Alma Mater Studiorum · Università di Bologna

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Master in Physics

Relatore:
Prof.
Armando Bazzani

Presentata da:
Riccardo Scheda

Anno Accademico 2017-2018

Contents

List of Figures

Introduction

Chapter 1

Cancer attractors

In questo capitolo viene fatta una piccola introduzione al modello classico preda-predatore di Lotka-Volterra.

1.1

Chapter 2

Random Boolean Networks

In this chapter we explain the basic concepts of Random Boolean Network proposed for the first time by Kauffman.

Random Boolean networks (RBNs) were introduced in 1969 by S. Kauffman as a simple model of genetic systems. Each gene was represented by a node that has two possible states, on (corresponding to a gene that is being transcribed) and off (corresponding to a gene that is not being transcribed). There are altogether N nodes, and each node receives input from K randomly chosen nodes, which represent the genes that control the considered gene. Furthermore, each node is assigned an update function that prescribes the state of the node in the next time step, given the state of its input nodes. This update function is chosen from the set of all possible update functions according to some probability distribution. Starting from some initial configuration, the states of all nodes of the network are updated in parallel. Since configuration space is finite and since dynamics is deterministic, the system must eventually return to a configuration that it has had before, and from then on it repeats the same sequence of configurations periodically: it is on an attractor.

2.1 The model

Let's consider a network of N nodes. The state of each node at a time t is given by $\sigma_i(t) \in 0, 1$ with $i = 1, ..., N$. The N nodes of the network can therefore together assume

2^N different states. The number of incoming links to each node i is denoted by k_i and is drawn randomly independently from the distribution $P(k_i)$. The dynamical state of each $\sigma_i(t)$ is updated synchronously by a Boolean function Λ_i:

$$\Lambda_i : \{0,1\}^{k_i} \to \{0,1\}$$

An update function specifies the state of a node in the next time step, given the state of its K inputs at the present time step. Since each of the K inputs of a node can be on or off, there are $M = 2^K$ possible input states. The update function has to specify the new state of a node for each of these input states. Consequently, there are 2^M different update functions. Fo example let's consider a network with $K = 1$, so all the functions Λ_i receives the input from one single node. We can see the different possible functions for all the possible networks with $K = 1$:

$$one : \Lambda_i(\sigma_j) = \begin{cases} 1 \text{ if } \sigma_j = 0 \\ 1 \text{ if } \sigma_j = 1 \end{cases}$$

$$zero : \Lambda_i(\sigma_j) = \begin{cases} 0 \text{ if } \sigma_j = 0 \\ 0 \text{ if } \sigma_j = 1 \end{cases}$$

$$copy : \Lambda_i(\sigma_j) = \begin{cases} 0 \text{ if } \sigma_j = 0 \\ 1 \text{ if } \sigma_j = 1 \end{cases}$$

$$invert : \Lambda_i(\sigma_j) = \begin{cases} 1 \text{ if } \sigma_j = 0 \\ 0 \text{ if } \sigma_j = 1 \end{cases}$$

Now, in general each element receives inputs from exactly K nodes, so we have a dynamical system defined from:

$$\sigma_i(t+1) = \Lambda_i(\sigma_{i_1}(t), \sigma_{i_2}(t), ..., \sigma_{i_K}(t)). \tag{2.1}$$

So, the randomness of these network appears at two levels: in the connectivity of the network (which node is linked to which) and the dynamics (which function is attributed to which node).

2.2 Topology

For a given number N of nodes and a given number K of inputs per node, a RBN is constructed by choosing the K inputs of each node at random among all nodes. If we construct a sufficiently large number of networks in this way, we generate an ensemble of networks. In this ensemble, all possible topologies occur, but their statis- tical weights are usually different. Let us consider the simplest possible example, $N = 2$ and $K = 1$, shown in Figure 2.1. There are 3 possible topologies.

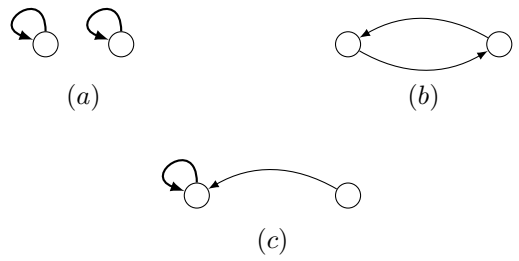

Figure 2.1: Set of all possible networks with $N = 2$ and $K = 1$.

Topologies (a) and (b) have each the statistical weight 1/4 in the ensemble, since each of the links is connected in the given way with probability 1/2. Topology (c) has the weight 1/2, since there are two possibilities for realizing this topology: either of the two nodes can be the one with the self-link.

While the number of inputs of each node is fixed by the parameter K, the number of outputs (i.e. of outgo- ing links) varies between the nodes. The mean number of outputs must be K, since there must be in total the same number of outputs as inputs. A given node becomes the input of each of the N nodes with probability $\frac{K}{N}$. In the

thermodynamic limit $N \to \infty$ the probability distribution of the number of outputs is therefore a Poisson distribution:

$$P_{out}(k) = \frac{K^k}{k!} e^{-K}$$

2.3 Dynamics

All nodes are updated at the same time according to the state of their inputs and to their update function. Starting from some initial state, the network performs a trajectory in state space and eventually arrives on an *attractor*, where the same sequence of states is periodically repeated. Since the update rule is deterministic, the same state must always be followed by the same next state. If we represent the network states by points in the 2^N-dimensional state space, each of these points has exactly one output, which is the successor state. We thus obtain a graph in state space. The size or length of an attractor is the number of different states on the attractor. The basin of attraction of an attractor is the set of all states that eventually end up on this attractor, including the attractor states themselves. The size of the basin of attraction is the number of states belonging to it. The graph of states in state space consists of unconnected components, each of them being a basin of attraction and containing an attractor, which is a loop in state space. The transient states are those that do not lie on an attractor. They are on trees leading to the attractors.

Let us illustrate these concepts by studying the small $K = 1$ network shown in Figure 2.2, which consists of 4 nodes:

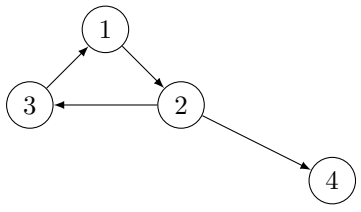

Figure 2.2: A small network with $N = 4$ and $K = 1$.

If we assign to the nodes 1,2,3,4 the functions invert, invert, copy, copy, an initial state 1111 evolves in the following way:

$$1111 \rightarrow 0011 \rightarrow 0100 \rightarrow 1111$$

This is an attractor of period 3. If we interpret the bit se- quence characterizing the state of the network as a number in binary notation, the sequence of states can also be written as

$$15 \rightarrow 3 \rightarrow 4 \rightarrow 15$$

The entire state space is shown in Figure 2.3:

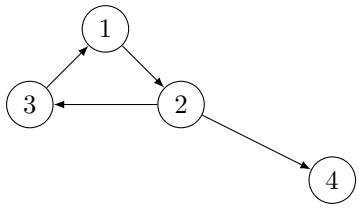

Figure 2.3: The state space of the network shown in Figure 2.2, if the functions copy, copy, invert, invert are assigned to the four nodes. The numbers in the squares represent states, and ar- rows indicate the successor of each state. States on attractors are shaded.

There are 4 attractors, two of which are fixed points (i.e., attractors of length 1). The sizes of the basins of attraction of the 4 attractors are 6,6,2,2. If the function of node 1 is a constant function, fixing the value of the node at 1, the state of this node fixes the rest of the network, and there is only one attractor, which is a fixed point. Its basin of attraction is of size 16. If the functions of the other nodes remain unchanged, the state space then looks as shown in Figure 2.4

Before we continue, we have to make the definition of attractor more precise: as the name says, an attractor attracts states to itself. A periodic sequence of states (which we also call cycle) is an attractor if there are states outside the attractor that lead to it. However, some netaworks contain cycles that cannot be reached from any state that is

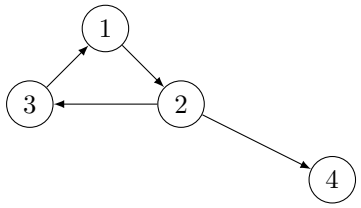

Figure 2.4: The state space of the network shown in Figure 2.2, if the functions 1, copy, invert, invert are assigned to the four nodes.

not part of it. For instance, if we removed node 4 from the network shown in Figure 2.2, the state space would only contain the cycles shown in Figure 2 C, and not the 8 states leading to the cycles. In the follow- ing, we will use the word cycle whenever we cannot be confident that the cycle is an attractor.

2.4 Applications

Let us now make use of the definitions and concepts introduced in this section in order to derive some results concerning cycles in state space. First, we prove that in an ensemble of networks with update rule 1 (biased functions) or rule 2 (weighted classes), there is on an average exactly one fixed point per network. A fixed point is a cycle of length 1. The proof is slightly different for rule 1 and rule 2. Let us first choose rule 2. We make use of the property that for every update function the inverted function has the same probability. The inverted function has all 1s in the output replaced with 0s, and vice versa. Let us choose a network state, and let us determine for which fraction of networks in the ensemble this state is a fixed point. We choose a network at random, prepare it in the chosen state, and perform one update step. The probability that node 1 remains in the same state after the update, is 1/2, because a network with the inverted function at node 1 occurs equally often. The same holds for all other nodes, so that the chosen state is a fixed point of a given network with probability 2^N . This means that each of the 2 N states is a fixed point in the proportion 2^N of all networks, and therefore

the mean number of fixed points per network is 1. We will see later that fixed points may be highly clustered: a small proportion of all networks may have many fixed points, while the majority of networks have no fixed point.

Next, we consider rule 1. We make now use of the property that for every update function a function with any permutation of the input states has the same probability. This means that networks in which state A leads to state B after one update, and networks in which another state C leads to state B after one update, occur equally often in the ensemble. Let us choose a network state with n 1s and Nn 0s. The average number of states in a network leading to this state after one update is $2^N p^n (1p^{Nn}$. Now, every state leads equally often to this state, and therefore this state is a fixed point in the proportion $p^n(1p)^{Nn}$ of all networks. Summation over all states gives the mean number of fixed points per network, which is 1.

Finally, we derive a general expression for the mean number of cycles of length L in networks with $K = 2$ inputs per node. The generalization to other values of K is straightforward. Let $\langle C_L \rangle_N$ denote the mean number of cycles in state space of length L, averaged over the ensemble of networks of size N . On a cycle of length L, the state of each node goes through a sequence of 1s and 0s of period L. Let us number the 2^L possible sequences of period L of the state of a node by the index j, ranging from 0 to $m = 2^L 1$. Let n_j denote the number of nodes that have the sequence j on a cycle of length$L$, and $(P_L)^j_{l,k}$ the probability that a node that has the input sequences l and k generates the output sequence j. This probability depends on the probability distribution of update functions.

Then

$$\langle C_L \rangle_N = \frac{1}{L} \sum_{n_j} \frac{N!}{n_0! \ldots n_m!} \prod_j \Big(\sum_{l,k} \frac{n_l n_k}{N^2} (P_L)^j_{l,k} \Big)^{n_j} \tag{2.2}$$

The factor 1/L occurs because any of the L states on the cycle could be the starting point. The sum is P over all possibilities to choose the values n j such that j n j = N . The factor after the sum is the number of different ways in which the nodes can be divided into groups of the sizes n 0 , n 1 , n 2 , . . . , n m . The product is the probability that each node with a sequence j is connected to nodes with the sequences l and k and has an update function that yields the output sequence j for the input sequences l and

k. This formula was first given in the beautiful paper by Samuelsson and Troein [10].

Chapter 3

Cancer attractors

In questo capitolo viene fatta una piccola introduzione al modello classico preda-predatore di Lotka-Volterra.

3.1 The model

We consider a physical system that can be described by an weighted interaction network among nodes that can assume different dynamical states (in the case of a gene network the states $\sigma \in [0, 1]$ and we have models similar to spin models). In the simplest case, we introduce a stochastic dynamics using the probability $p_i(t)$ that the node i is in the state $\sigma_i = 1$ (then $1 - p_i(t)$ is the probability to get $\sigma_i = 0$) and we define a linear equation for the probability evolution

$$\dot{p}_i(t) = \sum_j \mathcal{P}_{ij} p_j(t) - \gamma_i p_i(t) \tag{3.1}$$

where $\mathcal{P}_{ij}$ are transition probability rates and γ_i^{-1} defines the mean lifetime of the excited state. The meaning of the rates $\mathcal{P}_{ij}$ is the rate at which the excited state of the node j increases (or decreases if $\mathcal{P}_{ij} < 0$) the probability of a transition to the excited state of the node i. Since $0 \leq p_i \leq 1$ for all i, this space should be invariant for the dynamics. This condition depends on the spectral properties of the matrix

$$\mathcal{P}_{ij} - \gamma_j \delta_{ij} \tag{3.2}$$

associated to the system. Let consider the case $\mathcal{L}_{ij} \geq 0$ (i.e. we have no inhibitory link), the first quadrant is clearly invariant and if we define

$$\sum_i \mathcal{P}_{ij} = \hat{\gamma}_j > 0$$

the matrix

$$\mathcal{L}_{ij} = \mathcal{P}_{ij} - \hat{\gamma}_j \delta_{ij}$$

is a Laplacian matrix and the system (3.1) can be written in the form

$$\dot{p}_i(t) = \sum_j \mathcal{L}_{ij} p_j(t) - \Delta\gamma_i p_i(t) \qquad \Delta\gamma_i = \gamma_i - \hat{\gamma}_i$$

and by assumption we have $\gamma_i > \hat{\gamma}_i$. The eigenvalues of the matrix $\mathcal{L}_{ij}$ have all negative real part except the null eigenvalue. It follows that all the eigenvalue of the matrix (3.2) has negative real part and the dynamics is a contraction towards the origin: a stable solution (i.e. without any external stimulus the system relaxes to the $\sigma_i = 0$ state). A non trivial stationary can be achieved only if an external stimulus is inserted

$$\dot{p}_i(t) = \sum_j \mathcal{P}_{ij} p_j(t) - \gamma_i p_i(t) + \epsilon f_i(t) \tag{3.3}$$

The stationary solution has to satisfy $p_i \in [0, 1]$ so that $f_i(t) \geq 0$ otherwise we can have negative probability when $p_i \simeq 0$. The case of a Laplacian matrix

$$\hat{\gamma}_i = \gamma_i$$

we get another possible stationary solution for $\mathcal{L}_{ij} p_j^* = 0$ in the first quadrant and the subspace $\sum p_i = 0$ is invariant and the dynamics is a contraction in this subspace (in general). Then the system a stable stationary solution even in absence of an external stimulus.
The presence of inhibitory links complicates the model and one has to prove that

- 1) there exists a physical space: an invariant cone in the first quadrant where the dynamics is a contraction towards the origin;
- 2) the external stimulus maintains the solution in the physical space.

Another solution could be to introduce boundary conditions so that $p_i \geq 0$ in any case (the system is non linear in such a case).
The eigenvalues of the matrix (3.2) define the different relaxation time scale the process and determine its rectivity to the change of the external stimulus: in a typical problem one consider a slowly varying external stimulus so that the system could be considered i a quasi stationary state

$$\sum_j \mathcal{L}_{ij} p_j - \Delta\gamma_i p_i = -\epsilon f_i(t) \qquad \frac{df_i}{dt} \ll 1$$

the derivative is small with respect to the eigenvalues of th matrix (adiabatic approximation). On the other hand we have the effect of a correlated noise (we need to introduce a correlation in order have a continuous function $f_i(t)$). The problem is to study the relation between the solution and the spectral properties of the matrix $\mathcal{L}_{ij}$: we simplify the equation by assuming $\Delta\gamma_i = \Delta\gamma$ so that if λ is an eigenvalue of $\mathcal{L}_{ij}$ then $\lambda - \Delta\gamma$ is an eigenvalue of the matrix (3.2) and we assume that the dynamics is perturbed by

$$\dot{p}_i(t) = \sum_j \left(\mathcal{L}_{ij} + \Delta\mathcal{L}_{ij}\right) p_j(t) - \Delta\gamma p_i(t) + \epsilon f_i(t) \tag{3.4}$$

where the perturbation $\Delta\mathcal{L}_{ij}$ is a Laplacian matrix ($\sum_i \Delta\mathcal{L}_{ij} = 0$ and we assume $< \mathcal{L} >= 0$) that can represent an error in the measure of the transition rates $\mathcal{L}_{ij}$ or possible evolution of network due to in time. In the first case we have an ensemble of transition matrices and we have to study the eigenvalue distribution due to perturbation and the possible presence of bifurcation phenomena. In the second case we have a stochastic differential equation (since $\Delta\mathcal{L}_{ij}(t)$ can be represented as a realization of a stochastic process). The possible approach are Perturbation Theory, Random Matrix Theory and Statistical Physics Methods for random matrices. The external signal form the environment (the environmental node) can be considered in the adiabatic approximation (to be justified form a biological point of view).
The underlying stochastic process on the graph is defined by assigning the state $\xi_i(t) \in [0, 1]$ at each node i according to a probability distribution $\pi_i(t)$ that evolves as

$$\dot{\pi}_i(t) = \sum_j \mathcal{L}_{ij}\xi_j(t) - \Delta\gamma\xi_i(t) + \epsilon f_i(t)$$

By discretizing the dynamics for a time step Δt we have the evolution

$$\pi_i(t+\Delta t) = \pi(t) + \sum_j \mathcal{L}_{ij}\xi_j(t)\Delta t - \Delta\gamma\Delta t\xi_i(t) + \epsilon f_i(t)\Delta t$$

and $\xi(t+\Delta t)$ realized according to the distribution $\pi_i(t+\Delta t)$ (stochastic cellular automata). The average dynamics is computed by

$$\begin{aligned} \dot{<}\pi_i(t)> &= \sum_j \mathcal{L}_{ij} < \xi_j(t) > -\Delta\gamma < \xi_i(t) > +\epsilon f_i(t) \\ &= \sum_j \mathcal{L}_{ij} p_j(t) - \Delta\gamma p_i(t) + \epsilon f_i(t) = \dot{p}_i(t) \end{aligned}$$

and we recover the average equation (3.1). But the stochastic dynamics gives information on the applicability of the average approximation and the variability at the critical states (at bifurcation of the spectrum of $\mathcal{L}$). The stochastic dynamics can be studied for stochastic connection matrices $\mathcal{L} + \Delta\mathcal{L}$.

Appendix A

Approssimazione di campo medio

Bibliography

[1] B. Drossel,*Random Boolean Networks*, arXiv:0706.3351. (2008)

[2] M. Rybarsch and S. Bornholdt,*On the dangers of Boolean networks: Activity dependent criticality and threshold networks not faithful to biology*, arXiv:1012.3287v1. (2010)

[3] J. Park and M. E. J. Newman,*The statistical mechanics of networks*, DOI: 10.1103/PhysRevE.70.066117 (2004)

[4] C. Gershenso,*Introduction to Random Boolean Networks*, arXiv:nlin/0408006,(2004)

[5] B. Derrida and H.Flyvbjerg, *The random map model: a disordere model with deterministic dynamics*, J.Physique, (1987)

[6] R. V. Sol, B. Loque, *Phase transitions and antichaos in generalized Kauffman networs*,Physics Letters,(1994)

[7] J. T. Lizier, S. Pritam, M. Prokopenko, *Information dynamics in small-world Boolean networks*,, (2011)

[8] B. Derrida, *Spin glasses, random boolean networks and simple models of evolution*

[9] A. Rka and A-L. Barabsi, *Statistical mechanics of complex networks*, Reviewes of modern physics, Volume 74,(2002)

[10] N. Masuda , M. A. Porter, R. Lambiotte ,*Random walks and diffusion on networks*,Physics Reports 716717 158,(2017)

[11] T. Biyikoglu, J. Leydold, P. F. Stadler,*Laplacian Eigenvectors of Graphs*, Springer

[12] Fan R. K. Chung,*Spectral Graph Theory*, CBMS

[13] Sui Huang , Ingemar Ernberg, and Stuart Kauffman,*Cancer attractors: A systems view of tumors from a gene network dynamics and developmental perspective*, DOI:10.1016/j.semcdb.2009.07.003, (2009)

[14] M. Ali Al-Radhawi, Nithin S. Kumar, Eduardo D. Sontag, Domitilla Del Vecchio ,*Stochastic multistationarity in a model of the hematopoietic stem cell differentiation network*,DOI:10.1109/cdc.2018.8619300.,(2018)

[15] Rushina Shah, Domitilla Del Vecchio,*Reprogramming cooperative monotone dynamical systems*,DOI:10.1109/cdc.2018.8618649, (2018)

[16] Atefeh Taherian Fard and Mark A. Ragan, *Modeling the Attractor Landscape of Disease Progression: a Network-Based Approach*,DOI: 10.3389/fgene.2017.00048, (2017)

[17] Sui Huang, Yan-Ping Guo, Gillian May, Tariq Enver,*Bifurcation dynamics in lineage-commitment in bipotent progenitor cells*,Developmental Biology 305, (2007)

[18] Xin-She Yang and Young Z. L. Yang,*Cellular Automata Networks*,arXiv:1003.4958 , (2010)

[19] Christopher H. Joyner and Uzy Smilansky, *Dysons Brownian-motion model for random matrix theory - revisited*,arXiv:1503.06417,(2015)

[20] Cameron P. Gallivan, Honglei Ren and Elizabeth L. Read,*Analysis of Single-Cell Gene Pair Coexpression Landscapes by Stochastic Kinetic Modeling Reveals Gene-Pair Interactions in Development* ,doi: 10.3389/fgene.2019.01387,(2019)

[21] Xin Kang, Chunhe Li,*Landscape inferred from gene expression data governs pluripotency in embryonic stem cells*,omputational and Structural Biotechnology Journal 18 (2020) 366374,(2020)

[22] Genaro J. Martnez , Andrew Adamatzky , Bo Chen , Fangyue Chen , Juan C.S.T. Mora ,*Simple networks on complex cellular automata: From de Bruijn diagrams to jump-graphs*,(2017)

[23] Jifan Shi ID , Tiejun Li , Luonan Chen, Kazuyuki Aihara,*Quantifying pluripotency landscape of cell differentiation from scRNA-seq data by continuous birth-death process*, (2019)