_3 - K_4)$	$(0,4), hp_4$
(2,0)	(2,0),1	-	-	-	-
(2,1)	$(2,1), 1 - \Sigma p_i$	$(3,1), 2hK_1$	$(2,2), 2hK_4$	$(2,0), hK_3$	$(1,1), hp_4$
(2,2)	$(2,2), 1 - \Sigma p_i$	$(3,2), 2hK_1$	$(2,3), 4hK_4$	$(2,1), 2hK_3$	$(1,2), hp_4$
(2,3)	$(2,3), 1 - \Sigma p_i$	$(3,3), 2hK_1$	$(2,4), 6hK_4$	$(2,2), 3hK_3$	$(1,3), hp_4$
(2,4)	$(2,4), 1 - \Sigma p_i$	$(3,4), 2hK_1$	-	$(2,3), 4h(K_3 - 2K_4)$	$(1,4), hp_4$
(3,0)	$(3,0), 1$	-	-	-	-
(3,1)	$(3,1), 1 - \Sigma p_i$	$(4,1), 3hK_1$	$(3,2), 3hK_4$	$(3,0), hK_3$	$(2,1), hp_4$
(3,2)	$(3,2), 1 - \Sigma p_i$	$(4,2), 3hK_1$	$(3,3), 6hK_4$	$(3,1), 2hK_3$	$(2,2), hp_4$
(3,3)	$(3,3), 1 - \Sigma p_i$	$(4,3), 3hK_1$	$(3,4), 9hK_4$	$(3,2), 3hK_3$	$(2,3), hp_4$
(3,4)	$(3,4), 1 - \Sigma p_i$	$(4,4), 3hK_1$	-	$(3,3), 4h(K_3 - 3K_4)$	$(2,4), hp_4$
(4,0)	$(4,0), 1$	-	-	-	-
(4,1)	$(4,1), 1 - \Sigma p_i$	-	$(4,2), 4hK_4$	$(4,0), hK_3$	$(3,1), hp_4$
(4,2)	$(4,2), 1 - \Sigma p_i$	-	$(4,3), 8hK_4$	$(4,1), 2hK_3$	$(3,2), hp_4$
(4,3)	$(4,3), 1 - \Sigma p_i$	-	$(4,4), 12hK_4$	$(4,2), 3hK_3$	$(3,3), hp_4$
(4,4)	$(4,4), 1 - \Sigma p_i$	-	-	$(4,3), 4h(K_3 - 4K_4)$	$(3,4), hp_4$

The step-equivalent Bidimensional IL System has as alphabet the set of all possible configurations of the nodes (the number of individuals of species x and y in the node), the previous table gives the rules, and the axiom is a matrix of symbols s_i. The context is the current symbol itself.

References

- Alfonseca, M.: *Teoría de Lenguajes, Gramáticas y Autómatas*, Promosoft, Publicaciones R.A.E.C., Madrid 1997, ISBN: 84-605-6092-9.

- Boon, J.P.; Dab, D.; Kapral, R.; Lawniczak, A.: *Lattice Gas Automata for Reactive Systems*, Physics Reports 273, p. 55-147, 1996.

- Herman, G.T.; Rozenberg, G.: *Developmental Systems and Languages*, North Holland/American Elsevier, Amsterdam 1975, ISBN: 0-7204-2806-8.

- Rozenberg, G.; Salomaa, A. (Eds.): *Lindenmayer Systems. Impacts on Theoretical Computer Science, Computer Graphics, and Developmental Biology*, Springer-Verlag, Berlin 1992, ISBN: 3-540-55320-7.