Università degli Studi di Cagliari

DOTTORATO DI RICERCA
Ingegneria Elettronica e Informatica
CICLO: XXV

TITOLO TESI
Models and Frameworks for Studying Social Behaviors

Settore Scientifico Disciplinare di afferenza: ING-INF/05

Presentata da: Marco Alberto Javarone

Coordinatore Dottorato: Ch.mo Prof. Alessandro Giua

Relatore: Ch.mo Prof. Giuliano Armano

Esame finale anno accademico 2011 - 2012

To my family

Acknowledgements

First of all, I would like to thank my advisor Prof.Giuliano Armano who gave me the opportunity to investigate the fascinating field of complex networks during this PhD journey. Furthermore, I am grateful for his teachings and for all his support. I am also grateful to dott.Alessandro Chessa who introduced me the theory of complex networks and to dott.Vincenzo De Leo for his useful suggestions. I take this opportunity to thank Prof.Luciano Colombo, Prof.Gianni Mula and Prof.Giuseppina D'Ambra for their valuable advices. I would like to thank Prof.Gavino Mariotti for his kind welcome at the Dept. of Social and Human Sciences at University of Sassari, and Prof.Fiorenzo Toso for his helpfulness and for his valuable suggestions. A particular thank to my friends dott.Matteo Matteucci and dott.Davide Eynard, both from Politecnico di Milano, with whom I have had the opportunity to work. A special thank to dott.Eloisa Vargiu for her useful suggestions and to dott.Andrea Manconi and dott.Alessandro Giuliani for their helpfulness during this journey. Finally, many thanks to Fondazione Banco di Sardegna that supported a part of this work.

Summary

Studies on social systems and human behavior are typically considered domain of humanities and psychology. However, it appears that recently these issues have attracted a strong interest also from the scientific community belonging to the hard sciences –in particular from physics, computer science and mathematics. The network theory offers powerful tools to study social systems and human behavior. In particular, complex networks have gained a lot of prestige as general framework for representing and analyze real systems. From an historical perspective, complex networks are rooted in graph theory –which in turn is dated back to 1736, when Leonhard Euler wrote the paper on the seven bridges of Königsberg. After Euler's work, different mathematicians (e.g. Cayley) focused their research on graphs –opening the possibility of applying their results to deal with theoretical and real problems. As a result, complex networks emerged as multidisciplinary approach for studying complex systems. From a computational perspective, models based on complex networks allows to extract information on complex systems composed by a great number of interacting elements. A variety of systems can be modelled as a complex network (e.g. social networks, the World Wide Web, internet, biological systems, and ecological systems). To summarize, any such system should give the possibility of viewing its elements as simple (at some degree of abstraction), while assuming the existence of nonlinear interactions, the absence of a central control, and emergent behavior. Nowadays, scientists belonging to different communities use complex networks as a framework for dealing with their preferred research issues, from a theoretical and/or pratical perspective. This work is aimed at illustrating some models, based on complex networks, deemed useful to represent social behaviors like competitive dynamics, groups formation, and emergence of linguistics phenomena.

Preface

This dissertation illustrates the research work done during three years of Doctorate School at the University of Cagliari. Part of the work has been made on research topics established by the Dept. of Social and Human Sciences at University of Sassari, also thanks to the financial support given by Fondazione Banco di Sardegna.

Apart from the introductory chapter on complex networks (see Chapter 1), each chapter is focused on a specific research topic, with the common aim of illustrating the ability of complex networks to give a support in the task of analyzing and/or modelling complex systems. In particular:

- Chapter 2 illustrates a model, inspired by quantum statistics, to represent competitive dynamics (it is focused on quantum-classical transitions in complex networks);
- Chapter 3 describes a geometrical model to study the dynamics of social networks (it is aimed at analysing the geometry of social networks);
- Chapter 4 illustrates a model for studying the emergence of acronyms in populations (it is aimed at analysing a linguistics phenomenon);
- Chapter 5 illustrates a framework to clustering multidimensional datasets, as real datasets that store data of social networks users (it is aimed at performing clustering according to a complex networks perspective).

The final chapter (Chapter 6) ends the thesis –highlighting, by means of relevant examples, that complex behaviors can be successfully analysed and/or modelled using complex networks.

Contents

List of figures

List of tables

Chapter 1

Complex Networks

Many natural and man-made complex systems, as biological neural networks, social networks and the Web, can be represented as complex networks [1] [2]. In recent years, complex networks have been adopted as general framework in a wide set of studies, ranging from biology to computer science, or from physics to economics. Furthermore, the comparison of results achieved by analysing networks of different real systems, yielded surprising outcomes. The main outcome was the identification of a series of unifying principles and statistical properties common to most of the real networks considered [3]. For example, it has been found that the degree distribution of many real complex networks significantly deviates from that of a Poisson distribution expected for a random graph and, in many cases, exhibits a power-law tail with an exponent $\gamma \in [2, 3]$ [4]. Moreover, many real networks showed correlations in the node degrees, relative short paths between two randomly chosen nodes, and the presence of a large number of short cycles [3] (all these cited properties will be described later with more detail). All these findings lead to a strong interest in complex networks. Further, under the hypothesis that the structure of a system is tightly related to the dynamical mechanisms which affect the function of the system itself, many efforts have been focused on developing models to mimic the growth of a network and to reproduce the structural properties observed in real topologies [3]. Network theory puts its basis in classical theory of graphs. In particular, although theory of graphs has been rapidly replaced by network theory in modelling real world systems, graphs are still useful as mathematical models of network structures [5]. Hence, before introducing models and main properties measured in networks, let us spend some words on some basic concepts of graph theory.

1.1 Basic Concepts

A complex network is in fact a graph characterized by non-trivial topological features. In general, a graph is a mathematical entity that allows to represent specific relations among a collection of items. More formally, a graph G is described as $G = (N, E)$, with N set of nodes and E set of edges. Nodes are usually described by a label and represent elements of a system, e.g., people of a social network, genes of the DNA, or web-sites. Edges represent connections among nodes and usually map specific relations as friendship, gene interactions, links among web-sites. A graph can be "directed" or "undirected", i.e., symmetrical relations hold among nodes or not. In the former case, a simple (undirected) edge is drawn between two nodes, whereas in the second case the edge takes the form of an arrow. For example, if two people are friends (then, obviously know each other) a simple edge is drawn between them. Instead, if a web-site a offers a link to another web-site b, this relation is represented as an arrow from a to b. A graph can be "weighted" or "unweighted". In particular, if a numerical value is associated to edges, i.e., relations among nodes are weighted in some way, the graph is weighted. For example, let us consider the airline network, where each airport is represented as a node and every route among airports is represented as an edge. We could identify a weight for each edge as the geographical distance between respective airports. All edges, existing in a graph with N nodes, are collected in a $N \times N$ matrix, called "adjacency matrix" that characterizes the graph itself. Undirected graphs have a symmetric adjacency matrix. Unweighted graphs are represented by a binary adjacency matrix. In particular, the adjacency matrix A of an unweighted graph is composed by the elements:

$$a_{ij} = \begin{cases} 1 & \text{if } e_{ij} \text{ is defined} \\ 0 & \text{if } e_{ij} \text{ is not defined} \end{cases} \tag{1.1}$$

Instead, a weighted graph is represented by a real matrix.

Degree distribution

Node of a graph can have one or more connections with other nodes. In graph and in network theory, the number of connections of a node is called degree, and it is usually denoted as k. An important function, to investigate the structure of a network, is the degree distribution $P(k)$ [1]. Fundamentally, the $P(k)$ represents the probability that a randomly selected node had the degree equal to k, i.e., it is linked with k nodes.

Clustering Coefficient

The clustering coefficient allows to know if nodes of a network tend to cluster together. This phenomenon is common in many real networks as social networks, where it is possible to identify circles of friends or acquaintances in which every person knows all the other people. For example, in social networks if the node a is connected to the node b and the node b is connected to the node c, there is a high probability that a be connected to c. The clustering coefficient can be computed as:

$$C = \frac{3 \times Tn}{Tp} \tag{1.2}$$

Tn is the number of triangles in the network and Tp is the number of connected triples of nodes. A connected triple is a single node with edges running to an unordered pair of others. The value of C lies in the range $0 \leq C \leq 1$. Another definition of the clustering coefficient has been developed by Watts and Strogatz in [6], which defined this quantity as a local value:

$$C_i = \frac{Tn_i}{Tp_i} \tag{1.3}$$

with Tn_i number of triangles connected to node i and Tp_i number of triples centered on node i. In this case, the local C of nodes with a degree equal to 0 or 1 is set to 0. In so doing, the global clustering coefficient of a network is computed as:

$$C = \frac{1}{n} \sum_i C_i \tag{1.4}$$

This parameter allows to measure the density of triangles in a network, and can be computed for directed and undirected networks. On the other hand, the local definition of clustering C_i has been adopted in the sociological literature, where it is referred to as the "network density".

Betweenness Centrality

The betweenness centrality measures the centrality of a node in a network [7]. This parameter is quantified as the number of geodetics from all nodes to all others that pass through that node. In particular, it can be computed as follows:

$$B_i = \sum_{x \neq i \neq y} \frac{\sigma_{xy}(i)}{\sigma_{xy}} \tag{1.5}$$

with σ_{xy} total number of geodesics from x-th node to y-th node, $\sigma_{xy}(i)$ total number of geodesics from x-th node to y-th node passing through the node i.

Community structure

A community is a group of highly mutually interconnected nodes in a network [8]. If communities can be easily identified, the network is said to have community structure. Networks with this property can have non-overlapping communities or overlapping communities. In the former case, communities emerge naturally as are composed by groups of nodes with dense connections internally and sparser connections with nodes of other groups. In the second case, it is more difficult to identify communities.

Path Length

The distance between two nodes in the same network is defined as the minimum number of edges which connect them. As in other metrical spaces, the minimal distance is called "geodesic", whereas if there is not a path between two nodes, their distance is infinite. There are many different algorithms to find a geodesic in networks, e.g., the Dijkstra's algorithm [9] and the Floyd-Warshall algorithm [10].

Assortativity

The assortativity is a property of networks that allows to evaluate if their nodes prefer to attach to other nodes that are (not) similar [11]. This property affects the whole structure of a network, e.g., social networks can be divided into communities of users speaking the same language or having same hobbies. In general, the assortativity can be computed as follows:

$$r = \frac{\sum_i e_{ii} - \sum_i a_i b_i}{1 - \sum_i a_i b_i} \tag{1.6}$$

with e_{ij} fraction of edges in a network that connect a node of type i to one of type j, $a_i = \sum_j e_{ij}$ and $b_j = \sum_i e_{ij}$. A network is assortative when the assortativity value is positive and, on the contrary, it is disassortative when this value is negative. The similarity can also refers to the nodes degree, i.e., their amount of edges. In this case, Johnson et al. [12] showed a relation between assortativity and shannon entropy of networks. In particular, they found that scale-free networks have a high probability to be disassortative.

1.2 Network Structures

Erdös-Renyi graphs

One of the first works about random networks has been developed by Paul Erdös and Alfred Renyi [13]. These mathematicians defined a famous model known as Erdös-Renyi graph, or simply E-R graph (as we call hereinafter). This model considers a graph with N nodes and a probability p to generate

each edge, hence a E-R graph has around $p \cdot \frac{N(N-1)}{2}$ edges. E-R graphs have a binomial degree distribution, defined as follows:

$$P(k) = \binom{N-1}{k} p^k (1-p)^{n-1-k} \tag{1.7}$$

If $N \to \inf$ and $np = const$, their degree distribution converges to a Poissonian distribution:

$$P(k) \sim e^{-\eta n} \cdot \frac{(\eta n)^k}{k!} \tag{1.8}$$

To generate these E-R graphs we used the following simple algorithm:

1. Define the number of N of nodes and the probability p for each edge
2. Draw each potential-link with probability p

Figure 1.1 illustrates the $P(k)$ for a E-R graph with $N = 25000$ and $p = 4 \cdot 10^{-4}$. The network, whose $P(k)$ is shown in Figure 1.1, has an average

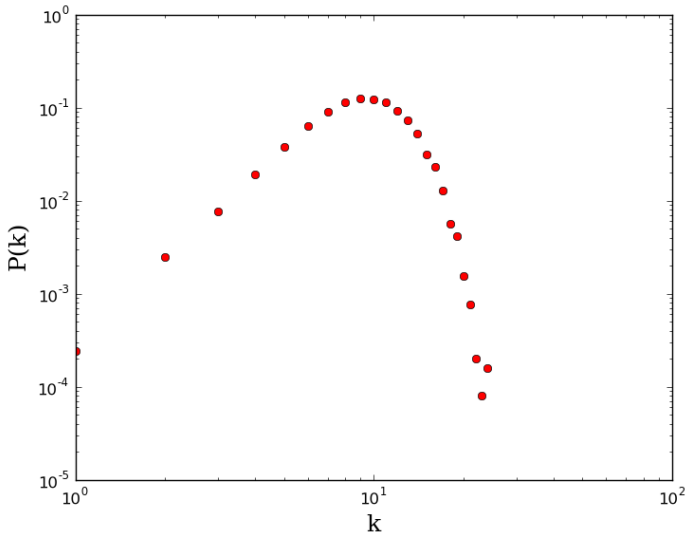

Figure 1.1: $P(k)$ (in loglog scale) of an E-R graph with $N = 25000$ and $p = 4 \cdot 10^{-4}$.

degree $\langle k \rangle \sim 10$. An example of a small E-R graph with $N = 100$ is given in Figure 1.2.

Scale-free networks

Many real complex networks show a $P(k)$ that follows a power-law function as:

$$P(k) \sim c \cdot k^{-\gamma} \tag{1.9}$$

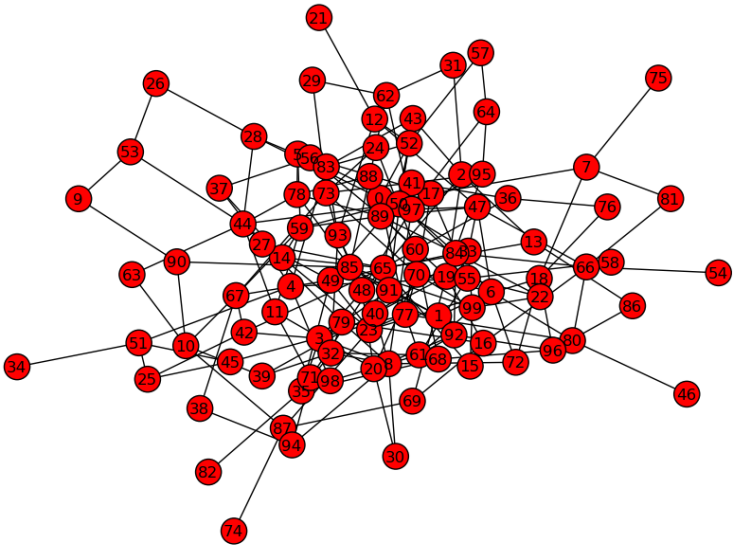

Figure 1.2: A small E-R graph with $N = 100$.

with c normalizing constant and γ parameter of the distribution known as the scaling parameter. These networks have a scale-free structure and are characterized by the presence of few nodes (called hubs) that have many connections (i.e, a high degree). The most famous model to generate scale-free networks is the Barabasi-Albert model (BA model hereinafter) [1], which considers N nodes and m minimum number of edges drawn for each node. The BA model can be summarized as follows:

1. Define N number of nodes and m minimum number of edges drawn for each node

2. Add a new node and link it with other m pre-exisisting nodes. Pre-existing nodes are selected according to the following equation:

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j} \tag{1.10}$$

   with $\Pi(k_i)$ probability that the new node generates a link with the i-th node having a k_i degree.

Figure 1.3 illustrates the $P(k)$ for a scale-free network with $N = 25000$ and $m = 5$. The network, whose $P(k)$ is shown in Figure 1.1, has an average degree $\langle k \rangle \sim 10$. A small scale-free structure is shown in Figure 1.4. As it is possible to see in Figure 1.4, all nodes have at least 2 connections and, in the center of the figure, it is possible to note the presence of few hubs with a high number of connections.

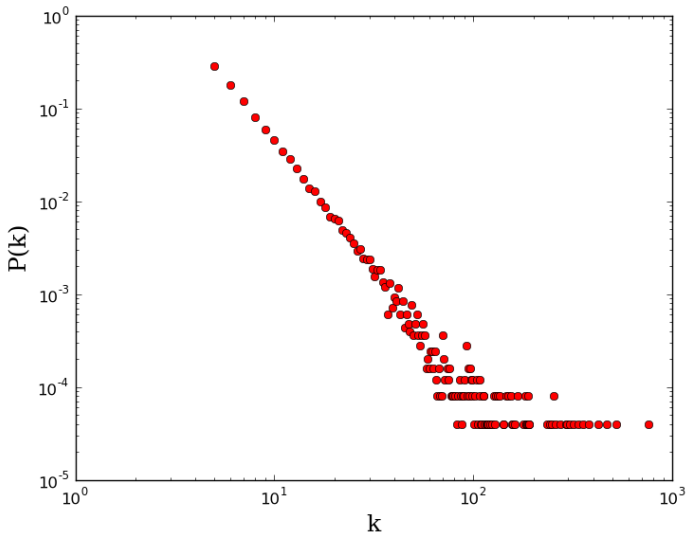

Figure 1.3: $P(k)$ (in loglog scale) of a scale-free network with $N = 25000$ and $m = 5$.

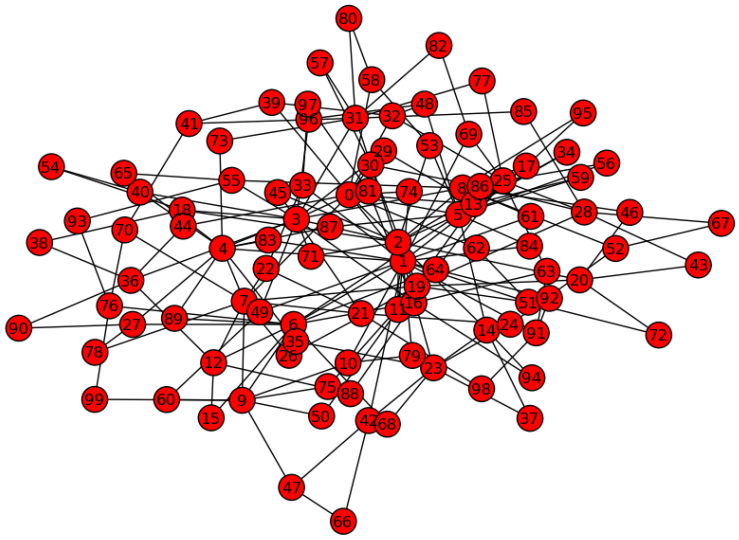

Figure 1.4: A small scale-free network generated by the BA model with $N = 100$ and $m = 2$.

Small-World networks

Many real complex networks show a small-world character [1], i.e., every nodes can be reached from any other in a small number of hops. More

formally, small-world networks are characterized by a distance L, between to randomly chosen nodes, equal to $L \propto \ln N$. Furthermore, two main properties allow to identify small-world networks, i.e., a short average path length and a relatively high clustering coefficient. In particular, the clustering coefficient of a small-world network is higher than that of its related classical random networks, i.e., the E-R graph generated with the same set of nodes. One of the first algorithms to generate random networks provided with a small-world character is the Watts-Strogatz model (WS hereinafter) [6]. This model can be summarized as follows:

1. Define a regular ring lattice with N nodes, each connected to k neighbors ($k/2$ on each side)

2. For every node i take every edge (i, j) with $i \leq j$ and rewire it with probability β. Rewiring is done by replacing the edge (i, j) with (i, k) with k chosen with uniform probability from all nodes avoiding loop and edge duplication

A small-world network with 1000 nodes generates by the WS model is shown in Figure 1.5. The WS model offers an interesting behavior studying the

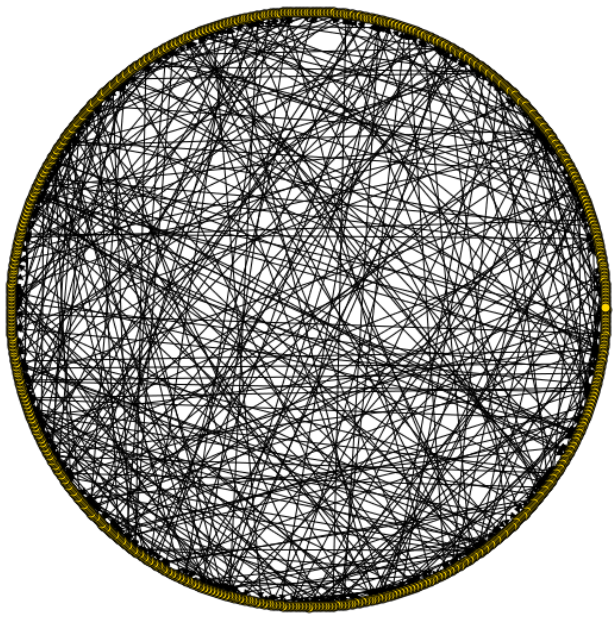

Figure 1.5: A small-world network generated by the WS model, with $N = 1000$ and a rewiring probability $\beta = 0.4$.

effect of the rewiring probability β. In particular, this model generates regular lattices for $\beta = 0$ and completely random networks at $\beta = 1$, where

all lattice-like features disappear. At intermediate values of β, the WS model generates networks that consist of a mixture of random and regular connections. This behavior is illustrated in Figure 1.6.

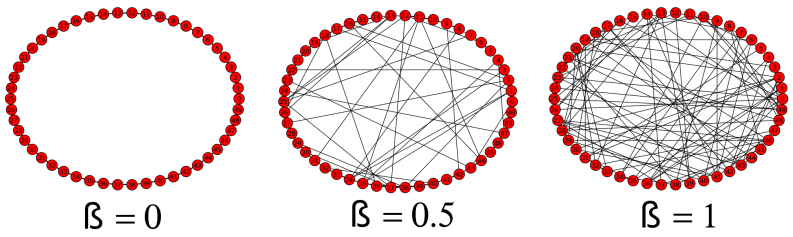

Figure 1.6: Small-world networks generated by the WS model. From left to right: network generated with $\beta = 0$, $\beta = 0.5$ and $\beta = 1$.

1.3 Community Detection

Community detection is the process of finding communities in a graph, also called "graph partitioning". From a computational perspective, this is not a easy task and many algorithms have been proposed, according to three main categories: divisive, agglomerative, and optimization algorithms. Furthermore, many algorithms adopted in machine learning, e.g., k-means, fuzzy C-means, and hierarchical (see [14]), are also used to perform community detection. Some famous algorithms to perform community detection are briefly discussed.

1.3.1 Louvain Algorithm

The Louvain method [16] is an optimization algorithm based on an objective function devised to measure the quality of partitions. At each iteration, the Louvain method tries to maximize the so-called *weighted-modularity*, defined as:

$$Q = \frac{1}{2m} \cdot \sum_{i,j} \left[a_{ij} - \frac{k_i k_j}{2m} \right] \cdot \delta(\mathbf{x}_i, \mathbf{x}_j) \tag{1.11}$$

where a_{ij} is the generic element of the adjacency matrix, k is the degree of a node, m is the total "weight" of the network (i.e., the sum of all weighted links of the network), and $\delta(\mathbf{x}_i, \mathbf{x}_j)$ is the Kronecker Delta, used to assert whether a pair of samples belong to same community or not. Given a weighted network with N nodes, this algorithm can be summarized as follows:

1. Define a community for each node
2. For each node i consider all its neighbors and compute the gain of modularity if i is moved to the community of its j-th neighbor
3. Place each node i in the community for which the gain of modularity is maximum
4. For each new community computed at STEP (3) define a node
5. For each node defined at STEP (4) define weighted links. Each link has a weight equal to the sum of weights of the links among nodes in the corresponding two communities; links among nodes of the same community generate self-loops for this community
6. Repeat from STEP 1 until a global maximum modularity is computed, then STOP

1.3.2 Girvan-Newmann Algorithm

The Girvan-Newmann algorithm [8], belonging to the family of divisive algorithms, is based on the concept of *betweenness*. In general, this concept is related to the frequency of the participation of an element to a process. It can be computed for nodes, as discussed before, and for edges. In particular, edge betweenness is the number of shortest paths between all nodes pairs that run along the considered edge. The Girvan-Newman algorithm, that makes use of the concept of edge betweenness, can be summarized as follows:

1. Compute the centrality of each edge
2. Remove the edge with largest centrality
3. The betweenness of all edges affected by STEP (2) is recomputed
4. Repeat STEPS (2) and (3) until all edges are removed

Different variants of this algorithm have been proposed –see [17].

1.3.3 Kernighan-Lin Algorithm

The Kernighan-Lin algorithm [18] is based on a greedy optimization of a benefit function Q. Once two groups of nodes of a network are defined, the function Q is computed as the number of edges that lie within the two groups minus the number of edges that lie between them. The user has to specify the size of the two groups and the relative configuration. The algorithm can be summarized as follows:

1. Define two groups of nodes (randomly or user-chosen)
2. Compute, for all possible pairs of nodes i and j belonging to the two groups, the ΔQ in the benefit function that would result if i was swapped with j
3. Select the pair which maximizes this change and perform the swap
4. Repeat from STEP 2 until all nodes in one group have been swapped once (the same node cannot be swapped twice)
5. Go back over the sequence of performed swaps to find the point during this sequence at which Q was highest. This is the bisection of the graph. Then, STOP

In principle, this algorithm computes only two partitions of a network, but its outcomes can be considered as an input for the algorithm to find more granular partitions.

1.3.4 Physical Models for Community Detection

Here, we illustrate some models inspired by physics to perform community detection tasks.

Spin Models

Spin models can be applied for clustering analysis, in particular by using the famous Potts model –see [19]. The Potts model is widely used in statistical mechanics for systems of spins that can be in q different states. The basic idea is that at the ground state, considering ferromagnetic and antiferromagnetic interactions, there are different spin values that form homogeneous clusters. To use this representation, a preliminary mapping of the original model to the network must be performed. In particular, spins are mapped to nodes and their interactions are mapped to edges. After this first modelling task, an algorithm of community detection can be applied to compute partitions (i.e., communities). The algorithm of Reichardt and Bornholdt [20] implements this model. In particular, this algorithm considers the Hamiltonian of the described model:

$$H = -J \sum_{i,j} a_{i,j} \delta(\sigma_i, \sigma_j) + \gamma \sum_{s=1}^{q} \frac{n_s(n_s - 1)}{2} \tag{1.12}$$

where $a_{i,j}$ is the adjacency matrix, σ_i is the spin-state of the node i, δ is the Kroenecker function, n_s is the number of spins in the state s, J and γ are some coupling parameters. The Hamiltonian (1.12) is composed by two main components: the classical ferromagnetic Potts model energy and favors spin alignment. The ratio $\frac{\gamma}{J}$ describes the relative importance of these components. Varying this ratio, the system can be analyzed at different levels of modularity, from a single whole community to a community for each node. This algorithm starts by randomly assigning spins to nodes and using

an appropriate number of states q, high enough for each considered problem. The ground state of the Hamiltonian corresponds to the community structure of the network and it is computed by simulated annealing.

Random Walk

Random walk methods can be easily applied to community detection tasks. In principle, if a network has a community structure, a random walker will spend long time inside each community because of the density of internal edges that they will be passed through. In the following, we illustrate one of these methods called Zhou algorithm [21]. The Zhou algorithm defines the distance $d_{i,j}$ between two nodes, $\mathbf{x}_i$ and $\mathbf{x}_j$, as follows: $d_{i,j}$ is the average number of steps that a random walker crosses to reach a node $\mathbf{x}_j$ starting from a node $\mathbf{x}_i$. As a consequence, close nodes have a high probability to belong to the same community. Zhou defines two kinds of attractors called *global* and *local*. For a node $\mathbf{x}_i$, the global attractor is a node $\mathbf{x}_j$ if $d_{i,j} \leq d_{i,k}$ for any $\mathbf{x}_k$ node of the network, whereas the local attractor of $\mathbf{x}_i$ is $\mathbf{x}_j$ if $\mathbf{x}_j$ belongs to $E_{\mathbf{x}_i}$, the set of the nearest-neighbors of $\mathbf{x}_i$, and $d_{i,j} \leq d_{i,l}$ for any $\mathbf{x}_l$ belonging to $E_{\mathbf{x}_i}$. Finally, the node $\mathbf{x}_i$ is put in the same community of its attractor with all nodes for which $\mathbf{x}_i$ itself is an attractor. Applications to real networks show that this method allows to find meaningful partitions.

Synchronization

Synchronization is a phenomenon occurring in systems of interacting components. In synchronized state, components are in the same or in similar state(s). Principles at the base of this phenomenon have been used to find communities in networks. As for other physical models, a preliminary mapping of the physical system to the network must be performed. In particular, each node is mapped to an oscillator, with an initial random phase and nearest-neighbor interactions. Considering that oscillators in the same community synchronize first, a full synchronization will take more time. By analysing the evolution of the process, it is possible to identify clusters of nodes in the same state. In this work [22], Arenas et al. introduced an algorithm based on these physical principles. In particular, the proposed algorithm makes use of the Kuramoto oscillators [23], where the phase Θ_i of an oscillator $\mathbf{x}_i$ evolves according to:

$$\frac{d\Theta_i}{dt} = \omega_i + \sum_j K \sin(\Theta_j - \Theta_i) \tag{1.13}$$

with ω_i natural frequency of $\mathbf{x}_i$, K strength of the coupling between oscillators and the sum is over all oscillators. If the interaction coupling overgrows a threshold (set depending on the width of the distribution of natural frequencies), the dynamics lead to the synchronization. In this model each

oscillator is coupled only to its nearest-neighbors. With the aim to compute the effect of local synchronization, authors introduced a local order parameter:

$$\rho_{ij}(t) =< \cos(\Theta_i(t) - \Theta_j(t)) > \tag{1.14}$$

This parameter measures the average correlation between pairs of nodes. Using the correlation matrix $\rho(t)$ at the time t, it is possible to find clusters of nodes that synchronize together. The clusters are identified by a binary matrix, computed by thresholding the entries of $\rho(t)$, called dynamic connectivity matrix $D_t(T)$. Using the spectrum of $D_t(T)$ it is possible to derive the number of disconnected components at time t. Analysing the number of components as function of time, some plateaus may appear at some characteristic time scales, indicating structural scales of the network with robust communities. The presence of plateaus at different time scales indicates a hierarchical organization of the network. After a large enough Δt all oscillators are synchronized and the whole systems behaves as a single component.

Chapter 2

Fermionic Networks: a Model for Competitive Dynamics

In this chapter, we illustrate a theoretical model, based on complex networks and inspired by quantum statistics, to study competitive dynamics [24][25]. This work shows that the emergence of different structures in complex networks, such as the scale-free and the winner-takes-all, can be represented in terms of a quantum-classical transition for quantum gases. In particular, we propose a model of fermionic networks that allows to investigate the network evolution and its dependence with the system temperature. In turn, the network evolution and the system temperature represent, respectively, the evolution of a social system and its level of competitiveness. Simulations, performed in accordance with the cited model, clearly highlight the separation between classical random vs. winner-takes-all networks, in full correspondence with the separation between classical vs. quantum regions for quantum gases. Consequently, classical random networks represent systems with a low level of competitiveness, whereas winner-takes-all represent systems with a high level of competitiveness. After a brief introduction to basic concepts of quantum statics, we discuss a model of networks that makes use of quantum statistics of bosons. Finally, we illustrate the proposed model and the results of related simulations.

2.1 Quantum Statistics

Statistical mechanics assumes a central role when dealing with systems composed by many particles, the underlying assumption being that particles are identical and indistinguishable. Moreover, their quantum energy levels are

extremely closely spaced, with a cardinality much greater then the number of particles. Energy levels can be grouped in bundles with the approximation that levels in the same bundle have the same energy. Particles with a symmetric wave function, called bosons, obey Bose-Einstein statistics, whereas particles with an antisymmetric wave function, called fermions, obey to Fermi-Dirac statistics [26]. Given a system with N particles of the same type, we can build an N-body wave function, with several admissible states. For each state α, the corresponding number of particles, say n_α (also called occupation number), is given by the following equation:

$$n_\alpha = \begin{cases} 0, 1, ..., \infty & \text{bosons} \\ 0, 1 & \text{fermions} \end{cases} \tag{2.1}$$

and $\sum_\alpha n_\alpha = N$. Considering a gas composed by N bosons, the number of microstates is computable according to the equation:

$$\Omega_b = \Pi_i \frac{(n_i + g_i)!}{n_i!\, g_i!} \tag{2.2}$$

with g_i representing the i-th bundle. The distribution of particles follows the Bose-Einstein statistics:

$$n_i^b = g_i \cdot \left(e^{\frac{\epsilon_i - \mu}{k_b T}} - 1 \right)^{-1} \tag{2.3}$$

where ϵ_i denotes the energy of the i-th bundle, μ the chemical potential, and k_b the Boltzmann constant. In the event that a gas is composed by fermions, we must consider also the Pauli exclusion principle. Hence, the number of microstates is computable according to the equation:

$$\Omega_f = \Pi_i \frac{g_i!}{n_i!\,(g_i - n_i)!} \tag{2.4}$$

Here, the distribution of particles follows the Fermi-Dirac statistics:

$$n_i^f = g_i \cdot \left(e^{\frac{\epsilon_i - \mu}{k_b T}} + 1 \right)^{-1} \tag{2.5}$$

Both these distributions approximate the classical behaviour in proximity of the high-temperature limit, showing a quantum-classical transition. This phenomenon occurs when particles sparsely occupy excited states. In particular, with λ thermal wavelength and ρ density, the following conditions hold:

$$\begin{cases} \rho\lambda^3 \gg 1 & \text{classical regime} \\ \rho\lambda^3 \approx 1 & \text{onset of quantum effects} \end{cases} \tag{2.6}$$

The classical regime is described by the Maxwell-Boltzmann distribution. In particular, with $Z = \sum_j g_j e^{-\frac{\epsilon_j}{k_b T}}$ partition function, we write:

$$n_i^{mb} = \frac{N}{Z} \cdot g_i \cdot e^{-\frac{\epsilon_i}{k_b T}} \tag{2.7}$$

2.2 Bosonic Networks

In this model, Bianconi and Barabasi [28] compared network evolution to a phase transition of bosonic gases. Two main structures, i.e., *fit-get-rich* and *WTA*, are identified as two different phases at low temperatures. In this model, each node is interpreted as an energy level and each link as a pair of particles. Starting from a fitness parameter η, energy is computed according to the following equation:

$$\epsilon = -\frac{1}{\beta} \cdot \log \eta \tag{2.8}$$

with $\beta = \frac{1}{T}$. Here, the fitness parameter η describes the ability of each node to compete for new links. In particular, for the i-th node, the probability of connection with new nodes is proportional to:

$$\Pi_i = \frac{\eta_i k_i}{\sum_j \eta_j k_j} \tag{2.9}$$

with k_i degree of the i-th node. Notably, new nodes tend to link with pre-existing nodes having high values of (η, k). The generation of a scale-free network in the *fit-get-rich* phase is characterized by Eq. (1.9) and entails the presence of a few nodes with a high degree connected to many others with low degree. In a bosonic gas, when the temperature decreases, particles aim to occupy lower energy levels. Then, at a temperature below the critical temperature T_c, the Bose-Einstein condensation takes place. In this model, as the temperature decreases, many particles move to lower levels while keeping the corresponding particles at upper levels. In so doing, links concentrate on a few nodes, until they condensate in the *WTA* phase, characterized by the fact that only one node predominates. In [29], Bianconi discussed the differences between bosonic and fermionic networks, showing that the former are scale-free, whereas the latter can be represented by Cayley trees.

2.3 Fermionic Networks

Let us now introduce a novel proposal for modeling network dynamics, inspired by the physics of fermions. Given a network $G = (V, E)$, with V non empty set of nodes and E non empty set of links, let us represent each link

as a particle and each node as a degenerate bundle of energy levels. Usually, the number g_i of available states in the i-th bundle is much larger than the number p_i of its particles Let us assume that the i-th bundle has an energy ϵ_i. This value can be assigned randomly or depends on a property of the system –e.g. a fitness parameter η or any other function deemed relevant, with the trivial constraint that it must be computable for each node of the network. In the proposed model, lower bundles have more energy levels. In particular, the first bundle has $n-1$ levels, the second has $n-2$ levels, and so on and so forth. Note that the link l_{ij}, which connects nodes i and j, is represented only by a single energy level, i.e., ϵ_{ij}, which in turn belongs to the i-th bundle (under the assumption that the i-th bundle is deeper than the j-th). In so doing, the last node, say y_0, is represented by a bundle without energy levels, although it can be linked in the event that a particle stays at the ϵ_{xy_0} level, with x corresponding to one of the other nodes.

2.3.1 Modelling Network Evolution

Let us consider an evolving network, i.e., a network that changes over time. Almost all real networks evolve over time; examples are social networks (where people find or lose friends or co-workers) and the web (where websites compete to gain more inlinks). Furthermore, let us consider this network a closed system, so that the number of nodes and the number of links remain constant over time. As discussed before, for every node, a bundle is defined –whose energy is computed with Eq. (2.8). In so doing, the relative position of each bundle depends on the value of its energy, so that deeper bundles embody more states. Considering the ability of the particles to jump between energy levels as the temperature varies, at high temperatures particles follow the classical Maxwell-Boltzmann distribution, being spread among the available states according to Eq. (2.7). On the other hand, as temperature decreases, many particles move to lower energy levels (see Figure 2.1).

In this work, we consider the evolution of a system caused by cooling and heating processes. A detailed analysis of both processes follows.

Cooling Process

During a cooling process, a few nodes gain new links and their degree k_i is increased. Given the number of particles, it is possible to compute the Fermi Energy E_f of the system as the energy of the bundle containing the last particle at $T \to 0$. Hence, as temperature decreases, and assuming that the number of particles approximates the number of bundles, the *WTA* phase takes place (see also [28]). For every variation of the temperature, the probability for a particle to jump from the i-th to the j-th bundle is computed

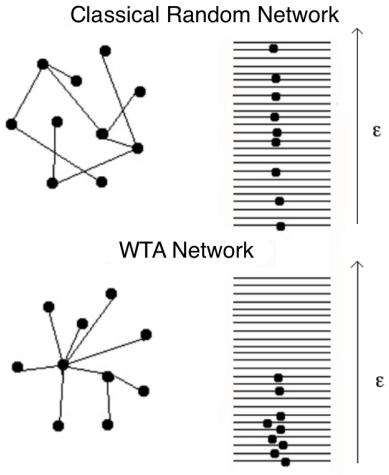

Figure 2.1: On the left, from top to bottom, the evolution of a network with 10 nodes and 9 links from a classical random network to a *WTA* network. On the right, their corresponding fermionic models, which result from a cooling process that pushes particles to low energy levels.

according to the following equation:

$$p(i \to j) = \frac{\Delta T}{T} \cdot \frac{1}{\Delta B(j,i)} \cdot f(g_j) \tag{2.10}$$

where T denotes the temperature of the system before the variation, ΔT the variation of temperature, $\Delta B(j,i)$ is the distance between the bundles j and i, and $f(g_j)$ is the function:

$$f(g_j) = \begin{cases} 0 & \text{if } g_j = 0 \\ 1 & \text{if } g_j \geq 1 \end{cases} \tag{2.11}$$

with g_j number of available states in the j-th bundle. Hence, considering that a particle in the i-th bundle can jump to $i-1$ underlying bundles, each with a probability given by Eq. (2.10), the probability p_J to jump from the i-th to another bundle is:

$$p_J(i) = \sum_{z=1}^{i-1} p(i \to z) \tag{2.12}$$

and the probability p_S to stay in the same bundle is

$$p_S(i) = 1 - p_J(i) \tag{2.13}$$

Then, the final bundle of each particle is chosen by a weighted random selection among all candidate bundles (including the bundle in which the particle is located).

Heating Process

Heating is performed after cooling. Particles can now move to higher energy levels, gradually generating vacancies at lower levels. Also in this case, for every variation of the temperature, the probability for a particle in the i-th bundle to jump to the j-th bundle is computed using a variant of Eq. (2.10), in which $f(g_j)$ is defined as:

$$f(g_j) = \begin{cases} 0 & \text{if } g_j = 0 \\ 1 - \frac{p_j}{g_j} & \text{if } g_j \geq 1 \end{cases} \tag{2.14}$$

with p_j number of particles located at the j-th bundle. Eq. (2.14) has been devised to prevent, at high temperatures, particles from filling high-energy levels densely. For each particle, the probability to jump is computed by the following equation:

$$p_J(i) = \sum_{z=i+1}^{n-1} p(i \to z) \tag{2.15}$$

and the probability to stay by Eq. (2.13). The same criterion adopted in the cooling process (i.e., weighted random selection) is applied for choosing the energy level of each particle. In our model, the temperature corresponds to the level of competitiveness of the system (e.g., a competition for new links in a social network among users, a competition for new customers among companies, or a competition for new inlinks among web-sites). To complete the model, let us assume that each network has the structure of an E-R graph when generated at time $t = 0$.

2.3.2 Simulations

The proposed fermionic model has been tested with many simulations. In particular, we generated networks of different sizes with an E-R graph structure. These networks have been implemented by connecting nodes randomly –giving rise to a graph $G(n, \zeta)$, where n is the number of nodes and ζ is the probability of an edge to be drawn (note that an edge is drawn independently from other edges). Their degree distribution is binomial, converging to a poissonian distribution for a large number of nodes–see Eq. (1.8). Simulations have been performed with a number of nodes ranging from 50 to 10000, $\zeta = \frac{\langle k \rangle}{n-1}$ with $\langle k \rangle$ average degree of the network (see [30]) and an initial temperature ranging from 100K to 500K. For each simulation the network evolves until all particles of the model reach their final position, for both cooling and heating process. At each time step, the temperature is increased (heating) or decreased (cooling) by 10%, then the algorithm computes new positions of the particles and analyzes the degree distribution, computing the scaling parameter γ and the normalizing costant c. The scaling parameters were estimated, as suggested in [31], by using the following

equation:

$$\hat{\gamma} = 1 + n \cdot \left[\sum_{i=1}^{n} \ln \frac{k_i}{k_m} \right]^{-1} \tag{2.16}$$

with k_m minimum degree estimated. The normalizing constant is computed as follows:

$$c = \frac{1}{\int_{k_m}^{\infty} k^{-\gamma} dk} \tag{2.17}$$

Figure 2.2 illustrates a transition between the *E-R graph* structure and the *WTA* structure, for a network with 10000 nodes and $\langle k \rangle = 20$, generated at 100K. As shown in the cited figure, a cooling process in an *E-R* graph

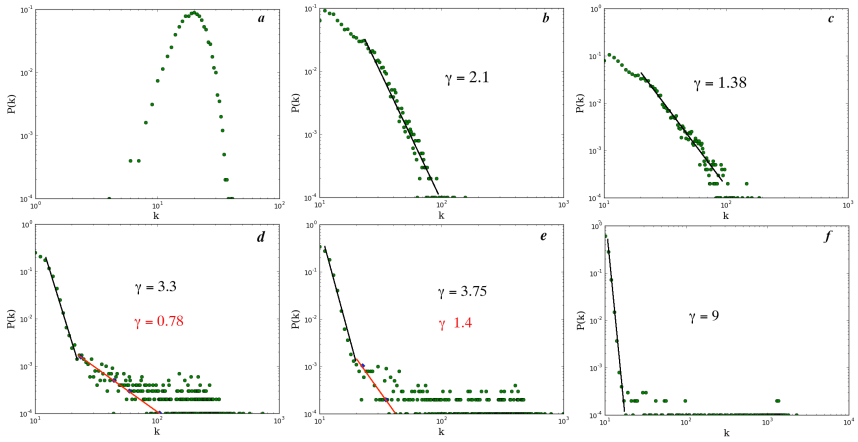

Figure 2.2: The evolution of the degree distribution of a network, during a cooling process, with 10000 nodes and $\langle k \rangle = 20$, generated at $T = 100$K. Each panel shows the network at different time steps t: **a)** at $t = 0$; **b)** at $t = 4$; **c)** at $t = 5$; **d)** at $t = 19$; **e)** at $t = 28$; and **f)** at $t = 50$. Note that for $t = 0$ the network has an *E-R graph* structure, whereas for $t = 50$ it has a *WTA* structure. Continuous black and red lines are used to highlight data interpolation. The corresponding scaling parameter(s) γ is (are) indicated in each panel.

entails a transition to a scale-free structure after 4 time steps. After 19 time steps all networks apparently converge to a *WTA* structure, showing in some cases composite distributions, which in turn can be identified by a process of logarithm data binning (see [32]). A small network having a *WTA* structure is shown in Figure 2.3, where only a few nodes have a great amount of links, i.e., their bundles contain the majority of particles. After the cooling process, the network with 10000 nodes is subject to the heating process and Figure 2.4 illustrates the evolution of the degree distribution. During the heating process (see Figure 2.4), the network loses its *WTA* structure. Then,

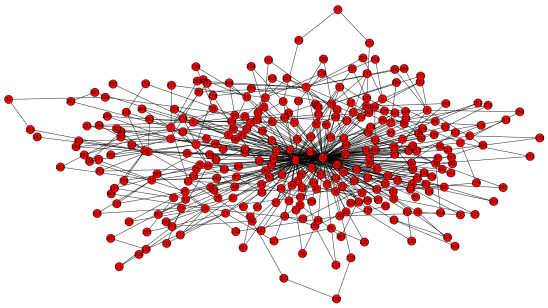

Figure 2.3: A network with $n = 300$ with a *WTA* structure, obtained by cooling an *E-R* graph until $T \approx 0$K. As highlighted by the figure, there are a few winning nodes (clearly visible in the center of the figure).

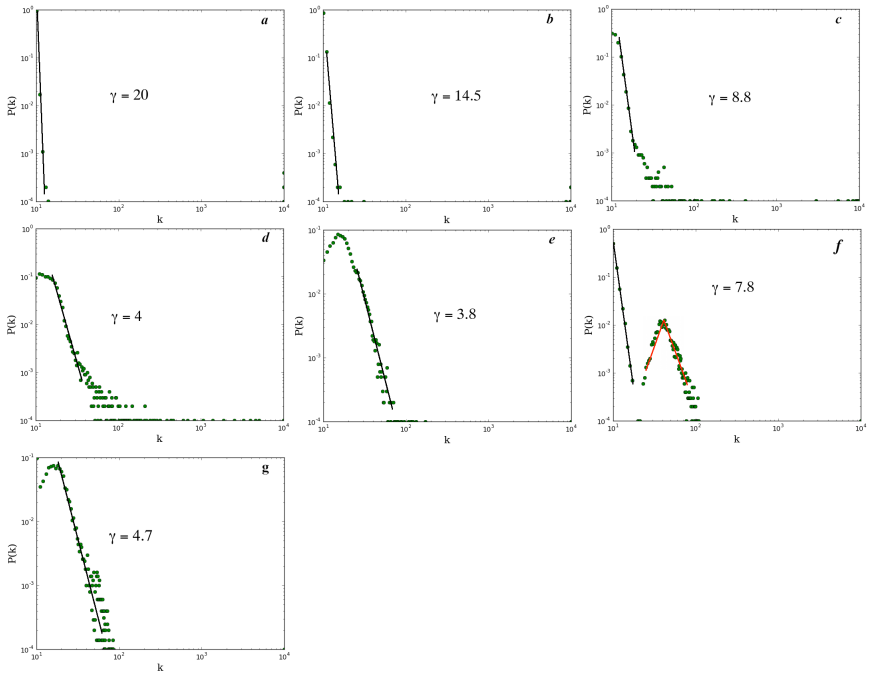

Figure 2.4: The evolution of the degree distribution of a network, during a heating process, with 10000 nodes and $\langle k \rangle = 20$. Each panel shows the network at different time steps t: **a)** at $t = 0$; **b)** at $t = 15$; **c)** at $t = 28$; **d)** at $t = 34$; **e)** at $t = 40$; **f)** at $t = 58$; and **g)** at $t = 65$. Note that for $t = 0$ the network has a *WTA* structure. Continuous black and red lines are used to highlight data interpolation. The corresponding scaling parameter γ is indicated in each panel.

its degree distribution apparently becomes scale-free (see panel **d** and **e** of Figure 2.4). As temperature further increases, it converges to a hybrid distribution (see panel **f** of Figure 2.4) characterized by two main distributions: exponential and gaussian. Eventually, a homogeneous structure ($\gamma = 4.7$) emerges. Similar results have been achieved in all simulations, varying the number of nodes and considering different initial temperatures. Figure 2.5 shows the number of particles that, at each time step during both processes, change their energy level. As temperature decreases, particles move to lower

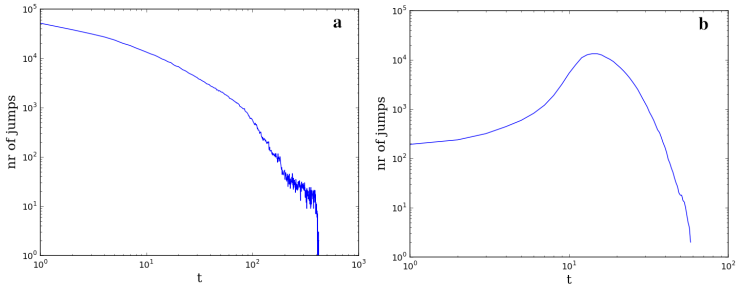

Figure 2.5: Number of particles that change their energy level (indicated as nr of jumps) along time (considering a network with $n = 3000$). **a** During the cooling process, the number of jumps rapidly decreases. **b** During the heating process, at the beginning, all particles are constrained to low-energy levels. After few time steps particles can find more available states in the upper bundles and the number of jumps increases. This curve reaches its maximum when all particles have available upper energy levels to reach, until these top levels become full and the number of jumps begins to decrease. At the end, all the particles are mainly arranged in the higher energy levels.

energy levels until they occupy the deeper bundles; then the number of particles that change their position falls to zero. On the other hand, while heating the system, particles slowly begin to jump to higher energy levels. At the beginning of this process, only few particles move, as the majority of particles are in fact constrained to their level due to the lack of available (close) upper levels. Then, all particles can move and the number of jumps get a maximum, until also the upper levels begin to fill. At the end of the process, all particles mainly occupy the upper energy levels and the number of jumps falls to zero.

2.3.3 Discussion

Fermionic networks show that the emergence of different structures can be represented as a quantum-classical transition for quantum gases. In particular, a *WTA* structure corresponds to a fermionic gas approximated by the quantum regime at low temperatures. On the other hand, a classi-

cal random network corresponds to the same gas in the classical regime at high temperatures. During a cooling process, at intermediate temperatures, a scale-free structure emerges. As shown in Figure 2.2, the *E-R* structure rapidly changes into a scale-free structure, with a scaling parameter of about 2.1. This parameter decreases to 1.38 and afterwards increases until the network loses a neat scale-free structure (see panel **d** of Figure 2.2) and begins to converge to the *WTA* structure characterized by a high value of the scaling parameter. In particular, a homogeneous structure emerges, with the presence of hubs (i.e., nodes with high degree). At the end of the cooling process, few nodes have a very high degree ($\sim n$) and the remaining nodes have low degree. Considering the heating process, we observed that the scaling parameter slowly decreases at the beginning of the process. During the first simulation steps the network apparently converges to a scale-free structure, while for values of the scaling parameter around $3.8 - 3.5$ the network converges to an hybrid structure, which follows an exponential distribution for low values of k and a gaussian distribution for high values of k (see panel **f** of Figure 2.4). Eventually, a homogeneous structure takes over at high temperatures. Surprisingly we found that the whole process, considering both cooling and heating, is not reversible when mapped to networks evolution. Nevertheless, the separation between classical random vs. winner-takes-all networks finds a full correspondence with the separation between classical vs. quantum regions for quantum gases. Other analyses about the connection between classical random and scale-free networks have been reported in [33]. In the cited paper, the authors show that, for cold regime, their network is scale-free, but as the temperature increases the network loses its metric structure and its hierarchical heterogeneous organization, becoming a classical random network.

Considering that many real complex networks are scale-free while others are not (see for example [34]), we deem that the proposed fermionic model can be considered a good candidate for representing their evolution, at low and high temperatures. As shown in Figure 2.5, we analyzed also the dynamics of particles during both processes. In each simulation we observed that the cooling process takes more time to let particles get their final position. During the cooling process, the number of particles changing position is very high from the first time step. Instead, during the heating process we found that, at the beginning, this number is small and rapidly increases after few (usually about 10) time steps. Then, this amount of jumps gets a maximum and begins to decreases until all particles stop moving. We deem that this behavior is an effect of the Eq. (2.14), since it has been defined to avoid that particles occupy densely high energy levels at high temperatures.

Chapter 3

Geometry of Social Networks

In this chapter we study social networks dynamics considering the individual perception, related to the concept of similarity, of people. Similarity is a concept investigated in different sciences, including computer science, cognitive sciences, and mathematics. From a computational perspective, the similarity is usually codified as a distance measure and different metrics can be adopted to compute it. As reported in previous works, some properties of social networks seem to be deeply influenced by the human behavior. For instance, social networks as physics and biology co-authorship show a degree-degree correlation, in contrast with almost all real complex networks that are degree-degree anticorrelated. We propose a model to map the people's behavior while they generate links, considering both similarity and popularity of people. Moreover, we hypothesize that each person has her/his own perception of similarity. To represent this difference among people, we use a hyperbolic model to compute distances, providing each person with individual geometric parameters. Simulations, in accordance with the proposed model, generate small-world networks that show a community structure. Before to introduce the proposed model, we illustrate some key-concepts of hyperbolic geometry and briefly discuss a geometrical approach to the studying of complex networks [33].

3.1 Hyperbolic Models of Complex Networks

In their seminal work, Krioukov et al. [33] investigated the underlying geometry of complex networks. Under proper assumptions, they demonstrate that scale-free networks emerge by connecting points spread in a hyperbolic space. In this section, after briefly recalling some key-concepts of hyperbolic geometry [33], we introduce the geometric model of complex networks. Hyperbolic geometry is a powerful framework used in different domains. Just

to cite few, it is possible to find hyperbolic models in physics [41], visual perception [42], computer science [43], and crystallography [44]. In general, hyperbolic geometry allows to study isotropic spaces with negative curvature. These spaces are difficult to represent and many models have been developed, e.g., the Poincare disk and the Hyperboloid model [45]. For each problem or application, it is possible to choose the most suitable or comfortable hyperbolic representation. The geometric model described in [33] is based on $2D$ hyperbolic spaces $\mathbb{H}^2_\zeta$ with curvature $K = -\zeta^2 < 0$ and $\zeta > 0$. The authors adopt the native representation of $\mathbb{H}^2_\zeta$; hence its the ground space is $\mathbb{R}^2$ and every point $p \in \mathbb{R}^2$ having polar coordinates (r_p, ρ_p) has a hyperbolic distance from the origin equal to r_p. Furthermore, the length of the circle $L(r)$ and the disk area $A(r)$ are computed as follows:

$$\begin{aligned} L(r) &= 2\pi \sinh \zeta r \\ A(r) &= 2\pi(\cosh \zeta r - 1) \end{aligned} \tag{3.1}$$

The distance x between two points (r, θ) and $(r^{'}, \theta^{'})$ is computed as:

$$\cosh \zeta x = \cosh \zeta r \cosh \zeta r^{'} - \sinh \zeta r \sinh \zeta r^{'} \cos \Delta\theta \tag{3.2}$$

with $\Delta\theta = \pi - |\pi - |\theta - \theta^{'}||$ angle between the two points. If $\zeta \to 0$, Eq. (3.1) converge to their Euclidean analogs. This model allows to simulate the network generation considering the similarity among nodes. Here, as usual, the concept of similarity is codified as a distance. Simulations are performed spreading N points in a $2D$ hyperbolic disk with constant curvature $K = -1$, assigning each point an angular coordinate $\theta \in [0, 2\pi]$ and a radial coordinate $r \in [0, R]$. The angular distribution density is uniform, whereas the radial distribution density is exponential:

$$\rho(r) = \alpha \frac{\sinh \alpha r}{\cosh \alpha R - 1} \approx e^{\alpha r} \tag{3.3}$$

with the exponential exponent $\alpha > 0$. The simple way to connect these points is by evaluating their hyperbolic distances using the Heaviside step function, so that the connections are generated when the distance x satisfy the relation $x \leq R$, with R disk radius. Then, a scale-free distribution emerges. Moreover, the authors show how to generate scale-free networks with a desired average degree. Further considerations, related to statistical mechanics models vs their geometric model of complex networks, are discussed.

3.2 Perception-based Model of Social Networks

Let us now introduce our model to study the dynamics of social networks. As in [33], we adopt the *native* representation of $\mathbb{H}^2_\zeta$. We map each person

(user hereinafter) to a point in a hyperbolic disk, identified by the coordinates (r, θ), with $r \in [0, R]$ (R disk radius) and $\theta \in [0, 2\pi]$. The disk radius is computed as $R = \ln(n)$, with n number of users. To generate a network, we let users connect considering their similarity and their popularity. In particular, the similarity is computed as a distance by Eq. (3.2) and the popularity as the node degree. Users are provided with two individual parameters, ζ and ϵ. The curvature ζ allows users to evaluate distances in her/his perceived geometric space (note that all spaces are hyperbolic). The parameter ϵ allows users to decide if other users are similar or not. In particular, users send requests of connection to all other users when the computed distance is $\leq \epsilon R$. After the first step, if two users send each other a request, an edge is drawn between them and they become friends. In so doing, once defined these connections, a first social network emerges. We hypothesize that users can be influenced while evaluating the similarity with other users that share common friends. Under this hypothesis, we let each user decide whether to define other connections with users linked with their friends. At this step, it is worth noting that users define a list of potential friend by a network metric, i.e., only one node (the common friend) separates them. After defining this list of potential friends, each xth user re-computes the distance with her/his yth potential friend (by Eq. (3.2), using ζ_x) and verifies whether the following relation is satisfied:

$$d_{xy} < \ln(f + 2) \cdot \epsilon_x \cdot R \tag{3.4}$$

with f number of common friends. If the constraint imposed by relation (3.4) is satisfied, the xth user sends a request of connection with probability:

$$p = 1 - \frac{1}{2} \ln(\zeta_x) \tag{3.5}$$

Once again, users send and receive requests of connections and, as before, define new connections for mutual requests. Finally, all users can have a list of applicants, i.e., users they consider unlike, which sent a request of connection. Hence, they can accept or decline requests and their decision is based on the popularity of these applicants, i.e., on the applicants degree. In particular, at this step, users accept these connections only if applicants have equal or higher degree. Each user can become an applicant if she/he sends a request to another user that considers her/him unlike. The proposed model can be summarized as follows:

1. Define n users and assign them polar coordinates ($\theta \in [0, 2\pi]$, r as defined in Eq. (3.3)), a curvature ζ and a parameter ϵ

2. Each xth user computes the distance d_{xy} with the yth user, by Eq. (3.2) using ζ_x, and sends a request of connection to y if $d_{xy} \leq \epsilon_x R$

3. If two users send reciprocally a request of connection, an edge is drawn between them and they become friends

4. Each user generate a list of potential friends, i.e., users linked with their friends

5. Each user re-computes the distance with her/his potential friend (by Eq. (3.2) and sends a request of connection, with probability of Eq. (3.5), if the condition of Eq. (3.4) is satisfied

6. If two users send reciprocally a request of connection, an edge is drawn between them and they become friends

7. Each user accepts requests of connection from users she/he considers unlike, if applicants have equal or higher degree

3.3 Simulations

Synthetic social networks generated by the proposed model have been analyzed considering different parameters, i.e., assortativity, average clustering coefficient, and average degree. Furthermore, we evaluated if these networks show a community structure as many real social networks do [8]. We performed many simulations with a number of agents in the range $[1500, 5000]$. Each agent is mapped to a point on a hyperbolic disk. In particular, we spread points with uniform angular density $\rho(\theta) = \frac{1}{2\pi}$ and with exponential radial density $\rho(r) \approx e^{\alpha r}$, having $\alpha \in [2, 3]$. Agents were provided with the curvature ζ in range $(1, \frac{\ln n}{3})$, and with the parameter ϵ in range $[0, 75 - 1]$. Figure 3.1 shows how these parameters influence the concept of similarity of each agent. Figure 3.2 shows the difference among networks generated by using different α in the radial distribution. It is worth noting the strong difference among networks generated with different values of α. Table 3.1 illustrates measured properties of simulated networks. As discussed before, it is possible to evaluate if simulated networks are small-world. Hence, for each network, we generated the corresponding E-R graph (i.e., the classical random network generated on the same set of nodes) and we measured the shortest path length and the average clustering coefficient. Furthermore, as the distance between randomly chosen nodes in a small-world network is $\approx \ln(n)$, we computed the average distance between randomly chosen nodes of simulated network. Results of this comparison are reported in the Table 3.2. Finally, we analyzed also the degree distribution of each network. Figure 3.3 shows the degree distribution of a network with $n = 2500$. As shown in Figure 3.3, simulated networks are not scale-free. On the contrary, for α equal to 2.5 and to 3, we achieved $P(k)$ functions similar to that of E-R graphs.

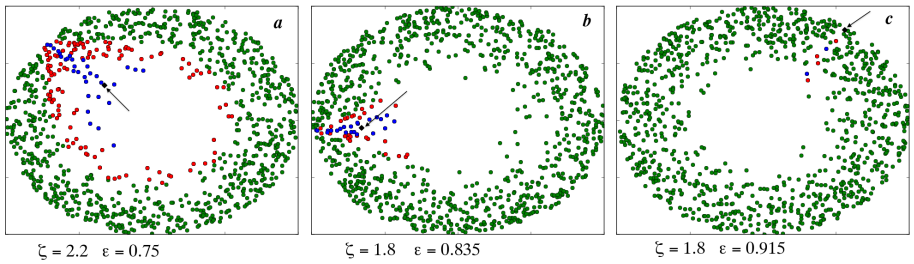

Figure 3.1: From left to right, distribution of 1000 agents in the hyperbolic disk, with $\alpha = 2$. In each panel the arrow points a randomly chosen agent, say x, indicated as a black diamond. Blue points have a hyperbolic distance less than ϵR from x. Red points have a hyperbolic distance less than R from x. ζ and ϵ values of each x point are indicated under the related panel. **a** The x point is far from the disk boundary, hence it has many neighbor points. **b** The x point is not far from the disk boundary. **c** The x point is almost on the disk boundary, hence it has very few neighbor points.

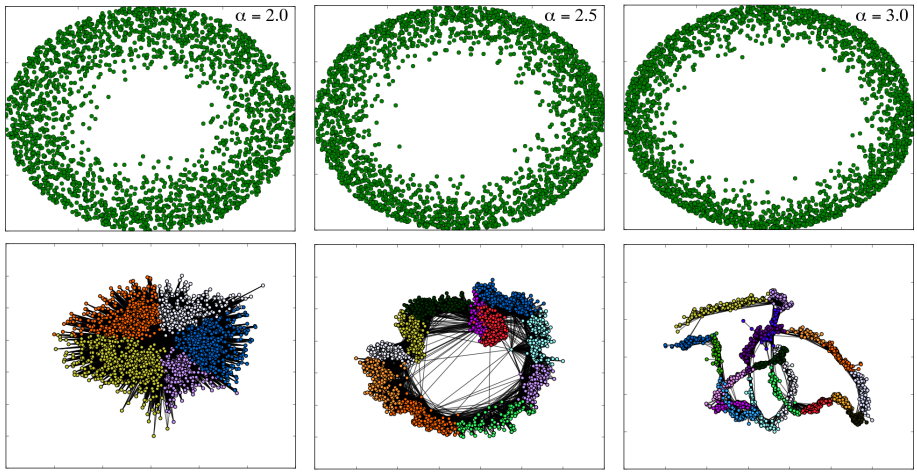

Figure 3.2: From left to right, distribution of 2500 agents in the hyperbolic disk varying the value of α (indicated in each panel). The related networks, generated with the proposed model, are placed below each disk. Each color identifies a community.

3.3.1 Discussion

Results of simulations clearly highlight that the network structure is deeply affected by the exponent α. In particular, the density of edges decreases as α increases. Reducing the density of edges the shortest path length increases, hence there is a correlation between α and the shortest path length –see Table 3.1. Also the number of communities depends on α, in particular,

n	α	Assortativity	avgCC	SPL	Communities	$\langle k \rangle$
1500	2.0	-0.12	0.748	2.18	6	92.12
1500	2.5	$+0.03$	0.74	5.37	11	22.59
1500	3.0	-0.11	0.75	10.71	16	14.16
2500	2.0	-0.15	0.756	2.15	5	131.78
2500	2.5	-0.12	0.77	3.39	11	32.6
2500	3.0	-0.2	0.746	10.7	19	17.28
3000	2.0	-0.126	0.747	2.22	5	138.77
3000	2.5	-0.124	0.761	3.87	13	31.85
3000	3.0	-0.18	0.757	8.7	17	19.1
5000	2.0	-0.17	0.76	2.1	5	231.72
5000	2.5	-0.11	0.781	3.28	10	46.06
5000	3.0	-0.2	0.763	8.05	20	22.5

Table 3.1: Properties of simulated networks. n indicates the number of nodes, α the exponent of the radial distribution, avgCC the average clustering coefficient, SPL the shortest path length, Communities the number of identified communities, and $\langle k \rangle$ the average degree.

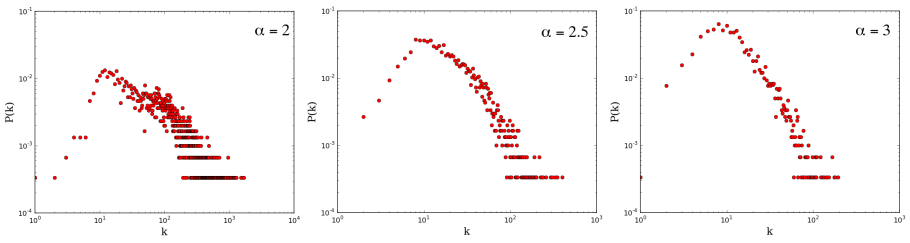

Figure 3.3: From left to right, degree distribution of networks with $n = 2500$ achieved varying the value of α (indicated in each panel).

this number increases with α. On the other hand, the assortativity and the average clustering coefficient seem to do not depend on α. It is interesting to note that the assortativity is almost always negative, then the related networks are disassortative. Although in [12] authors stated that social networks use to be assortative, in [34] author reports that there are social networks as "student relationship" that are disassortative. As observed before, we found that simulated networks are not scale-free. In particular, for $\alpha = 2.5$ and $\alpha = 3$ the $\langle k \rangle$ is at the top of $P(k)$ and the degree distribution decays exponentially for large values of k. These results are in accordance with those reported in [1]. Also the parameters ζ and ϵ play an important role in the network structure. In [33], simulated networks get a scale-free structure using a generic $\alpha > 0$, hence the assigning of different curvatures ζ to agents yields this difference with the theoretical model of Krioukov

n	α	Δ avgCC	Δ SPL	Δrand. Distance
1500	2.0	0.686	0.245	1.02
1500	2.5	0.733	2.68	2.41
1500	3.0	0.74	7.68	7.05
2500	2.0	0.7	0.21	1.26
2500	2.5	0.75	0.76	0.755
2500	3.0	0.74	7.69	7.63
3000	2.0	0.7	0.27	1.253
3000	2.5	0.75	1.18	0.925
3000	3.0	0.751	5.71	5.61
5000	2.0	0.713	0.154	1.58
5000	2.5	0.772	0.644	0.724
5000	3.0	0.759	5.04	4.68

Table 3.2: Comparison between simulated network and their related E-R graphs. n and α identifies the simulated network, Δ avgCC is the difference between average clustering coefficients, Δ SPL is the difference between shortest path length, Δrand. Distance is the difference between the expected distance in a small-world network of that size, i.e., $\approx \ln(n)$, and that computed in simulated networks considering randomly chosen nodes.

et al.. To state that only ζ is responsible for this behavior, we performed simulations setting $\epsilon = 1$ for all agents, achieving degree distributions similar to that of Figure 3.3. On the other hand, the parameter ϵ strongly affects the number of possible friends of each agent, as it can be observed in Figure 3.1. Both parameters, ζ and ϵ, contribute to the generation of the potential friends list. ζ represents the individual perception of similarity, whereas ϵ allows to represent the individual maximum distance to consider someone similar. Results of numerical simulations, shown in Table 3.2, suggest that for $\alpha = 2.5$ it is possible to achieve small-world networks. In particular, comparing simulated networks with their related E-R graphs (as discussed before), the difference of average clustering coefficient is always quite high (about 2 orders of magnitude). On the other hand, the shortest path lengths are not always equivalent or quit similar. Furthermore, the difference in distance between randomly chosen nodes and the theoretical value of $\ln(n)$ strongly increases for $\alpha = 3$.

Chapter 4

Emergence of Acronyms

In this chapter we propose a model constructed by the framework of complex networks to study a specific problem in language dynamics [46], i.e., the emergence of acronyms in a community of language users. Language is a complex system that evolves over time, due to several phenomena. In recent years, new communication media are affecting interpersonal written communication. In particular, mobile phones and internet-based communication media are leading people to use a small number of characters when message writing. Hence, in most cases acronyms or abbreviations are used. For instance, a mobile phone message is usually composed of short phrases, the social network Twitter only allows 140 characters for each message (called tweet) and in many online forums users have limited space for each question or answer. Although the use of acronyms dates back to ancient times, nowadays this type of linguistic sign is gaining prestige. In this work, we study the introduction of acronyms in social systems. In particular, we define a simple game for the purpose of analysing how the use of an acronym spreads in a population, considering its ability to create shared meaning. We performed many numerical simulations using the proposed model, showing the creation of acronyms to be the result of collective dynamics in a population. After a brief introduction to the modelling of linguistics phenomena, we discuss two famous language games. Then, we illustrate the proposed model.

4.1 Modelling Linguistics Phenomena

Language can be considered as a complex system that evolves over time, involving several different phenomena, in every community. In general, studies of language are strongly interdisciplinary as language can be analysed from different perspectives, e.g., linguistics, philosophical, cognitive, computational and also statistical mechanics [47][48][49][50][46]. The relatively new

field of language dynamics [46] deals with language evolution, mainly using computational tools and the statistical mechanics framework. One of the insightful results achieved is modelling the creation of a common linguistic convention in a community of language users as the result of their interactions in a complex dynamical system. In particular, the creation of a language is modeled by a game among agents. The first observations in this direction were developed by Wittgenstein [51] and, later, Steels [52] developed a model representing the linguistics behavior as a series of language games. Baronchelli et al. [53] introduced a microscopic model of communicating autonomous agents inspired by the Naming Game [52]. Following this, mapping communities of language users to complex networks became natural, as social relations such as friendship [54][30][55] can be represented by links among nodes. For example, Dall'Asta et al. [56] studied the dynamics of the Naming Game on complex networks. One of the phenomena involved in language evolution is lexical innovation, i.e., new terms added to people's vocabulary. Lexical innovations, or neologisms, have a meaningful role and usually an aesthetic effect. Furthermore, they are adopted from a language for different needs, such as for new inventions (e.g., the term television), or if a bilingual person introduces a loanword. Nowadays, written communication takes place largely using mobile phones and internet-based media. Both categories are leading people to use a small number of characters. Hence, acronyms are commonly used. Acronyms are linguistic signs (signs, hereinafter) composed of the initial components of other signs or of a phrase. Each component can be an individual character or part of a sign. Formally, a sign [47] is the fundamental unit of language and it comprises two elements, a signifier and a signified. The former is the shape of a word and its phonic component, whereas a signified is the ideational component, the meaning or the concept appearing in our mind when we hear or read the signifier. The signified is related to a referent, the actual object or concept. Although the creation of acronyms dates back to antiquity, e.g., SPQR (Senatus PopulusQue Romanus) or CEO (Chief Executive Officer), it seems that nowadays this kind of linguistic sign is gaining prestige. Acronyms such as ASAP (As Soon As Possible), BTW (By The Way) and LOL (Laughing Out Loud) commonly appear in web sites, chat-rooms or web-forums. Therefore, when a new acronym is defined by a writer, it can be regarded as a lexical innovation. Although acronyms are signs, they were generated and evolved differently from other signs. Usually, a normal reader is able to recognize whether a sign is an acronym, so she/he has to think of all possible meanings considering the related context. Therefore, more than one meaning can be associated with an acronym and, after some time, a common meaning is shared among people. This latter consideration means that people must converge to a common opinion. Similar problems have already been studied, e.g., Starnini et al. defined a Voter Model with a number of states from 2 to ∞ [57], Sood et al. analyzed the Voter Model on graphs [58] and Krapivsky

et al. defined a model of opinion dynamics [59]. In this work we study the introduction of acronyms in a community of language users. In particular, we illustrate a model where people, by their interactions, converge to a common meaning for each new acronym. After a brief theoretical analysis of the Shannon entropy of acronyms, we show results of numerical simulations to analyse the average number of meanings related to an acronym and the evolution of the system as its use spreads, up until the definition of a common meaning. The remainder of the chapter is organized as follows: Section 4.2 gives a brief introduction to the most famous language games. Section 4.3 introduces our model on the emergence of acronyms. Section 4.4 shows a theoretical analysis of the Shannon entropy of acronyms, and results of the numerical simulations.

4.2 Language Games

In this section, we briefly introduce two famous models, Naming Game and Category Game [60], to study the emergence of language in populations.

4.2.1 Naming Game

The Naming Game is played by N players with the aim to define a common vocabulary for M objects present in their environment. Here, the term object takes a generic meaning, which includes physical objects, concepts, and people. Each player has a vocabulary to represent all objects she/he knows. All vocabularies are empty at $t = 0$. At each time step, two randomly selected players interact, one as speaker and one as listener. The following list of rules governs their interactions:

- the speaker selects an object from the context;
- the speakers searches the word, from its vocabulary, associated with the object and, if it does not exist, she/he invents a new word;
- the speaker transmits the word to the listener;
- if the listener knows the word and she/he associates it with the intended object, the interaction is a success and both players maintain in their vocabulary only the winning word;
- if the interaction is a failure, the listener adds the word with that meaning in her/his vocabulary.

Later on, a simpler model, called the minimal language game, has been defined by Baronchelli et al. [53] This game is played by N players, arranged in a fully connected network.

4.2.2 Category Game

The Category Game is a minimal model for linguistic categorization. This game involves N players engaged in categorising a single analogic perceptual channel. Each stimulus is coded as a real number in the range $[0, 1]$. The categorisation is identified by partitioning the interval $[0, 1]$, defining sub-intervals called perceptual categories. Each player has a dynamic inventory of form-meaning associations, linking perceptual categories to words (forms), which represent their linguistic counterpart. Perceptual categories and the corresponding words co-evolve dynamically through elementary interactions among players. At $t = 0$, all players have only the perceptual category $[0, 1]$, with no name associated to it. At each time step, two randomly selected players interact in front of a set of objects. One player (the speaker) says the name of one object, while the other player (the listener) tries to guess the named object. The game is successful when the listener gives the right answer.

4.3 Acronyms Game

Let us now introduce a novel model to study the emergence of acronyms in a community of language users. The proposed model, called Acronyms Game (AG hereinafter), considers a system with N interacting agents which communicate by a linguistic convention. We assume that this convention is based on a common vocabulary of signs. Relations among signs are mapped to a network, that has been given the name of *NetSigns*. The edges of this network are generated between signs that can be used together, since their combination has a logic meaning. Agents can play the role of writer or reader and, in both cases, they know the rules for generating/codifying an acronym, i.e., using only the first character of each sign. Provided that more than one meaning can be associated with an acronym, many different signs can be created with the same signifier. When agents converge to a common meaning, only the related sign is kept in their vocabulary. The game comprises two main phases: a spreading phase and a converging phase. During the first phase, the acronym spreads through the system, while during the second, agents try to converge to a common meaning.

Spreading Phase. This phase begins with an agent, in the role of writer, sending to a reader a message that contains an acronym. Both players are randomly selected, on the condition that the reader still does not know the acronym. In particular, the writer generates a message of random length and decides to introduce an acronym related to the last z signs. For example, it wants to send a friend the following message: "Do not wait for me, because I'm still working" and it writes: "Do not wait for me, because I'm **sw**". The reader receives the message and thinks of some possible meanings for the acronyms. At each step, all agents who know the acronym become

writers. This process continues until all agents have read and saved the acronym with a list of possible meanings. *Converging Phase.* This phase begins with a random writer which sends a message, containing the acronym, to a random reader. The writer, before sending the message, chooses the most appropriate meaning among those saved. If the reader has the same meaning for that acronym, both agents keep only that meaning and delete the others. If not, the reader adds the chosen meaning to its list of meanings. This phase ends when all agents converge to a common meaning for the acronym. Summarizing, AG can be described as follows:

- *Spreading Phase.* An agent generates an acronym to be spread throughout the system.

    1. Two agents are randomly selected, one as writer and one as reader;
    2. The writer generates a message (in accordance with the constraints defined in *NetSigns*), which contains an acronym related the last z signs, and sends it to the reader;
    3. The reader reads the message and thinks about some possible meanings for the acronym;
    4. If the reader finds at least one possible meaning, it becomes writer (and, in this phase, it will never play again the role of reader);
    5. Each writer of the system sends a message using the acronym (choosing the most appropriate meaning among those saved) to a randomly selected reader. In the event that no reader exists, the spreading phase is over.

- *Converging Phase.* All agents know the acronym (with a list of possible meanings). They can play the role of writer and reader, and must converge to a common meaning.

    1. Two agents are randomly selected, one as writer and one as reader;
    2. The writer generates a message that contains the acronym. To use the acronym, it must choose the most appropriate meaning among those saved;
    3. The reader reads the message. If it has the chosen meaning in its list of meanings, the communication is successful and both agents keep only that meaning and delete the others; otherwise the reader adds the chosen meaning to its list of meanings;
    4. Repeat from (i) until all agents converge to a common meaning for the acronym.

To complete the model let us assume that agents play AG in a fully-connected network.

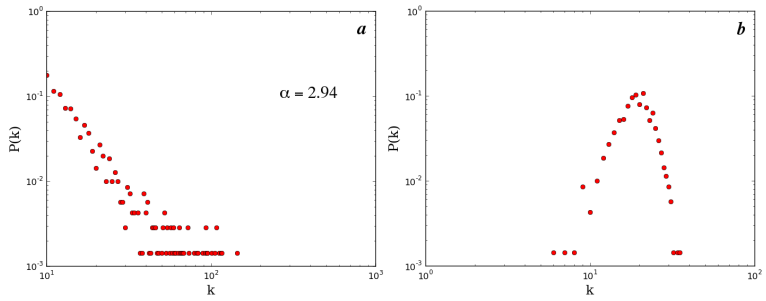

Figure 4.1: Degree distribution of *NetSigns* used in simulations: **a**) Scale-free structure. **b**) E-R graph structure.

4.4 Results

We performed many numerical simulations to study the dynamics of AG, considering a number of agents $N \in [100, 1600]$ and acronyms of length $z \in [2, 4]$. Agents interact in a fully-connected network using a common vocabulary of 700 signs. In order to make the game as realistic as possible, we limited the number of solutions (i.e., the saved meanings) for each agent, as well as the number of attempts to codify it. Thus, when an agent reads a new acronym, it tries to codify it in at most 100 attempts, saving a maximum of 5 different possible meanings. If it is not able to find a meaning, it will be considered again, at the next time step, as a possible reader. As signs must be linked to one another in a network, we generated *NetSigns* in accordance with two main structures: scale-free and Erdos-Renyi graph (E-R graph hereinafter). The use of a scale-free structure has been inspired from the works of Motter et al. [61] and from that of i Cancho et al. [62], whereas the use of E-R graph structures has been chosen for the sake of comparison. To generate networks with scale-free structure, we adopted the BA model [1], setting $m_0 = 10$. In so doing, we computed a value of around 2.94 for the scaling parameter α (related to the degree distribution $P(k) \sim k^{-\alpha}$) and an average degree $\langle k \rangle \sim 20$ –see panel **a** of Figure 4.1. Then, we generated networks with E-R graph structure, provided with the same set of nodes and with the same density of edges achieved in the related scale-free networks –see panel **b** of Figure 4.1. We hypothesize that the structure of *NetSigns* can affect the Shannon entropy of generated acronyms. In turn, we deem that Shannon entropy can affect the spreading phase of AG. Hence, before studying AG we developed a preliminary analysis of the relations between the Shannon entropy of acronyms and the structure of *NetSigns*.

4.4.1 Shannon entropy of acronyms

This analysis allows to compare the two structures adopted for *NetSigns*. To perform this comparison we used a fully-connected network as reference. Generally speaking, when any such network is used, there exists a very large set of possible solutions for each acronym, with a cardinality equal to:

$$|\Omega_{fc}| = \prod_{i=1}^{z} \omega_i \tag{4.1}$$

where z is the acronym length, $|\Omega_{fc}|$ is the cardinality of the set of all possible solutions considering a fully-connected *NetSigns*, and ω_i is the number of signs having the i-th character as initial character. In the proposed model, the number of possible solutions can be substantially reduced as we used scale-free or E-R graph structures. In particular, $|\Omega_{sf}| \leq |\Omega_{fc}|$ and $|\Omega_{er}| \leq |\Omega_{fc}|$ (with $|\Omega_{sf}|$ and $|\Omega_{er}|$ cardinality of the set of all possible solutions considering a *NetSigns* with the scale-free and E-R graph structure). The reason of the reported inequalities hold is that not all signs in these network structures are reciprocally linked. Hence, when one or more characters of the acronym are initials of signs with low degree (i.e., with few neighbours), the number of possible solutions decreases. In the case of fully-connected networks, the Shannon entropy [63] can be computed as follows:

$$H_{fc} = \sum_{i=1}^{z} h_i \tag{4.2}$$

where h_i is the entropy of each character of the acronym, computed as:

$$h_i = \sum_{j=1}^{\omega_i} s_j \log_2 \frac{1}{s_j} = \log_2 \omega_i \tag{4.3}$$

with s_j probability of the j-th sign. In the case of a scale-free or E-R graph structure, the Shannon entropy H_r can be computed as follows:

$$H_r = \sum_{j_1} \sum_{j_2} ... \sum_{j_z} s_{j_1,j_2,...,j_z} \log_2 \frac{1}{s_{j_1,j_2,...,j_z}} \tag{4.4}$$

with z acronym length and $s_{j_1,j_2,...,j_z}$ probability of all the signs occurring together. The utilization of these two structures, playing the AG game, offers a computational disadvantage in the task of codifying an acronym, if compared to the utilization of a fully-connected *NetSigns*. The task of codifying an acronym entails two steps:

- select signs in agreement with letters of the acronym
- verify that the selected signs are linked (i.e., there is a path in the network of signs that connects them)

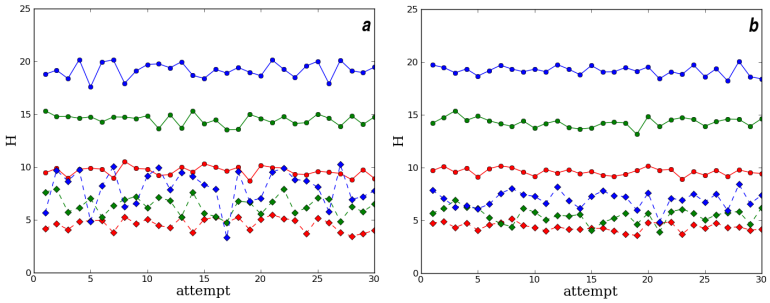

Figure 4.2: Comparison between Shannon entropies achieved at different attempts (i.e., random generated acronyms). Red lines indicate acronyms of length 2, green lines indicate acronyms of length 3 and blue lines indicate acronyms of length 4. The figure reports: **a**) fully-connected networks (continuous lines) vs scale-free structure (dotted lines). **b**) fully-connected networks (continuous lines) vs E-R graph structure (dotted lines).

As a consequence, when signs are linked in scale-free networks or E-R graph, agents spend more time to identify sequences of linked signs, whereas in a fully-connected network a path for any sequence of signs always exists. Given these considerations, we analysed the Shannon entropy for a set of randomly generated acronyms, comparing the difference between scale-free networks or E-R graph and fully-connected networks–see Figure 4.2. Then, we compared the relative gain, in terms of Shannon entropy of randomly generated acronyms, achieved by using scale-free or E-R graph structures instead of fully-connected networks–see Table 4.1. Results achieved in this

length of acronym	$\%G$ scale-free	$\%G$ E-R graph
2	55.32	54.44
3	55.95	61.93
4	58.24	64.43

Table 4.1: Average gains, in terms of Shannon entropy of randomly generated acronyms, obtained between *NetSigns* with scale-free and E-R graph structure. The table reports the relative gain of Shannon entropy achieved by using scale-free "$\%G$ scale-free" or E-R graph "$\%G$ E-R" structures.

preliminary analysis suggest that performing AG with a *NetSigns* provided set with a scale-free structure should be more convenient than using an E-R graph structure. In particular, the Shannon entropy gain for scale-free structures is about 55.51%, whereas the gain for E-R graph is about 59.93%. A higher gain implies a lesser Shannon entropy. We hypothesize that scale-free networks yield acronyms with higher Shannon entropy due to the presence of hubs, i.e., of signs with many connections. Although from

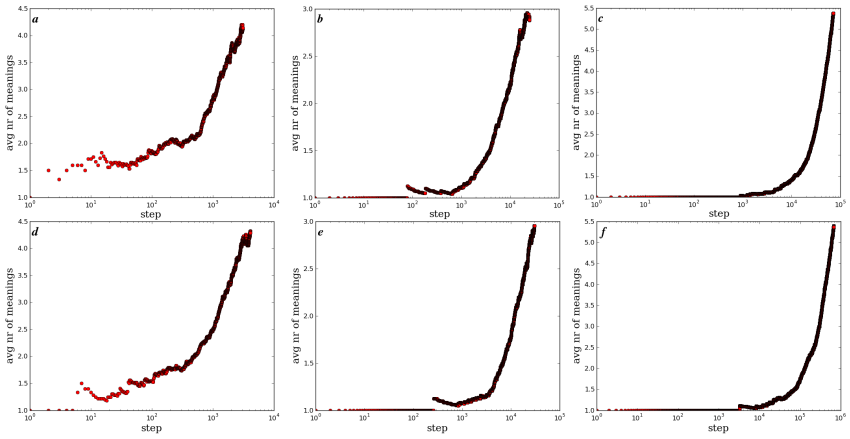

Figure 4.3: Average number of meanings for an acronym over time, in a population of 400 agents during the spreading phase of AG. On top, results achieved by using a scale-free *NetSigns*: **a**) Acronym of 2 characters; **b**) Acronym of 3 characters; **c**) Acronym of 4 characters. At the bottom, results achieved by using an E-R graph structure for *NetSigns*: **d**) Acronym of 2 characters; **e**) Acronym of 3 characters; **f**) Acronym of 4 characters.

a computational perspective it is better to manage data with low entropy, in AG the problem is not to find a particular solution, but to find a generic fitting solution, i.e., a sequence of linked signs. Hence, higher Shannon entropy implies a greater amount of potential fitting solutions.

4.4.2 Numerical simulations of AG

As discussed before, AG is composed by two main phases: a spreading phase and a converging phase. Since agents must codify acronyms during the spreading phase, we expect during this phase, AG be faster using *NetSigns* with a scale-free structure than using an E-R graph structure. On the other hand, using scale-free structures, it is less likely that the meaning of the acronym defined by the first random player will be equal to that defined by the whole population at the end of the game. Once discussed these theoretical issues, we ran AG considering both structures, scale-free and E-R graph, for the *NetSigns*. Figure 4.3 shows results of the spreading phase of AG for a population of 400 agents. During the spreading phase, the average number of meanings for an acronym increases over time steps, as agents can save up to 5 meanings. As shown in Figures 4.3 and 4.4, theoretical considerations made in the preliminary analysis of Shannon entropy are fundamentally confirmed. The most trivial observations are that the number of time steps increases with the number of agents, and that the number of time steps required to complete the spreading phase increases with the length of the acronym (see panels **a-b-c** and panels **d-e-f** of Fig-

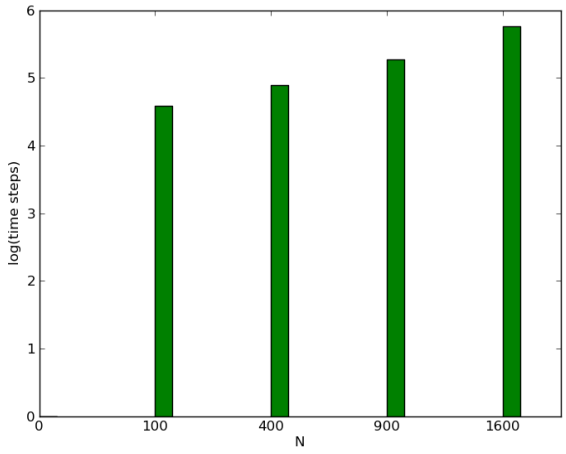

Figure 4.4: Number of time steps (on a logscale) to complete the game, varying the number of agents from 100 to 1600, with acronyms of length 4.

ure 4.3). Furthermore, apparently the length of acronyms affects also the first time steps of this phase. In particular, the average number of meanings increases more slowly than the increasing length of the acronym. This behavior can be caused by the increasing difficulties that agents have to dealt with while trying to identify solutions, as acronyms increase their length. As a consequence, the average number of meanings increases when more agents are involved. Notwithstanding this phenomenon, it is worth noting that the maximum number of average meanings does not seem to depend on the length of acronyms. Further observations can be made by comparing the results achieved by using different structures of *NetSigns*. As expected, after the Shannon entropy analysis, the duration of the spreading phase is higher for agents which use a *NetSigns* with an E-R graph structure, as the entropy of its acronyms is lower. Furthermore, using the E-R graph structure more agents must be involved before that the average number of meanings increases. This last phenomenon is clear shown in panels **b-e** and **c-f** of Figure 4.3. On the other hand, contrary to expectation given by Shannon entropy analysis, the maximum average number of meanings does not seem to depend on the structure of *NetSigns*, although scale-free networks offer more fitting solutions. It is worth highlighting that this last achievement could be affected by the constraint we imposed in the system, i.e., each agent can save no more than 5 meanings for each acronym. Figure 4.5 shows results of the whole dynamics of AG. In diagrams of Figure 4.5 the two phases of AG are well characterized. In particular, the first phase ends when the curve achieves its maximum (or after few time steps –see panel **b** of Figure 4.3).The converging phase of AG is much more faster than the

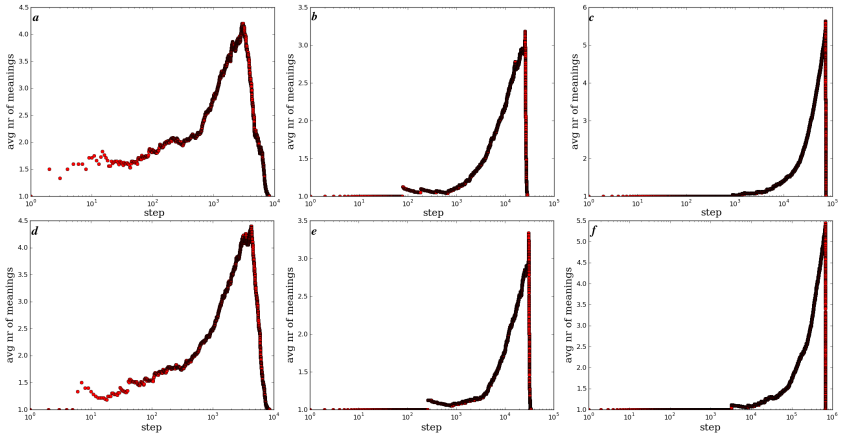

Figure 4.5: Average number of meanings for an acronym over time, in a population of 400 agents, considering the spreading and the converging phase. On top, results achieved by using a scale-free *NetSigns*: **a**) Acronym of 2 characters; **b**) Acronym of 3 characters; **c**) Acronym of 4 characters. At the bottom, results achieved by using a E-R graph structure for *NetSigns*: **d**) Acronym of 2 characters; **e**) Acronym of 3 characters; **f**) Acronym of 4 characters.

first one and ends when the curve (i.e., the average number of meanings) is 1. The converging phase has no relations with the Shannon entropy of acronyms, as each agent knows at least one possible meaning. Furthermore, there are no relations between the length of acronyms and the duration of the converging phase. On the other hand, this second phase can be affected only by the number of agents playing AG. Eventually we found that, in all simulations, the proposed model allowed agents to converge to a unique common solution. We then analysed the evolution of the system, as illustrated in Figure 4.6. We represented the information that each agent has about the acronym, e.g., whether it knows it or not, by a color. In particular, agents are red if they have never heard about the acronym (or have no meanings for it); black if they know it and have several meanings for it; cyan if they know it and have only one meaning for it. In the latter case, if the assigned meaning is equal to that assigned by the acronym's inventor, the color is turned to green. It is worth recalling that, despite of the appearance of Figure 4.6, we analysed AG considering agents linked in a fully-connected network. In some cases the final meaning adopted is equal to that assigned by its inventor, but in general we did not observe any particular bias toward this specific state.

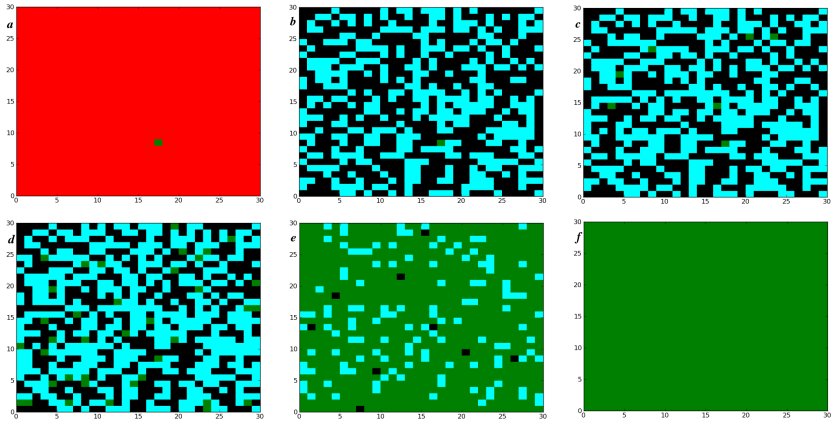

Figure 4.6: Evolution of a population composed by 900 agents, which use a scale-free *NetSigns*, during the introduction of a 3-character acronym. Colors codify the state of each agent. Red if she/he does not know the acronym; green if she/he knows the acronym with its initial meaning; black if the agent assigns more than one meaning and cyan assigns a meaning different from the original one. **a**) The system at the beginning of the game; **b**) The end of the first phase; **c**) Second phase after 500 time steps; **d**) Second phase after 1000 time steps; **e**) Second phase after 5000 time steps; **f**) The system at the end of the game.

Chapter 5

Clustering Datasets by Community Detection

In this chapter, we propose a method based on complex networks analysis, devised to perform clustering on multidimensional datasets [64]. For example, datasets containing information about users of a social networks. The proposed method maps the elements of the dataset in hand to a weighted network according to the similarity that holds among data. Network weights are computed by transforming the Euclidean distances measured between data according to a Gaussian model. Notably, this model depends on a parameter that controls the shape of the actual functions. Running the Gaussian transformation with different values of the parameter allows to perform multiresolution analysis, which gives important information about the number of clusters expected to be optimal or suboptimal. Solutions obtained running the proposed method on simple synthetic datasets allowed to identify a recurrent pattern, which has been found in more complex, synthetic and real, datasets.

5.1 Clustering vs Community Detection

Community detection is one of the most important processes in complex network analysis, aimed at identifying groups of highly mutually interconnected nodes, called communities [65], in a relational space. From a complex network perspective, a community is identified after modelling any given dataset as graph. For instance, a social network inherently contains communities of people linked by some (typically binary) relations –e.g., friendship, sports, hobbies, movies, books, or religion. On the other hand, from a machine learning perspective, a community can be thought of as a cluster. In this case, elements of the domain are usually described by a set

of features, or properties, which permit to assign each instance a point in a multidimensional space. The concept of similarity is prominent here, as clusters are typically identified by focusing on common properties (e.g., age, employment, health records).

The problem of clustering multidimensional datasets without a priori knowledge about them is still open in the machine learning community (see, for example, [66][67][68]). Although complex networks are apparently more suited to deal with relations rather than properties, nothing prevents from representing a dataset as complex network. In fact, the idea of viewing datasets as networks of data has already been developed in previous works. Just to cite few, Heimo et al. [69] studied the problem of multiresolution module detection in dense weighted networks, using a weighted version of the q-state Potts method. Mucha et al. [70] developed a generalized framework to study community structures of arbitrary multislice networks. Toivonen et al. [71] used network methods in analyzing similarity data with the aim to study Finnish emotion concepts. Furthermore, a similar approach has been developed by Gudkov et al. [72], who devised and implemented a method for detecting communities and hierarchical substructures in complex networks. The method represents nodes as point masses in an $N-1$ dimensional space and uses a linear model to account for mutual interactions.

The motivation for representing a dataset as graph lies in the fact that very effective algorithms exist on the complex network side to perform community detection. Hence, these algorithms could be used to perform clustering once the given dataset has been given a graph-based representation. Following this insight, in this paper we propose a method for clustering multidimensional datasets in which they are first mapped to weighted networks and then community detection is enforced to identify relevant clusters. A Gaussian transformation is used to turn distances of the original (i.e. feature-based) space to link weights of the complex networks side. As the underlying Gaussian model is parametric, the possibility to run Gaussian transformations multiple times (while varying the parameter) is exploited to perform multiresolution analysis, aimed at identifying the optimal or sub-optimal number of clusters.

The proposed method, called *DAN* (standing for Datasets as Networks), makes a step forward in the direction of investigating the possibility of using complex network analysis as a proper machine learning tool.

5.2 Clustering Datasets

Before to introduce *DAN*let us spend few words on classical clustering algorithms. Cluster analysis (or simply *clustering*) is an *unsupervised learning* approach, directly exploiting regularities in the data to be analysed, that builds a higher level representation to be used for reasoning or prediction;

this higher level description is usually in the form of groups or, in a more abstract view, it represents a partitioning of the data.

K-means

As centroid-based clustering is one of the most acknowledged clustering strategies, the $k-Means$ algorithm (e.g., [73]), which belongs to this family, has been selected as one of the comparative tools. For the sake of completeness, let us briefly summarize it:

1. Randomly place k centroids in the given metric space;
2. Assign each sample to the closest centroid, thus identifying tentative clusters;
3. Compute the Center of Mass (CM) of each cluster;
4. IF CMs and centroids (nearly) coincide THEN STOP;
5. Let CMs become the new centroids;
6. REPEAT from STEP 2.

The evaluation function of $k - Means$, called *distortion* and usually denoted as J, is computed according to the formula:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \left| s_i^{(j)} - c_j \right|^2 \tag{5.1}$$

where n_j is the number of samples that belong to the j-th cluster, $s_i^{(j)}$ is the i-th sample belonging to j-th cluster, and c_j its centroid. Note that different outputs of the algorithm can be compared in terms of distortion only after fixing k –i.e., the number of clusters. In fact, comparisons performed over different values of k are not feasible, as the more k increases the lower the distortion is. For this reason, the use of $k - Means$ entails a main issue: how to identify the optimal number k of centroids (see [74]).

Spectral Clustering

Spectral clustering [15] algorithms use the spectrum of the similarity matrix to identify relevant clusters (the generic element of a similarity matrix measures the similarity between the corresponding data). These methods allow to perform dimensionality reduction, so that clustering can be enforced along fewer dimensions. Similarity matrices can be generated in different ways – e.g., ϵ-neighborhood graph, k-nearest neighbor graphs and fully connected graph. The main tools for spectral clustering are graph Laplacian matrices.

In particular, in this work we used the unnormalized graph Laplacian matrix defined as:

$$L = D - W \tag{5.2}$$

where D is the degree matrix (i.e., a diagonal matrix with the degrees $d_1, ..., d_n$ on the diagonal) and W is the adjacency (or similarity) matrix of the similarity graph. The following algorithm has been used to perform unnormalized spectral clustering:

1. Generate the fully connected similarity graph and let W be its adjacency matrix;
2. Compute the unnormalized Laplacian L;
3. Compute the first k eigenvectors $u_1, ..., u_k$ of L;
4. Let $U \in \Re^k$ be the matrix containing the eigenvectors $u_1, ..., u_k$ as columns;
5. For $i = 1, ..., n$, let $y_i \in \Re^k$ be the vector corresponding to the i-th row of U;
6. Cluster the points $(y_i)_{i=1,...,n}$ in $\Re^k$ with the k-means algorithm into clusters $C_1, ..., C_k$.

Notably, also in this case the number k of cluster is required as input.

5.3 Datatasets as Networks

The first step of the *DAN* method consists of mapping the dataset in hand to a complex network. The easiest way to use a complex network for encoding a dataset is to let nodes denote the elements of the dataset and links denote their similarity. In particular, we assume that the weight of a link depends only on the distance among the involved elements. To put the model into practice, we defined a family of Gaussian functions –used for computing the weight between two elements.

Computing similarity among data

Let us briefly recall that a metric space is identified by a set $\mathcal{Z}$, together with a distance function $d : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R}$, like Euclidean, Manhattan and Chebyshev distances. In *DAN*, the underlying assumption is that a sample s can be described by N features $f_1, f_2, ..., f_N$, encoded as real numbers. In other words, the sample can be represented as a vector in an N-dimensional metric space $\mathcal{S}$. Our goal is to generate a fully connected weighted network taking into account the distances that hold in $\mathcal{S}$. Conversely, the complex network space will be denoted as $\mathcal{N}$, with the underlying assumption that for each sample $s_i \in \mathcal{S}$ a corresponding $n_i \in \mathcal{N}$ exists and vice versa. This assumption makes easier to evaluate the proximity value L_{ij} between two

$n_i, n_j \in \mathcal{N}$, according to the distance d_{ij} between the corresponding elements $s_i, s_j \in \mathcal{S}$.

Without loss of generality, let us assume that each feature in $\mathcal{S}$ is normalized in $[0, 1]$ and that a function $\psi : \mathbb{R} \to \mathbb{R}$ exists for computing the similarity among data in $\mathcal{N}$, starting from the value of the distance function in $\mathcal{S}$. In symbols:

$$L(n_i, n_j) = L_{ij} \stackrel{\triangle}{=} \psi(d_{ij}) = \psi(d(s_i, s_j)) \tag{5.3}$$

Evaluating similarity for all pairs of samples in $\mathcal{N}$ (i.e., evaluating their weighted links) allows to generate a fully connected complex network. Moreover, recalling that $\mathcal{S}$ is normalized in $[0, 1]$, we expect $L_{ij} \approx 0$ when $d_{ij} \approx \sqrt{N}$, N being the number of features of the space $\mathcal{S}$. The value $\sqrt{N}$ comes from the following inequality, which holds for any pair of samples $s_i, s_j \in \mathcal{S}$ (represented by their vector representation in terms of the given set of features $\mathbf{r_i}, \mathbf{r_j}$):

$$d_{ij} = \sqrt{\sum_{k=1}^{N} (\mathbf{r_i}[k] - \mathbf{r_j}[k])^2} \leq \sqrt{N} \tag{5.4}$$

where $\mathbf{r_i}[k]$ denotes the k-th component of $\mathbf{r_i}$.

Multiresolution Analysis

Let us recall that multiresolution analysis is performed with the goal of extracting relevant information, useful for identifying the optimal or suboptimal number of communities (hence, of clusters). To perform multiresolution analysis on the network space, a parametric family $\Psi(\lambda) : \mathbb{R} \to \mathbb{R}$ of functions is required, where λ is a parameter that controls the shape of each ψ function. After setting a value for λ, the corresponding ψ can be used to convert the distance computed for each pair of samples in the given dataset into a proximity value. In particular, the following parametric family of Gaussian functions has been experimented:

$$\Psi(\lambda; x) = e^{-\lambda x^2} \tag{5.5}$$

As a consequence, L_{ij}, i.e. the weight of the link between two nodes $n_i, n_j \in \mathcal{N}$, can be evaluated according to Equation 5.5 as follows:

$$L_{ij} \stackrel{\triangle}{=} \psi(\lambda; d_{ij}) = e^{-\lambda d_{ij}^2} \tag{5.6}$$

where the λ parameter is used as a constant decay of the link.

Following the definition of $\Psi(\lambda; x)$ as $e^{-\lambda x^2}$, multiresolution analysis takes place varying the value of the λ parameter. The specific strategy adopted for varying λ is described in the experimental section. As for now,

let us note that an exponential function with negative constant decay ensures that distant points in an Euclidean space are loosely coupled in the network space and vice versa. Moreover, this construction is useful only if $\Psi(\lambda; x)$ models local neighborhoods, which gives further support to the choice of Gaussian functions [15].

5.4 Results

Experiments have been divided in three main groups: i) *preliminary tests*, aimed at running *DAN* on few and relatively simple synthetic datasets, ii) *proper tests*, aimed at running *DAN* on more complex datasets, and iii) *comparisons*, aimed at assessing the behavior of *DAN* with reference to $k-Means$ and spectral clustering. Furthermore, we implemented *DAN* by using the Louvain method [16] to perform the community detection.

Almost all datasets used for experiments (except for Iris) are synthetic and have been generated according to the following algorithm:

Inputs: number of samples (n), dimension in the Euclidean space (N), number of clusters (k), and radius of a cluster (r)

1. For each cluster $j = 1, 2, ..., k$, choose a random position c_j in the normalized Euclidean space;

2. Equally subdivide samples among clusters and randomly spread them around each position c_j, with a distance from c_j in $[0, r]$.

Preliminary Tests

A first group of 4 synthetic datasets, called *TS/1* (i.e., Testing Set 1) hereinafter, has been generated. Their main characteristics are summarized in Table 5.1.

Group	*Dim*	N_s	N_c	μ_r	σ_r
TS/1	2D	1897	5	0.4	0.3
	3D	1683	3	0.09	0.04
	3D	1500	10	0.42	0.22
	4D	1680	6	0.62	0.45

Table 5.1: Features of datasets used for preliminary tests (*TS/1*) – *Dim*, N_s, and N_c denote the dimension of datasets, the number of samples, and the intrinsic number of clusters. Moreover, μ_r and σ_r denote the average radium and the variance of samples.

Figure 5.1 shows the datasets with 3 and 10 clusters, together with the optimal solutions achieved by *DAN*.

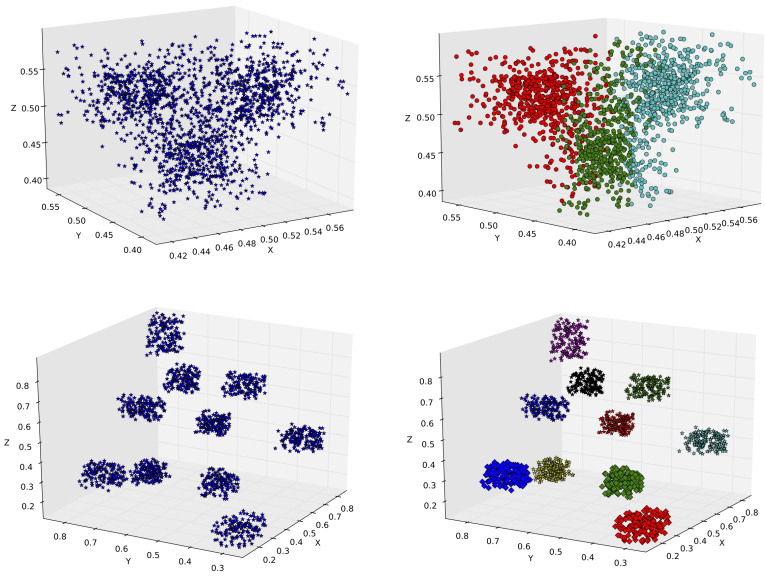

Figure 5.1: Second and third datasets of *TS/1*, together with the solutions achieved by *DAN* using $\log_{10}(\lambda) = 3$ (each cluster has been colored with a different color).

Multiresolution analysis has been performed varying the value of λ according to Equation 5.5. A logarithm scaling has been used for λ, as we experimentally found that small changes had a negligible impact on the corresponding algorithm for community detection. In particular, for each dataset, we calculated the adjacency matrix for all values of λ such that $\log_{10}(\lambda) = 0, 1, 2, 3, 4$. It is worth pointing out that the maximum value of $\log_{10}(\lambda)$ is expected to depend on the cardinality of the dataset in hand –the greater the cardinality, the greater the value of $\log_{10}(\lambda)$. However, for most datasets, a value of $\log_{10}(\lambda) = 4$, i.e., $\lambda = 10,000$, appears to be large enough to include all relevant information by means of multiresolution analysis. Table 5.2 shows the results of multiresolution analysis for preliminary tests.

As for the capability of identifying the optimal or suboptimal solutions[1] by means of multiresolution analysis, we observed the following pattern to occur: the optimal number of communities is robust with respect to

[1] As pointed out by Arenas et al. [75], it may not appropriate to speak of correct vs. incorrect solutions for multiresolution analysis. In a context of community detection we deem more appropriate to speak of optimal or suboptimal solutions (see also [76] for more information on this issue).

Group	N_c	Number of Clusters				
TS/1	5	2	3	**5**	**5**	**5**
	3	**3**	**3**	**3**	**3**	103
	10	2	3	**10**	**10**	151
	6	2	4	**6**	**6**	37
		0	1	2	3	4
		$\log_{10}(\lambda)$				

Table 5.2: Results of multiresolution analysis achieved during preliminary tests. The number of communities is reported, calculated for $\log_{10}(\lambda) = 0, 1, 2, 3, 4$. Optimal values are highlighted in bold.

the values of $\log_{10}(\lambda)$, as highlighted in Table 5.2. Our hypothesis was that this recurrent pattern could be considered as a decision rule for identifying the optimal number of communities (and hence of λ).

Proper Tests (*TS/2*)

We generated a second group of datasets, characterized by an increasing complexity with respect to *TS/1*. This second group of datasets is denoted as *TS/2* (i.e., Testing Set 2) hereinafter. We run *DAN* also on these new datasets, with the goal of verifying the validity of the pattern identified during preliminary tests. Moreover, we performed experiments using *Iris*, a well-known multivariate real dataset available at the UCI ML repository [77]. Iris contains 50 samples (described by 4 attributes) belonging to 3 species of Iris: setosa, virginica and versicolor. Table 5.3 summarizes the main characteristics of *TS/2* and *Iris*. The corresponding results, obtained with *DAN*, are shown in Table 5.4.

Looking at these results, we still observe the pattern identified by preliminary tests. Furthermore, one may note that a correlation often exists between the cardinality of the dataset in hand and the order of magnitude of its optimal λ (typically, the former and the latter have the same order of magnitude). It is also interesting to note that in some datasets of *TS/1* (i.e., 2nd, 3rd and 4th) and of *TS/2* (i.e., 4th, 5th and 6th) the optimal λ precedes a rapid increase in the number of communities. As a final note, we found no significant correlation between the optimal λ and the weighted-modularity parameter, notwithstanding the fact that this parameter is typically important to assess the performance of the adopted community detection algorithm.

Group	*Dim*	N_s	N_c	μ_r	σ_r
TS/2	3D	350	5	0.35	0.19
	3D	2000	20	0.44	0.2
	3D	5000	30	0.51	0.24
	4D	535	4	0.64	0.46
	8D	1680	6	0.86	0.62
	12D	930	8	1.22	0.88
Iris	4D	150	3	0.49	0.26

Table 5.3: Characteristics of datasets used for proper tests (*TS/2*), listed out according to the group they belong to. *Dim*, N_s, and N_c denote the dimension of datasets, the number of samples, and the intrinsic number of clusters. Moreover, μ_r and σ_r denote the average radium and the variance of samples.

Group	N_c	Pattern	Number of Clusters				
TS/2	5	✓	3	**5**	**5**	8	84
	20	✓	3	4	16	**20**	**21**
	30	✓	4	5	21	**30**	**30**
	4	✓	2	**4**	**4**	105	181
	6	✓	2	4	**6**	**6**	1186
	8	✓	3	5	**8**	**8**	875
Iris	3	✓	**3**	**3**	10	82	147
			0	1	2	3	4
						$\log_{10}(\lambda)$	

Table 5.4: Results of multiresolution analysis on the selected datasets during proper tests, listed out according to the group they belong to. The number of communities is reported, calculated for $\log_{10}(\lambda) = 0, 1, 2, 3, 4$. Optimal values are reported in bold. The patterns observed on synthetic datasets (and reported in the table for the sake of completeness), allows to easily compute the expected optimal number of communities also for *Iris*.

Comparison: *DAN* vs. k-Means and Spectral Clustering

We run the k-Means algorithm (using the Euclidean metric) and the spectral clustering algorithm on the selected datasets –with the goal of getting new insights on the results of the partitioning procedure defined in *DAN*. Both algorithms used for comparison purposes have been run using the optimal

values of k identified by means of multiresolution analysis. The comparison has been performed considering the distortion J computed for each solution. Figure 5.2 reports comparative results and clearly shows that, in around 72.2 percent of the cases, *DAN* achieves the best result.

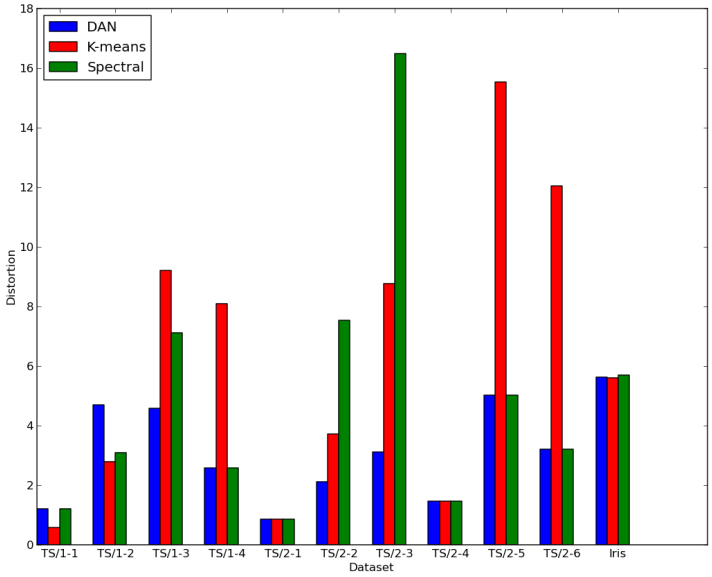

Figure 5.2: Comparison, in terms of distortion, among solutions achieved by *DAN*, blue bars, $k-Means$, red bars and spectral clustering, green bars (the lesser the better).

These results highlight the validity of the proposed framework, also considering that *DAN* computes partitions without any a priori knowledge about the datasets, as the optimal (or suboptimal) number of clusters is typically found by applying the previously described pattern. Although $k-Means$ is faster than *DAN*, it is important to stress that its results, at each attempt, depend tightly on the initial position of the k centroids. Hence, in absence of a strategy for identifying the initial disposal of centroids, $k-Means$ should be (and it is in fact) run several times –the solution with the smaller distortion being selected as optimal. The spectral clustering algorithm showed its effectiveness many times, although bad solutions have been computed with datasets 2 and 3 of *TS/2*, characterized by 20 and 30 clusters, respectively.

Chapter 6

Conclusions

In this work we illustrate different models of social behaviors as competitive dynamics, social networks dynamics, and emergence of linguistics phenomena. Furthermore, we introduce a novel framework, based on complex networks, to clustering datasets, as real datasets containing information about social networks users. In Chapter 2 a new theoretical model has been introduced, based on complex networks and inspired by quantum statistics, to study competitive dynamics. We define a fermionic network model that allows to represent complex networks as quantum gases. Using this model, we show that network evolution is a temperature-dependent process characterized by three main phases: classical random, scale-free and winner-takes-all. The network evolution and the system temperature represent, respectively, the evolution of a social system and the level of competitiveness of the system itself. Performing a cooling process, the transition from classical random to scale-free networks takes place. Notably, the system achieves equilibrium when a winner-takes-all structure is reached, despite the non-equilibrium nature of the network evolution. On the other hand, performing a heating process which starts from a winner-takes-all structure, the network evolution follows a slightly different path. In particular, a pure scale-free structure is not reached, although the actual structure is very similar. Surprisingly, we found that the whole process, considering both cooling and heating, is not reversible when mapped to networks evolution. Finally, we observe that classical random networks (i.e., quantum gases in the classical region) can represent social systems where components (i.e., people) do not compete among themselves, scale-free networks (i.e., quantum gases between the classical and the quantum region) can represent social systems with a medium level of competitiveness, whereas the winner-takes-all networks (i.e., quantum gases in the quantum region) can represent social systems characterized by a high level of competitiveness. In Chapter 3 we study social networks dynamics considering the individual perception, related to the concept of similarity, of people. To study these dynamics we developed a hyperbolic model of

social networks. In particular, we represent the dynamics of link generation among people, considering both similarity and popularity. Similarity is codified as a hyperbolic distance between people, which in turn are embedded in their individual hyperbolic space. This concept has been put into practice by providing people with an individual space curvature ζ. Furthermore, we provide people with a coefficient that allows them to evaluate the maximum distance to consider someone else similar. On the other hand, we codified the popularity as the number of peoples links, hence it is not subject to individual evaluation as for similarity. Results of simulations show that the proposed model yields networks with a community structure, a degree distribution like that of E-R graphs, and for $\alpha \geq 2.5$ with small-world behavior as many real social networks show. In Chapter 4 we study the emergence of acronyms in a community of language users. The study of the introduction of acronyms by means of a statistical mechanics approach makes possible to represent macroscopic collective dynamics, today (and in the past), existing in many human languages. We introduce a model, called Acronyms Game, where agents interact following a set of rules in a fully-connected network. These simple interactions involve the creation of an acronym with a shared common meaning. Agents use a common vocabulary of signs that are linked in a network, with a scale-free structure or E-R graph structure. The use of a network of signs allows us to represent real scenarios as, usually, when people try to codify an acronym they do not think of all possible signs for each character but, where possible, try to consider only signs related to the specific context of the reading. In this study, we develop a brief analysis of the Shannon entropy of acronyms. In particular, we highlight the effects of the structure of a network of signs for agents, during the task of codifying acronyms. Numerical simulations show that the final adopted meaning of an acronym is usually different from the one assigned by its inventor. As for future work, we are planning to analyse the Acronyms Game in social networks, in particular, examining different network structures and rules for spreading the acronym and its evolution. In Chapter 5 we describe a method for clustering multidimensional datasets, able to find the most appropriate number of clusters also in absence of a priori knowledge. We have shown that community detection can be effectively used also for data clustering tasks, provided that datasets are viewed as complex networks. The proposed method, called *DAN*, makes use of transformations between metric spaces and enforces multiresolution analysis. A comparative assessment with other wellknown clustering algorithms (i.e., k-Means and spectral clustering) has also been performed, showing that *DAN*often computes better results. In the light of these results, we deem *DAN*can be adopted as general framework to clustering datasets of social networks users, since each user is described by a large number of features. Finally, in this thesis we offered some examples for modelling social behaviors by an interdisciplinary approach. In particular, we adopted the modern network theory as general

framework and, when useful, we used powerful tools of statistical mechanics. In the light of these results, we deem that the application of "hard sciences" to social sciences can be useful to represent many complex dynamics and to achieve important information about social and economical systems.

List of Publications Related to the Thesis

1. Marco Alberto Javarone and Giuliano Armano, "Quantum Classical Transitions in Complex Networks", Journal of Statistical Mechanics: Theory and Experiment (JSTAT), 2013

2. Marco Alberto Javarone and Giuliano Armano, "Phase Transitions in Fermionic Networks", 11th International Conference on Adaptive and Natural Computing Algorithms (ICANNGA13), LNCS Springer 2013

3. Giuliano Armano and Marco Alberto Javarone, "Clustering Datatsets by Complex Networks Analysis", Complex Adaptive Systems Modeling (CASM), 1:5, 2012

4. Marco Alberto Javarone and Giuliano Armano, "Perception of Similarity: a Model for Social Networks Dynamics" (submitted to JSTAT)

5. Marco Alberto Javarone and Giuliano Armano, " Emergence of Acronyms in a Community of Language Users" (submitted to JSTAT)

Bibliography

[1] Albert,R. and Barabasi, A.L., Statistical Mechanics of Complex Networks. *Reviews of Modern Physics* **74**, 47–97 (2002)

[2] Guimer, R., Danon, L., Diaz-Guilera, A., Giralt, F. and Arenas, A, Self-similar community structure in a network of human interactions. *Physical Review E* **68** (2003)

[3] Boccaletti, S., Latora, V., Moreno, Y.Chavez, M., HWang, D.U., Complex networks: Structure and dynamics. *Physics Reports* **424**, 175 – 308, (2006)

[4] Barabasi, A.L., Albert, R., Emergence of Scaling in Random Networks. *Science* **15**, 509–512 (1999)

[5] Easley, D., Kleinberg, J., Networks Crowds and Markets *Cambridge University Press* (2010)

[6] Watts, D. J. and Strogatz, S. H., Collective dynamics of "small-world" networks. *Nature* 440-442 (1998)

[7] Brandes, O., A Faster Algorithm for Betweenness Centrality *The Journal of Mathematical Sociology* **25 - 1** (2001)

[8] Girvan, M., Newman, M.E.J., Community structure in social and biological networks. *PNAS* **99** 7821–7826 (202)

[9] Mohring, R.H., Schilling, H., Schutz, B., Wagner, D., Willhalm, T., Partitioning Graphs to Speed Up Dijkstras Algorithm. Experimental and Efficient Algorithms LNCS, 3503, 189–202 (2005)

[10] Han, S.C., Franchetti, F., Pushel, M., Program Generation for all-pairs shortes path problem. Proc. of the 15th Int.Conf. on Parallel architectures and compilation techniques, 22–232 (2006)

[11] Newman, M.E.J., Mixing Patterns in Networks. *Physical Review E* **2** 026126 (2003)

[12] Johnson, S., Torres, J. J., Marro, J., Munoz, M.A., Entropic Origin of Disassortativity in Complex Networks. *Physical Review Letters*, **104 - 10** 108702 (2010)

[13] P. Erdos, P. and Renyi, A., On the Evolution of Random Graphs. *publication of the mathematical institute of the hungarian academy of sciences* 17–61 (1960)

[14] Fortunato,S, Community detection in graphs. *Physics Reports* **486** 2566–2572 (2002)

[15] Luxburg, U., A tutorial on spectral clustering. *Statistics and Computing* **17 - 4** 395–416 (2007)

[16] Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E., Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* P10008 (2008)

[17] Tyler, J.R., Wilkinson, D.M., Huberman, B.A., Email as Spectroscopy: Automated discovery of Community Structure within Organizations. *Proc. Communities and technologies* 81– 96(2003)

[18] Kernighan, B.W., Lin, S., An Efficient Heuristic Procedure for Partitioning Graphs. *Bell Sys. Tech. J.g* **49 - 2** 291– 308(1970)

[19] Wu, F.Y.,, The Potts Model. *Reviews of Modern Physics* **54 - 1** 235–268 (1982)

[20] Reichardt, J., Bernholdt, S., Detecting Fuzzy Community Structures in Complex Networks with a Potts Model. *Physical Review Letters* **93 - 21** 218701 (2004)

[21] Zhou, H., Network landscape from a Brownian particle's perspective. *Physical Review E* **67 - 4** 041908 (2003)

[22] Arenas, A., Diaz-Guilera, A.,Perez-Vicente, C.J., Synchronization Reveals Topological Scales in Complex Networks. *Physical Review Letters* **96 - 11** 114102 (2006)

[23] Kuramoto, Y., Chemical Oscillations, Waves, and Turbulence. *Springer-Verlag New York* 164 (1984)

[24] Javarone, M.A., Armano, G., Quantum-Classical Transitions in Complex Networks. *Journal of Statistical Mechanics: Theory and Experiment* (2013)

[25] Javarone, M.A., Armano, G., Phase Transitions in Fermionic Networks. *Proc. of International Conference on Adaptive and Natural Computing Algorithms*, Lausanne, Switzerland (2013)

[26] Huang, K., Statistical Mechanics. *John Wiley and Sons* (1987)

[27] Hartonen,T., Annila, A., Natural Networks as Thermodynamic Systems. *Complexity* (2012)

[28] Bianconi,G., Barabasi,A.L., Bose-Einstein Condensation in Complex Networks. *Physical Review Letters* **86** 5632–5635 (2001)

[29] Bianconi, G., Quantum statistics in complex networks. *Physical Review E* **66** (2002)

[30] Newman, M.E.J., Watts, D.J., Strogatz, S.H., Random Graphs Models of Social Networks. *PNAS* **99** 2566–2572 (2002)

[31] Clauset, A., Shalizi, C.R., Newman, M.E.J., Power-Law distributions in empirical data. *SIAM Review* **51** 661–703 (2009)

[32] Milojevic, S., Power-law Distributions in Information Science - Making the Case for Logarithmic Binning. *Journal of the American Society for Information Science (JASIST)* **61 - 12** 24172425 (2010)

[33] Krioukov, D., Papadopoulos, F., Kitsak, M., Vahdat, A. and Boguna, M., Hyperbolic Geometry of Complex Networks. *Physical Review E* (2010)

[34] Newman, M.E.J., The structure and function of complex networks. *SIAM Reviews* **45** 167–256 (2003)

[35] Arthur,D. Motwani, R., Sharma, A., Xu, Y., Pricing Strategies for Viral Marketing on Social Networks. Internet and Network Economics LNCS, 5929, 101–112 (2009)

[36] Pempek, T., Yermolayeva, A.Y., Calvert, S.L., College students' social networking experiences on Facebook. Journal of Applied Developmental Psychology, 30, 227–238 (2009)

[37] Sparrow, M.K., The application of network analysis to criminal intelligence: An assessment of the prospects. Social Networks, 3, 251–274 (1991)

[38] Morselli, C., Giguere, C., Petit, K., The application of network analysis to criminal intelligence: An assessment of the prospects. Social Networks, 1, 143–153 (2007)

[39] McPherson, M., Smith-Lovin, L., Cook, J.M., Birds of a Feather: Homophily in Social Networks. Ann. Rev. Sociol., 27, 415–444 (2001)

[40] Papadopoulos, F., Kitsak, M., Serrano, M.A., Boguna, M., Krioukov, D., Popularity versus similarity in growing networks. Nature, 489, 537 - 540 (2012)

[41] Barbera, E., Curro, C., Valenti, G., A hyperbolic model for the effects of urbanization on air pollution. Applied Mathematical Modelling, 34, 2192 - 2202 (2010)

[42] Indow, T., Hyperbolic Representation of Global Structure of Visual Space. Journal of Mathematical Psychology, 41, 89 - 98 (1997)

[43] Munzner, T., Exploring large graphs in 3D hyperbolic space. Computer Graphics and Applications, IEEE 18, 18 - 23 (1998)

[44] Hyde, S.T., Friedrichs, O.D., Ramsden, S.J., Robins, V., Towards enumeration of crystalline frameworks: the 2D hyperbolic approach. Solid State Science, 8, 740 - 752 (2006)

[45] Anderson, J.W., Hyperbolic Geometry. Springer, (2005)

[46] Loreto, V., Baronchelli, A., Mukherjee, A., Puglisi, A., Tria, F., Statistical Physics of Language Dynamics. *Journal of Statistical Mechanics: Theory and Experiment* (2011)

[47] de Saussure, F., Course in General Linguistics. *Lectures at the University of Geneve 1906 - 1911* (1916)

[48] Hauser M.D., Chomsky, N., Fitch, W.T., The Faculty of Language: What Is It, Who Has It, and How Did It Evolve?. *Science* **298-5598** 1569–1579 (2002)

[49] Arbib, M.A., From Mirror Neurons to Computational Neurolinguistics. *IJCNN - International Joint Conference on Neural Networks. IEEE* 184–190 (2009)

[50] Lycan, W.G., Philosophy of language: A contemporary introduction. *Taylor and Francis* (2012)

[51] Wittgenstein, L., Philosophical Investigations. (1974)

[52] Steels, L. A self-organizing spatial vocabulary. *Artificial Life Journal* **2-319** (1995)

[53] Baronchelli A., Felici M., Loreto, V., Caglioti, E., Steels, L., Sharp transition towards shared vocabularies in multi-agent systems. *Journal of Statistical Mechanics: Theory and Experiment* (2006)

[54] Jin, E.M., Girvan, M., Newman, M.E.J., Structure of growing social networks. *Physical Review E* **64** (2001)

[55] Newman, M.E.J., Scientific collaboration networks. I. Network construction and fundamental results. *Physical Review E* **64** (2001)

[56] Dall'Asta, L., Baronchelli, A., Barrat, A., Loreto, V., Non-equilibrium dynamics of language games on complex networks. *Physical Review E* **74** (2006)

[57] Starnini, M., Baronchelli, A., Pastor-Satorras, R., Ordering dynamics of the multi-state voter model. *Journal of Statistical Mechanics: Theory and Experiment* (2012)

[58] Sood, V., Redner, S., Voter Model on Heterogeneous Graphs. *Physical Review Letters* **94 - 17** 178701 (2005)

[59] Krapivsky, P. L., Redner, S., Dynamics of Majority Rule in Two-State Interacting Spin Systems. *Physical Review Letters* **90 - 23** 238701 (2003)

[60] Puglisi, A., Baronchelli, A., Loreto, V., Cultural route to the emergence of linguistic categories. *PNAS* (2008)

[61] Motter, Adilson E., de Moura, Alessandro, P. S., Lai, Ying-Cheng, Dasgupta, P., Topology of the conceptual network of language. *Physical Review E* **65 - 6** 065102 (2002)

[62] i Cancho,R.F., Sol, R.V., The Small World of Human Language. *Proceedings of The Royal Society of London. Series B, Biological Sciences* **268-1482** 2261–2266 (2001)

[63] Annick, L., Jean-Luc, B., Pezard, L., Entropy estimation of very short symbolic sequences. *Physical Review E* **79 - 4** 046208 (2009)

[64] Armano, G., Javarone, M.A., Clustering datasets by complex networks analysis. *Complex Adaptive Systems Modeling* **1:5** (2013)

[65] Newman, M.E.J., Girvan, M., Finding and evaluating community structure in networks. *Physical Review E* **69** 026113 (2004)

[66] Jain, A.K., Data clustering: 50 years beyond Kmeans. *Pattern Recognition Letters* **31 - 8** 651-666 (2010)

[67] Tibshirani, R., Walther, G., Hastie, T., Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **63 - 2** 411-423 (2001)

[68] Hansen, H.M., Bin, Y., Model Selection and the Principle of Minimum Description Length. *Journal of the American Statistical Association* **96** 746-774 (1998)

[69] Heimo, T., Kaski, K., Kumpula, J.M., Saramaki, J., Detecting modules in dense weighted networks with the Potts method. *Journal of Statistical Mechanics: Theory and Experiment* **08** 08007 (2008)

[70] Mucha, P., Richardson, T., Macon, K., Porter, M., Onnela, J., Community Structure in TimeDependent, Multiscale, and Multiplex Networks. *Science* **328 - 5980** 876-878 (2010)

[71] Toivonen, R., Kivela, M., Saramaki, J., Viinikainen, M., Vanhatalo, M., Sams, M., Networks of Emotion Concepts. *PLoS ONE* **7 - 1** e28883 (2012)

[72] Gudkov, V., Montealegre, V., Nussinov, S., Nussinov, Z., Community detection in complex networks by dynamical simplex evolution. *Physical Rev E* **78** 016113 (2008)

[73] Alsabti, K., An efficient kmeans clustering algorithm. *Proc. of IPPS/SPDP Workshop on High Performance Data Mining* (1998)

[74] Eick, C., Zeidat, N., Zhao, Z., Supervised Clustering Algorithms and Benefits. *Proc. of ICTAI* (2004)

[75] Arenas, A., Fernandez, A., Gomez, S., Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics* **10 - 5** 053039 (2008)

[76] Li, Z., Hu, Y., Xu, B., Di, Z., Fan, Y., Detecting the optimal number of communities in complex networks *Physica A: Statistical Mechanics and Its Applications* **391** 1770-1776 (2011)

[77] Frank, A., Asuncion, A., UCI Machine Learning Repository. *http://archive.ics.uci.edu/ml* (2010)