Alma Mater Studiorum
Università di Bologna

DOTTORATO DI RICERCA IN
MATEMATICA
Ciclo XXXI

Settore concorsuale: 01/A4 Fisica matematica
Settore scientifico-disciplinare: MAT/07

STATISTICAL MECHANICS AND LEARNING PROBLEMS IN NEURAL NETWORKS

Presentata da:
RACHELE LUZI

Coordinatore Dottorato:
Giovanna Citti

Supervisore:
Armando Bazzani

Esame finale anno 2019

Contents

Chapter 1

Introduction

Statistical mechanics has greatly contributed in studying many-body systems with the purpose of deriving the properties of macroscopic bodies, as their thermodynamical behaviour. Complex systems research is becoming ever more important in many fields of science and other disciplines. A complex system, is a system made from a large number of interacting component, evolving in a non-linear way with respect to initial conditions or with respect to small perturbations. The evolution occurs through multiple pathways and often interactions are not known. The system can be modelled by a graph which is defined by a set of nodes (or vertices) and a set of links (or edges) that identify relations between nodes. In addiction they are useful to model different kinds of relations and processes in physical, biological, social and information systems and can represent many pratical problems.

In sociology graphs are used to measure actors'prestige or to explore rumor spreading generally using social network analysis softwares [1, 2]. Social networks can be modelled by *friendship graphs* which connect people if they know each other, while *influence graphs* model whether certain people can influence the behaviour of their followers. In statistical physics a graph can be used to represent a system of interacting particles (which are identified by the nodes) putting an edge between two of them if they interact. The importance of interactions is moduled with an eventual weight. Also neural networks can be naturally modelled by networks since they trivially introduce a dynamics over the network structure coming from neurons dynamics.

Similarly, in computational neuroscience, graphs model functional connections between brain areas that interact to give rise to various cognitive processes. For what it concerns computer science, graphs are used to represent network of communication, web network, data organization, etc. For example, the link structure of a website can be represented by a directed graph where web pages are identified by nodes and the directed edges are assigned if a link connect one page to another one. A similar approach can be applied to problems that concern social media, travel, biology, biochemical networks, ecological systems etc. In chemistry, graphs make a natural model for the interaction in molecular structures, where vertices and edges respectively represent atoms and bonds, while in biology they can be used to model the migration of species between regions.

In recent years, statistical mechanics is developing its interest in studying learning systems. Neural network systems are made by a large number of interacting simple elements called neurons, which are binary devices that switch from one state to another when input exceeds a particular threshold value. More precisely, a neuron receives an input value which is transformed to an output value that acts as an input for the successive layer of neurons and so on. Each neuron is a mathematical continuous function that models the functioning of a biological neuron: it computes the weighted average of the given input and the sum is passed through a non linear function, often called activation function (one common activation function is the sigmoid function [3].

Neural networks due their name to the structure of the nervous system and are used as model to study the rise of cognitive and learning behaviors like it happens in brain (for instance phychologists use neural networks to understand cognitive processes in the human mind). As it happens in the brain, such systems learn from examples: this concept has extensively been studied using models and methods of statistical physics [4, 5]. These models are accessible to investigations and may be helpful to understand some principles in biological systems.

The first contribute was given by Warren McCulloch and Walter Pitts who proposed in 1943 a network of two-state threshold elements to perform logic operations [6]. In 1949, Donald Hebb proposed a learning rule that describes how the neuronal activities influence the connection between neurons [7]. It provides an algorithm to update weight of neuronal connection within neural network. Different versions of the rule have been proposed to make the updating rule more realistic. The weight of connection between neurons is a function of the neuronal activity and the classical Hebb's rule indicates "neurons that fire together, wire together".

The first feedforward network, namely the *perceptron*, was proposed by Frank Rosenblatt in 1960 [8, 9]: the perceptron is a model based on a supervised learning algorithm of binary classifiers which makes its predictions based on a linear predictor function combining a set of weights with the feature vector.

In the 80's the contribute of John Hopfield and Elizabeth Gardner was fundamental to create a match between neural networks and statistical mechanics. J. Hopfield pointed out the equivalence between the long-time behavior of networks with symmetric connections and equilibrium properties of magnetic systems such as spin glasses and used it to create neural circuits for associative memory and other computational tasks [10, 11]. The units in Hopfield nets are in fact binary threshold units whose value is determined by whether or not the units'input exceed the fixed threshold. During the training, for different values of the weights, which handle interactions between neurons, we associate to the network a quantity which reminds to the energy of a statistical model. Under repeating updatings of the weights, the network will converge to a state which is a local minimum in the energy function. E. Gardner, instead, became the pioneer of applications to networks of methods from spin glass theory and opened the door to many investigations into learning systems showing how to analyze the space of all the networks in order to memorise a certain number of patterns, using spin-glass techniques and ideas [12, 13]. Spin glasses are magnetic systems with randomly distributed ferromagnetic and antiferromagnetic interactions and the low temperature phase, i.e. the spin glass phase, is a prototype for consensation in disordered systems with conflicting constraints. Theoretical studies show that in the case of long-range interactions between spins, the energy surface presents many local minima very close to energy to the ground state. Neural systems present many features in common with long-range spin-glass models: the coupling constants between spins can assume both positive and negative values, as it happens between neurons, and both spins and neurons are connected with long-range interactions.

Neural networks can be used without knowing precisely how the training process works but we can observe and measure its response to a set of inputs. The weights of a neural network with hidden layers are interdependent: the slight modification of a weight will impact the neuron it propagates directly and as a consequence all the neurons it is connected to affecting all the outputs. Hence the core problem of neural networks is essentially one of the statistical inference, since we aim to find the optimal weight configuration. The most naive approach to do that is setting an initial random configuration of weights, tuning them evaluating the accuracy of the provided results. One way to measure the amount of the error is the loss, or cost, function: there are several definitions for the loss function but all of them penalize the distance between the predicted value and the given value in the dataset.

Suppose to have a rule implemented by some machine according to which it gives a response as a consequence of a specific input information: the natural question which borns is how it is possible to infer the learning function on the basis of this input-output pairs. Physicists and mathematicians focusing on neural network models tried to answer how is possible to infer a learning function to a machine.

In this thesis we present and discuss how the statistical mechanics, through the network theory, can be useful to solve many different mathematical problems. In particular, it's not worth to observe that from a mathematical point of view, training a neural network model equals to solve an inverse problem for a spin model. In the first case, we aim to compute a set of weights and biases that best fit the model or, equivalently, that minimize the cost function which is associated to the model. At the same time, in the second case, we focus on the inference of the coupling constants and of the magnetic fields which minimize the Gibbs energy of a given set of spins.

This thesis is organized as follows.

In the second chapter we test the inverse problem method for a class of monomer-dimer statistical mechanics models that contain also an attractive potential and display a mean-field critical point at a boundary of a coexistence line. We obtain the inversion by analytically identifying the parameters in terms of the correlation functions and via the maximum-likelihood method. The precision is tested in the whole phase space and, when close to the coexistence line, the algorithm is used together with a clustering method to take care of the underlying possible ambiguity of the inversion.

In the third chapter we perform some analysis in order to characterize statistical properties of the observed mobility of drosophilas expressing different kinds of proteins. In order to do this, we identify such insects as interacting particles of a system and we define an interaction network with the aim to understand how they behave according to the different proteins expression. After having observed and analyzed how their walking speed changes, we study how the interaction dynamic undergoes variations. From our results, it emerges that wild type drosophilas are more induced in having a more active social exchanges with respect to the other ones, which seem to be focused only on their route.

In the fourth chapter we give an overview of the already existing algorithm *Replicated Belief Propagation (RBP)* deeply analyzing the equations which define the model. Belief propagation, also known as sum-product message-passing algorithm, is a message-passing algorithm which is used to perform inference on graphical models. It calculates the marginal distribution for each unobserved node conditional on any observed nodes. Many variants of such algorithm exist but we focus on that which operates on a factor graph, namely a bipartite graph containing nodes corresponding to variables and factors and where the edges connect variables and factors where they appear in. The algorithm works by passing *messages* along the edges. Each message contains the influence that one variable exerts on another. Messages are updated till convergence is achieved and are used to estimate each node marginal distribution which is proportional to the product of all messages from adiacent factors.

In the fifth chapter we apply the RBP in order to predict the congestion formation in the framework of complex systems physics. Traffic is a complex system where vehicle interactions and finite volume effects produce different collective regimes and phase transition phenomena. Such prediction can be a difficult problem due to the heterogenous behaviour of drivers when the vehicle density increases. We propose a novel pipeline to classify traffic slowdowns by analyzing the features extracted from the fundamental diagram of traffic. We train the RBP and we provide a forewarning time of prediction related to the training set size. In conclusion we compare our performances with those of the most common classifiers used in machine learning analysis.

Chapter 2

Inverse problem for the mean-field monomer-dimer model with attractive interaction

In this chapter we test the inverse problem method for a class of monomer-dimer statistical mechanics models that have also an attractive potential and display a mean-field critical point at a boundary of a coexistence line. The inversion is obtained by analytically identifying the parameters in terms of the correlation functions and via the maximum-likelihood method. The precision is tested in the whole phase space and, when close to the coexistence line, the algorithm is used together with a clustering method to take care of the underlying possible ambiguity of the inversion. The results which are presented in this chapter have been published in [14].

2.1 Introduction

In the last decade a growing corpus of scientific research has been built that focus on the attempt to infer parameters by reconstructing them from statistical observations of systems. The problem itself is known as statistical inference and traces back to the times when the mathematical-physics description of nature became fully operative thanks to the advances of mechanics and calculus, i.e. with the French mathematicians Laplace and Lagrange. In recent times this field and its most ambitious problems have deeply connected with statistical physics [15, 16, 17] at least in those cases in which the structure of the problem includes the assumption of an underlying model to describe the investigated phenomena. The aforementioned connection is surely related to the ability that statistical physics has acquired to describe phase transitions. In this paper we study the inverse problem for a model of interacting monomer-dimers in the mean-field, i.e. in the complete, graph. The denomination comes from the fact that the standard calculation in statistical mechanics, i.e. the derivation of the free energy and correlation from the assignment of the parameters is called the direct problem. Monomer-dimer models appeared in equilibrium statistical mechanics to describe the process of absorption of monoatomic or diatomic molecules in condensed matter lattices [18]. From the physical point of view monomers and dimers cannot occupy the same site of the lattice due to the hard-core interaction i.e. the strong contact repulsion generated by the Pauli exclusion principle. Beside such interaction though, as first noticed by Peierls [19], the attractive component of the Van der Waals potentials might influence the phase structure of the model and the thermodynamic behaviour of the material. In the mean field setting analysed here the monomer-dimer model displays the phenomenon of phase coexistence among the two types of particles [20, 21, 22]. This makes the inverse problem particularly challenging since in the presence of phase coexistence the non uniqueness of its solution requires a special attention in identifying the right set of configurations. Under mean-field theory, the monomer-dimer model can be solved for the monomer densities and the correlations between monomers and dimers: the mean-field solution is inverted to yield the parameters of the model (external field and imitation coefficient) as a function of the empirical observables. The inverse problem has also been known for a long time as Boltzmann machine learning [23]. Its renewed interest is linked to the large number of applications in many different scientific fields like biology [24, 25, 26, 27], computer science for the matching problem [28, 29, 30] and also social sciences [31, 32].

In this chapter we follow an approach to the inverse problem similar to the one introduced for the multi-species mean-field spin model in the work [33].

The chapter is organised in the following sections and results. In the second section we recall briefly the monomer-dimer model and we review the basic properties of its solution [20, 22]. In the third section we solve the inverse problem: using the monomer density and the susceptibility of the model, we compute the values of the two parameters, here called coupling constants, J and h. The first measures the preference for a vertex to be occupied by a monomer (respectively dimer), by imitating his neighbours. Firstly we identify the analytical inverse formulas providing an explicit expression of the free parameters in terms of the mentioned macroscopic thermodynamic variables. Then we use the maximum likelihood estimation procedure in order to provide an evaluation of the macroscopic variables starting from real data. The fourth section presents and discusses a set of numerical tests for finite number of particles and finite number of samples. The dependence of the monomer density and the susceptibility is studied with respect to the system size. We find that both of them have a monotonic behavior which depends on the parameters value and reach their limiting values with a correction that vanishes as at the inverse volume. We then investigate how the experimental monomer density and susceptibility at fixed volume depend on the number of samples. The effectiveness of the inversion is tested for different values of the imitation coefficients and external fields. After observing that the error of the inversion does not vanish when the parameters are close to the coexistence phase we investigate the effectiveness of clustering algorithms to overcome the difficulty. We find in all cases that the inverse method reconstructs, with a modest amount of samples, the values of the parameters with a precision of a few percentages. The chapter has two technical appendices: the first on the rigorous derivation of the exact inverse formulas, the second that supports the first and studies the non homogeneous Laplace method convergence to the second order.

2.2 Definition of the model

Let $G = (V, E)$ be a finite simple graph with vertex set V and edge set $E = \{uv \equiv \{u, v\} | u \neq v \in V\}$.

Definizione 2.1. A *dimer configuration* D on the graph G is a set of *dimers* (pairwise non-incident edges):

$$D \subseteq E \quad \text{and} \quad (uv \in D \Rightarrow uw \notin D \quad \forall w \neq v).$$

The associated set of *monomers* (dimer-free vertices), is denoted by

$$\mathcal{M}(D) := \mathcal{M}_G(D) := \{u \in V | uv \notin D, \forall v \in V\}.$$

Given a dimer configuration $D \in \mathcal{D}_G$, we set for all $v \in V$ and $e \in E$

$$\alpha_v(D) := \begin{cases} 1, & \text{if} \quad v \in \mathcal{M}(D) \\ 0, & \text{otherwise} \end{cases}$$

and

$$\alpha_e(D) := \begin{cases} 1, & \text{if} \quad e \in D \\ 0, & \text{otherwise.} \end{cases}$$

Definizione 2.2. Let $\mathcal{D}_G$ be the set of all possible dimer configurations on the graph G. The *imitative monomer-dimer model* on G is obtained by assigning an external field $h \in \mathbb{R}$ and an imitation coefficient $J \geq 0$ which gives an attractive interaction among particles occupying neighbouring sites. The *Hamiltonian* of the model is defined by the function $H_G^{\text{IMD}} : \mathcal{D}_G \to \mathbb{R}$ such that

$$H_G^{\text{IMD}} := -\sum_{v\in V} h\alpha_v - \sum_{uv\in E} J(\alpha_u\alpha_v + (1-\alpha_u)(1-\alpha_v)). \tag{2.1}$$

The choice of the Hamiltonian naturally induces a Gibbs probability measure on the space of configuration $\mathcal{D}_G$:

$$\mu_G^{\text{IMD}}(D) := \frac{\exp(-H_G^{\text{IMD}}(D))}{Z_G^{\text{IMD}}} \quad \forall D \in \mathcal{D}_G, \tag{2.2}$$

where the *partition function*

$$Z_G^{\text{IMD}} = \sum_{D\in\mathcal{D}_G} \exp(-H_G^{\text{IMD}}(D))$$

is the normalizing factor.
The natural logarithm of the partition function is called *pressure function* and it is related to the free energy of the model.

The normalized expected fraction of monomers on the graph is called *monomer density*. It can also be obtained computing the derivative of the pressure per particle with respect to h:

$$m_G^{\text{IMD}} := \sum_{D\in\mathcal{D}_G} \frac{|\mathcal{M}(D)|}{|V|}\mu_G^{\text{IMD}}(D) = \frac{\partial}{\partial h}\frac{\log Z_G^{\text{IMD}}}{|V|}.$$

It is easy to check that

$$2|D| + |\mathcal{M}(D)| = |V|. \tag{2.3}$$

In this paper we study the imitative monomer-dimer model on the complete graph, that is

$$G = K_N = (V_N, E_N)$$

with $V_N = \{1,\dots,N\}$ and $E_N = \{\{u,v\}|u,v\in V_N, u<v\}$.
In order to keep the pressure function of order N, it is necessary to normalize the imitation coefficient by $\frac{1}{N}$ because the number of edges grows like N^2

and to subtract the term $\log N \sum_{e\in E_N} \alpha_e$ to the external field. Thus we will consider the Hamiltonian $H_N^{\text{IMD}} : \mathcal{D}_N \to \mathbb{R}$,

$$H_N^{\text{IMD}} := -\sum_{v\in V_N} h\alpha_v + \log N \sum_{e\in E_N} \alpha_e - \sum_{uv\in E_N} \frac{J}{N}(\alpha_u\alpha_v + (1-\alpha_u)(1-\alpha_v)) \ . \quad (2.4)$$

All the thermodynamic quantities will therefore be functions of N and we are interested in studying the large volume limits.

Before studying the inverse problem, we briefly recall the main properties of the model (see [20, 22]).
Taking $m \in [0,1]$, the following variational principle holds

$$p^{\text{IMD}} = \sup_m \tilde{p}(m),$$

where p^{IMD} is the pressure of the model at the thermodynamic limit and

$$\tilde{p}(m(J,h),J,h) := -Jm^2 + \frac{1}{2}J + p^{\text{MD}}((2m-1)J+h) \quad \forall m \in \mathbb{R},$$

with $p^{\text{MD}}(\xi) := -\frac{1-g(\xi)}{2} - \frac{1}{2}\log(1-g(\xi)) = -\frac{1-g(\xi)}{2} - \log(g(\xi)) + \xi \quad \forall \xi \in \mathbb{R}$ and $g(\xi) := \frac{1}{2}(\sqrt{e^{4\xi}+4e^{2\xi}} - e^{2\xi}) \quad \forall \xi \in \mathbb{R}$. The solution of the model reduces to identify the value m^* that maximizes the function $\tilde{p}$ and it is found among the solutions of the consistency equation $m = g((2m-1)J+h)$ that include, beside the equilibrium value, also the unstable and metastable points. It is possible to prove that m^* (which represents the monomer density) is a smooth function for all the values of J and h with the exception of the coexistence curve $\Gamma(J,h)$. Such curve is differentiable in the half-plane (J,h) which originates at the critical point $(J_c,h_c) = (\frac{1}{4(3-2\sqrt{2})}, \frac{1}{2}\ln(2\sqrt{2}-2) - \frac{1}{4})$.

2.3 The inverse problem

The evaluation of the parameters of the model starting from real data is usually called inverse problem and amounts of two steps. The analytical part of the inverse problem is the computation of the values J and h starting from those of the first and second moment of the monomer (or dimer) density. The statistical part instead is the estimation of the values of the moments starting from the real data and using the maximum likelihood principle [34] or the equivalent formulations in statistical mechanics terms [35]. For what it concerns the analytical part, using the results of Appendix A and B, it can be proved that in the thermodynamic limit the imitation coefficient and the external field can be respectively computed as

$$J = -\frac{1}{2\chi} + \frac{2-m^*}{4m^*(1-m^*)}, \tag{2.5}$$

and

$$h = g^{-1}(m^*) - J(2m^*-1) = \frac{1}{2}\log\left(\frac{m^{*2}}{1-m^*}\right) - J(2m^*-1). \tag{2.6}$$

We denote by m_N and χ_N the finite size monomer density average and susceptibility $N(\langle m_N^2\rangle - \langle m_N\rangle^2)$, while their limiting values are denoted without the subscript N.

For the statistical part we use the maximum likelihood estimation procedure. Given a sample of M independent dimer configurations $D^{(1)},\dots,D^{(M)}$ all distributed according to the Gibbs measure (2.2), the maximum likelihood function is defined by

$$L(J,h) = \mu_N^{\text{IMD}}\{D^{(1)},\dots,D^{(M)}\} = \prod_{i=1}^{M} \frac{\exp(-H_N^{\text{IMD}}(D^{(i)}))}{\sum_{D\in\mathcal{D}_{K_N}}\exp(-H_N^{\text{IMD}}(D))}.$$

The function $L(J,h)$ reaches its maximum when the first and the second momentum of the monomer density are calculated from the data according to the following equations:

$$\begin{cases} m_N &= \frac{1}{M}\sum_{i=1}^{M} m_N(D^{(i)}), \\ m_N^2 &= \frac{1}{M}\sum_{i=1}^{M} m_N^2(D^{(i)}). \end{cases} \tag{2.7}$$

The inverse problem is therefore solved by the composition of (2.7) with (2.5) and (2.6). In particular, denoting by m_{exp} and χ_{exp} respectively the average monomer density and the susceptibility computed from the sample

$$m_{exp} = \frac{1}{M}\sum_{i=1}^{M} m_N(D^{(i)}) \quad \text{and} \quad \chi_{exp} = N\left(\frac{1}{M}\sum_{i=1}^{M} m_N^2(D^{(i)}) - m_{exp}^2\right), \tag{2.8}$$

the estimators of the model's free parameters are

$$J_{exp} = -\frac{1}{2\chi_{exp}} + \frac{2-m_{exp}}{4m_{exp}(1-m_{exp})} \tag{2.9}$$

and

$$h_{exp} = \frac{1}{2}\log\left(\frac{m_{exp}^2}{1-m_{exp}}\right) - J_{exp}(2m_{exp}-1). \tag{2.10}$$

2.4 The inversion at finite volume and finite sample size

The aim of this chapter is to study the robustness of the inversion procedure, i.e. the computation of the parameters from real data. The idea is to infer the value of J and h from the configurations generated according to the distribution of the model. In order to compute efficiently the values of the statistical estimators m_{exp} and χ_{exp} and in order to obtain a good approximation of the analytical inverse formulas in terms of finite size thermodynamic variables, we have to choose a large number of configurations of the sample and a large number of vertices of the graph, which are respectively identified by M and N. Since in real data we have a finite number of vertices and a finite number of configurations, the robustness will be studied with respect to both these two quantities.
The data that we are going to use are extracted from a virtually exact simulation of the equilibrium distribution. In fact, the mean-field nature of the model allows to rewrite the Hamiltonian (2.1) as a function of the dimer, or monomer, density (see equation (2.3)):

$$H_N^{\text{IMD}}(d_N) = -N\left(J\left(16d_N^2 - 4d_N + \frac{N-1}{2N}\right) + h\,(1-4d_N) - 2d_N \log N\right), \tag{2.11}$$

where $d_N = d_N(D) = \frac{|D|}{2N}$, or equivalently

$$H_N^{\text{IMD}}(m_N) = -N\left(J\left(m_N^2 - m_N + \frac{N-1}{2N}\right) + hm_N + \frac{1}{2}\log N(m_N - 1)\right). \tag{2.12}$$

In particular we use the following definition of the partition function:

$$Z_N^{\text{IMD}} = \sum_{|D|=0}^{[N/2]} c_N(D) e^{-H_N^{\text{IMD}}(d_N(D))}, \tag{2.13}$$

where the term $c_N(D) = \frac{N!}{|D|!(N-2|D|)!}2^{-|D|}$ is the number of the possible configurations with $|D|$ dimers on the complete graph with N vertices. Using the previous representation of the partition function we extract large samples of dimer densities values according to the equilibrium distribution. Those will be used for the statistical estimation of the first two moments (2.7). We are going to illustrate the results with some examples. Figure 2.1 shows the finite size average monomer density m_N and finite size susceptibility χ_N for the monomer-dimer model at different N's for different couples of parameters

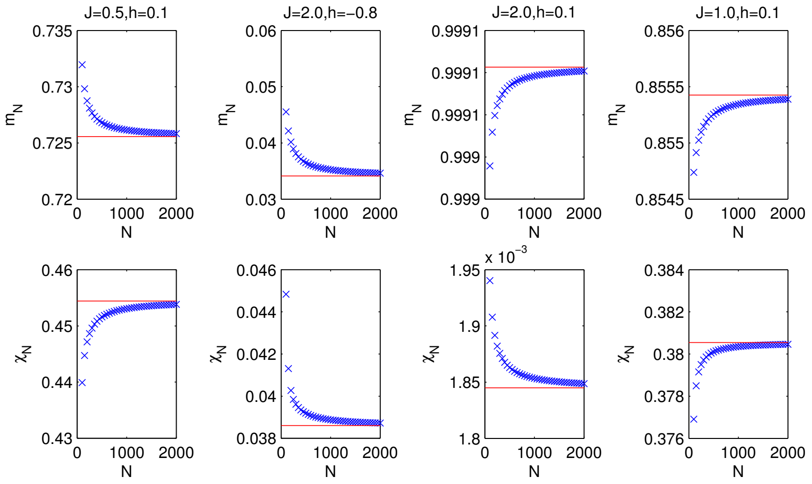


Figure 2.1: Finite size average monomer density m_N (upper panels) and susceptibility χ_N (lower panels) as a function of N for the monomer-dimer model at different values of J and h. The red continuous lines represent the values in the thermodynamic limit.

(J, h). The figure highlights the monotonic behavior of m_N and χ_N as function of N. We point out that the different monotonic behaviors of the finite size monomer density and susceptibility provide a useful information about the phase space region at which the system is found before applying the full inversion procedure. Figure 2.2 shows the power-law fits of the behavior of the finite size corrections both for monomer density and susceptibility. In order to test numerically our procedure, we consider 20 $M-$samples for each couple (J, h) and we solve the inverse problem for each one of them independently; then we average the inferred values over the 20 $M-$samples. We denote by $\overline{m}_{exp}$, $\overline{\chi}_{exp}$, $\overline{J}_{exp}$ and $\overline{h}_{exp}$ such averaged quantities. The two panels of figure 2.3 represent the statistical dependence of the estimators $\overline{m}_{exp}$ and $\overline{\chi}_{exp}$ on the number of the configurations of the sample. To check out that dependence on the sample $D^{(1)}, \dots, D^{(M)}$, we computed the values of the experimental estimators over a set of 20 independent instances of such samples. The errors are standard deviations on 20 different $M-$samples of the same simulation: we find numerical evidence that $M \geq 5000$ stabilizes the estimations.

To test numerically the inversion procedure, we take a sample of $M = 5000$ dimer configurations $\{D^{(i)}\}$, $i = 1, \dots, M$ over a complete graph with $N = 2000$ vertices. We consider $J \in [0.1, 1.5]$ and we fix $h = 0.1$; the obtained values for this case are shown in the left panel of figure 2.4, where $\overline{J}_{exp}$ and

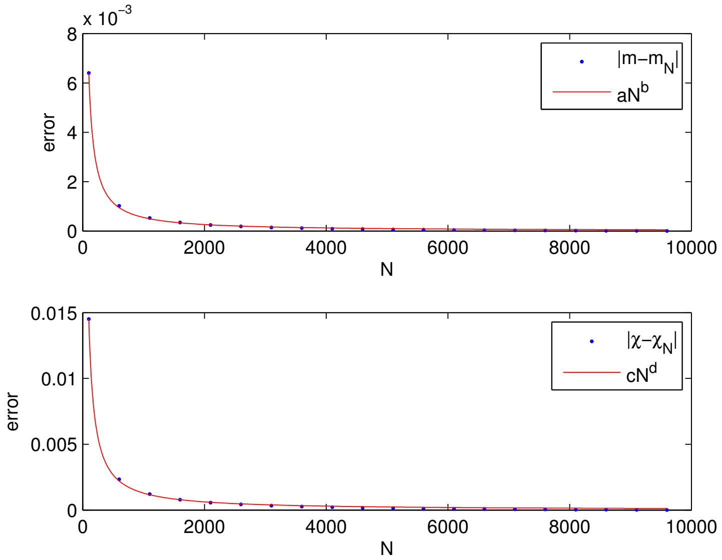


Figure 2.2: $J = 0.5, h = 0.1$. Upper panel: $|m_N - m|$ as a function of N together with the best fit aN^b for the data in the left upper panel of figure 2.1. We obtain $a = 0.306, a \in (0.1703, 0.4418)$ and $b = -0.8549, b \in (-0.9459, -0.7639)$ with a goodness of fit $R^2 = 0.9815$. Lower panel: $|\chi_N - \chi|$ as a function of N together with the best fit cN^d for the data in the left lower panel of figure 2.1. We obtain $c = 1.277, c \in (0.9883, 1.566)$ and $d = -0.9765, d \in (-1.024, -0.929)$ with a goodness of fit $R^2 = 0.9971$.

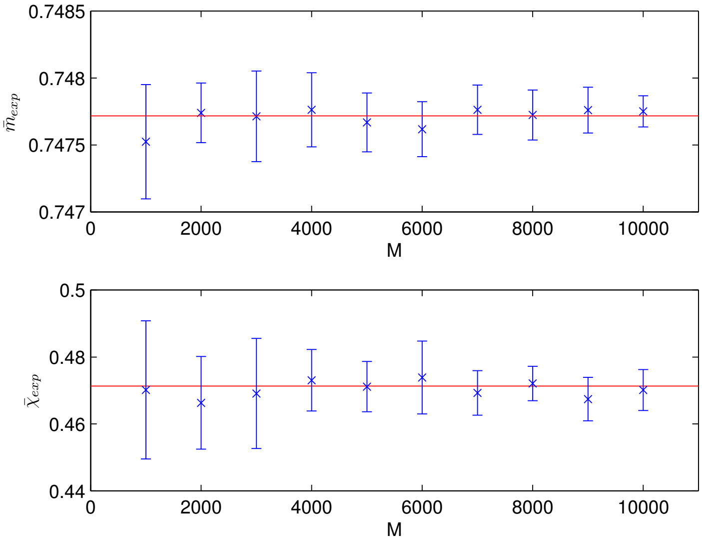


Figure 2.3: $N = 2000$, $J = 0.6$ and $h = 0.1$. Error bars are standard deviations on 20 different $M-$samples of the same simulation. Upper panel: average monomer density $\overline{m}_{exp}$ (blue crosses) as a function of M (number of the configurations in the sample). The red continuous line represents the finite size monomer density m_N. Lower panel: susceptibility $\overline{\chi}_{exp}$ (blue crosses) as a function of M (number of the configurations in the sample. The red continuous line represents the finite size susceptibility χ_N.

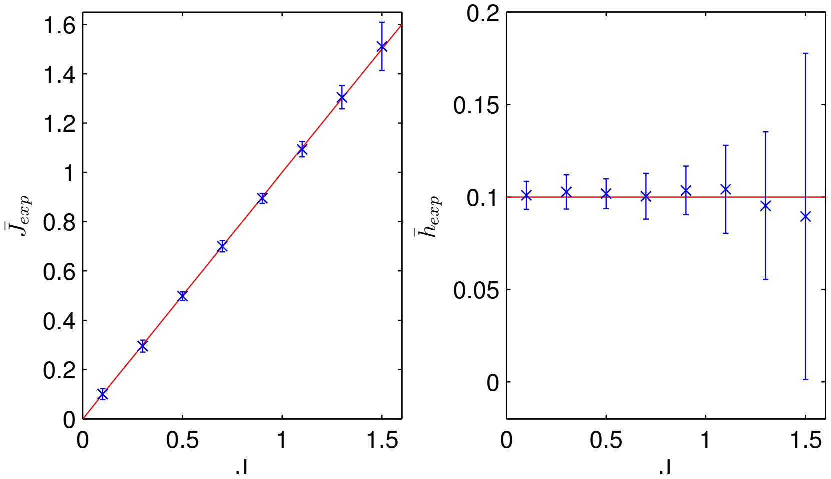


Figure 2.4: Error bars are standard deviations on 20 different $M-$samples of the same simulation. Left panel: $\overline{J}_{exp}$ as a function of $J \in [0.1, 1.5]$ (blue crosses). The red continuous line corresponds to the exact value of the imitation coefficient. Right panel: the value of $\overline{h}_{exp}$ (blue crosses) calculated from (2.10) for the values of J_{exp} in the left panel, as a function of $J \in [0.1, 1.5]$. The red continuous line corresponds to the exact value of h.

$\overline{h}_{exp}$ are plotted as functions of J. Note that the inferred values of the parameters are in optimal agreement with the exact values. Observe that for large values of J, the reconstruction get worse since the interaction between particles grows.
In figure 2.5 we report the absolute errors as a function of the imitation coefficient in reconstructing J and h in the cases of figure 2.4.

Figures 2.6 and 2.7 show relative errors in recostructing parameters for increasing sizes of the graph. They highlight that for large values of N and M, the inference of parameters doesn't give good results only in the case that the couple (J, h) is close to the coexistence line, but when we deal with real data, it may happen that we don't have a model defined over a graph with a large number of vertices or numerous configurations of the sample. In these cases, when J and h take values in the region of metastability, the inversion at finite volume and finite sample size can't be made using the method descripted above and we need another procedure to solve the problem, as it is shown in the following section.

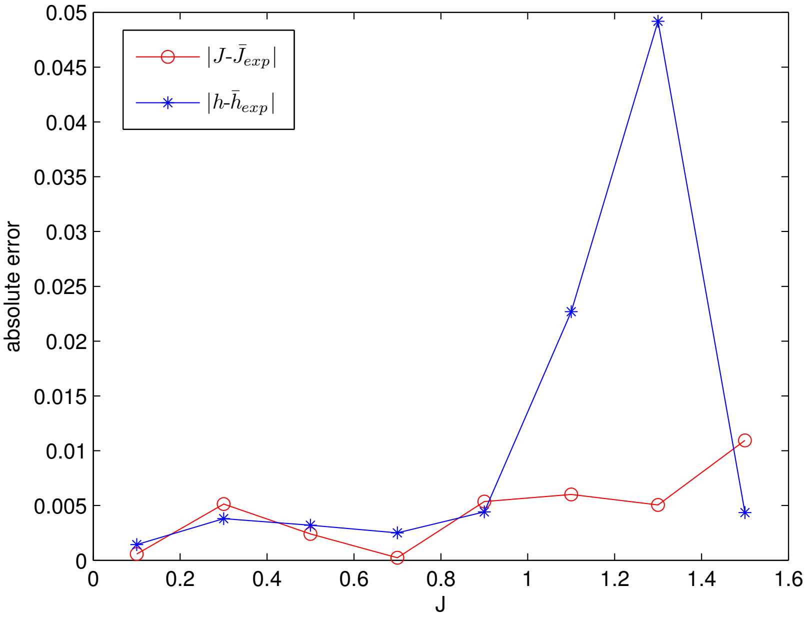
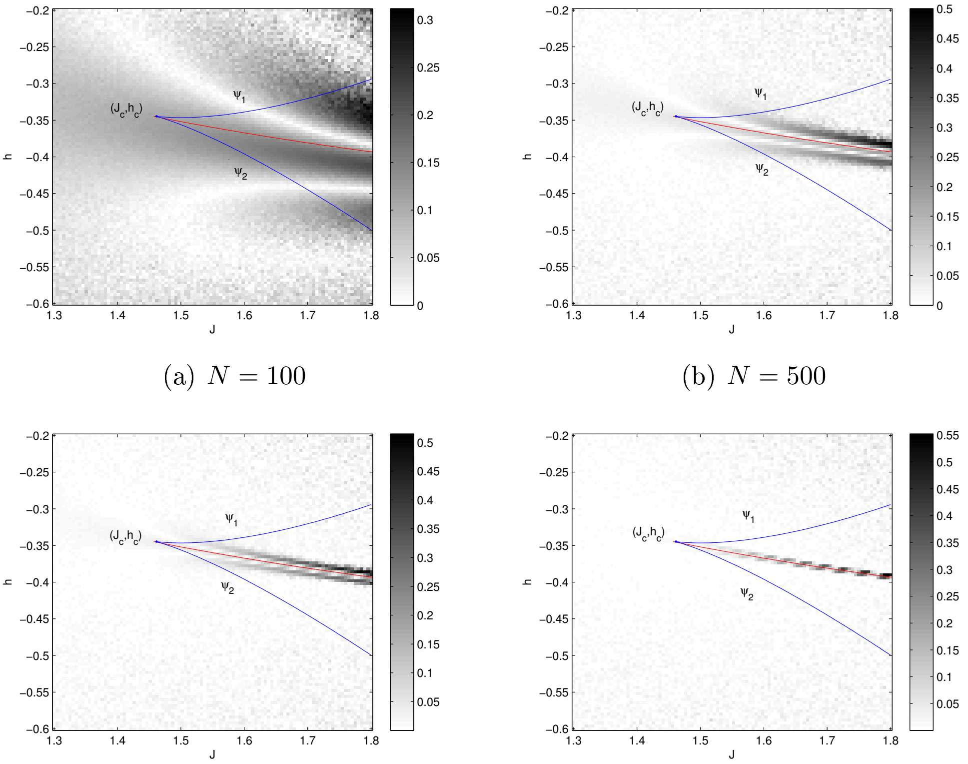


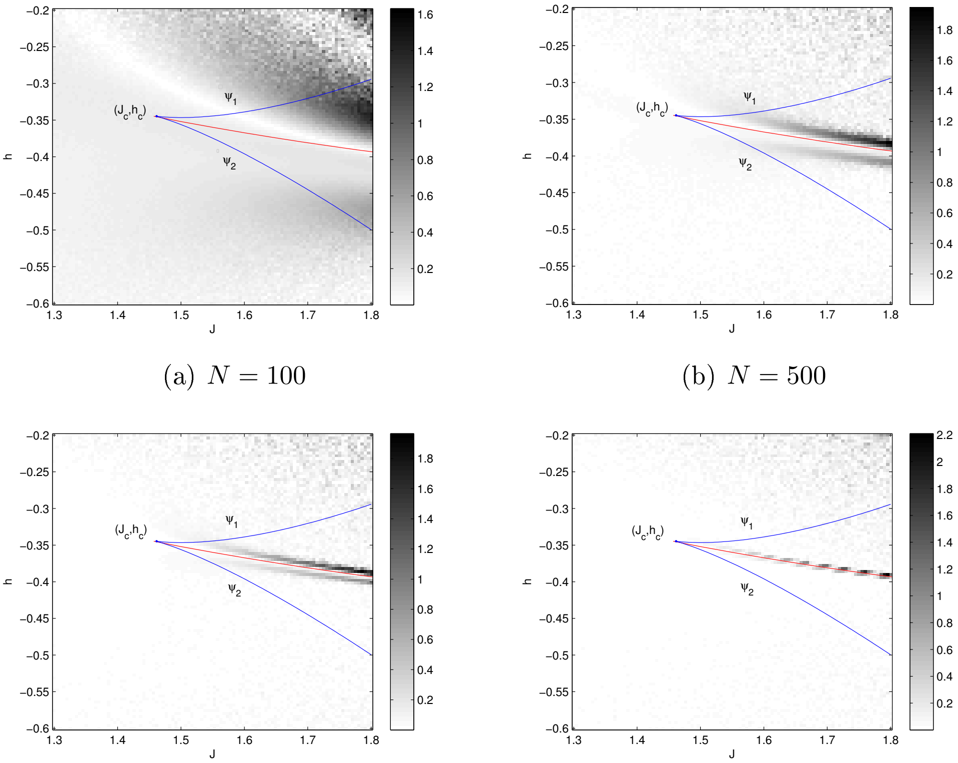
Figure 2.5: Inference of parameters of the monomer-dimer model on 20 different $M-$samples of the same simulation. Absolute errors in reconstructing J and h, where $J \in [0.1, 1.5]$ and $h = 0.1$.



(a) $N = 100$ (b) $N = 500$

(c) $N = 1000$ (d) $N = 3000$

Figure 2.6: Relative errors in reconstructing the imitation coefficient J. The points of the phase space are coloured with respect to the errors which assume the highest values along the coexistence line. The graybar on the right gives a range for the computed errors: the scale goes from white for the lowest to black for the highest. The blue curves ψ_1 and ψ_2 define the region of metastability (see [20, 21, 22]), the red curve is the coexistence line while (J_c, h_c) is the critical point; the blue and red colors do not identify any error. The number of configurations of the sample is set to be $M = 500$.



(a) $N = 100$

(b) $N = 500$

(c) $N = 1000$

(d) $N = 3000$

Figure 2.7: Relative errors in reconstructing the external field h. The points of the phase space are coloured with respect to the errors which assume the highest values along the coexistence line. The graybar on the right gives a range for the computed errors: the scale goes from white for the lowest to black for the highest. The blue curves ψ_1 and ψ_2 define the region of metastability (see [20, 21, 22]), the red curve is the coexistence line while (J_c, h_c) is the critical point; the blue and red colors do not identify any error. The number of configurations of the sample is set to be $M = 500$.

2.5 The inversion at finite volume and finite sample size with clustered phase space

We are now going to work on the monomer-dimer inverse problem when the phase space doesn't present only one equilibrium state, i.e. when the system undergoes a phase transition. We explain how to modify the mean-field approach we have seen above. If the model is defined for the parameters J and h such that the couple $(J, h) \in \Gamma$, the Gibbs probability density of the model presents two local maxima and we cannot study the inversion problem in a global way as we have done in the second section but we have to understand what happens in a local neighborhood of each maximum. Given M independent dimer configurations $D^{(1)}, \dots, D^{(M)}$ all distributed according to the Gibbs probability measure for this model, we can understand their behavior around m_1 and m_2 separating them in two sets, before applying formulas (2.9) and (2.10), i.e. we divide the configurations of the sample in clusters using the so called clustering algorithms which classify elements into classes with respect to their similarity (see [36, 37, 38, 39]). The clustering algorithms we use are based on the distance between the monomer density of the configurations: we put them in the same group if they are close enough and far from the other clusters (the concept of distance between clusters will be discussed later).

The method we use is the density clustering [36], which is based on the idea that the cluster centers are encircled by near configurations with a lower local density and that they are relatively far from any other configuration with a high local density. For each configuration we compute two quantities: its local density ρ_i and its distance δ_i from configurations with higher density. These quantities depend on the euclidean distance $d_{ij} = |m^{(i)} - m^{(j)}|$, where $m^{(i)}$, for $i = 1, \dots, M$ is the monomer density of the configuration $D^{(i)}$.

The local density ρ_i of $D^{(i)}$ is defined by

$$\rho_i = \sum_{j=1}^{M} \varphi(d_{ij} - d_c), \tag{2.14}$$

where d_c is an arbitrary cutoff distance (we will discuss later the choice of d_c) and

$$\varphi(x) = \begin{cases} 1 & \text{if} \quad x < 0 \\ 0 & \text{otherwise.} \end{cases}$$

In other words, the local density ρ_i corresponds to the number of configurations that are closer than d_c to the configuration $D^{(i)}$.

Remark 2.1. The choice of the cutoff distance d_c is crucial for the results of the algorithm: if we take a too large or a too small value for d_c it is possible that the algorithm is not able to find correctly the cluster centers. From the results of our simulations it emerges that, if we want to solve the inverse problem over a complete graph with $N = 3000$ vertices working with a sample made of $M = 10000$ dimer configurations, we have the best reconstruction of the free parameters when d_c is setted to be equal to 0.01. Obviously the choice depends on the range where the clusters centers have to be found and on the number of configurations of which the sample is made. More in general we have seen that for large values of M, the minimum absolute error in reconstructing parameters occurs when the cutoff distance is equal to $\frac{C}{M}$.

The distances δ_i are the minimum distance between the configuration $D^{(i)}$ and any other configuration with higher local density:

$$\delta_i = \min_{j:\rho_j>\rho_i} d_{ij}, \tag{2.15}$$

while for the configuration with the highest local density we take $\delta_{\bar{i}} = \max_j d_{ij}$. Observe that the quantity δ_i is much larger than the typical nearest neighbor distance only for the configurations that are local or global maxima in the density. Thus cluster centers are recognised as configurations for which the δ_i is anomalously large (this situation will be illustrated in example 2.1 in the following).
After the cluster centers have been found, each remaining configuration is assigned to its closest neighbor with higher density.

Remark 2.2. We tested our inversion formulas using two other clustering algorithms, obtaining analogous results, which put a number of data points into K clusters starting from K random values for the centers $x^{(1)}, \dots, x^{(K)}$: the K-means clustering algorithm and the soft K-means clustering algorithm [37]. However the results we are going to talk about have been obtained using the density clustering algorithm: by using this algorithm we do not have to specify the number of clusters since it finds them by itself.

Remark 2.3. From the results of our simulations, according to the example 2.1 in the following, it emerges that, if the couple of parameters which defines the model is not close enough to the coexistence line, we have a better reconstruction of the parameters applying equations (2.9) and (2.10) to the configurations which belong to the largest cluster.
On the other hand, when the couple (J, h) is near to the coexistence line $\Gamma(J, h)$, we solve the problem applying equations (2.9) and (2.10) to each cluster and averaging the inferred values as follows. We define the respective

observables of the two classes as

$$m_{exp}^{(k)} = \frac{1}{M_k} \sum_{i \in \mathcal{C}_k} m_i$$

and

$$\chi_{exp}^{(k)} = N \left(\frac{1}{M_k} \sum_{i \in \mathcal{C}_k} m_i^2 - (m_{exp}^{(k)})^2 \right),$$

where $k \in \{1, 2\}$, $\mathcal{C}_k$ is the set of indices of the configurations belonging to the k^{th} cluster and $M_k = |\mathcal{C}_k|$ is its cardinality.
We now apply (2.9) separately to each group in order to obtain two different estimators $J_{exp}^{(1)}$ and $J_{exp}^{(2)}$; finally we take the weighted average of all the different estimates

$$J_{exp} = \frac{1}{M_1 + M_2}(M_1 J_{exp}^{(1)} + M_2 J_{exp}^{(2)}) \tag{2.16}$$

in order to obtain the estimate for the imitation coefficient.
To estimate the parameter h, we first compute the values $h_{exp}^{(1)}$ and $h_{exp}^{(2)}$ within each cluster using equation (2.10) and the corresponding $J_{exp}^{(k)}$; the final estimate for h is given by the weighted average over the clusters

$$h_{exp} = \frac{1}{M_1 + M_2}(M_1 h_{exp}^{(1)} + M_2 h_{exp}^{(2)}). \tag{2.17}$$

We now focus on some cases of clustered phase space and we solve the inverse problem applying the density clustering algorithm.
In order to test numerically the inversion procedure for the monomer-dimer model, we consider a sample of $M = 10000$ dimer configurations $\{D^{(i)}\}$, $i = 1, \dots, M$ over a complete graph with $N = 3000$ vertices. We denote by the bar averaged quantities and the errors are standard deviations over $20-M$ samples.

Example 2.1. Consider a monomer-dimer model defined by the couple

$$(J, h) = (2.001, -0.4145);$$

the Gibbs probability distribution of the monomer densities for this choice of parameters is represented in figure 2.8. Given $M = 10000$ independent dimer configurations $D^{(1)}, \dots, D^{(M)}$ all distributed according to the Gibbs probability measure for this model, we use the density clustering algorithm in order to divide them in two sets to reconstruct the parameters.
As we can see by figures 2.8 and 2.9, configurations are divided in two

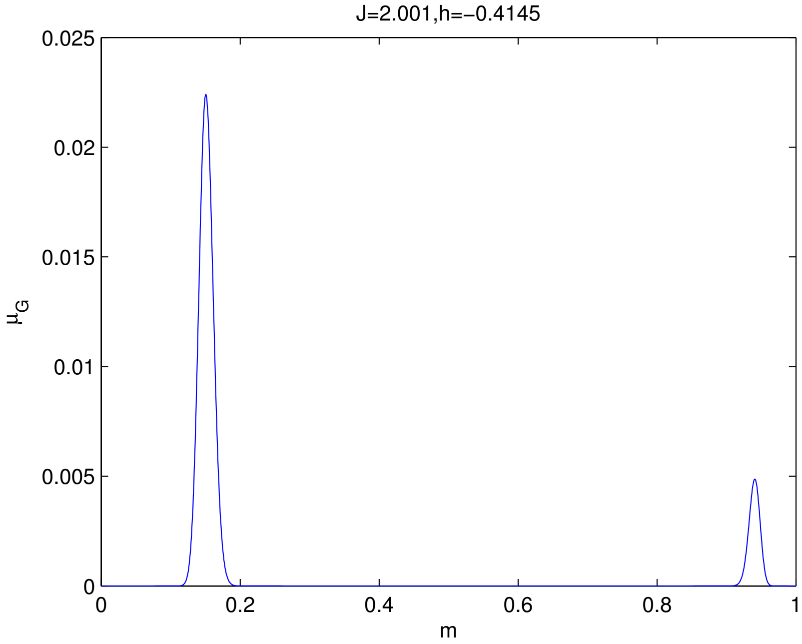


Figure 2.8: Gibbs probability distribution of the monomer densities for the dimer configurations of the monomer-dimer model defined by the couple of parameters $(J, h) = (2.001, -0.4145)$.

clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively centered in $\overline{m}_1 = 0.1507 \pm 5.7 \cdot 10^{-17}$ and $\overline{m}_2 = 0.9402 \pm 9.9 \cdot 10^{-4}$; moreover the cluster centered in m_1 contains more configurations than that centered in m_2. Let start observing that the reconstructed parameters are better solving the problem only respect to the largest cluster.
Applying equations (2.9) and (2.10) both to the configurations in $\mathcal{C}_1$ and $\mathcal{C}_2$ according to remark 2.3, by formulas (2.16) and (2.17) we obtain the following reconstructed values of parameters:

$$\overline{J}_{exp} = 2.0141 \pm 0.0802 \quad \text{and} \quad \overline{h}_{exp} = -0.4196 \pm 0.0828. \tag{2.18}$$

Applying instead equations (2.9) and (2.10) only to the configurations in the largest cluster $\mathcal{C}_1$, we obtain the following reconstructed values of parameters:

$$\overline{J}_{exp} = 2.0036 \pm 0.0353 \quad \text{and} \quad \overline{h}_{exp} = -0.4091 \pm 0.0247. \tag{2.19}$$

In order to justify our choice for the cutoff distance, we focus on figure 2.10, which shows the euclidean distances between $\overline{J}_{exp}$ and the true parameter J (blue stars) and between $\overline{h}_{exp}$ and the true parameter h (red circles)

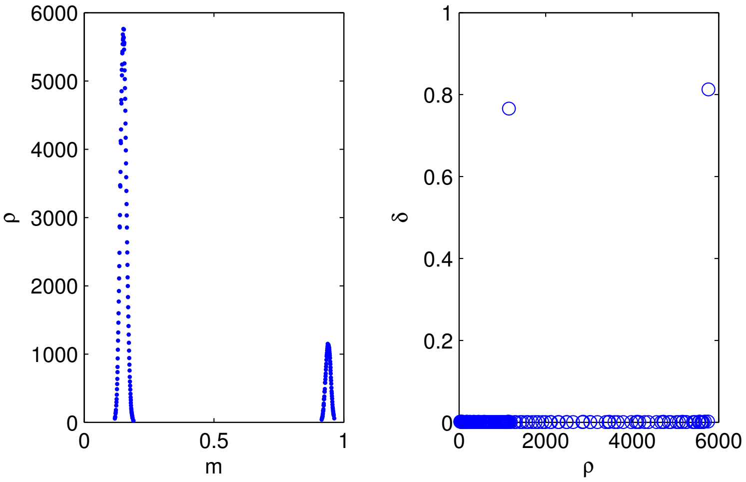


Figure 2.9: Density clustering algorithm. Left panel: plot of the vector ρ, whose components are computed according to (2.14), of the density of configurations around each configurations of the considered sample as a function of the monomer densities. Right panel: decision graph, plot of the vector δ, whose components are computed according to (2.15), as a function of the vector ρ.

for each choice of d_c, that takes value 10^{-j}, for $j = 1, \dots, 6$. We can see that, taking a sample of $M = 10000$ dimer configurations over a complete graph with $N = 3000$ vertices, we obtain the minimum absolute error considering $d_c = 0.01$. According to what we have told above, the choice is arbitrary and it depends on the range of values of the monomer densities and on the number of configurations in the sample: obviously, working with a larger set of dimer configurations we have more freedom in the choice of the cutoff distance. In conclusion we have seen that in the case where the couple of parameters (J, h) belongs to the region of metastability and is far enough from the coexistence line, at finite volume and at finite sample size, there are two clusters and one of them is much larger than the other one. According to remark 2.3, the obtained results confirm that the reconstruction of the parameters is better if we apply formulas (2.9) and (2.10) only to the largest set of configurations. The goodness of results is estimated comparing (2.18) and (2.19): the distance between the reconstructed parameters $\overline{J}_{exp}$ and the true value J is smaller in the first case, while the respective recontructions of h are equivalent.

We proceede considering ten different couples of parameters which are nearby the coexistence line $\Gamma(J, h)$ descripted above. In order to define them

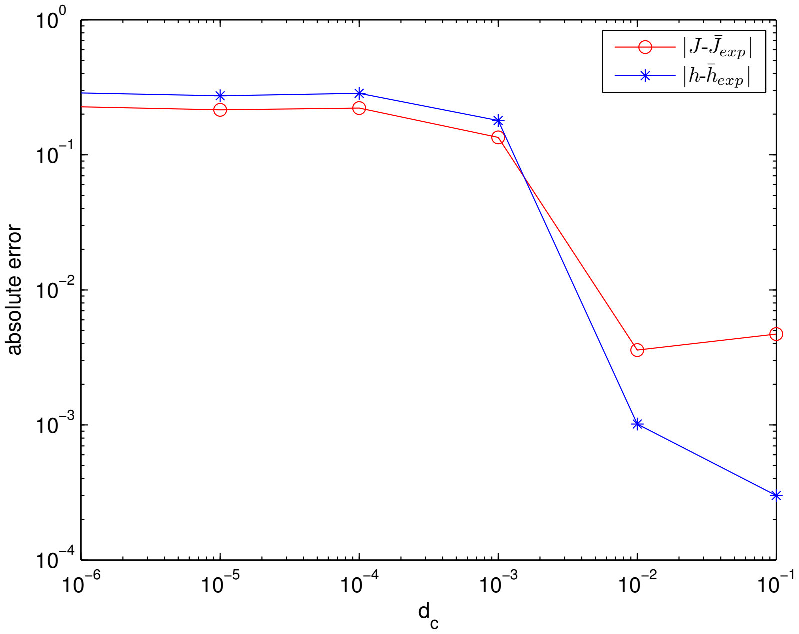


Figure 2.10: Density clustering algorithm: choice of the cutoff distance. Absolute errors in reconstructing J and h. Distance between the reconstructed $\overline{J}_{exp}$ and the true value J (blue stars) and distance between $\overline{h}_{exp}$ and h (red circle) for each choice of the cutoff distance d_c, which takes value 10^{-j}, for $j = 1, \ldots, 6$. The values of J_{exp} and h_{exp} are averaged across 20 $M-$samples. The errors are plotted as a function of d_c.

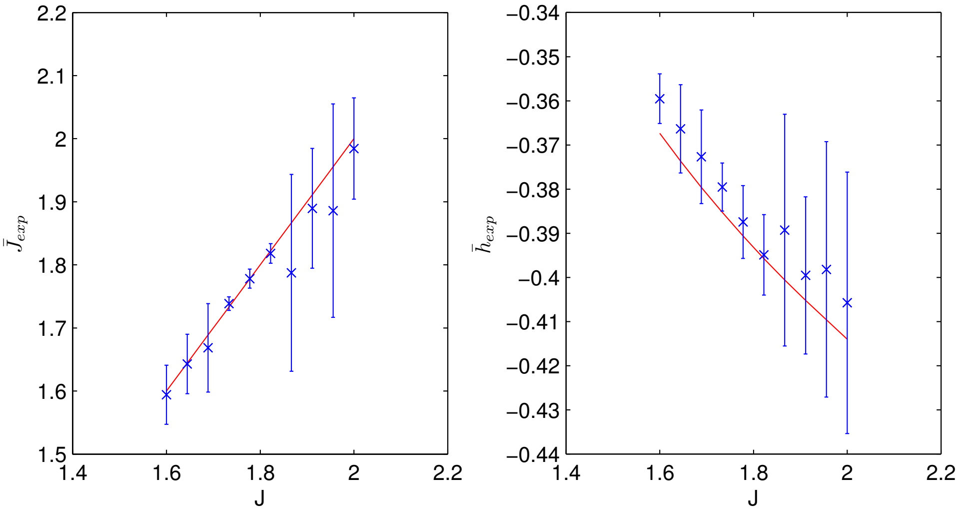


Figure 2.11: $N = 3000$, $J \in [1.6, 2]$, h takes values over the coexistence line. Error bars are standard deviations on 20 different $M-$samples of the same simulation. Parameters are reconstructed using the density clustering algorithm. Left panel: $\overline{J}_{exp}$ (blue crosses) calculated from (2.16) as a function of J. The red continous line represents the true value of J. Right panel: the value of $\overline{h}_{exp}$ (blue crosses) calculated from (2.17) for the values of $\overline{J}_{exp}$ in the left panel, as a function of J. The red continuous line corresponds to the exact value of h.

we take ten equispaced values for the imitation coefficient J in the interval $[1.6, 2]$ and we compute the corresponding values for the parameter J using equations (2.16) and (2.17). The obtained values are shown in figure 2.11, where $\overline{J}_{exp}$ and $\overline{h}_{exp}$ are plotted as a function of J. In figure 2.12 we can see the results in reconstructing parameters crossing the coexistence line $\Gamma(J, h)$. Fixed $J = 1.8$ we take increasing values of the parameter h in the interval $[-0.3940, -0.3924]$. In figure 2.13 we can see how the distribution of Gibbs of the monomer densities changes for different values of h.

In figure 2.14 the euclidean distances between $\overline{J}_{exp}$ and the value $J = 1.8$ (blue stars) and between $\overline{h}_{exp}$ and $h \in [-0.3940, -0.3924]$ (red circles) are shown for each of the nine couples (J, h).

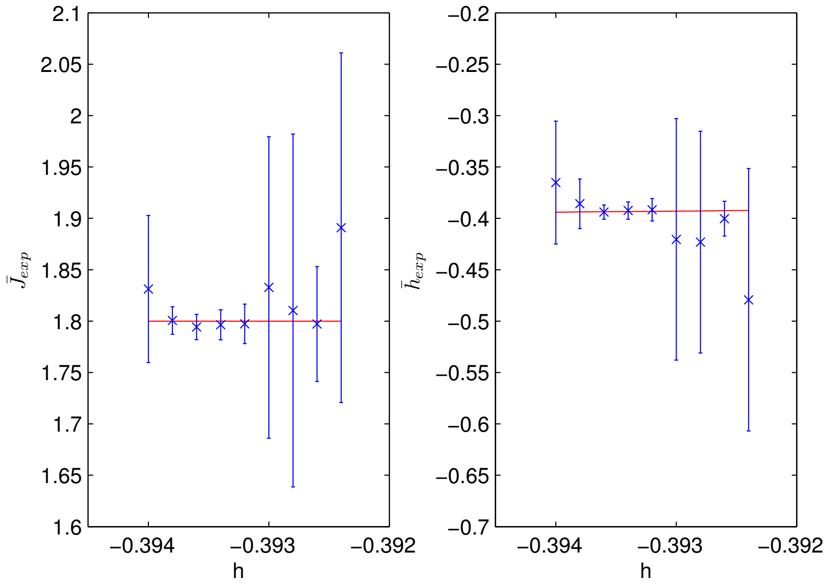


Figure 2.12: $N = 3000$, $J = 1.8$, $h \in [-0.3940, -0.3924]$. Error bars are standard deviations on 20 different $M-$samples of the same simulation. Parameters are reconstructed using the density clustering algorithm. Left panel: the value of $\overline{J}_{exp}$ (blue crosses) calculated from (2.16) as a function of h together with the statistical error. The red continous line represents the true value of J. Right panel: the value of $\overline{h}_{exp}$ (blue crosses) calculated from (2.17) for the values of $\overline{J}_{exp}$ in the left panel, as a function of h together with the statistical error. The red continuous line corresponds to the exact value of h.

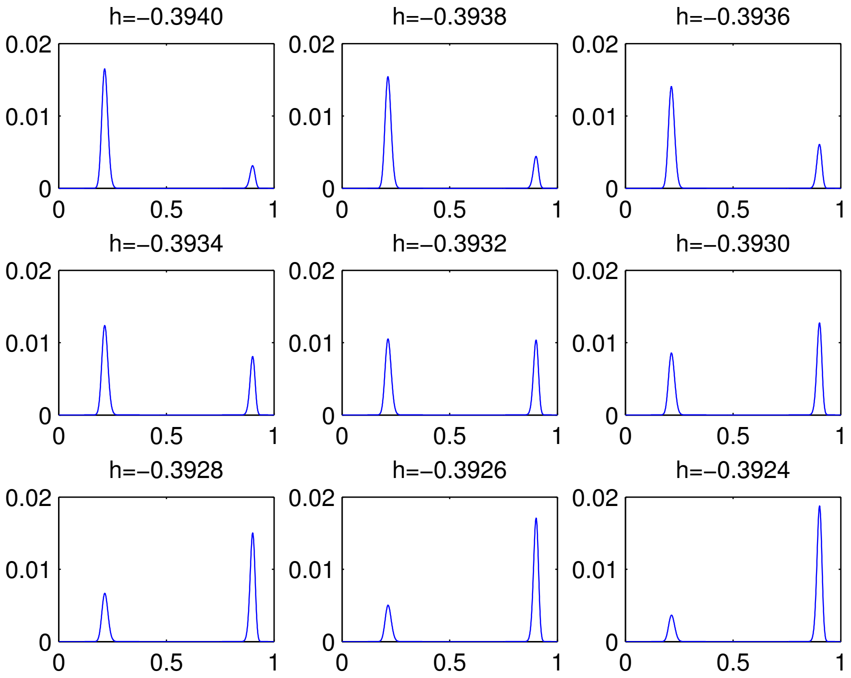


Figure 2.13: $N = 3000$, $J = 1.8$, $h \in [-0.3940, -0.3924]$. Gibbs probability distribution of the monomer densities for the dimer configurations of the monomer-dimer model defined by each couple of parameters (J, h) defined in figure 2.12.

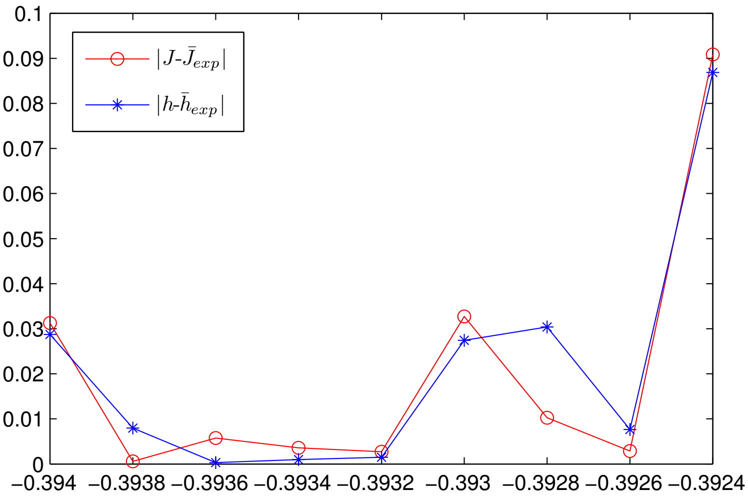


Figure 2.14: $N = 3000$, $J = 1.8$, $h \in [-0.3940, -0.3924]$. Absolute errors in reconstructing J and h using the density clustering algorithm. Distance between the reconstructed $\overline{J}_{exp}$ and the true value J (blue stars) and distance between $\overline{h}_{exp}$ and h (red circle) for each couple of parameters. The values of $\overline{J}_{exp}$ and $\overline{h}_{exp}$ are averaged across 20 $M-$sample. The errors are plotted as a function of h.

Chapter 3

Statistical properties of group mobility in an invertebrate model: Drosophila melanogaster

In this chapter we apply the network theory to a biological environment. The results have been reached in collaboration with the Department of Biomedical Sciences and the Department of Biology of Padova.

3.1 Introduction

Behavior can be described as internally coordinated responses (actions or inactions) of whole organisms to internal and/or external stimuli [40] which requires a complex and high level neural processing activity. From the operational point of view, each nervous system, whether simple and made by few neurons like in invertebrates (bees, flies, shrimps) or complex with many neurons like in vertbrates, mammals and humans, adaptive behavior is regulated by similar steps and basic mechanisms. Selective attention picks specific sensory informations which couple perception to action and process them at perceptual level by selecting which object will be the target of the ongoing action and which action utilize to reach the goal [41].

It is therefore interesting from the mathematical point of view analyse and define these similar mechanisms in behavioral responses, and evaluate how they evolved in organisms (and brains) far away from the evolutive point of view.

An extremely interesting adaptive behavior is navigation or mobility between other individuals in the same environment. In this case, among other informations to be processed at perceptual level, social interactions, i.e. the exchange of informations between individuals in a group, play a crucial role.

Mathematical models of the operational mechanisms which underlie navigation and mobility, are therefore interesting in helping to understand how we move among individuals and individual relative importance in regulating our navigation. Models of human mobility were developed, relying on statistical approaches [42] or on cognitive anlyses [43]. These studies approached the problem from two distinct point ov views: i) a mathematical approach with the aim to gain further knowledge for solving human mobility in social economical contexts and, ii) a psychophysiological approach with the aim to know the higher neurophysiological basis of human navigation. Our aim is to try to unify these to approaches in a unique approach with the idea to find a possible mathematical model describing basic and evolutively maintained mechanisms of animal navigation and mobility, as a background for future experiments. In this paper we investigated from a mathematical point of view, walking locomotion of limited groups of flies (Drosophila melanogaster) freely moving together in a uniform and homogeneously illuminated arena, without any external visual target. Utilizing the extremely sophisticated genetic tools which can be used on this animal model, we carried out investigations on a group of normal control wild type individuals and on a group of mutated individuals where brain level of dopamine, a neurotransmitter responsible of reward decision-making both in invertebrates and in vertebrates, was genetically increased.

By applying a specific definition of interaction, from our results, it emerged a clear separation between the two groups which can be classified simply using a bayesian classifier. Comparing our results with a null model composed by particles, we found that individuals move according to a particular dynamic which is not comparable to a random walk. Our considerations were finally applied to a mixed arena and we find out that the behavior and the nature of such insects can be classified according to our criteria.

The chapter is organized in the following sections and results.

In the second section we present the structure of the dataset and the recording methods which have been used. In the third section we analyze trajectories of drosophilas and their flying speed. From our results, it emerges that there is a clear separation between the two groups which can be classified simply using a bayesian classifier. In the fourth section we study the interaction graph that is originated from drosophilas according to our rule of interaction. We also compare results with a null model which is composed by particles which move according to a random walk inside of an arena of the same dimensions of the arena of our experiments. In order to reproduce an analogous experiment we lead particles to move with the same speed of drosophilas speed. It emerges that insects move according to a particular dynamic which has no links with a random walk. In the fifth section we apply our considerations to an heterogeneous arena and we find out that the behavior and the nature of such insects can be classified according to our criteria.

3.2 Data collection

Animals

The experiments were performed on adult wild-type fruit flies (Drosophila melanogaster; Oregon-R strain) and pale mutant flies in which an alteration of Tyrosine Hydroxilase function leads to reduced Dopamine synthesis [44, 45]. All flies were reared on standard cornmeal-sucrose-yeast medium at 22°C in a 12 h light/12 h dark cycle at 60% relative humidity. Fly crowding was also controlled (20-30 flies each vial) to avoid competition for food. Only individual 5-7 day-old male flies were used. Flies were kept in their food vials until the beginning of the experiment. Thus for the experiment flies were not starved nor were their wings clipped. All experiments were conducted between zeitgeber time 2 and 4 at room temperature 22°-23°C.

Experimental setup

Walking locomotor behavior of individual flies was tested in a circular plexiglass arena. The arena (maximum height at the centre = 3.5 mm; diameter = 110 mm) was designed so as to i) confine flies in 2D space, ii) not allow the flies to reach the edge of the arena and iii) to impede flight by means of a glass *ceiling*[46]. The arena was placed on a plexiglas platform inside a closed box and illuminated with a cold white leds from below. A transparent opaque paper around the arena made illumination homogenous. No visual targets were present inside the arena or accessible to fly vision. Flies walking locomotion inside the arena was recorded from above using a CCD camera (Chameleon 3, FLIR System Inc, Wilsonville, OR, USA) with 1288 × 964 pixel resolution, fitted with a 2.8-8 mm varifocal lens (Fujifilm, Tokyo, JP) mounted 36 cm above the arena. Videos of flies moving in the arena were recorded at 21 frames s-1, following selection of a 700 × 700 pixel region of interest which included the entire arena.

Procedure

Ten individual male flies, selected as above described, were placed inside the arena and left to adpt in complete darkness for 5 minutes. At the end of the "adaptation period", lights were turned on and continuous video recording of flies walking activity for a period of 10 minutes was acquired on PC. The same procedure was utilized to record walking activity of a group of 10 adult male Oregon R (wild type) flies and to record walking activity of a group of 10 adult male mutant flies.

Data analysis

Uncompressed AVI files of recorded videos were offline analysed using the CTRAX open source software [47]. By means of this software we obtained, for each frame, a precise 2D position and body orientation for each fly. Errors occurring during tracking were fixed manually using appropriate available MATLAB scripts (CTRAX, FixErrors Toolbox) [47]. Finally, other available MATLAB scripts (CTRAX, Behavioral Microarray Toolbox) were used to compute a suite of speed and acceleration properties [47].

3.3 Walking locomotor analysis

We started out by asking if walking locomotor properties can identify a phenotype in mutant flies.

We started simply analyzing trajectories which have been defined by flies inside of the arena. From these quantitative analyses it emerges that locomotor behaviors are totally different: the wild group fly without following a precise scheme, while the mutated group seems to prefer to move along the borders of the arena and have a more regular flying way. Figure 3.1 represents the trajectories which have been defined respectively by the wild group and the mutated one, both from a total point of view and from a singular point of view.

These first images leaded us to consider the fly locomotor speed: in any experiment we found a statistically significant relationship between the respective genetic mutations and the vectorial moving speed. It emerges that the wild type insects generally fly with a lower speed than the mutated flies. In figure 3.2, the upper panels present the histograms of the mean speed in the arena for each timestamp recording. The wild type data are distributed according to a skew distribution to the right: they are not symmetrically distributed around the mean since the larger values tend to be farther away from

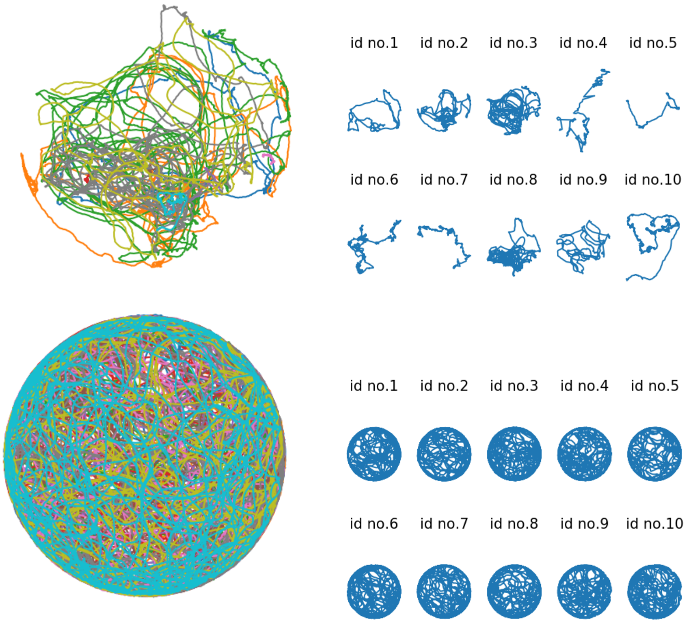


Figure 3.1: Flies tracking. Upper panels: we respectively represent the trajectories which have been defined by the OREGON group (left figure) and the trajectory which have been singularly defined by each fly (right figure). Lower panels: we respectively represent the trajectories which have been defined by the PALE group (left figure) and the trajectory which have been singularly defined by each fly (right figure).

the mean than the smaller ones. The mutant type data instead are more symmetrically distributed around the mean which assumes a value clearly higher to the first one. Precisely the wild type speed distribution is best fitted by an exponential distribution $Exp(\lambda)$ with $\lambda = 1.7808$ while PALE type speed distribution is best fitted by a Gaussian distribution $N(\mu, \sigma)$ with $\mu = 47.5375$ and $\sigma = 4.7838$. For that reason it is easy to define a threshold to separate the two groups which can be easily classified using a bayesian classifier. In figure 3.3 we measure the average speed of each fly in the respective arena finding again a clear separation.

3.4 Interaction graphs

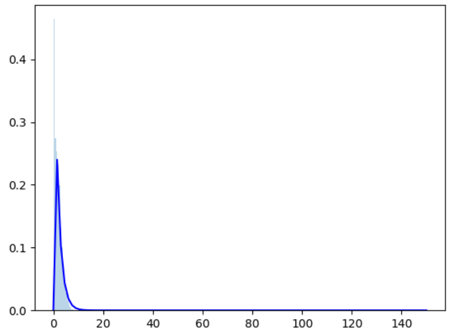
As second discriminant feature, we focus on the social behavior of drosophilas: our main goal is to separate the Pale class from the wild type by considering the interaction between insects in the same experiment. From our observations the wild group is the most interactive while flies which express proteins or have undergone some mutations almost don't get in contact one with each other. We say that two drosophilas have an interaction if they remain in the same square of 12 pixels side at least for 3 frame instants.

We show in figure 3.4 the interaction network of the two different experiments: we highlight with the line width different interaction weights which have been computed acording to the number of instants of contact. It emerges that wild type drosophilas present a higher attitude to get and remain in contact with other ones for a longer time. In order to characterize vertices and edges we study the indicators of centrality of the graph: we focused on the degree, which is defined as the number of links incident upon a node. In order to give some easy definition and to set the notation, consider a graph $G = (V, E)$, with V set of vertices and E set of edges. The degree centrality of a vertex v is defined as

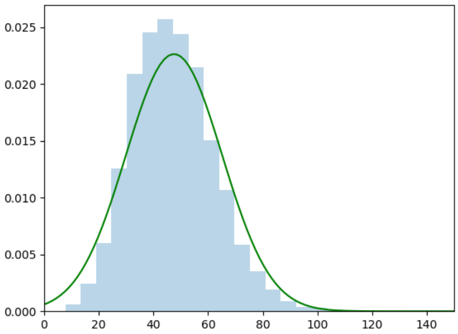
$$C_D(v) = deg(v).$$

Figure 3.5 shows the plots of each node's degree.

After having studied such graphs, we wanted to understand if insects move around the arena according to a particular dynamic or randomly. For this reason we defined a system of particles, and let them evolve in a square of side 700 pixels, according to the arena dimensions, with a speed which is near dorsophilas speed but in accordance to a random walk. So we reproduced two different arenas as follow. The arena which reproduces OREGON type behavior presents particles which move with a speed $v_O \sim Exp(\lambda)$ and that which reproduces PALE type behavior presents particles which move



(a) Oregon



(b) Pale

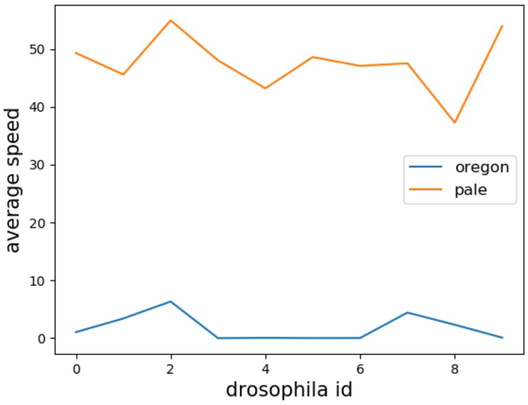


Figure 3.2: Flying speed. Upper panels. The histogram represents the average speed of each frame in different homogeneous experiments. Upper left panel: Oregon speed histogram. The blue line represents the best fitting distribution Exp(λ) with $\lambda = 1.7808$. Upper right panel: Pale speed histogram. The blue line represents the best fitting distribution $N(\mu, \sigma)$ with $\mu = 47.5375$ and $\sigma = 4.7838$. Lower panel: Flying speed graph. Different colors highlight different experiments. The upper red line represents the average flying speed of the Pale drosophilas while the blue line represents the average flying speed of the wild type drosophilas.

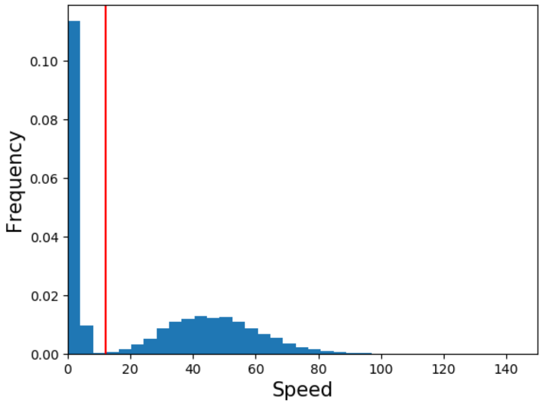


Figure 3.3: Cumulative speed histogram. The red line is a bayesian threshold which separates the wild type from the other one.

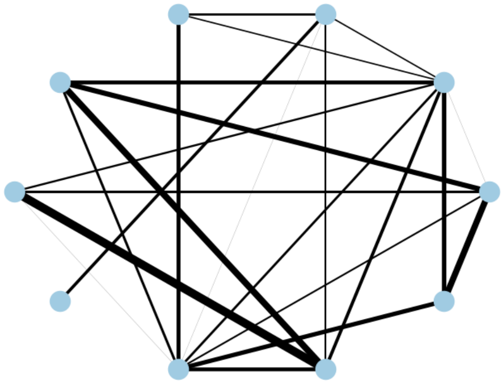
with a speed $v_P \sim N(\mu, \sigma)$ as we mentioned above. We build in this way a dataset which contains the spatial coordinates of the positions assumed in each timeframe by the particles and we process such data in the same way as we did with flies experiments. We report in figures 3.6 the different interaction graphs defined by such random particles: they lead us to think that interactions are due to specific and characteristic dynamics.

3.5 Blind dataset

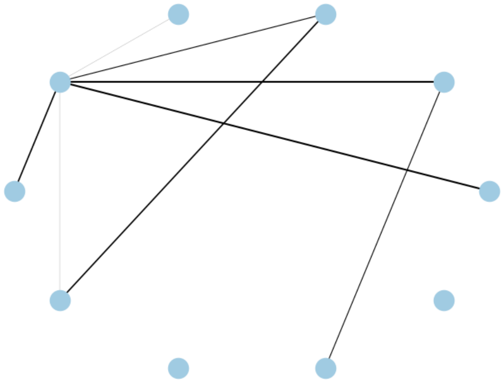
In order to confirm the integrity of the studied discriminant features, we considered an heterogeneous experiment and we tried to classify the nature of drosophilas applying criteria descripted above. In this experiment the arena is composed by ten drosophilas: seven of them belong to the OREGON type and the other three ones belong to the PALE type. We started analyzing each insect average flying speed (figure 3.7). Each point represents an insect and we plot the mean of their flying speed. Simply looking at each respective mean we find that two groups can be easily identified: the first three drosophilas fly with an evident higher velocity than the other seven ones. It leaded us to start thinking about the composition of the arena.

As a second step we built the interaction graph which is generated by this heterogeneous group and we represent it in figure 3.8. There is a big connected component composed by seven nodes while the threeother nodes of the graph don't interact.

Considerations we made above support and confirm the nature of the



(a) Oregon



(b) Pale

Figure 3.4: Interaction networks. The graphs represent the interactions between drosophilas in different homogeneous experiments. Upper panels: OREGON type interaction network. Lower panel: PALE type interaction network.

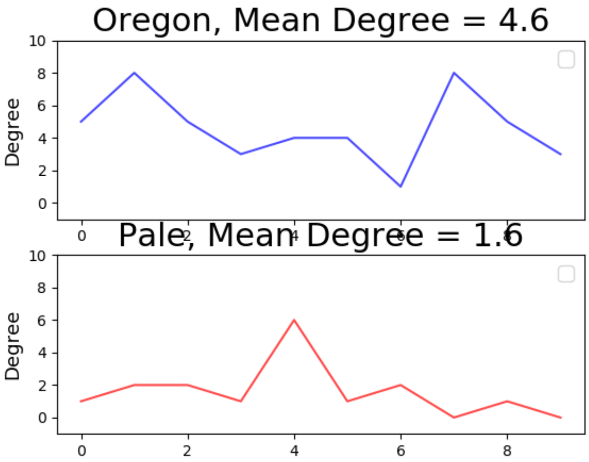
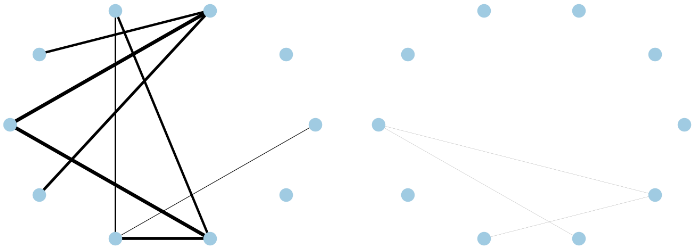


Figure 3.5: Interaction networks degree. The graphs represent the interaction graph degree centrality. Upper panel: first wild type group, mean degree is equal to 5.4. Middle panel: second wild type group, mean degree is equal to 4.6. Lower panel: Pale group, mean degree is equal to 1.6.

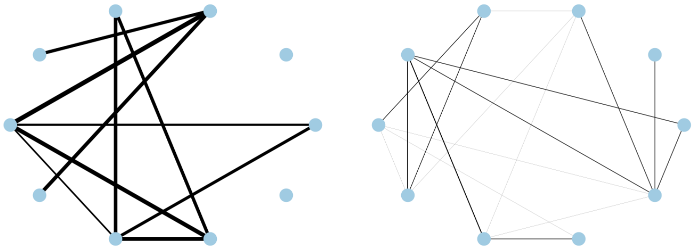
dataset and we are effectively able to classify insects according to their motor and social behavior.

3.6 Conclusions

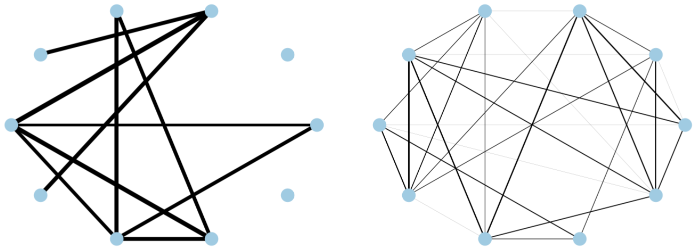
Assigning behavioral functions to neural structures has long been a central goal in neuroscience and is a necessary first step to understand how the brain generates behavior. Here, we study locomotion and social behaviors for Drosophila melanogaster using the network theory. We combine genetic mutations with our behavioral analysis to create correlation maps from which we generate hypotheses on how the behavior, and consequently the brain, reacts to such mutations. In conclusion, our maps provide a preliminary tool to distinguish PALE drosophilas from the wild type.



(a) side = 12 pixels (b) side = 12 pixels



(c) side = 20 pixels (d) side= 20 pixels



(e) side = 25 pixels (f) side = 25 pixels

Figure 3.6: Interaction random networks. Left panels: Simulation which represent the Oregon type drosophilas. Right panels: Simulation which represent the Pale type drosophilas. Each row presents the interaction graphs which have been defined according to different pixel dimensions for the side of the interaction square.

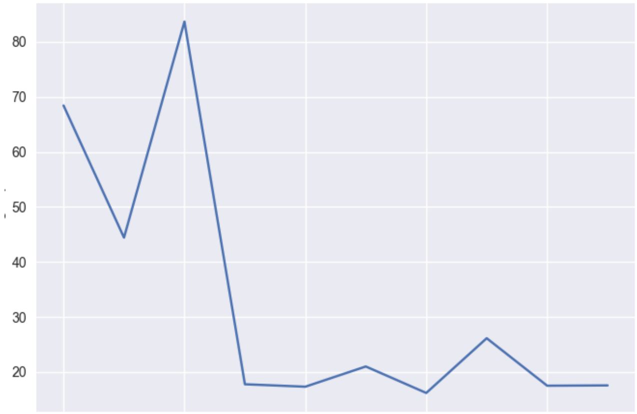


Figure 3.7: Flying speed heterogeneous arena graph. We represent to average flying speed for each drsophila in the arena.

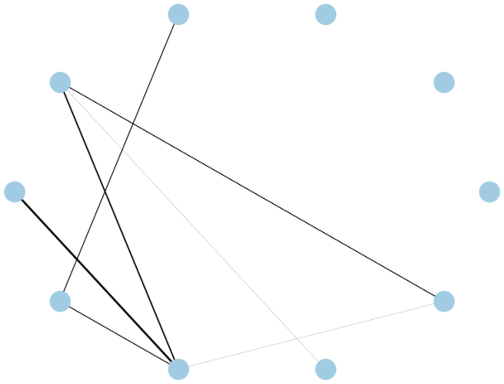


Figure 3.8: Interaction blind network. The graphs represent the interactions between drosophilas in an heterogeneous experiment. All and only the wild type insects compose the big connected component.

Chapter 4

Neural networks: Replicated Belief Propagation

In this chapter we introduce the general architecture of a neural network and give an overview to the Replicated Belief Propagation algorithm, whose structure is presented in [48]. After the presentation of the model we deeply explore the equations which characterize it.

4.1 Introduction

In artificial neural networks, learning from data can be performed through several methods. Basically training refers to determine the best set of weights for maximize the neural network's accuracy. A feed-forward neural network presents many layers of neurons connected together: the first layer takes in an input which is processed through the sucessive layers returning out an output value. More formally, consider a network composed by K layers each of them presenting N_K neurons (see figure 4.1). The algorithm is trained using a sample of M couples composed by an $N-$input vector ξ_i and the associated output τ_{ξ_i}, for $i = 1, \dots, M$. We define a_j^i the activation of the j^{th} neuron in the i^{th} layer which is computed according to the following relation:

$$a_j = \alpha\left(\sum_k \left(w_{jk}^i \cdot a_k^{i-1}\right) + b_j^i\right), \tag{4.1}$$

where:

- α is the activation function,
- w_{jk}^i is the weight of the link which connect the k^{th} neuron in the $(i-1)^{th}$ layer to the j^{th} neuron in the i^{th} layer,

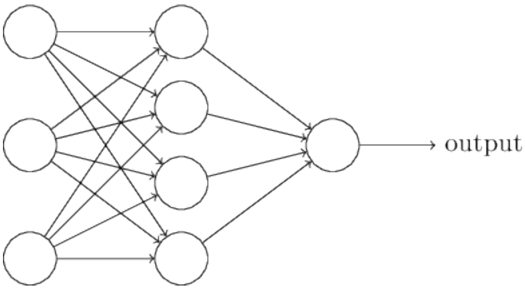


Figure 4.1: The picture represents a neural network structure. The network in the picture takes input vectors of size $N = 3$ and presents $K = 1$ hidden layer of size $N_1 = 4$.

- b^i_j is the bias of the j^{th} neuron in the i^{th} layer,
- a^i_j is the activation value of the j^{th} neuron in the i^{th} layer.

The activation function of a node defines the output of that node given an input and represents the state of the neuron which can fires or not, namely it acts as the Heaviside function of a given real value $z \in \mathbb{R}$. In order to smooth the error propagation during the training, it has been introduced the sigmoid activation function which is defined as $\alpha(z) = \dfrac{1}{1 + \exp(-z)}$.

With the aim to quantify how well the found the set of weights and biases approximates the output related to a specific input vector we define a cost function. The definition of a cost function is not universal but one example can be provided considering the relation

$$C(W, b) = \frac{1}{2M} \sum_{i \in M} ||y_M - \tau_{\xi_i}||^2, \tag{4.2}$$

where y_M is the output provided by the network using the set of weights and biases $\{W, b\}$. Connection weights are thus iteratively tuned through processes over the cost-function. One of the possible risks training a neural network, is that the found solution can be trapped in local minima with the result that the classification doesn't give optimal results [51]. In fact, it is not worth to remark that the cost function could present several points of minimum and not necessarily the found solution sits on the gllobal minimum: this is due to the nonlinearity introduced by the activation functions. The cost function of a network infact presents many hills, valleys and other irregularities which cause the presence of local minima, which can produce good but not excellent results. If we imagine the cost function as the energy function associated to dynamical system, we can find a parallelism between

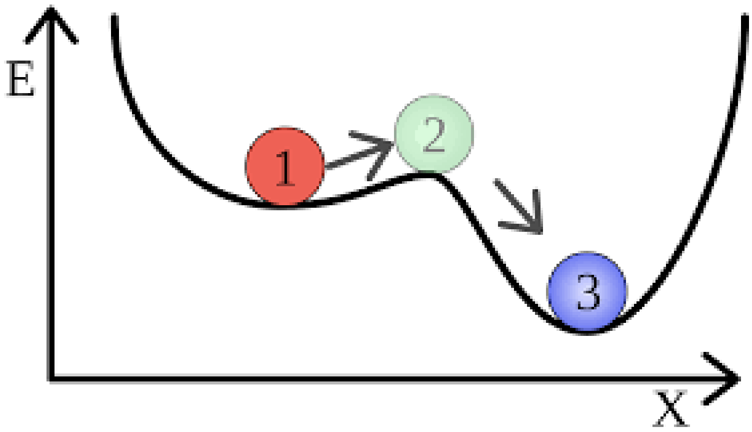


Figure 4.2: The picture presents three different energy states of a system: the red ball identifies a metastable state, the green one identifies an unstable state while the blue sits on the stable state.

local minima of the cost function and the metastable states. In physics, metastability is a stable state other than the system's state of least energy. An example can be provided thinking about a ball resting on a slope: if it is slightly pushed it will probably remain into its hollow, but if the push is stringer it probably rolls down the slope (see figure 4.2).

In order to avoid configurations which are trapped in local minima and to reduce the computational cost (which generally grows according to the size of the network), in [52, 53] it has been introduced a method that confirms experimentally the existence of subdominant and extremely dense regions of solutions for learning systems: in particular it has been proved that such dense regions are accessible by learning protocols and that synaptic configurations are robust to perturbations and generalize better than typical solutions.

This chapter is organized as follows. In the second section we provide a statistical mechanical contestualization of the model and we introduce the replica theory, which is deeply analyzed in appendix C. In the third section we give the definition of the energy of the mode. In the fourth section we explain the concept of the factor graph and we derive Belief Propagation equations. In the fifth section we analyze equations which rule the model.

4.2 Theoretical background of the model

As we introduced above, finding a solution which is not trapped in low performance configurations seems to be one of the main problems training a neural network with descrete synapses. One example can be provided by the K-statisfiability problem (Ksat) which deals with an ensemble of N boolean variables submitted to M constraints. Each constraint is an 'OR' function of K variables (or their negations) of the ensemble and the problem is to find a configuration which satisfies all the constraint. Because of the binary nature of the variables, the solution has to be found between 2^N possibilities and this was the first problem which has been classified as NP-complete becoming a central problem in combinatorial optimization. This problem can also model spin-glass problems, which deal with a set of N spins interacting with coupling constants: each couple of spins can be seen as a constraint and the minimal state of energy corresponds to minimize the number of violated constraints. In [54, 55] it has been proved that in the space of such solutions there could be one, many or few connected clusters. In any case, efficient algorithms seem to exist only when the system has extensive and connected regions of solutions, called *unfrozen* clusters. On the contrary, solutions which are isolated are called *locked* (see [56] for more details). For that reason in [57], it has been introduced a different measure of probability for the weights configurations: it ignores isolated solutions and enhances the statistical weight of clusters of solutions.

In order to introduce the training model, we start considering a system composed by a finite set of interacting spin particles $\sigma = \{\sigma_i\}_{i\in\Lambda}$, where Λ is a finite subset of $\mathbb{N}$. Suppose that particles interact according to the Hamiltonian

$$H(\sigma) = -\sum_{i,j\in\Lambda} J_{ij}\sigma_i\sigma_j - \sum_{i\in\Lambda} h_i\sigma_i, \tag{4.3}$$

where J_{ij} are the interaction parameters between particles σ_i and σ_j and h_i is the magnetic field associated to σ_i. Let us introduce the parameter $\beta = \dfrac{1}{k_B T}$ that corresponds to the thermodynamics temperature, where k_B is the Boltzmann constant and T is the system temperature. The equilibrium distribution of the configuration of spins is given by the measure of Boltzmann-Gibbs:

$$P(\sigma,\beta) = \frac{e^{-\beta H(\sigma)} \prod_{i\in\Lambda} d\rho(\sigma_i)}{Z(\beta)}, \tag{4.4}$$

where $Z(\beta)$ is the partition function of the system and is defined by

$$Z(\beta) = \sum_{i\in\Lambda} \exp\left(-\beta H(\sigma)\right). \tag{4.5}$$

As we have anticipated above, for learning problems with a discrete number of synapses, from numerical experiments it emerges the fact that efficient algorithms also seek unfrozen solutions, in fact the standard statistical analysis shows that the problem of learning random patterns with binary synapses in single layer networks is exponentially dominated by isolated solutions that are extremely hard to find algorithmically. In [58] it has been introduced a method which highlights analytical evidence for the existence of subdominant and extremely dense regions of solutions. Such results have been confirmed by numerical experiments and the found configurations are robust to perturbations and generalize better than typical solutions. In any cases it has been shown that the equilibrium description in these problems is not sufficient since it predicts that the problem is always in the regions with locked solutions. In [59], starting from an equilibrium solution of weights, the space structure around it has been explored: geometrically such space is composed by isolated solutions. This justifies the introduction of a new measure, which ignores isolated solutions increasing statistical weight of large and accessible regions of solutions. The idea is based on the Franz-Parisi potential [60] according to which we select a possible solution from the distribution of Gibbs and count how many other possible solutions are located around it. Given a configuration σ with energy $E(\sigma)$, such measure is defined by

$$P(\sigma,\beta,y,\gamma) = \frac{\exp(y\Phi(\sigma,\beta,\gamma))}{Z(\beta,y,\gamma)}, \tag{4.6}$$

where y has the formal role of the inverse of temperature. The function $\Phi(\sigma,\gamma,\beta)$ is the local free entropy associated to the configuration σ and satisfies the equation

$$\Phi(\sigma,\gamma,\beta) = \log \sum_{\sigma'} \exp(-\beta E(\sigma') - \gamma d(\sigma,\sigma')), \tag{4.7}$$

where σ' identifies other configuration solutions. The function $d(\cdot,\cdot)$ indicates the distance between configurations. Such distance is defined according to the model under consideration: during our implementation we consider the squared distance

$$d(\sigma,\sigma') = \frac{1}{2}\sum_{i=1}^{N} (\sigma_i - \sigma'_i)^2. \tag{4.8}$$

We start observing that for large values of β, the function (4.7) counts the number of minima of energy, weighting them via the parameter γ by the distance from the reference configuration σ^*, so that for large values of y, a configuration σ^* has a non-negligible weight only if it is surrounded by an exponential number of local minima. By increasing the value of the parameter γ, it is possible to focus on the configurations around σ^* which will have with high probability the same properties of the surrounding minima.

These large-deviation statistics seems to capture very well the behavior of efficient algorithms on discrete neural networks, which find solutions belonging to high-density regions when these regions exist, and fail otherwise. These solutions could not emerge in standard equilibrium description but there existe some algorithms that are able to find them. Moreover the method is robust since solutions are immersed in clusters, i.e. regions of other good solutions.

The energy landscape presents many valleys with the results that there are several local minima points: this justifies the introduction of the local entropy Φ, which permits to simplify the problem. To this aim, the increasing of the parameter γ plays a fundamental role. In fact, for small values of γ, the entropy landscape appears smooth and the dense region is identifiable only on a coarse-grained-scale. Increasing the value of γ, the landscape assumes the same shape of the energy landascape smoothing its irregularities so that the global minimum point of the entropy, which is situated in a dense region of minima points, corresponds to the global minimum point of the energy.

Hence, the aim of the method is to find the equilibrium configuration σ^* which minimizes Φ, with the result that the main difficulty becomes the estimation of this quantity: this problem is addressed using the Belief Propagation algorithm (see [62]) which will be described in the third section of this chapter.

The introduction of interacting replicas provides a very useful tool for seeking dense regions of solutions (see appendix C for more details about Replica theory): the structure of such model leads in fact good solutions to be organized in clusters. If the considered parameter y, which has been introduced above, is a non-negative integer, it counts the number of replicas of a given configuration σ^*. Making some simple computations we can rewrite the partition function as:

$$
\begin{aligned}
Z(\beta, y, \gamma) &= \sum_{\sigma^*} \exp(y\Phi(\sigma^*, \beta, \gamma)) = \\
&= \sum_{\sigma^*} (\exp(\Phi(\sigma^*, \beta, \gamma)))^y = \\
&= \sum_{\sigma^*} \left[\exp\left(\log \sum_{\sigma'} \exp(-\beta E(\sigma') - \gamma d(\sigma^*, \sigma')) \right) \right]^y = \\
&= \sum_{\sigma^*} \left(\sum_{\sigma'} \exp(-\beta E(\sigma') - \gamma d(\sigma^*, \sigma')) \right)^y = \\
&= \sum_{\sigma^*} \sum_{\sigma'_a} \prod_{a=1}^{y} \exp(-\beta E(\sigma'_a) - \gamma d(\sigma^*, \sigma'_a)) = \\
&= \sum_{\sigma^*} \sum_{\sigma'_a} \exp(-\beta \sum_{a=1}^{y} E(\sigma'_a) - \gamma \sum_{a=1}^{y} d(\sigma^*, \sigma'_a))
\end{aligned} \tag{4.9}
$$

The partition function (4.9) describes a system of $y+1$ interacting replicas of the system: the configuration σ^* acts as reference while the other y replicas are identical and interact with σ^*. Replicating the model under the addition of an interaction term provides a very useful tool to explore accessible regions of the energy landscape. This simple procedure can be applied to a variety of different algorithms and in particular we will focus on the Belief Propagation algorithm.

For our experiments, we consider a neural network composed by identical threshold units which are arranged in a feed-forward architecture. The network is trained on a training set composed by αN couples (ξ^μ, τ^μ), where $\xi^\mu \in \{-1, +1\}^N$ are binary input vectors and $\tau^\mu \in \{-1, +1\}$ is the associated scalar output. The parameter $\alpha > 0$ is called capacity of the network and gives informations on the maximum possible size of training set with respect to the length of the training vectors.

4.3 The energy function

The neural network we are going to consider is composed by identical threshold units which are arranged in a feed-forward architecture: each unit, which is parametrized by a vector of synapic weights W, has many input channels and issues only one output that is computed as the sign of the scalar product $\langle W, \xi^\mu \rangle$. The aim of training a neural network is to find a matrix of synaptic weights W so that the network outputs equals τ^μ for every pattern ξ^μ. This condition can be written in different ways according to the structure of the architecture.

If the model is a single-layer network, namely a perceptron, the synaptic weights are arranged in a vector with the same size of the input vector and the previous condition can be written as

$$E(W) = \sum_{\mu=1}^{\alpha N} \Theta(-\tau^\mu \langle W, \xi^\mu \rangle) = 0, \tag{4.10}$$

or equivalently as:

$$\chi(W) = \prod_{\mu=1}^{\alpha N} \Theta(\tau^\mu \langle W, \xi^\mu \rangle) = 1, \tag{4.11}$$

where $\Theta(x)$ is the Heaviside function.

Assuming our variables to be binary, the Ising model formalism results to be valid. Putting equation (4.10) into the Boltzmann Gibbs distribution $P(W, \beta) = Z(\beta)^{-1} \exp(-\beta E(W))$ we obtain the probability distribution for our network model, which is defined by:

$$P(W, \beta) = Z(\beta)^{-1} \exp(-\beta \sum_{\mu=1}^{\alpha N} \Theta(-\tau^\mu \langle W, \xi^\mu \rangle)). \tag{4.12}$$

Under the hypothesis that the learning procedure can be satisfied, i.e. if there exist some weights configurations such that the energy is null, when the temperature is very low, the following limit holds

$$\lim_{\beta \to \infty} \exp(-\beta \Theta(-u)) = \Theta(u)$$

and the distribution reads

$$P(W) = Z^{-1} \prod_{\mu=1}^{\alpha N} \Theta(\tau^\mu \langle W, \xi^\mu \rangle) = Z^{-1} \chi(W), \tag{4.13}$$

where the partition function

$$Z = \sum_W \prod_{\mu=1}^{\alpha N} \Theta(\tau^\mu \langle W, \xi^\mu \rangle) = \sum_W \chi(W) \tag{4.14}$$

counts the number of solutions of the problem.

The previous equations can be easily extended for a network which is composed by several layers. If the model is a fully-connected two-layer neural network, named committee machine, and presents one hidden layer of size K, then the energy of the weight configuration becomes

$$E(W) = \sum_{\mu=1}^{\alpha N} \Theta(-\tau^{\mu} \sum_{k=1}^{K} \mathrm{sign}\langle W^k, \xi^{\mu} \rangle), \tag{4.15}$$

where W^k is the vector of synaptic weights associated to the k^{th} hidden unit and has the same length of the input vector ξ.

As we have introduced above, in order to avoid isolated solutions, we build the architecture according to the replica theory and we compute the distance function between replicas of the system using the definition in (4.8).

4.4 The Belief Propagation equations

The Belief Propagation algorithm, also known as sum-product message-passing, is an iterative message-passing method which can be used for performing inference on graphical models [61, 64]. It calculates the marginal distribution of each unobserved node, conditional on any observed nodes. One example of graphical model is given by factor graphs.

In order to define a factor graph, we consider N random variables x_i and a real function $g(x_1, \dots, x_N) = g(\vec{x})$. For each node there exists a marginal function g_i such that

$$g_i(x_i) = \sum_{\bar{x_i}} g(\vec{x}), \tag{4.16}$$

where $\bar{x_k}$ means that the sum runs over all the variables except x_k. Suppose that there exist m functions f_j and m subset $S_j \in \{x_1, \dots, x_n\}$, for $j = 1 \dots, m$, such that the function g can be factorized as

$$g(\vec{x}) = \prod_{j=1}^{m} f_j(S_j). \tag{4.17}$$

Under these conditions, the function corresponds to a factor graph $G = (X, F, E)$, where $X = \{x_1, \dots, x_N\}$ is the set of variable vertices, $F = \{f_1, \dots, f_m\}$ is the set of factor vertices and E is the set of edges which connect the variable vertex x_k and the factor vertex f_j if and only if $x_k \in S_j$ (see figure 4.3).

Factor graphs are related to message-passing algorithms since they are useful to compute certain characteristics of the function g like the marginal

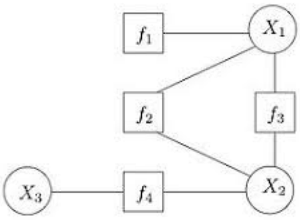


Figure 4.3: Example of factor graph which corresponds to the factorization of a function $g(x_1, x_2, x_3) = f_1(x_1)f_2(x_1, x_2)f_3(x_1, x_2)f_4(x_2, x_3)$.

distributions: nodes are considered as processing units and edges as a channel to exchange messages. We identify with $\mu_{x_i \to f_j}$ messages which go from node variables to factor variables and with $\mu_{f_j \to x_i}$ the inverse message.

We start observing that message passing algorithms are usually exact for trees, but only approximate for graphs with cycles (see [63]): for that reason we are going to give equations for the computation of messages for graph trees and then we provide some alternative for general kinds of graphs.

We start describing how to compute the marginal of the node x_0, which is the root of the graph tree. Starting from leaves, messages are recursively exchanged level by level with the upper nodes. Given the node x_k, we define in general $\partial x_k = \{j \in \{1, \dots m\}$ such that f_j is connected to $x_k\}$. Then the marginal is obtained as

$$g_0(x_0) = \prod_{k \in \partial x_0} \mu_{f_k \to x_0}(x_0), \tag{4.18}$$

where $\mu_{f_k \to x_0}$ are the messages from factor nodes f_k to the variable node x_0 that are computed after the convergence of the algorithm which is described in the following.

This procedure can be extended for the computation of marginals of the other nodes of the graph. When a node is situated at an intermediate level, it plays both the role of being a father and a son and its marginal is computed as the product of both messages from factors to variables and viceversa.

Since when this approach is applied on graphs with cycles it doesn't give the correct solution, there have been provided many approaches to deal this problem.

One possible approach to decode messages in this case is to agglomerate the graph enough until cycles are deleted, and then apply the sum-product algorithm, which will now be exact.

A second approach consists in applying the procedure described above locally and recursively, hoping that the method converges. Messages from nodes x_i to factors f_a and their opposites are respectively updated at each iteration step t according to the followings equations:

$$\mu^t_{x_i \to f_a}(x_i) = \prod_{k \in \partial x_i \setminus a} \mu^t_{f_k \to x_i}(x_i) \tag{4.19}$$

$$\mu^{t+1}_{f_a \to x_i}(x_i) = \sum_{j \neq i} \left[f_a(S_a) \prod_{j \in \partial f_a \setminus i} \mu^t_{x_j \to f_a}(x_j) \right], \tag{4.20}$$

where $\partial f_a = \{j \in \{1, \dots n\}$ such that x_j is connected to $f_a\}$. Latter equations are updated alternatively until convergence is achieved, which is ensured by the following statement (see [64] for more details):

Theorem 4.4.1. *Consider a tree-graphical model with diameter t^*, i.e. suppose that t^* is the maximum distance between any two variable nodes. Then:*

1. *the BP update equations* (4.19) *and* (4.20) *converge at most t^* iterations,*
2. *the fixed point messages provide the exact marginals for any variable node i.*

When convergence is achieved, messages can be used to compute marginals, which are called *beliefs*, as follows:

$$g_i(x_i) = \prod_{k \in \partial x_i} \mu_{f_k \to x_i}(x_i). \tag{4.21}$$

We are now going to apply this procedure to the network model in order to derive Belief Propagation equations.

Consider a spin glass model where N spins σ_i taking value in a finite set Ω interact according to the Hamiltonian

$$H(J) = \sum_a J_a \prod_{i \in \partial J_a} \sigma_i, \tag{4.22}$$

where J_a are the interaction parameters and ∂J_a indicates the set of spin indices which interact through J_a Under these conditions, the computation of local magnetizations consists in solving an inference Bayesian problem where the spins distribution depends by interaction parameters.

Suppose that interaction parameters are extracted from a uniform distribution $P(J)$. Using the Bayes' theorem, given the interaction parameters, the conditional probability of the spins can be factorizable as follows:

$$P(\overline{\sigma}|\{J_a\}) = P(\{J_a\}|\overline{\sigma}) \frac{P(\overline{\sigma})}{P(\{J_a\})} \propto P(\{J_a\}|\overline{\sigma}) = \prod_a P(J_a|\overline{\sigma}), \tag{4.23}$$

where

$$P(\{J_a\}|\overline{\sigma}) = \frac{\exp\left(\beta J_a \prod_{j\in\partial J_a} \sigma_j\right)}{\sum_{J_a} \exp\left(\beta J_a \prod_{j\in\partial J_a} \sigma_j\right)}.$$

The structure of the probability distribution in (4.23), leads us to represent the system as a factor graph where the factor and the variable nodes are respectively identified by the interaction parameters and the spins. Under these conditions, equations (4.19) and (4.20) become:

$$\mu^t_{\sigma_i\to J_a}(\sigma_i) = \prod_{k\in\partial\sigma_i\setminus a} \mu^t_{J_k\to\sigma_i}(\sigma_i) = P(\sigma_i|\{J_{k\neq a}\}) \tag{4.24}$$

$$\mu^{t+1}_{J_a\to\sigma_i}(\sigma_i) = \sum_{\Omega,\sigma_j\neq\sigma_i} \left[P(\{J_a\}|\sigma_j) \prod_{j\in\partial J_a\setminus i} \mu^t_{\sigma_j\to J_a}(\sigma_j) \right]. \tag{4.25}$$

Previous equations can be rewritten using probability formalism as:

$$\mu^t_{\sigma_i\to J_a}(\sigma_i) = P(\sigma_i|\{J_{k\neq a}\}) \tag{4.26}$$

$$\mu^t_{J_a\to\sigma_i}(\sigma_i) \propto P(J_a|\sigma_i,\{J_{k\neq a}\}) = \sum_{\Omega,\sigma_j\neq\sigma_i} P(\{J_a\}|\sigma_j)P(\sigma_j|\{J_{k\neq a}\}). \tag{4.27}$$

In order to eliminate the dependence by the value assumed by the spin σ_i, we introduce the notion of *cavity magnetization*, which is simply the average of the respective message (4.26) and (4.27) at each iteration step:

$$\begin{aligned} m^t_{\sigma_i\to J_a} &= \frac{\sum_\Omega \sigma_i \mu^t_{\sigma_i\to J_a}(\sigma_i)}{\sum_\Omega \mu^t_{\sigma_i\to J_a}(\sigma_i)} \\ \hat{m}^t_{J_a\to\sigma_i} &= \frac{\sum_\Omega \sigma_i \mu^t_{J_a\to\sigma_i}(\sigma_i)}{\sum_\Omega \mu^t_{J_a\to\sigma_i}(\sigma_i)}. \end{aligned}$$

In the specific case of the Ising model, since variables can take value in $\Omega = \{-1,+1\}$, previous equations respectively read

$$m^t_{\sigma_i\to J_a} = \frac{\mu^t_{\sigma_i\to J_a}(+1) - \mu^t_{\sigma_i\to J_a}(-1)}{\mu^t_{\sigma_i\to J_a}(+1) - \mu^t_{\sigma_i\to J_a}(-1)} \tag{4.28}$$

$$\hat{m}^t_{J_a\to\sigma_i} = \frac{\mu^t_{J_a\to\sigma_i}(+1) - \mu^t_{J_a\to\sigma_i}(-1)}{\mu^t_{J_a\to\sigma_i}(+1) + \mu^t_{J_a\to\sigma_i}(-1)} \tag{4.29}$$

giving the following definitions for the messages:

$$\mu^t_{\sigma_i\to J_a}(\sigma_i) = \frac{1 + m^t_{\sigma_i\to J_a}\sigma_i}{2} \tag{4.30}$$

$$\mu^{t+1}_{J_a\to\sigma_i}(\sigma_i) \propto \frac{1 + \hat{m}^t_{J_a\to\sigma_i}\sigma_i}{2}. \tag{4.31}$$

Proposition 4.4.2. *Consider a set of spins which interact according to the Ising model. Magnetizations* (4.28) *and* (4.29) *can be exactly computed and are respectively defined by:*

$$m^t_{\sigma_i \to J_a} = \tanh\left[\sum_{k \in \partial\sigma_i \setminus a} \tanh^{-1}(\hat{m}^t_{J_a \to \sigma_i})\right] \tag{4.32}$$

$$\hat{m}^{t+1}_{J_a \to \sigma_i} = \tanh(\beta J_a) \prod_{j \in \partial J_a \setminus i} m^t_{\sigma_j \to J_a}. \tag{4.33}$$

Proof. In order to derive equation (4.32), we recall the equation (4.19), since the message from the variable node σ_i to the factor node J_a satisfies both the cavity magnetization (4.28) and the equation

$$\mu^t_{\sigma_i \to J_a} = \prod_{k \in \partial\sigma_i \setminus a} \mu^t_{J_k \to \sigma_i}. \tag{4.34}$$

Making some simplifications and putting (4.28) in (4.34), we obtain:

$$\begin{aligned}
m^t_{\sigma_i\to J_a} &= \frac{\mu^t_{\sigma_i\to J_a}(+1)-\mu^t_{\sigma_i\to J_a}(-1)}{\mu^t_{\sigma_i\to J_a}(+1)+\mu^t_{\sigma_i\to J_a}(-1)} = \\
&= \frac{1-\dfrac{\mu^t_{\sigma_i\to J_a}(-1)}{\mu^t_{\sigma_i\to J_a}(+1)}}{1+\dfrac{\mu^t_{\sigma_i\to J_a}(-1)}{\mu^t_{\sigma_i\to J_a}(+1)}} = \\
&= \frac{1-\exp\left(2\cdot\dfrac{1}{2}\log\left(\dfrac{\mu^t_{\sigma_i\to J_a}(+1)}{\mu^t_{\sigma_i\to J_a}(-1)}\right)\right)}{1+\exp\left(2\cdot\dfrac{1}{2}\log\left(\dfrac{\mu^t_{\sigma_i\to J_a}(+1)}{\mu^t_{\sigma_i\to J_a}(-1)}\right)\right)} = \\
&= \tanh\left[\frac{1}{2}\log\left(\frac{\mu^t_{\sigma_i\to J_a}(+1)}{\mu^t_{\sigma_i\to J_a}(-1)}\right)\right] = \\
&= \tanh\left[\frac{1}{2}\log\left(\frac{\prod_{k\in\partial s_i\setminus a}\mu^t_{J_k\to\sigma_i}(+1)}{\prod_{k\in\partial s_i\setminus a}\mu^t_{J_k\to\sigma_i}(-1)}\right)\right] = \\
&= \tanh\left[\frac{1}{2}\sum_{k\in\partial s_i\setminus a}\log\left(\frac{\mu^t_{J_k\to\sigma_i}(+1)}{\mu^t_{J_k\to\sigma_i}(-1)}\right)\right] = \\
&= \tanh\left[\frac{1}{2}\sum_{k\in\partial s_i\setminus a}\log\left(\frac{1+\dfrac{\mu^t_{J_k\to\sigma_i}(+1)-\mu^t_{J_k\to\sigma_i}(-1)}{\mu^t_{J_k\to\sigma_i}(+1)+\mu^t_{J_k\to\sigma_i}(-1)}}{1-\dfrac{\mu^t_{J_k\to\sigma_i}(+1)-\mu^t_{J_k\to\sigma_i}(-1)}{\mu^t_{J_k\to\sigma_i}(+1)+\mu^t_{J_k\to\sigma_i}(-1)}}\right)\right] = \\
&= \tanh\left[\sum_{k\in\partial s_i\setminus a}\frac{1}{2}\log\left(\frac{1+\hat{m}^t_{J_k\to\sigma_i}}{1-\hat{m}^t_{J_k\to\sigma_i}}\right)\right] = \\
&= \tanh\left[\sum_{k\in\partial s_i\setminus a}\tanh^{-1}(\hat{m}^t_{J_k\to\sigma_i})\right],
\end{aligned}$$

where $m^t_{J_k\to\sigma_i}$ satisfies equation (4.29).

In order to derive equation (4.33), we recall the equation (4.20), since the message from the the factor node J_a to variable node σ_i satisfies both the cavity magnetization (4.29) and the equation

$$\mu^{t+1}_{J_a\to\sigma_i}(\sigma_i) = \sum_{\Omega,\sigma_j\neq\sigma_i}\left[\exp\left(\beta J_a\prod_{k\in\partial J_a}\sigma_k\right)\prod_{k\in\partial J_a\setminus i}\mu^t_{\sigma_k\to J_a}(\sigma_k)\right]. \qquad (4.35)$$

Putting (4.35) into (4.29) we obtain:

$$\hat{m}_{J_a\to\sigma_i} = \frac{\sum_\Omega \left[\exp\left(\beta J_a \prod_{j\in\partial J_a\setminus i}\sigma_j\right) - \exp\left(-\beta J_a \prod_{j\in\partial J_a\setminus i}\sigma_j\right)\prod_{j\in\partial J_a\setminus i}\mu_{\sigma_i\to J_a}(\sigma_j)\right]}{\sum_\Omega \left[\exp\left(\beta J_a \prod_{j\in\partial J_a\setminus i}\sigma_j\right) + \exp\left(-\beta J_a \prod_{j\in\partial J_a\setminus i}\sigma_j\right)\prod_{j\in\partial J_a\setminus i}\mu_{\sigma_i\to J_a}(\sigma_j)\right]}. \tag{4.36}$$

The numerator of the latter equation can be obviously rewritten as

$$\begin{aligned}
&\sum_\Omega \left[\exp\left(\beta J_a \prod_{j\in\partial J_a\setminus i}\sigma_j\right) - \exp\left(-\beta J_a \prod_{j\in\partial J_a\setminus i}\sigma_j\right)\prod_{j\in\partial J_a\setminus i}\mu_{\sigma_j\to\sigma_a}(\sigma_j)\right] = \\
=&\sum_\Omega \left\{\left[\left(\exp\left(\beta J_a \prod_{j\in\partial J_a\setminus\{i,k\}}\sigma_j\right) - \exp\left(-\beta J_a \prod_{j\in\partial J_a\setminus\{i,k\}}\sigma_j\right)\right)\mu_{\sigma_k\to J_a}(1) + \right.\right. \\
&+\left.\left(\exp\left(-\beta J_a \prod_{j\in\partial J_a\setminus\{i,k\}}s_j\right) - \exp\left(\beta J_a \prod_{j\in\partial J_a\setminus\{i,k\}}\sigma_j\right)\right)\mu_{\sigma_k\to J_a}(-1)\right]\cdot \\
&\left.\cdot\prod_{j\in\partial J_a\setminus\{i,k\}}\mu_{\sigma_j\to J_a}(\sigma_j)\right\} = \\
=&\sum_\Omega \left\{\left[\exp\left(\beta J_a \prod_{j\in\partial J_a\setminus\{i,k\}}\sigma_j\right) - \exp\left(-\beta J_a \prod_{j\in\partial J_a\setminus\{i,k\}}\sigma_j\right)\right]\cdot\right. \\
&\left.\cdot\left(\mu_{\sigma_k\to J_a}(1) - \mu_{\sigma_k\to J_a}(-1)\right)\prod_{j\in\partial J_a\setminus\{i,k\}}\mu_{\sigma_j\to J_a}(\sigma_j)\right\} = \\
&= \left(\mu_{\sigma_k\to J_a}(1) - \mu_{\sigma_k\to J_a}(-1)\right)\cdot \\
&\cdot\sum_\Omega \left\{\left[\exp\left(\beta J_a \prod_{j\in\partial J_a\setminus\{i,k\}}\sigma_j\right) - \exp\left(-\beta J_a \prod_{j\in\partial J_a\setminus\{i,k\}}\sigma_j\right)\right]\prod_{j\in\partial J_a\setminus\{i,k\}}\mu_{\sigma_j\to J_a}(\sigma_j)\right\}.
\end{aligned}$$

Applying the previous procedure to every variable σ_j, the numerator and the denominator of (4.36) can be respectively written as

$$\left(\exp(\beta J_a) - \exp(-\beta J_a)\right)\prod_{j\in\partial J_a\setminus i}\left(\mu_{\sigma_k\to J_a}(1) - \mu_{\sigma_k\to J_a}(-1)\right) \tag{4.37}$$

and

$$\left(\exp(\beta J_a) + \exp(-\beta J_a)\right)\prod_{j\in\partial J_a\setminus i}\left(\mu_{\sigma_k\to J_a}(1) + \mu_{\sigma_k\to J_a}(-1)\right), \tag{4.38}$$

so that (4.36) is equivalent to

$$\begin{aligned}\hat{m}_{J_a\to\sigma_i} &= \frac{(\exp(\beta J_a)-\exp(-\beta J_a))\prod_{j\in\partial J_a\setminus i}(\mu_{\sigma_k\to J_a}(1)-\mu_{\sigma_k\to J_a}(-1))}{(\exp(\beta J_a)+\exp(-\beta J_a))\prod_{j\in\partial J_a\setminus i}(\mu_{\sigma_k\to J_a}(1)+\mu_{\sigma_k\to J_a}(-1))} = \\ &= \frac{\exp(\beta J_a)-\exp(-\beta J_a)}{\exp(\beta J_a)+\exp(-\beta J_a)}\prod_{j\in\partial J_a\setminus i}\frac{\mu_{\sigma_k\to J_a}(1)-\mu_{\sigma_k\to J_a}(-1)}{\mu_{\sigma_k\to J_a}(1)+\mu_{\sigma_k\to J_a}(-1)} = \\ &= \tanh(\beta J_a)\prod_{j\in\partial J_a\setminus i} m_{\sigma_j\to J_a},\end{aligned}$$

where

$$m_{\sigma_j\to J_a} = \frac{\mu_{\sigma_k\to J_a}(1)-\mu_{\sigma_k\to J_a}(-1)}{\mu_{\sigma_k\to J_a}(1)+\mu_{\sigma_k\to J_a}(-1)}.$$

□

4.5 Theoretical details: the equations which rule the model

We summarize in the following the main steps of the method. We are going to deal with three different kinds of nodes (see figure 4.4): we denote with

- σ_i, the variable nodes which represent weights,
- η^μ the factor node defined by the function $\Theta(\tau^\mu\langle W^k,\xi^\mu\rangle)$,
- τ^μ the variable node which represents the otuput associated to the pattern ξ^μ

It is not worth to remark that in the network model we are going to describe, the weights play the same role of the spins in the previous section. In fact, as we have said already, we aim to find the weight configuration which minimizes the energy of the model, which corresponds to the equilibrium spins configuration announced above.

In order to train the neural network model, we aim to solve a system of equations: the involved quantities are called messages and represent *cavity marginal probabilties.* Each edge of the graph presents two messages going in opposite directions which are respectively represented by an equation of the system and expressed as a function of the neighbouring messages. Such system is solved iteratively, updating the equations until a fixed point is reached. In according to the structure of the graph, we have to define four

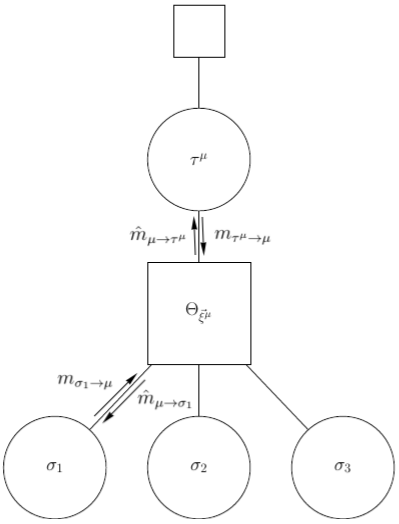


Figure 4.4: The figure shows the factor graph associated to a perceptron which takes in input a vector ξ^μ of size 3. Circles on the bottom represents the sinaptic weights, the square represents the factor node η^μ and the circle on the top represents the output τ^μ.

different kinds of messages: two messages respectively start from the variable nodes σ_i and τ^μ and are directed to the factor node η^μ, while the opposite messages start from the factor node η^μ and respectively directed to σ_i and τ^μ.

Messages from variable (input or output) nodes to the factor node are defined according to (4.32) at each iteration step t by the following equations

$$m^t_{\sigma_i\to\eta^\mu} = \tanh\left[\sum_{k\in\partial\sigma_i\setminus\eta^\mu} \tanh^{-1}(m^t_{k\to\sigma_i})\right] \tag{4.39}$$

$$m^t_{\tau^\mu\to\eta^\mu} = \tanh\left[\sum_{k\in\partial\tau^\mu\setminus\eta^\mu} \tanh^{-1}(m^t_{k\to\tau^\mu})\right] \tag{4.40}$$

while messages from factor nodes to input or output variable nodes are defined according to (4.29) by the following equations

$$m_{\eta^\mu\to\sigma_i} = \frac{\sum_{\sigma_i} \sigma_i \nu^t_{\eta^\mu\to\sigma_i}(\sigma_i)}{\sum_{\sigma_i} \nu^t_{\eta^\mu\to\sigma_i}(\sigma_i)} \tag{4.41}$$

$$m_{\eta^\mu\to\tau^\mu} = \frac{\sum_{\tau^\mu} \tau^\mu \nu^t_{\eta^\mu\to\tau^\mu}(\sigma_i)}{\sum_{\tau^\mu} \nu^t_{\eta^\mu\to\tau^\mu}(\tau^\mu)} \tag{4.42}$$

where $\nu^t_{\eta^\mu \to \sigma_i}(\sigma_i)$ and $\nu^t_{\eta^\mu \to \tau^\mu}(\tau^\mu)$ come from (4.20) and are equal to

$$\nu^t_{\eta^\mu \to \sigma_i}(\sigma_i) = \sum_{\tau^\mu, \{\sigma_j\} j \neq i} \left\{ \Theta\left[\tau^\mu \left(\sum_{k \in \partial\eta^\mu \setminus i} \sigma_k \xi_k^\mu \right) \right] \left(\frac{1 + \tau^\mu m^t_{\tau^\mu \to \eta^\mu}}{2} \right) \prod_{k \in \partial\eta^\mu \setminus i} \frac{1 + \sigma_k m^t_{\sigma_k \to \eta^\mu}}{2} \right\}$$
$$\nu^t_{\eta^\mu \to \tau^\mu}(\tau^\mu) = \sum_{\{\sigma_j\}} \left\{ \Theta\left[\tau^\mu \left(\sum_{k \in \partial\eta^\mu} \sigma_k \xi_k^\mu \right) \right] \prod_{k \in \partial\eta^\mu} \frac{1 + \sigma_k m^t_{\sigma_k \to \eta^\mu}}{2} \right\}$$

Even if equations above give the exact expression of the messages, the result can be obtained in $O(N^3)$ operations, where N is the size of the input, using either a partial convolution scheme or discrete Fourier trainsforms. When N is sufficiently large it is possible to use an approximated fast version: the result can be computed in $O(N)$ operations through the central limit theorem (see [65]). Given an input vector ξ^μ, let us define the following quantities:

$$\begin{cases} a^t_{\eta^\mu} = \sum_{i \in \partial\eta^\mu} \xi_i^\mu m^t_{i \to \eta^\mu} \\ b^t_{\eta^\mu} = \sum_{i \in \partial\eta^\mu} \left(1 - (m^t_{i \to \eta^\mu})^2\right) \\ g^t_{\eta^\mu \to j}(\sigma) = \operatorname{erf}\left(\dfrac{a^t_{\eta^\mu \to j} + \sigma \xi_j^\mu}{\sqrt{2 b^t_{\eta^\mu \to j}}} \right) \\ a^t_{\eta^\mu \to j} = a^t_{\eta^\mu} - \xi_j^\mu m^t_{j \to \eta^\mu}, \\ b^t_{\eta^\mu \to j} = b^t_{\eta^\mu} - \left(1 - (m^t_{j \to \eta^\mu})^2\right) \end{cases}$$

According to results in [48], equations (4.41) and (4.42) become

$$m^{t+1}_{\eta^\mu \to \tau} = \operatorname{erf}\left(\frac{a^t_{\eta^\mu}}{\sqrt{2 b^t_{\eta^\mu}}} \right), \tag{4.43}$$
$$m^{t+1}_{\eta^\mu \to j} = m^{t+1}_{\eta^\mu \to \mu} \frac{g^t_{\eta^\mu \to j}(+1) - g^t_{\eta^\mu \to j}(-1)}{2 + m^t_{\tau \to \eta^\mu}\left(g^t_{\eta^\mu \to j}(+1) + g^t_{\eta^\mu \to j}(-1)\right)}, \tag{4.44}$$
$$m^t_{\eta^\mu \to j} = \tanh\left(\sum_{\nu \in \partial j \setminus \eta^\mu} \tanh^{-1} m^t_{\nu \to j} \right). \tag{4.45}$$

After the convergence, the single site magnetizations can be computed as:

$$m_j = \tanh\left(\sum_{\eta^\mu \in \partial j} \tanh^{-1} m_{\eta^\mu \to j} + \tanh^{-1}(m_{* \to j}) \right), \tag{4.46}$$

where $m_{*\to j}$ is the contribute given by replicas described in appendix D.

The optimal configuration of weights can now be computed from the single site magnetizations as $W_j = \mathrm{sign}(m_j)$. After having trained the model, i.e. when the weight configuration has been computed, given any other input vector, the output associated is computed as the scalar product $\langle W, \xi \rangle$.

Chapter 5

Traffic slowdowns prediction

In this chapter we show an application of the RBP algorithm which has been analyzed in the previous chapter.

5.1 Introduction

Traffic is a complex system where the vehicle interactions and finite volume effects imply the existence of different collective regimes and phase transition phenomena. In particular the congestion formation has been studied in the framework of complex systems physics and its prediction can be a difficult problem due to the hetetogenous behaviour of drivers when the vehicle density increases. We propose a novel pipeline to classify traffic slowdowns by analyzing the features extracted from the fundamental diagram of traffic. We train a deep learning neural network providing a forewarning time of prediction related to the training set size. Then we compare our performances with those of the most common classifiers used in machine learning analysis.

The chapter is organized in the following sections and results.

In the second section we provide an historical background about studies which have been performed for what it concerns traffic congestion. In the third section we describe the analyzed dataset and we discuss the theoretical framework of the proposed classification and prediction methods. In the fourth section we present the results of our analysis and we make a comparison with other common classifiers. The fifth section contains the conclusion of our analysis.

5.2 Historical background

Understanding the traffic dynamics is fundamental task to improve the life quality and to reduce environmental impact of congestions. The traffic congestions are a cause of wasted fuel, air pollution and billions of wasted dollars in each year due to time lost. The research have tackled this problem by developing increasingly complex models that simulate the vehicles interaction and the driver behaviour in the different traffic regimes [66]. These models succeeded to point out the existence of critical values for the traffic density [67], the physical nature of formation of the stop and go regime [68, 69] and the dynamics of the traffic waves [70]. Many studies have been performed to estimate the urban traffic congestion and to implement prediction methods [71, 72, 73, 74, 75, 76, 77, 78], focusing on various different features of classification. In any cases the study of transient states that precede the congestion formation problem, the nature of traffic fluctuations near a critical states and the role of heterogeneous behavior of driver are still open problems. Therefore research is attempting to handle this problem or at least ease its deleterious effects [79, 80], also by using an apporach based on artificial intelligence point of view [81, 82, 83, 84, 85]. There are several research methods proposed in the field of traffic prediction like deterministic, non-deterministic and stochastic methods [86]. In this paper we develop an approach based on a deep learning algorithm to forecast the rising of the congestions related to stop and go regimes by classifying the traffic slowdowns that may precede the congestion formation. To apply our method we measure the traffic slowdowns by using the data from magnetic coils that record passage time a vehicle velocity. We consider samples of 100 vehicles during rush hours in an italian country road (MTS system) and we classify the signals depending on whether a congestion is observed in a sequent sample. The congestion is detected as a sudden decrease of the flow together with a increase of the traffic density according to the fundamental traffic diagram [87]. Besides the classification, our approach trains a deep learning algorithm to forecast the congestion formation from the feautures of the anticipatory traffic slowdowns. In this way one is able to define a warning time that could be used to perform strategies which reduce the congestion effects.

Traditional studies have developed simulation techniques to model traffic congestion dynamics [88], but many approaches have limitations, mainly due to errors during the calibration process of parameters and unrealistic assumptions. Deep learning techniques [51] are considered some of the most promising techniques to process huge high-dimensional data. The proposed method is based on the Binary Committee Machine Replicated Focusing Belief Propagation neural network (RFBP) [48]. The RFBP is an iterative message-passing algorithm for performing inference on graphical models: it calculates the marginal distribution for each unobserved node, conditional on any observed nodes [89, 57]. This method requires that the input data are binary signals. Therefore we have preprocessed the data from the magnetic coils to get binary signals that represent the dynamical features of the traffic slowdowns. Detectors record every car passing and generate an high time resolutions in the amount of data.

One of the main tool to study the congestion problem is the fundamental diagram that empirically defines a nonlinear relation between traffic flow and density [90]. Recently the existence of a fundamental diagram for a whole road network have been proposed and its application to study the congestion regime has been considered [91].

The chapter is organized as follows. In the second section we describe the dataset used in our analysis and we discuss the theoretical framework of the proposed classification and prediction methods. In the third section we present the results of our analysis and we make a comparison with other common classifiers. The fourth section contains the conclusion of the paper.

5.3 Materials and Method

In order to effectively train the model to make forecasts under the normal and congested traffic conditions observed in a road, a big dataset of past situations is necessary. In particular, in this paper, we use three datasets in the different steps performed to defined our model. Firstly we need to fit the parameters of the model using a training set: in this stage we train the model using a supervised learning method. The training set is composed by a set of coupled signals: an input vector, which consists in a binary vector representing the dynamical state of the traffic on the considered road in a given time interval, and the corresponding target, that identifies if in the following time interval the road will present a congestion or not. More precisely, we identified different time intervals with different input vectors sizes: different lengths correspond to the respective interval time of forecasting which ranges from 10 to 20 minutes. Secondly, the RFPB weight matrix obtained during the training is used to predict the targets for the observations in the second dataset, namely the validation dataset, which allows an unbiased evaluation of fitting procedure on the training set. Finally the third dataset, which is the test set, is used to provide an unbiased evaluation of the performances of the model.

The car data have been collected by the MTS system installed by Emilia Romagna region [92] and refer to the traffic of four country roads during

May 2011: magnetic coils have been installed on each side of a road and register the time and the velocity of moving of each vehicle. The dataset contains a list of vehicles with the corresponding informations: date, time, lane, direction and speed (see table 5.1).

Date	Time	Lane	Direction	Speed
05/01/2011	06.00.32	0	0	58
05/01/2011	06.00.36	0	0	65
05/01/2011	06.02.16	0	0	60
05/01/2011	06.02.44	2	1	71
05/01/2011	06.03.00	0	0	89

Table 5.1: The table shows the first five rows of data which have been recorded by a detector located on the SS72 country road near Rimini (lat-lon coordinates 44.0352066,12.5393341). The first and the second column contain the date and recording time. The third column contains the integer which ranges from 0 to the total number of lanes which is associated to the lane where the car drives. The fourth column contains the value 0 or 1 which is assigned according to the driving direction (obviously different lanes can be run in the same direction). The fifth column contains the driving speed which is measured by km/h.

Our analysis have been performed selecting data recorded from four different roads where different traffic dynamical regime are observed during the considered period. As expected, during weekdays we obviously find a daily amount of cars which is greater than the weekend's one. The daily amount of each lane reaches in average fifteen thousand of cars during weekdays and reduces to ten thousand cars during the weekend. Since we are interested in high traffic daily regime, we only analyzed weekdays data. According to empirical observations, each road is characterized by a massimal capacity: if the number of cars that enter the road is greater than this quantity, a slowdown will occur. In particular we present results coming from the analysis of data which have been recorded by detector number 454 which is sit in a road with four lanes (the dataset is available in *link*).

We define the flow as the ratio between a fixed number of cars and the time interval they need to pass and the density is the ratio between the flow and the average speed of the interval. In order to do that we consider the driving speed v_i and the driving time t_i of each car. Then, fixed the number of cars N, we define by $\Delta T(N)$ the time needed by the selected group of N cars to pass, i.e. $\Delta T(N) = \sum_{i=1}^{N-1}(t_{i+1} - t_i) = t_N - t_0$, and we normalize it by N obtaining $\bar{T}(N) = \frac{\Delta T(N)}{N}$. We compute the average driving speed of the group $\bar{v}(N) = \frac{\sum_{i=1}^{N} v_i}{N}$. The flow and the density are respectively computed as

$$\phi(N) = \frac{1}{\bar{T}(N)} \quad \text{and} \tag{5.1}$$

$$\rho(N) = \frac{\phi(N)}{\bar{v}(N)}. \tag{5.2}$$

More precisely, after some trials with different vaues of N from 80 to 120, we fix a range of 100 cars which is shifted each time a new vehicle arrives in order to define a time dependent flow during the day. Then a "moving density" has been computed as the ratio between the corresponding value of the flow of each set of cars and its average speed. In the data processing, we focus on two traffic dynamics features: the traffic flow and the vehicle density which are related according to the fundamental diagram (FD) of traffic (Figure 5.1) in homogeneous steady conditions.

The traffic flow is in a stable dynamical regime when the flow increases according to the traffic density. At low traffic densities, the effect of interactions among vehicles are negligible, the flow grows almost linearly with the density until a critical density value is reached, at which the flow reaches the road capacity taking its maximum value. In this situation the flow suddenly decreases as the density increases producing a local congestion. The FD highligths the interaction between vehicles and it is consistent with the existence of an optimal velocity model for vehicle dynamics [93]. When a congestion situation is rising, the traffic flow fluctuations are amplified since drivers are not able to mantain the optimal velocity. In figure 5.2 we show the behavior of the average speed, flow and density both when a congestion formation is observed and when we have regular traffic during the same hours in the same road. Focusing on the speed behaviour, in the first case people drive with an average speed which ranges from 60 km/h to 85 km/h whereas in the second case a large fluctuation is observed and the average velocity drops down at 10 km/h. Our goal is to analyze the fluctuations in the dynamical quantities that anticipate the congestion formation in order to forecast the occurrence of the congestion itself. The heterogeneous behavior of drivers makes the solution of this problem quite difficult if one uses average dynamical features, so that we decided to apply an AI system.

We first preprocess the data in order to eliminate noisy sample points: since we are interested in understanding high traffic regimes, we focus on the daily hours from 6 a.m. to 10 p.m. avoiding the night period.

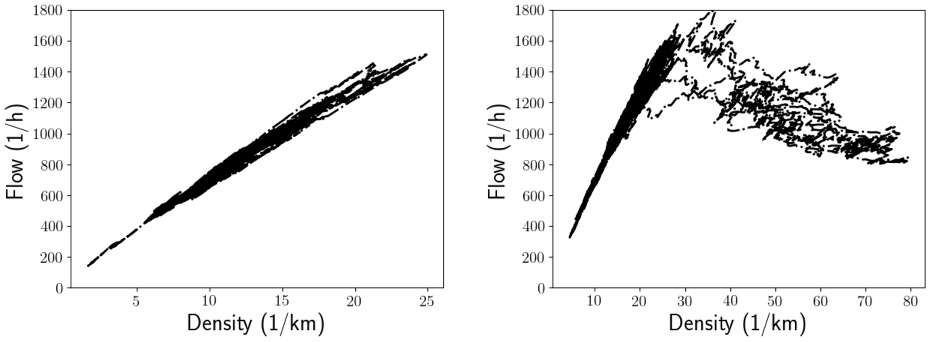
Secondly we transform the original signals in a binary signal keeping the informations coming from the FD. This structure is required by the theoretical model linked to the RFBP [48, 89, 57]. Indeed this deep learning method can be used to derive marginal probabilities on a system within the Bethe-Peierls approximation [94]: it works with binarized input arrays and is able to learn without getting trapped in configurations with low computational performance. The aim of the method is to solve a system of equations that involve quantities called messages: they represent single-variable cavity marginal probabilities. Neurons can interact with neurons of the previous and successive layer through messages going in opposite directions. Each equation of the system gives the expression of one of the messages as a function of its neighbouring messages and the resulting system of equations is then solved iteratively by initializing the messages in some arbitrary configuration. If convergence is achieved, the messages are used to compute the optimal configuration of weights. In order to classify slowdowns we binarize the data as follows: a slowdown occurrence is identified when the road density is greater than the daily average road density and the traffic flow is decreasing so it depends on the critical value of the density of the road. We associate to each day a binary ordered set such that each element corresponds to a car and assumes value $+1$ if the car is involved in a slowdown either -1.

Thirdly we split each daily sequence into subsets of arbitrary length which produce a different time interval of prediction. In our experiments four different length settings were applied: RFBP has been trained several different times using input vectors respectively with length 151, 201, 251 and 301, that corresponds to time intervals of forecasting from 10 to 20 minutes. We label each subset applying a majority rule to the values of the following subset. In figure 5.3 we show the results by processing data associated to the FD shown in figure 5.1 and to the relative graphs of flow and density in figure 5.2. The blue points represents the targets we give to each set of car.

5.4 Results and Performances

We use a cross-validation approach in order to estimate the parameters of the predictive model and to assess how the results of our statistical analysis can be applied to an unknown independent dataset [95]. We define a dataset to test the model during the training phase with the aim to limit overfitting and to understand how the model is generalizable to an indipendent dataset. Each round of cross-validation method partionates a sample of data into complementary subsets performing the analysis on the training set and validating the analysis on the test set.

In particular, to reduce variability, we consider 10 rounds of a 10-fold cross-validation which are performed using different partitions and we combine the validation results over the rounds to estimate the final predictive



(a) 01/05/2011

(b) 02/05/2011

Figure 5.1: Flow $\phi(N)$ versus density $\rho(N)$ daily recorded by a magnetic coil of the MTS system. Left panel: uncongested day, the diagram present a linear behavior. The evolution shows a direct proportionaly between flow and density of the road. The flow does not reach a critical value. Right panel: congested day, the diagram presents a cloud of points stemming from the straight line. The appearence of the fuzzy behavior is a feature of the congested behavior when the uniform stationary state for traffic becomes unstable.

model. Our performances in predicting the congestions are shown in figure 5.4 and 5.5. We remark that there are some cases where our prediction is not correct, but our approach aims to point out the existence of traffic features that may anticipate the congestion formation, which in some cases is not observed since its rising depends on the amplification of local traffic fluctuations. These cases happen when the traffic dynamics in the FD is near the congestion conditions, but finally the density decreases so that the congestion is escaped and the traffic comes to be regular. By the way our system is trained to forecast when the traffic dynamical state enters in the cloud region even if the congestion can be deceived by the decreasing of the density.

To benchmark and validate our results, we compare the prediction performances of our approach with the prediction of 10 different classifiers: KNeighbors (KNN), linear Support Vector Machine (lSVM), RBF Support Vector Machine (SVM), Gaussian Process, Decision Tree (DT), Random Forest (RF), MLP, AdaBoost (AB), Naive Bayes diagQuadratic Discriminant Analysis, Bayesian Quadratic Discriminant Analysis.

As performances estimator we use the area under the receiver operating characteristic curve (AUC), i.e. the trapezoidal approximation of the area under the Receiver Operating Characteristic curve. The AUC corresponds to the probability that the classifier ranks a true instance against a false one.

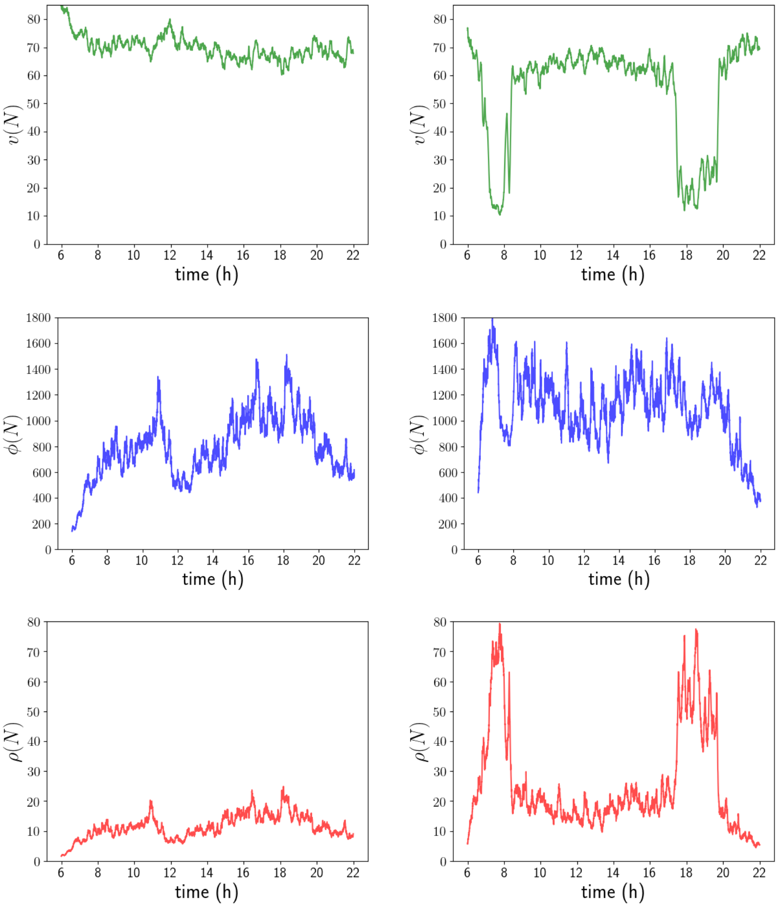


Figure 5.2: Left panel: evolution of speed, flow and density during a non congested day with respect to the FD on the left of figure 5.1 (flow and density are respectively defined by formulas (5.1) and (5.2)). Right panel: congested day with respect to the FD on the right of figure 5.1, density and flow have a different behavior which is due to the fluctuations of the speed. We remark that during time intervals from 7 a.m. to 9 a.m. and from 5 p.m. to 8 p.m density increases and flow decreases, in fact the road is involved in a congestion situation.

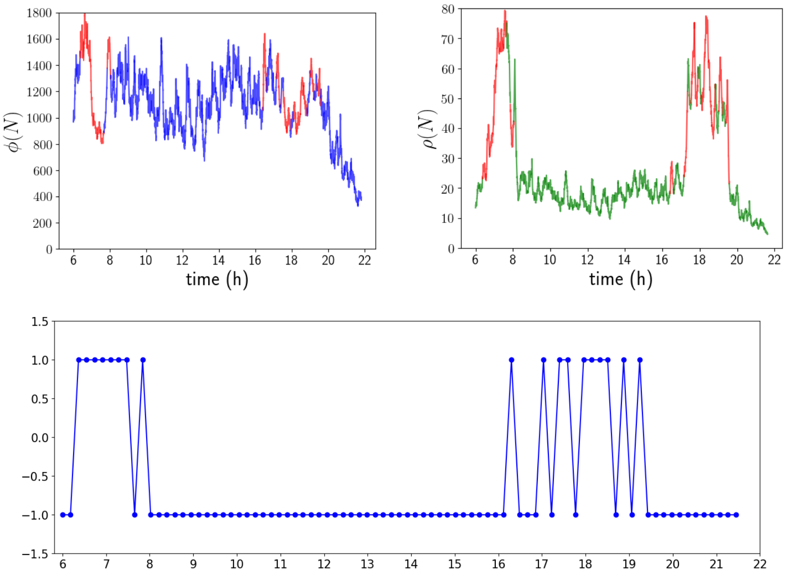


Figure 5.3: Upper panels: figures represent the evolution of flow and density according to figure 5.2. We highlight with red colour the regions where density increases and flow decreases, producing a congestion. Lower pane: According to the critical regions which have been highlighted in the upper panel, the blue points represents the targets we give to each set of car using our binarization method. In particular we assign +1 to the sets of cars which precede a congestion situation either -1.

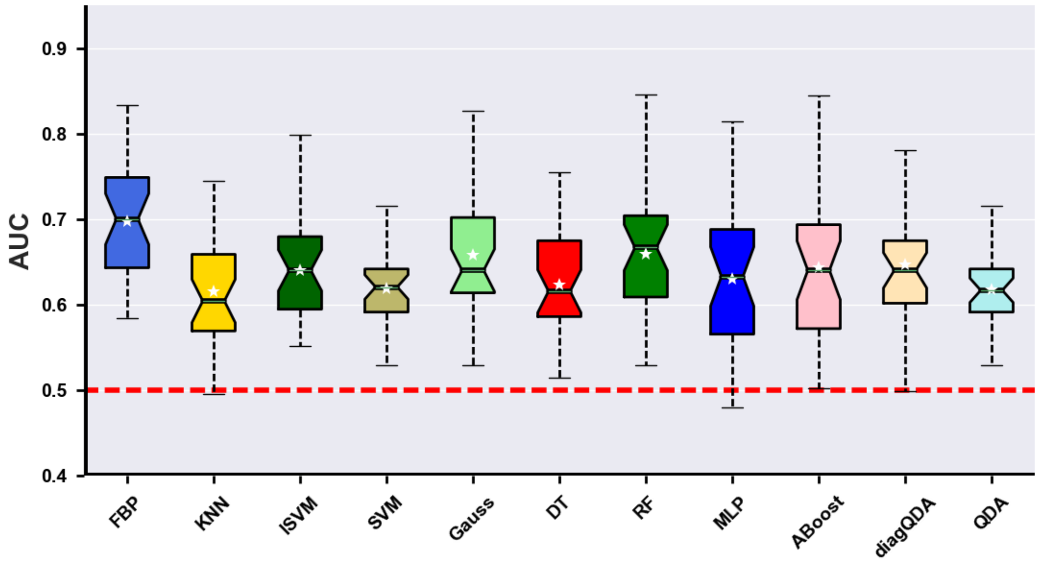


Figure 5.4: Comparison of AUC values. We represents the results with respect to each classifier reported along the x-axis. The first boxplot refers to the RFBP method and presents the highest performance of classification.

We plot in figure 5.4 the comparison of AUC values.

Secondly we analyze the Matthew's Correlation Coefficient (MCC) which provides a balanced measure of the quality of a binary classification taking into account true and false, positive and negatives even if classes have very different sizes. The MCC can take values in the range $[-1, +1]$. The value $+1$ represent a total right classification, the value -1 a complete disagreement of prediction and the value 0 is associated to a random classifier. We represent in figure 5.5 the comparison of MCC values.

5.5 Conclusions

The arise of a congested regime in traffic flow can be understood as the growth of an instability from a uniform stationary traffic flow. The mathematical models based on the optimal velocity concept point out the existence of critical thresholds for the vehicle density after which the congestion instability can develop. However in real situations the traffic along a road is rarely in a stationary dynamical condition and the local density fluctuations also

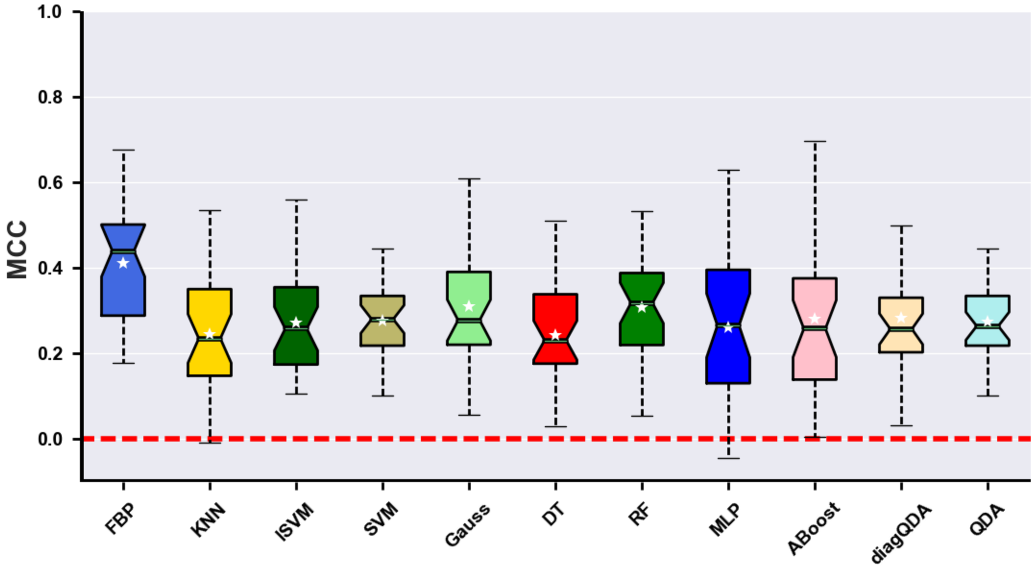


Figure 5.5: Comparison of MCC values. We represents the results with respect to each classifier reported along the x-axis. The first boxplot refers to the RFBP method and presents the highest performance of classification.

depend on the drivers behaviour. Moreover in nonstationary dynamical conditions the time scales for the congestion development play a crucial role. As a consequence, forecasting the arise of congestion becomes a very difficult problem that cannot be coped by simply using the fundamental diagram.

In this chapter we have shown that a deep learning neural network architercture (RFBP) is able to perform a short term forecasting of the traffic congestion from real data collected by magnetic coils installed on a country road network in Italy. Our approach aims to classify the slowdowns traffic dynamics that anticipate the congestion. Starting from vehicle speed and time headways data, we perform a binarization of the slowdown signal suitable to be classified by a deep learning architecture to distinguish between slowdowns that cause a congestion from the slowdowns that are recovered in the succesive evolution. By comparing our results with other classifying methods, it emerges that RFBP has definitely better performances definitely in congestion forecasting from the data. This is probably due to the theoretical structure which supports this algorithm and which leads to find configurations without getting trapped in local minima. The findings of this research coud be relevant in order to mitigate congestion effects and to reduce the traffic costs in extra-urban roads and could be used to give real-time information and suggestions to drivers (i.e. to reduce the traffic fluctuations by uniformizing the velocity). Moreover in the next future the artificial-intelligent approaches to forecast traffic congestion could be implemented in the automatic driving systems.

Appendix A

Monomer-dimer model. Thermodynamic limit of the susceptibility.

In this appendix, using the extended Laplace's method studied in Appendix B, we prove that

$$\lim_{N\to\infty} \frac{\partial}{\partial h}\langle m_N(J,h)\rangle = \frac{\partial}{\partial h} m(J,h).$$

We have used this result in the fourth section.

Theorem A.0.1. *Given an imitative monomer-dimer model defined by a couple of parameters (J,h) over a complete graph of N vertices, it holds:*

$$\lim_{N\to\infty} \frac{\partial}{\partial h}\langle m_N(J,h)\rangle = \frac{\partial}{\partial h} m(J,h). \tag{A.1}$$

Remark A.1. According to results in [21], write the partition function of the monomer-dimer model as

$$Z_N(J,h) = \int_{\mathbb{R}} e^{NF_N(x)} dx,$$

where

$$F_N(x) = -Jx^2 + p_N^{(0)}((2x-1)J+h), \tag{A.2}$$

$$p_N(J,h) = \frac{1}{N}\log\left(\int_{\mathbb{R}} e^{NF_N(x)} dx\right), \tag{A.3}$$

$$p_N^{(0)}(J,h) = \left. p_N(J,h)\right|_{J=0}.$$

Let c_N be the maximum point of the function $F_N(x)$. In order to simplify the notations set $\bar{x} := (2x-1)J+h$ and $\bar{c} := (1-2c)J+h$.

Proof. Let start computing the expectation of the monomer density using the definition of the pressure function given in (A.3):

$$\begin{aligned}\langle m_N(J,h)\rangle =&\frac{\partial p_N(J,h)}{\partial h} = \frac{1}{N}\frac{\partial}{\partial h}\log\left(\int_{\mathbb{R}} e^{NF_N(x)}dx\right) = \\ =&\frac{\displaystyle\int_{\mathbb{R}} e^{NF_N(x)}\frac{\partial}{\partial h}p_N^{(0)}(\bar{x})dx}{\displaystyle\int_{\mathbb{R}} e^{NF_N(x)}dx}.\end{aligned}$$

The finite size susceptibility can be written as:

$$\begin{aligned}&\chi_N(J,h) = \frac{\partial\langle m_N(J,h)\rangle}{\partial h} = \\ &=\frac{\displaystyle\int_{\mathbb{R}} e^{NF_N(x)}\left[N\left(\frac{\partial p_N^{(0)}(\bar{x})}{\partial h}\right)^2 + \frac{\partial^2 p_N^{(0)}(\bar{x})}{\partial h^2}\right]dx}{\displaystyle\int_{\mathbb{R}} e^{NF_N(x)}dx} - N\frac{\left(\displaystyle\int_{\mathbb{R}} e^{NF_N(x)}\frac{\partial}{\partial h}p_N^{(0)}(\bar{x})dx\right)^2}{\left(\displaystyle\int_{\mathbb{R}} e^{NF_N(x)}dx\right)^2} = \\ &=\frac{\displaystyle\int_{\mathbb{R}} e^{NF_N(x)}\frac{\partial^2}{\partial h^2}p_N^{(0)}(\bar{x})dx}{\displaystyle\int_{\mathbb{R}} e^{NF_N(x)}dx} + \qquad (A.4) \\ &\quad + N\left[\frac{\displaystyle\int_{\mathbb{R}} e^{NF_N(x)}\left(\frac{\partial}{\partial h}p_N^{(0)}(\bar{x})\right)^2 dx}{\displaystyle\int_{\mathbb{R}} e^{NF_N(x)}dx} - \left(\frac{\displaystyle\int_{\mathbb{R}} e^{NF_N(x)}\frac{\partial}{\partial h}p_N^{(0)}(\bar{x})dx}{\displaystyle\int_{\mathbb{R}} e^{NF_N(x)}dx}\right)^2\right]. \qquad (A.5)\end{aligned}$$

Now we are going to use the extended Laplace's method in order to evaluate the behavior of (A.4) and (A.5) at the thermodynamic limit.
Observe that, since all the quantities computed above are limited, the second order extended Laplace's method suffices to study the behavior of the finite size susceptibility as $N \to \infty$.

As $N \to \infty$, the numerator of (A.4) can be approximated as:

$$\sqrt{\frac{2\pi}{-NF''(c)}}e^{NF_N(c_N)}\left\{\left.\frac{\partial^2 p^{(0)}(\bar{x})}{\partial h^2}\right|_{\bar{x}=\bar{c}} + \frac{1}{N}\left[-\frac{\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left.\frac{\partial^2 p^{(0)}(\bar{x})}{\partial h^2}\right|_{\bar{x}=\bar{c}}}{2F''(c)} + \right.\right.$$
$$\left.\left. + \frac{\left.\frac{\partial^2 p^{(0)}(\bar{x})}{\partial h^2}\right|_{\bar{x}=\bar{c}} F^{(iv)}(c)}{8(F''(c))^2} + \frac{\frac{\mathrm{d}}{\mathrm{d}x}\left.\frac{\partial^2 p^{(0)}(\bar{x})}{\partial h^2}\right|_{\bar{x}=\bar{c}} F'''(c)}{2(F''(c))^2} - \frac{5\left.\frac{\partial^2 p^{(0)}(\bar{x})}{\partial h^2}\right|_{\bar{x}=\bar{c}} (F'''(c))^2}{24(F''(c))^3}\right]\right\}. \tag{A.6}$$

As $N \to \infty$, the numerator of the first fraction in (A.5) can be approximated as:

$$\sqrt{\frac{2\pi}{-NF''(c)}}e^{NF_N(c_N)}\left\{\left.\left(\frac{\partial p^{(0)}(\bar{x})}{\partial h}\right)^2\right|_{\bar{x}=\bar{c}} + \frac{1}{N}\left[\frac{-\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left.\left(\frac{\partial p^{(0)}(\bar{x})}{\partial h}\right)^2\right|_{\bar{x}=\bar{c}}}{2F''(c)} + \right.\right.$$
$$+ \frac{\left.\left(\frac{\partial p^{(0)}(\bar{x})}{\partial h}\right)^2\right|_{\bar{x}=\bar{c}} F^{(iv)}(c)}{8(F''(c))^2} + \frac{\frac{\mathrm{d}}{\mathrm{d}x}\left.\left(\frac{\partial p^{(0)}(\bar{x})}{\partial h}\right)^2\right|_{\bar{x}=\bar{c}} F'''(c)}{2(F''(c))^2} +$$
$$\left.\left. - \frac{5\left.\left(\frac{\partial p^{(0)}(\bar{x})}{\partial h}\right)^2\right|_{\bar{x}=\bar{c}} (F'''(c))^2}{24(F''(c))^3}\right]\right\} =$$
$$= \sqrt{\frac{2\pi}{-NF''(c)}}e^{NF_N(c_N)}\left[\left.\left(\frac{\partial p^{(0)}(\bar{x})}{\partial h}\right)^2\right|_{\bar{x}=\bar{c}} + \frac{A(J,h)}{N}\right]. \tag{A.7}$$

As $N \to \infty$, the numerator of the second fraction in (A.5) can be approximated as:

$$\left(\sqrt{\frac{2\pi}{-NF''(c)}}e^{NF_N(c_N)}\left\{\left.\frac{\partial p^{(0)}(\bar{x})}{\partial h}\right|_{\bar{x}=\bar{c}}+\frac{1}{N}\left[\frac{-\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left.\frac{\partial p^{(0)}(\bar{c})}{\partial h}\right|_{\bar{x}=\bar{c}}}{2F''(c)}+\right.\right.\right.$$

$$\left.\left.\left.+\frac{\left.\frac{\partial p^{(0)}(\bar{x})}{\partial h}\right|_{\bar{x}=\bar{c}}F^{(iv)}(c)}{8(F''(c))^2}+\frac{\frac{\mathrm{d}}{\mathrm{d}x}\left(\left.\frac{\partial p^{(0)}(\bar{x})}{\partial h}\right|_{\bar{x}=\bar{c}}\right)^2F'''(c)}{2(F''(c))^2}-\frac{5\left.\frac{\partial p^{(0)}(\bar{x})}{\partial h}\right|_{\bar{x}=\bar{c}}(F'''(c))^2}{24(F''(c))^3}\right]\right\}\right)^2=$$

$$=\left(\sqrt{\frac{2\pi}{-NF''(c)}}e^{NF_N(c_N)}\left[\left.\frac{\partial p^{(0)}(\bar{x})}{\partial h}\right|_{\bar{x}=\bar{c}}+\frac{B(J,h)}{N}\right]\right)^2=$$

$$=\left(\sqrt{\frac{2\pi}{-NF''(c)}}e^{NF_N(c_N)}\right)^2\left[\left.\left(\frac{\partial p^{(0)}(\bar{x})}{\partial h}\right)^2\right|_{\bar{x}=\bar{c}}+2\left.\frac{\partial p^{(0)}(\bar{x})}{\partial h}\right|_{\bar{x}=\bar{c}}\frac{B(J,h)}{N}+\frac{B^2(J,h)}{N^2}\right]. \tag{A.8}$$

As $N\to\infty$, the integral $\int_{\mathbb{R}}e^{NF_N(x)}dx$ can be approximated as:

$$\sqrt{\frac{2\pi}{-NF''(c)}}e^{NF_N(c_N)}\left\{1+\frac{1}{N}\left[\frac{F^{(iv)}(c)}{8(F''(c))^2}-\frac{5(F'''(c))^2}{24(F''(c))^3}\right]\right\}=$$

$$=\sqrt{\frac{2\pi}{-NF''(c)}}e^{NF_N(c_N)}\left[1+\frac{C(J,h)}{N}\right]. \tag{A.9}$$

Putting together (A.6) and (A.9) we obtain:

$$\text{(A.4)}\xrightarrow{N\to\infty}g'(\bar{x}). \tag{A.10}$$

Putting together (A.7),(A.8) and (A.9), we obtain:

$$\text{(A.5)}\xrightarrow{N\to\infty}\frac{-8J^2(g'(\bar{c}))^2}{2(-2J+4J^2g'(\bar{c}))}. \tag{A.11}$$

Using (A.10) and (A.11), we find that as $N\to\infty$

$$\chi_N(J,h)\xrightarrow{N\to\infty}g'+\frac{8J^2(g')^2}{2(-2J+4J^2g')}=\frac{4Jg'(1-2Jg')+8J^2(g')^2}{4J(1-2Jg')}=\frac{(g')^2}{1-2Jg'}. \tag{A.12}$$

At the thermodynamic limit, the susceptibility is the partial derivative of the solution $m(J,h)$ of the consistency equation with respect to the parameter h, so that:

$$\begin{aligned}\chi &= \frac{\partial m(J,h)}{\partial h} = \frac{\mathrm{d}}{\mathrm{d}h} g((2m-1)J+h)\left(1+2\frac{\partial m(J,h)}{\partial h}J\right) = \\ &= g'((2m-1)J+h)(1+2\chi J) \\ \Rightarrow \chi &= \frac{(g')^2}{1-2Jg'}.\end{aligned}$$

Hence, (A.1) is proved. □

Appendix B

Extended Laplace's method. Control at the second order.

The usual Laplace method works with integrals of the form

$$\int_{\mathbb{R}} (\psi(x))^n u(x) dx$$

as $n \to \infty$. In this appendix we prove an extension at the second order of the previous method when the functions ψ and u may depend on n (see [21] for the control at first order). We have used that in Appendix A.

Theorem B.0.1. *For all $n \in \mathbb{N}$, let $\psi_n : \mathbb{R} \to \overline{\mathbb{R}}$ and $u_n : \mathbb{R} \to \overline{\mathbb{R}}$. Suppose that there exists a compact interval $K \subset \mathbb{R}$ such that $\psi_n, u_n > 0$ on K, so that in particular*

$$\psi_n(x) = e^{f_n(x)} \quad \forall x \in K.$$

Suppose that $f_n \in C^4(K)$ and that $u_n \in C^2(K)$.
Moreover suppose that

1. $f_n \xrightarrow{n\to\infty} f$ *uniformly on K with its derivatives;*
2. $u_n \xrightarrow{n\to\infty} u$ *uniformly on K with its derivatives;*
3. *there exixts a positive constant $c_1 < \infty$ such that $|u_n| < c_1$;*
4. $\max_K f_n$ *is attained in a unique point $c_n \in int(K)$;*
5. $\max_K f$ *is attained in a unique point $c \in int(K)$;*
6. $\limsup_{n\to\infty} \left(\sup_{\mathbb{R}\setminus K} \log|\psi_n| - \max_K f_n \right) < 0;$

7. $f''(c) < 0$*;*

8. $\limsup\limits_{n\to\infty} \int_{\mathbb{R}} |\psi_n(x)|dx < \infty$.

Then, as $n \to \infty$*,*

$$\int_{\mathbb{R}} (\psi_n(x))^n u_n(x)dx = \sqrt{\frac{2\pi}{-nf''(c)}} e^{nf_n(c_n)} \left\{ u(c) + \frac{\Lambda}{n} + o\left(\frac{1}{n}\right) \right\}, \qquad \text{(B.1)}$$

where

$$\Lambda = -\frac{u''(c)}{2f''(c)} + \frac{u(c)f^{(iv)}(c)}{8(f''(c))^2} + \frac{u'(c)f'''(c)}{2(f''(c))^2} - \frac{5u(c)(f'''(c))^2}{24(f''(c))^3}.$$

In the proof we use the following elementary fact:

Lemma B.0.2. *Let* $(f_n)_{n\in\mathbb{N}}$ *be a sequence of continuous functions uniformly convergent to* f *on a compact set* K*. Let* $(I_n)_{n\in\mathbb{N}}$ *and* I *be subsets of* K *such that*

$$\max_{x\in I_n, y\in I} dist(x,y) \to 0, \quad as \quad n \to \infty.$$

Then

a) $\max\limits_{I_n} f_n \xrightarrow{n\to\infty} \max\limits_{I} f$

b) arg $\max\limits_{I_n} f_n \xrightarrow{n\to\infty}$ *arg* $\max\limits_{I} f$*, provided that* f *has a unique global maximum point on* I*.*

We proceed with the proof of the theorem.

Proof. Since c_n is an internal point of maximum of f_n (*hypothesis 4*), $f_n'(c_n) = 0$. Moreover $\forall x \in K$

$$f_n(x) = f_n(c_n) + \frac{1}{2}f_n''(c_n)(x-c_n)^2 + \frac{1}{6}f_n'''(c_n)(x-c_n)^3 + \frac{1}{24}f_n^{(iv)}(\xi'_{x,n})(x-c_n)^4, \qquad \text{(B.2)}$$

with $\xi'_{x,n} \in (c_n, x) \subset K$, and

$$u_n(x) = u_n(c_n) + u_n'(c_n)(x-c_n) + \frac{1}{2}u_n''(\xi''_{x,n})(x-c_n)^2, \qquad \text{(B.3)}$$

with $\xi''_{x,n} \in (c_n, x) \subset K$.

Fix $\frac{\epsilon}{2} = \bar{\epsilon}$ and N_ϵ such that $|f_n^{(i)}(\xi) - f^{(i)}(\xi)| < \bar{\epsilon}, i = 1,2,3,4$ and $|u_n^{(j)}(\xi) -$

$u^{(j)}(\xi)| < \overline{\epsilon}, j = 1, 2 \; \forall \xi \in K$ and $\forall n > N_\epsilon$. Since f and u and their respective derivatives are continuous in c, there exists $\overline{\delta}_\epsilon > 0$ such that $B(c, \overline{\delta}_\epsilon) \subset K$ and $\forall \xi : |\xi - c| < \overline{\delta}_\epsilon$

$$|f^{(i)}(\xi) - f^{(i)}(c)| < \frac{\epsilon}{2}, i = 1, 2, 3, 4,$$

and

$$|u^{(j)}(\xi) - u^{(j)}(c)| < \frac{\epsilon}{2}, j = 1, 2.$$

By lemma B.0.2, $c_n \xrightarrow{n \to \infty} c$ because c is the unique maximum point of f on K(*hypothesis 5*). Thus there exists $\overline{N}_{\delta_\epsilon}$ such that

$$|c_n - c| < \delta_\epsilon = \frac{\overline{\delta}_\epsilon}{3} \quad \forall n > \overline{N}_{\delta_\epsilon}. \tag{B.4}$$

Observe that by *hypothesis 7* and for $n > N_\epsilon \vee \overline{N}_{\delta_\epsilon}$, $f_n''(c_n) < 0$.
Moreover, for $n > N_\epsilon \vee \overline{N}_{\delta_\epsilon}$, $\forall x \in B(c, \delta_\epsilon) \subset B(c, \overline{\delta}_\epsilon)$ and $\forall \xi_{x,n} \in (c_n, x) \subset K$, it holds:

$$|\xi_{x,n} - c| \leq |\xi_{x,n} - x| + |x - c| \leq |c_n - x| + |x - c| \leq |c_n - c| + |c - x| + |x - c| < 3\delta_\epsilon = \overline{\delta}_\epsilon \Rightarrow$$

$$\begin{cases} |f_n^{(i)}(\xi'_{x,n}) - f^{(i)}(c)| \leq |f_n^{(i)}(\xi'_{x,n}) - f^{(i)}(\xi'_{x,n})| + |f^{(i)}(\xi'_{x,n}) - f^{(i)}(c)| < \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon \\ |u_n^{(j)}(\xi''_{x,n}) - u^{(j)}(c)| \leq |u_n^{(j)}(\xi''_{x,n}) - u^{(j)}(\xi''_{x,n})| + |u^{(i)}(\xi''_{x,n}) - u^{(i)}(c)| < \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon \end{cases} \tag{B.5}$$

By substituing (B.5) in (B.2) and in (B.3), we obtain that for $n > N_\epsilon \vee \overline{N}_{\delta_\epsilon}$ and $x \in B(c, \delta_\epsilon)$

$$f_n(x) \begin{cases} \leq & f_n(c_n) + \frac{1}{2} f_n''(c_n)(x - c_n)^2 + \frac{1}{6} f_n'''(c_n)(x - c_n)^3 + \\ & + \frac{1}{24}(f^{(iv)}(c) + \epsilon)(x - c_n)^4 \\ \geq & f_n(c_n) + \frac{1}{2} f_n''(c_n)(x - c_n)^2 + \frac{1}{6} f_n'''(c_n)(x - c_n)^3 + \\ & + \frac{1}{24}(f^{(iv)}(c) - \epsilon)(x - c_n)^4 \end{cases} \tag{B.6}$$

and

$$u_n(x) \begin{cases} \leq & u_n(c_n) + u_n'(c_n)(x - c_n) + \frac{1}{2}(u''(c) + \epsilon)(x - c_n)^2 \\ \geq & u_n(c_n) + u_n'(c_n)(x - c_n) + \frac{1}{2}(u''(c) - \epsilon)(x - c_n)^2 \end{cases} . \tag{B.7}$$

Now split the integral into two parts:

$$\int_{\mathbb{R}} e^{nf_n(x)} u_n(x)dx = \int_{\mathbb{R}\setminus B(c_n,\delta_\epsilon)} (\psi_n(x))^n u_n(x)dx + \int_{B(c_n,\delta_\epsilon)} e^{nf_n(x)} u_n(x)dx. \tag{B.8}$$

To control the first integral on the r.h.s. of (B.8) we claim that there exists $\eta_{\delta_\epsilon} > 0$ and $\hat{N}_{\delta_\epsilon}$ such that

$$\log|\psi_n(x)| < f_n(c_n) - \eta_{\delta_\epsilon} \quad \forall x \in \mathbb{R}\setminus B(c_n,\delta_\epsilon) \quad \forall n > \hat{N}_{\delta_\epsilon}; \tag{B.9}$$

this implies that

$$\limsup_{n\to\infty} \sup_{x\in\mathbb{R}\setminus B(c_n,\delta_\epsilon)} (\log|\psi_n(x)| - f_n(c_n)) < 0.$$

Indeed, using lemma B.0.2:

$$\begin{gathered}\limsup_{n\to\infty} \sup_{\mathbb{R}\setminus B(c_n,\delta_\epsilon)} (\log|\psi_n(x)| - f_n(c_n)) = \\ \left(\limsup_{n\to\infty} \sup_{x\in K\setminus B(c_n,\delta_\epsilon)} (f_n(x) - f_n(c_n))\right) \vee \left(\limsup_{n\to\infty} \sup_{x\in\mathbb{R}\setminus K} (\log|\psi_n(x)| - f_n(c_n)\right) = \\ \left(\sup_{x\in K\setminus B(c_n,\delta_\epsilon)} (f(x) - f(c))\right) \vee \left(\limsup_{n\to\infty} \sup_{x\in\mathbb{R}\setminus K} (\log|\psi_n(x)| - f_n(c_n))\right).\end{gathered}$$

Moreover, since c is the unique maximum point of the continuous function f on the compact set K,

$$\sup_{x\in K\setminus B(c_n,\delta_\epsilon)} (f(x) - f(c)) < 0$$

and this proves the claim.
Now using (B.9) and *hypothesis 8* we can say that there exist C_1 and N such that for all $n > N \vee \hat{N}_{\delta_\epsilon}$

$$\begin{aligned}\int_{\mathbb{R}\setminus B(c_n,\delta_\epsilon)} e^{nf_n(x)} u_n(x)dx &\le e^{(n-1)(f_n(c_n)-\eta\delta_\epsilon)} \int_{\mathbb{R}} |\psi_n(x)||u_n(x)|dx \le \\ &\le e^{(n-1)(f_n(c_n)-\eta\delta_\epsilon)} \int_{\mathbb{R}} |\psi_n(x)|c_1 dx \le \\ &\le C_1 e^{n(f_n(c_n)-\eta\delta_\epsilon)}. \end{aligned} \tag{B.10}$$

In order to find an upper bound for the second integral of the r.h.s. of (B.8), we proceed as follows:

$$\begin{aligned}
&\int_{B(c_n,\delta_\epsilon)} e^{nf_n(x)}u_n(x)dx \leq \\
\leq &\int_{B(c_n,\delta_\epsilon)} e^{n\left(f_n(c_n)+\frac{1}{2}f''_n(c_n)(x-c_n)^2+\frac{1}{6}f'''_n(c_n)(x-c_n)^3+\frac{1}{24}(f^{(iv)}(c)+\epsilon)(x-c_n)^4\right)}u_n(x)dx = \\
= &\int_{B(c_n,\delta_\epsilon)} e^{nf_n(c_n)+\frac{n}{2}f''_n(c_n)(x-c_n)^2}e^{n\left(\frac{1}{6}f'''_n(c_n)(x-c_n)^3+\frac{1}{24}(f^{(iv)}(c)+\epsilon)(x-c_n)^4\right)}u_n(x)dx.
\end{aligned} \tag{B.11}$$

Since δ_ϵ may be chosen small, the second exponential term can be expanded as

$$\begin{aligned}
\exp&\left[n\left(\frac{1}{6}f'''_n(c_n)(x-c_n)^3+\frac{1}{24}(f^{(iv)}(c)+\epsilon)(x-c_n)^4\right)\right] \leq \\
\leq& 1+n\left[\frac{1}{6}f'''_n(c_n)(x-c_n)^3+\frac{1}{24}(f^{(iv)}(c)+\epsilon)(x-c_n)^4\right]+ \\
+&\frac{n^2}{72}(f'''_n(c_n))^2(x-c_n)^6+n^2C_2|x-c_n|^7,
\end{aligned} \tag{B.12}$$

where C_2 is a positive real constant. Substitute (B.12) and (B.7) in (B.11). Collecting powers of $(x-c_n)$ and observing that odd powers don't contribute to the integral, we claim that:

$$\begin{aligned}
\text{(B.11)} \leq & e^{nf_n(c_n)}\int_{B(c_n,\delta_\epsilon)} e^{\frac{n}{2}f''_n(c_n)(x-c_n)^2}\left[u_n(c_n)+(x-c_n)^2\frac{u''(c)+\epsilon}{2}+\right. \\
&+(x-c_n)^4\left(n\frac{u'_n(c_n)f'''_n(c_n)}{6}+n\frac{u_n(c_n)(f^{(iv)}(c)+\epsilon))}{24}\right)+ \\
&+(x-c_n)^6\left(n^2\frac{u_n(c_n)(f'''_n(c_n))^2}{72}+n\frac{(u''(c)+\epsilon)(f^{(iv)}(c)+\epsilon)}{48}\right)+ \\
&\left.+(x-c_n)^8n^2\left(\frac{(u''(c)+\epsilon)(f'''_n(c_n))^2}{144}+u'_n(c_n)C_2\right)\right]dx+ \\
+&e^{nf_n(c_n)}\int_{B(c_n,\delta_\epsilon)} e^{\frac{n}{2}f''_n(c_n)(x-c_n)^2}n^2C_2|x-c_n|^7dx.
\end{aligned}$$

Making the change of variable

$$t=\sqrt{-nf''_n(c_n)}(x-c_n),$$

we obtain:

$$
\begin{aligned}
\text{(B.11)} \leq & \frac{e^{nf_n(c_n)}}{\sqrt{-nf''_n(c_n)}} \int_{B(0,\sqrt{-nf''_n(c_n)}\delta_\epsilon)} e^{-\frac{t^2}{2}} \left\{ u_n(c_n) + \frac{1}{n} \left[-t^2 \frac{u''(c)+\epsilon}{2f''_n(c_n)} + \right.\right. \\
& \left. + t^4 \left(\frac{u_n(c_n)(f^{(iv)}(c)+\epsilon)}{24(f''_n(c_n))^2} + \frac{u'_n(c_n)f'''_n(c_n)}{6(f''_n(c_n))^2} \right) - t^6 \frac{u_n(c_n)(f'''_n(c_n))^2}{72(f''_n(c_n))^3} \right] + \\
& \left.+ \frac{1}{n^2} \left[-t^6 \frac{(u''(c)+\epsilon)(f^{(iv)}(c)+\epsilon)}{48(f''_n(c_n))^3} + t^8 \left(\frac{(u''(c)+\epsilon)(f'''_n(c_n))^2}{144(f''_n(c_n))^4} + \frac{u'_n(c_n)C_2}{(f''_n(c_n))^4} + \right) \right] \right\} dt \\
& + 2\frac{e^{nf_n(c_n)}}{\sqrt{-nf''_n(c_n)}} \int_{t\in B(0,\sqrt{-nf''_n(c_n)}\delta_\epsilon):t\geq 0} e^{-\frac{t^2}{2}} t^7 \frac{C_2}{n^{3/2}(f''_n(c_n))^{7/2}} dt = \\
= & \frac{e^{nf_n(c_n)}}{\sqrt{-n(f''_n(c_n))}} \int_{B(0,\sqrt{-nf''_n(c_n)}\delta_\epsilon)} e^{-\frac{t^2}{2}} \left\{ u_n(c_n) + \frac{a^{(1)}_{n,\epsilon}(t)}{n} + \frac{b^{(1)}_{n,\epsilon}(t)}{n^2} \right\} dt + \\
& + 2\frac{C_2 e^{nf_n(c_n)}}{n^2(f''_n(c_n))^4} \int_{t\in B(0,\sqrt{-n(f''_n(c_n))}\delta_\epsilon):t\geq 0} e^{-\frac{t^2}{2}} t^7 dt, \qquad \text{(B.13)}
\end{aligned}
$$

where $a^{(1)}_{n,\epsilon}(t)$ and $b^{(1)}_{n,\epsilon}(t)$ are the arguments inside square brackets which are respectively multiplied by $\frac{1}{n}$ and by $\frac{1}{n^2}$.

In order to find a lower bound for the second integral of the r.h.s. of (B.8), we proceed as follows:

$$
\begin{aligned}
& \int_{\mathbb{R}} (\psi_n(x))^n u_n(x) dx \geq \\
\geq & \int_{B(c_n,\delta_\epsilon)} e^{nf_n(c_n)+\frac{n}{2}f''_n(c_n)(x-c_n)^2} e^{n\left(\frac{1}{6}f'''_n(c_n)(x-c_n)^3+\frac{1}{24}(f^{(iv)}(c)-\epsilon)(x-c_n)^4\right)} u_n(x) dx. \qquad \text{(B.14)}
\end{aligned}
$$

Since δ_ϵ may be chosen small, the second exponential term satisfies

$$
\begin{aligned}
& \exp\left[n \left(\frac{1}{6} f'''_n(c_n)(x-c_n)^3 + \frac{1}{24}(f^{(iv)}(c)-\epsilon)(x-c_n)^4 \right) \right] \geq \\
\geq & 1 + n \left[\frac{1}{6} f'''_n(c_n)(x-c_n)^3 + \frac{1}{24}(f^{(iv)}(c)-\epsilon)(x-c_n)^4 \right] + \\
& + \frac{n^2}{72}(f'''_n(c_n))^2(x-c_n)^6 - n^2 C_3 |x-c_n|^7, \qquad \text{(B.15)}
\end{aligned}
$$

where C_3 is a positive real constant.
Analogously as above, expand the second exponential term of (B.14) as in (B.12) and the function $u_n(x)$ as in (B.3). Collecting powers of $(x-c_n)$ and

making the change of variable

$$t = \sqrt{-nf_n''(c_n)}(x - c_n),$$

we obtain:

$$\begin{aligned}
\text{(B.14)} \geq & \frac{e^{nf_n(c_n)}}{\sqrt{-nf_n''(c_n)}} \int_{B(0,\sqrt{-nf_n''(c_n)}\delta_\epsilon)} e^{-\frac{t^2}{2}} \left\{ u_n(c_n) + \frac{1}{n} \left[-t^2 \frac{u''(c) - \epsilon}{2f_n''(c_n)} + \right. \right. \\
& \left. + t^4 \left(\frac{u_n(c_n)(f^{(iv)}(c) - \epsilon)}{24(f_n''(c_n))^2} + \frac{u_n'(c_n) f_n'''(c_n)}{6(f_n''(c_n))^2} \right) - t^6 \frac{u_n(c_n)(f_n'''(c_n))^2}{72(f_n''(c_n))^3} \right] + \\
& \left. + \frac{1}{n^2} \left[-t^6 \frac{(u''(c) - \epsilon)(f^{(iv)}(c) - \epsilon)}{48(f_n''(c_n))^3} + t^8 \left(\frac{(u''(c) - \epsilon)(f_n'''(c_n))^2)}{144(f_n''(c_n))^4} - \frac{u_n'(c_n) C_3}{(f_n''(c_n))^4} \right) \right] \right\} dt + \\
& - 2 \frac{e^{nf_n(c_n)}}{\sqrt{-nf_n''(c_n)}} \int_{t \in B(0,\sqrt{-nf_n''(c_n)}\delta_\epsilon) : t \geq 0} e^{-\frac{t^2}{2}} t^7 \frac{C_3}{n^{3/2}(f_n''(c_n))^{7/2}} dt = \\
= & \frac{e^{nf_n(c_n)}}{\sqrt{-nf_n''(c_n)}} \int_{B(0,\sqrt{-nf_n''(c_n)}\delta_\epsilon)} e^{-\frac{t^2}{2}} \left\{ u_n(c_n) + \frac{a_{n,\epsilon}^{(2)}(t)}{n} + \frac{b_{n,\epsilon}^{(2)}(t)}{n^2} \right\} dt + \\
& - 2 \frac{C_3 e^{nf_n(c_n)}}{n^2 (f_n''(c_n))^4} \int_{t \in B(0,\sqrt{-n(f_n''(c_n))}\delta_\epsilon) : t \geq 0} e^{-\frac{t^2}{2}} t^7 dt, \qquad \text{(B.16)}
\end{aligned}$$

where $a_{n,\epsilon}^{(2)}(t)$ and $b_{n,\epsilon}^{(2)}(t)$ are the arguments inside square brackets which are respectively multiplied by $\frac{1}{n}$ and by $\frac{1}{n^2}$.

It is easy to verify that:

$$\begin{aligned}
& \int_{B(0,\sqrt{n}\delta_\epsilon)} e^{-\frac{t^2}{2}} u_n(c_n) dt \xrightarrow{n \to \infty} \sqrt{2\pi} u(c), \\
& \int_{B(0,\sqrt{n}\delta_\epsilon)} e^{-\frac{t^2}{2}} t^{2k} dt \xrightarrow{n \to \infty} \int_{\mathbb{R}} e^{-\frac{t^2}{2}} t^{2k} dt = \sqrt{2\pi}(2k-1)(2k-3)\dots(3)(1), \quad \forall k \in \mathbb{N}, \\
& \int_{t \in B(0,\sqrt{n}\delta_\epsilon) : t \geq 0} e^{-\frac{t^2}{2}} t^7 dt \xrightarrow{n \to \infty} \int_0^{+\infty} e^{-\frac{t^2}{2}} t^7 dt = 48. \qquad \text{(B.17)}
\end{aligned}$$

In conclusion, using (B.8),(B.10),(B.13), (B.16) and (B.17), we obtain that for $\epsilon \in (0, \epsilon_0]$ and $n > N \vee N_\epsilon \vee \overline{N}_{\delta_\epsilon} \vee \hat{N}_{\delta_\epsilon}$

$$\frac{\displaystyle\int_{\mathbb{R}} (\psi_n(x))^n u_n(x) dx - \sqrt{\frac{2\pi}{-nf''(c)}} e^{nf_n(c_n)} u(c)}{\displaystyle\sqrt{\frac{2\pi}{-nf''(c)}} e^{nf_n(c_n)} \frac{\Lambda}{n}} \leq$$

$$\leq \frac{\dfrac{\displaystyle\int_{B(0,\sqrt{-nf''_n(c_n)}\delta_\epsilon)} e^{-\frac{t^2}{2}} \left\{ u_n(c_n) + \frac{a^{(1)}_{n,\epsilon}(t)}{n} + \frac{b^{(1)}_{n,\epsilon}(t)}{n^2} \right\} dt}{\sqrt{-f''_n(c_n)}} - \sqrt{\dfrac{2\pi}{-nf''(c)}} u(c)}{\displaystyle\sqrt{\frac{2\pi}{-f''(c)}} \frac{\Lambda}{n}} +$$

$$+ \frac{\displaystyle 2 \frac{C_2}{n^2 (f''_n(c_n))^5} \int_{t \in B(0,\sqrt{-nf''_n(c_n)}\delta_\epsilon) : t \geq 0} e^{-\frac{t^2}{2}} t^7 dt + \frac{C_1}{e^{n\eta\delta_\epsilon}}}{\displaystyle\sqrt{\frac{2\pi}{-f''(c)}} \frac{\Lambda}{n}} \xrightarrow{n\to\infty}$$

$$\xrightarrow{n\to\infty} \frac{\sqrt{\dfrac{2\pi}{-f''_n(c_n)}}}{\sqrt{\dfrac{2\pi}{-f''(c)}}} \xrightarrow{\epsilon\to 0} 1 \tag{B.18}$$

and

$$\frac{\displaystyle\int_{\mathbb{R}} (\psi_n(x))^n u_n(x)dx - \sqrt{\frac{2\pi}{-nf''(c)}}e^{nf_n(c_n)}u(c)}{\displaystyle\sqrt{\frac{2\pi}{-nf''(c)}}e^{nf_n(c_n)}\frac{\Lambda}{n}} \geq$$

$$\geq \frac{\dfrac{\displaystyle\int_{B(0,\sqrt{nf_n''(c_n)}\delta_\epsilon)} e^{-\frac{t^2}{2}}\left\{u_n(c_n) + \frac{a^{(2)}_{n,\epsilon}(t)}{n} + \frac{b^{(2)}_{n,\epsilon}(t)}{n^2}\right\}dt}{\sqrt{-f_n''(c_n)}} - \sqrt{\dfrac{2\pi}{-nf''(c)}}u(c)}{\displaystyle\sqrt{\frac{2\pi}{-f''(c)}}\frac{\Lambda}{n}} +$$

$$- \frac{\displaystyle 2\frac{C_3}{n^2(f_n''(c_n))^5}\int_{t\in B(0,\sqrt{-nf_n''(c_n)}\delta_\epsilon):t\geq 0} e^{-\frac{t^2}{2}}t^7dt + \frac{C_1}{e^{n\eta\delta_\epsilon}}}{\displaystyle\sqrt{\frac{2\pi}{-f''(c)}}\frac{\Lambda}{n}} \xrightarrow{n\to\infty}$$

$$\xrightarrow{n\to\infty} \frac{\sqrt{\dfrac{2\pi}{-f_n''(c_n)}}}{\sqrt{\dfrac{2\pi}{-f''(c)}}} \xrightarrow{\epsilon\to 0} 1. \tag{B.19}$$

Hence, by (B.18) and (B.19), (B.1) is proved. □

Appendix C

Replica Theory

In this chapter we present an introduction to replica theory which has been widely studied from G. Parisi in [96]. We derive the replica theory equations from the SK model since many other statistical models can be reduced to this one by simple assumptions on the parameters.

The SK model is dxefined by the following Hamiltonian

$$H(\sigma) = -\sum_{\langle i,j \rangle}^{N} J_{ij}\sigma_i\sigma_j - h\sum_{i}^{N}\sigma_i, \tag{C.1}$$

where $\sigma_i \in \{-1, 1\}$, h is the uniform magnetic field and the couplings are random variables extracted from the distribution

$$P(J) = \sqrt{\frac{N}{2\pi}}\exp\left(-\frac{NJ^2}{2}\right). \tag{C.2}$$

Because of the randomicity of the interaction, thermodynamic properties must be computed by fixing an instance of the disorder. Such property is called *quenched disorder* and was widely studied by Parisi in [97]. When we are dealing with the thermodynamic limit, thanks to the *self-averaging* properties, the free energy of a specific realization of the disorder can be computed by the average over such realization of the J-dependent free energy, which means that

$$\lim_{N\to\infty} f_J^{(N)} = \lim_{N\to\infty} \overline{f_J^{(N)}}, \tag{C.3}$$

where $\overline{f_J^{(N)}}$ is the average over the disorder of the $J-$dependent free energy. Relation (C.3) is satisfied with probability one since the set of sequences of $J's$ for which the previous result doesn't hold have zero measure [97]. Thus the solution of the statics of a system under the presence of the quenched disorder requires the computation of the logarithm of the partition function

$$Z_J = \sum_{\{\sigma\}} \exp\left(-\beta H(\sigma)\right) = \sum_{\{\sigma\}} \exp\left(\beta \sum_{\langle i,j\rangle}^{N} J_{ij}\sigma_i\sigma_j + \beta h \sum_{i}^{N} \sigma_i\right) \tag{C.4}$$

which cannot be done analytically: the *replica trick* is used to overcome this difficulty. The idea is based on the following identity:

$$\lim_{n\to 0} \frac{\overline{Z^n} - 1}{n} = \overline{\log Z} \tag{C.5}$$

We introduce n non interacting replicas of the system of N interacting spins living in the same realization of the disorder. The n-th power of the partition can be written as follows:

$$\begin{aligned} Z_J^n &= \left(\sum_{\{\sigma\}} \exp\left(\beta \sum_{\langle i,j\rangle}^{N} J_{ij}\sigma_i\sigma_j + \beta h \sum_{i}^{N} \sigma_i\right)\right)^n = \\ &= \prod_{a=1}^{n} \sum_{\{\sigma^a\}} \exp\left(\beta \sum_{\langle i,j\rangle}^{N} J_{ij}\sigma_i^a\sigma_j^a + \beta h \sum_{i}^{N} \sigma_i^a\right) = \\ &= \sum_{\{\sigma\}} \exp\left(\beta \sum_{a=1}^{n}\sum_{\langle i,j\rangle}^{N} J_{ij}\sigma_i^a\sigma_j^a + \beta h \sum_{a=1}^{n}\sum_{i}^{N} \sigma_i^a\right), \end{aligned} \tag{C.6}$$

so that its average with respect to the couplings can be computed as:

$$\overline{Z_J^n} = \int \left(\prod_{i<j} P(J_{ij}) dJ_{ij}\right) \sum_{\{\sigma\}} \exp\left(\beta \sum_{a=1}^{n}\sum_{\langle i,j\rangle}^{N} J_{ij}\sigma_i^a\sigma_j^a + \beta h \sum_{a=1}^{n}\sum_{i}^{N} \sigma_i^a\right). \tag{C.7}$$

In order to compute the replicated partition function we have to keep on mind the following identities:

$$\int \exp(-Ax^2 + Bx) dx = \sqrt{\frac{\pi}{A}} \exp\left(\frac{B^2}{4A}\right) \tag{C.8}$$

$$\int \exp\left(-\sum_{i,k=1}^{M} A_{ik}x_ix_k + \sum_{i=1}^{M} B_i x_i\right) d^M x = \sqrt{\frac{\pi}{\det(A)}} \exp\left(\frac{1}{4}\sum_{i,k=1}^{M} (\hat{A}^{-1})_{ik} B_i B_k\right). \tag{C.9}$$

We can get an expression of the replicated partition function starting from the equation (C.7):

$$\overline{Z_J^n} = \int \left(\prod_{i<j} P(J_{ij}) dJ_{ij} \right) \sum_{\{\sigma\}} \exp\left(\beta \sum_{a=1}^{n} \sum_{\langle i,j \rangle}^{N} J_{ij} \sigma_i^a \sigma_j^a \right) \exp\left(\beta h \sum_{a=1}^{n} \sum_{i}^{N} \sigma_i^a \right) =$$

$$= \sum_{\{\sigma\}} \exp\left(\beta h \sum_{a=1}^{n} \sum_{i}^{N} \sigma_i^a \right) \int \left(\prod_{i<j} \sqrt{\frac{N}{2\pi}} e^{-\frac{NJ_{ij}^2}{2}} dJ_{ij} \right) \exp\left(\beta \sum_{a=1}^{n} \sum_{\langle i,j \rangle}^{N} J_{ij} \sigma_i^a \sigma_j^a \right) =$$

$$= \sum_{\{\sigma\}} \exp\left(\beta h \sum_{a=1}^{n} \sum_{i}^{N} \sigma_i^a \right) \sqrt{\frac{N}{2\pi}} \int \left(\prod_{i<j} e^{(\beta \sum_{a=1}^{n} \sum_{\langle i,j \rangle}^{N} J_{ij} \sigma_i^a \sigma_j^a - \frac{NJ_{ij}^2}{2}} dJ_{ij} \right).$$

By applying identities (C.8) and (C.9), the integral in latter equation can be written as

$$\sqrt{\frac{2\pi}{N}} \frac{\beta^2}{2N} \exp\left(\sum_{i<j} \sum_{a,b} \sigma_i^a \sigma_i^b \sigma_j^a \sigma_j^b \right) \tag{C.10}$$

so that (C.7) reads:

$$\overline{Z_J^n} = \sum_{\{\overline{\sigma}\}} \exp\left(\beta h \sum_{a=1}^{n} \sum_{i}^{N} \sigma_i^a + \frac{\beta^2}{2N} \sum_{i<j} \sum_{a,b} \sigma_i^a \sigma_i^b \sigma_j^a \sigma_j^b \right). \tag{C.11}$$

In the latter equation, interaction between replicas has been introduced and the interaction term can be rewritten as

$$\sum_{i<j} \sum_{a,b} \sigma_i^a \sigma_i^b \sigma_j^a \sigma_j^b = N^2 \sum_{a<b} \left(\frac{1}{N} \sum_{i} \sigma_i^a \sigma_i^b \right)^2 + \frac{N^2 n - N n^2}{2}. \tag{C.12}$$

We show the latter identity starting from the right term:

$$
\begin{aligned}
&N^2\sum_{a<b}\left(\frac{1}{N}\sum_i \sigma_i^a\sigma_i^b\right)^2+\frac{N^2n-Nn^2}{2}=\\
=&N^2\sum_{a<b}\frac{1}{N^2}\left(\sum_i(\sigma_i^a\sigma_i^b)^2+2\sum_{i<j}\sigma_i^a\sigma_i^b\sigma_j^a\sigma_j^b\right)+\frac{N^2n-Nn^2}{2}=\\
=&N^2\sum_{a<b}\left(\frac{1}{N}+\frac{2}{N^2}\sum_{i<j}\sigma_i^a\sigma_i^b\sigma_j^a\sigma_j^b\right)+\frac{N^2n-Nn^2}{2}=\\
=&N^2\frac{n(n-1)}{2N}+2\sum_{a<b}\sum_{i<j}\sigma_i^a\sigma_i^b\sigma_j^a\sigma_j^b+\frac{N^2n-Nn^2}{2}=\\
=&N^2\frac{n(n-1)}{2N}+2\left(\frac{1}{2}\sum_{ab}\sum_{i<j}\sigma_i^a\sigma_i^b\sigma_j^a\sigma_j^b-\frac{1}{2}\sum_{a}\sum_{i<j}(\sigma_i^a\sigma_j^a)^2\right)+\frac{N^2n-Nn^2}{2}=\\
=&\frac{Nn^2-Nn}{2}+\sum_{ab}\sum_{i<j}\sigma_i^a\sigma_i^b\sigma_j^a\sigma_j^b-\frac{nN(N-1)}{2}+\frac{N^2n-Nn^2}{2}=\\
=&\sum_{i<j}\sum_{a,b}\sigma_i^a\sigma_i^b\sigma_j^a\sigma_j^b.
\end{aligned}
$$

Putting (C.12) into (C.11),

$$
\overline{Z_J^n}=\sum_{\{\bar{\sigma}\}}\exp\left(\beta h\sum_{a=1}^{n}\sum_{i}^{N}\sigma_i^a\right)\exp\left(\frac{\beta^2}{4}(Nn-n^2)\right)\prod_{a<b}\exp\left(\frac{\beta^2}{2N}\left(\sum_i\sigma_i^a\sigma_i^b\right)^2\right), \tag{C.13}
$$

and applying (C.8) with $4A=2N\beta^2$ and $B=\beta^2\sum_i\sigma_i^a\sigma_i^b$ the partition function reads:

$$
\begin{aligned}
\overline{Z_J^n}=&\sum_{\{\bar{\sigma}\}}\exp\left(\beta h\sum_{a=1}^{n}\sum_{i}^{N}\sigma_i^a+\frac{\beta^2}{4}(Nn-n^2)\right)\left(\frac{2\pi\beta^2}{N}\right)^{\frac{n(n-1)}{2}}\cdot\\
&\cdot\prod_{a<b}\int\exp\left(-\frac{N\beta^2}{2}Q_{ab}^2+\beta^2\sum_i\sigma_i^a\sigma_i^bQ_{ab}\right)dQ_{ab}=\\
=&\exp\left(\frac{\beta^2}{4}(Nn-n^2)\right)\left(\frac{2\pi\beta^2}{N}\right)^{\frac{n(n-1)}{2}}\int\exp\left(-\frac{N\beta^2}{2}\sum_{a<b}Q_{ab}^2\right)\cdot\\
&\cdot\left(\sum_{\{\bar{\sigma}\}}\exp\left(\beta h\sum_a\sigma^a+\beta^2\sum_{a<b}\sigma^a\sigma^bQ_{ab}\right)^N\right)dQ_{ab}.
\end{aligned} \tag{C.14}
$$

As we told above, we are interested in computing the free energy

$$f(\beta, h) = \lim_{n\to 0}\lim_{N\to\infty}\left(-\frac{1}{\beta N n}\log \overline{Z_J^n}\right). \tag{C.15}$$

Observe that

$$\overline{Z_J^n} \propto \int e^{-NS(Q,h)}dQ, \tag{C.16}$$

where

$$S(Q, h) = -\frac{\beta^2 n}{4} + \frac{\beta^2}{2}\sum_{a<b} Q_{ab}^2 - W(Q)$$

and

$$W(Q) = \log \sum_{\{\overline{\sigma}\}} \exp\left(\beta h \sum_{a} \sigma^a + \beta^2 \sum_{a<b} \sigma^a \sigma^b Q_{ab}\right).$$

Using the approximation Laplace method to solve the integral (C.16), we find that

$$f(\beta, h) = \lim_{n\to 0}\frac{1}{\beta n}\mathrm{extr}_Q S(Q, h), \tag{C.17}$$

where $\mathrm{extr}_Q S(Q, h)$ is the extremal point of the function S with respect to Q.

Appendix D

Interacting replicas: the contribute to the model.

The main peculiarity of the descripted algorithm is its attitude in seeking dense regions of solutions since the process progessively focuses on smaller and smaller regions of the configuration space. This approach is possible replicating the system: we consider N vector variables $\{W_j^a\}_{a=1}^y$ of length y for $j = 1, \ldots, N$ and we assume an internal simmetry for each variable so that all the marginals are invariant for any permutation of the replica indices, which means that

$$P_j\left(\{W_j^a\}_{a=1}^y\right) = P_j\left(\sum_{a=1}^{y} W_j^a\right). \tag{D.1}$$

As we said in the previous chapters, we keep one configuration W^* as reference configuration and we replicate it y times. For this reason starting from (4.6) is possible to write the new probability distribution associated to the model which corresponds to:

$$\begin{aligned} P(\overline{W}^*, \{W_j^a\}_{a=1}^y, \beta, y, \gamma) &= Z(\beta, y, \gamma)^{-1} \exp\left(-\beta \sum_{a=1}^{y} E(W^a) + \gamma \sum_{a=1}^{y} W^a \cdot W^*\right) = \\ &= Z(\beta, y, \gamma)^{-1} \prod_{a=1}^{y} \exp\left(-\beta E(W^a) + \gamma W^a \cdot W^*\right). \end{aligned} \tag{D.2}$$

As $\beta \to \infty$, (D.2) becomes

$$P(\overline{W}^*, \{W_j^a\}_{a=1}^y, y, \gamma) = Z(y,\gamma)^{-1} \prod_{a=1}^{y} \left(\chi(W^a) \exp\left(\gamma W^a \cdot W^*\right)\right) = \\ = Z(y,\gamma)^{-1} \prod_{a=1}^{y} \left(\chi(W^a) \prod_{i=1}^{N} \exp\left(\gamma W_i^a W_i^*\right)\right),$$

so that the probability distributions associated to the simple perceptron, equivalent to (4.12), is defined by

$$P(\overline{W}^*, \{W_j^a\}_{a=1}^y, y, \gamma) = Z(y,\gamma)^{-1} \prod_{a=1}^{y} \prod_{\mu=1}^{M} \left(\Theta\left(\tau^\mu \sum_{i=1}^{N} \sigma_i^a \xi_i^\mu\right) \prod_{i=1}^{N} \exp\left(\gamma W_i^a W_i^*\right)\right). \tag{D.3}$$

Latter distribution is obviously a factorizable function and can be represented on a factor graph: under the presence of y replicas, the factor graph is composed by y replicas of the initial graph and configurations of weights interact through the term $e^{\gamma W^a \cdot W^*}$. For that reason we introduce a factor node γ to build such interaction (see figure D.1).

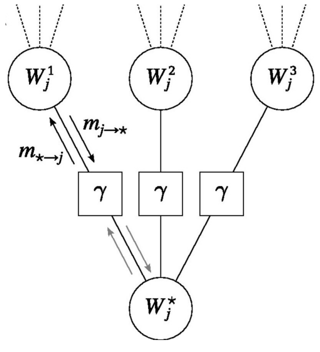


Figure D.1: The figure represents a portion of a BP factor graph with $y = 3$ replicas of the variable W_j. The node W_j^* represents the reference configuration value. Using the same notation of the third chapter, the squares represent the the factor nodes which interactions given by the term $e^{\gamma W^a \cdot W^*}$, namely the factor node γ. The dashed lines identify the links with the rest of the graph.

Inside of every single graph, messages continue to be exchange in the same way descripted in the previous chapters but every replica gives its contribute to the others sending at an extra- message $m_{*\to\sigma_i}^{t+1}$ which satisfies the equation:

$$m_{*\to\sigma_i}^{t+1} = \tanh\left[(y-1)\tanh^{-1}(m_{\sigma_i\to *}^{t} \tanh\gamma)\right] \tanh\gamma, \tag{D.4}$$

where $m^t_{\sigma_i\to *}$ is the message which has been sent at t^{th} iteration from the variable node to the factor node γ (i.e. with interaction strength γ). Under these conditions, the message (4.39) assumes the form

$$m^t_{\sigma_i\to\mu} = \tanh\left[\sum_{k\in\partial\sigma_i\setminus\mu}\tanh^{-1}(\hat{m}^t_{k\to\sigma_i}) + \hat{m}^t_{*\to\sigma_i}\right] \tag{D.5}$$

while $m_{\mu\to\sigma_i}$ and $m_{\tau^\mu\to\mu}$ takes the same expression as in (4.41) and (4.40). A message which is exchanged between a factor node γ_a and a variable node σ_i^a satisfies the equation

$$\hat{m}_{\gamma^a\to\sigma_i^a} = \tanh(\gamma) m_{\sigma_i^a\to\gamma^a} \tag{D.6}$$

while its opposite satisfies

$$m_{\sigma_i^a\to\gamma^a} = \tanh\left[\sum_{b\in\partial\sigma_i\setminus a}\tanh^{-1}(\hat{m}_{\gamma^b\to\sigma_i^a})\right]. \tag{D.7}$$

Putting (D.6) in (D.7), we obtain:

$$\begin{aligned}\hat{m}_{\gamma^a\to\sigma_i^a} &= \tanh(\gamma)\tanh\left[\sum_{b\in\partial\sigma_i\setminus a}\tanh^{-1}(\hat{m}_{\gamma^b\to\sigma_i^a})\right] = \\ &= \tanh(\gamma)\tanh\left[(y-1)\tanh^{-1}(\hat{m}_{\gamma^b\to\sigma_i^a})\right],\end{aligned}$$

As we told in the third chapter, the BP fixed points can be reached for large values of γ. In particular, by increasing simultaneously the interaction strength γ and the number of replicas y, the only fixed points of the iterative message passing procedure are completely polarized, and situated in regions of maximal density $y \to \infty$ at small distances $\gamma \to \infty$.

Bibliography

[1] W.S. Bainbridge, 1995, Neural network models of religoious beliefs, Sociologial Perspectives, vol. 38, no. 4, pp. 483-495.

[2] F. Chierchetti, S. Lattanzi, A. Panconesi, 2011, *Rumor spreading in social networks*, Theoretical Computer Science vol. 412, issue 24, pp. 2602-2610.

[3] Han J., Moraga C., 1995, *The influence of the sigmoid function parameters on the speed of backpropagation learning*, In: Mira J., Sandoval F. (eds) From Natural to Artificial Neural Computation. IWANN 1995. Lecture Notes in Computer Science, vol. 930. Springer, Berlin, Heidelberg.

[4] J. Hertz, A. Krogh, R.G. Palmer, 1991, *Introduction to the Theory of Neural Computation*, Addison Wesley, Redwood City.

[5] A. Engel, C. Van den Broeck, 2001, *Statistical Mechanics of Learning*, Cambridge University Press.

[6] W.S. McCulloch, W.H. Pitts, 1949, *A logical calculus of the ideas immanent in nervous activity*, Bullettin of Mathematical Biophysics, vol. 5, 115-133.

[7] D.O. Hebb, 1949, *The Organization of Behavior*, Wiley, New York.

[8] F. Rosenblatt, 1961, *Principles of Neurodynamics*, Spartan, Washington, D. C.

[9] M. Minsky, S. Papert , 1988, *Perceptrons*, MIT P., Cambridge, Mass.

[10] J. Hopfield, 1982, *Neural networks and physical systems with emergent collective computational abilities*, PNAS, 79 (8) 2554-2558; https://doi.org/10.1073/pnas.79.8.2554

[11] J. Hopfield, D. W. Tank, 1986, *Computing with Neural Circuits: A Model*, Science 233, p. 625.

[12] E. Gardner, 1988, *The space of interactions in neural network models*, Journal of Physics A: Mathematical and General, vol. 21, no. 1.

[13] E. Gardner, D. Wallace, N. Stroud, 1989, *Training with noise and the storage of correlated patterns in a neural network model*, Journal of Physics A: Mathematical and General, vol. 22, no. 12.

[14] P. Contucci, R.Luzi, C. Vernia, *Inverse problem for the mean-field monomer-dimer model with attractive interaction*, Journal of Physics A, Mathematical and theoretical, 2017, 50, pp. 205002 - 205028.

[15] H. Nishimori, 2001, *Statistical Physics of Spin Glasses and Information Processing: An Introduction*, Oxford University Press, Oxford.

[16] M. Mezard and A. Montanari, 2009, *Information, Physics, and Computation*, Oxford University Press, Oxford.

[17] F.E.Harrel, 2001, *Regression modeling strategies: with applications to linear models, logistic regression and survival analysis.* Springer Series in Statistics.

[18] J.K. Roberts, 1938, *Some properties of mobile and immobile adsorbed films.* Proceedings of the Cambridge Philosophical Society 34, pp. 399-411.

[19] R. Peierls, 1936, *On Ising's model of ferromagnetism.* Mathematical Proceedings of the Cambridge Philosophical Society, vol. 32, issue 03, pp. 477-481.

[20] D. Alberici, P. Contucci and E. Mingione, 2014, *A mean-field monomer-dimer model with attractive interaction. Exact solution and rigorous results.* Journal of Mathematical Physics, vol. 55, 063301:1-27.

[21] D. Alberici, P. Contucci, M. Fedele and E. Mingione, 2016, *Limit theorems for monomer-dimer model with attractive potential.* Communications in Mathematical Physics, 1-19.

[22] D. Alberici, P. Contucci and E. Mingione, 2014, *The exact solution of a mean-field monomer-dimer model with attractive potential.* Europhysics Letters, vol. 106, 10001-10005.

[23] D.H. Ackley, G.E. Hinton and T.J. Sejnowski, 1985, *A learning algorithm for Boltzmann machines.* Cognitive science, 9(1):147-169.

[24] K.T. O'Neil and W.F. De Grado, 1991, *A thermodynamic scale for the helix-forming tendencies of the commonly occurring amino acids.* Science, 253(5023):952.

[25] R. Monasson and S. Cocco, 2011, *Fast inference of interactions in assemblies of stochastic integrate-and-fire neurons from spike recordings.* Journal of computational neuroscience, 31, 199-227.

[26] Y. Roudi, J. Tyrcha and J. Hertz, 2009, *Ising model for neural data: Model quality and approximate methods for extracting functional connectivity.* Physical Review E, 79, 051915.

[27] V. Sessak and R. Monasson, 2009, *Small-correlation expansions for the inverse Ising problem.* Journal of Physics A: Mathematical and Theoretical, 42, 055001.

[28] R. Karp and M. Sipser, 1981, *Maximum matching in sparse random graphs.* Proceedings of the 22nd Annual Symposium of the FOCS (IEEE Computer Society Press), pp. 364âĂŞ375.

[29] L. Zdeborova and M. Mezard, 2006, *Thermodynamic origin of order parameters in mean-field models of spin glasses.* Journal of Statistical Mechanics, P05003.

[30] C. Bordenave, M. Lelarge and J. Salez, 2013, *Matchings on infinite graphs.* Probability Theory and Related Fields 157 (1-2), pp. 183-208.

[31] A. Barra, P. Contucci, R. Sandell and C. Vernia, 2014, *An analysis of a large dataset on immigrant integration in Spain. The Statistical Mechanics perspective on Social Action.* Nature Scientific Reports, 4, Article number: 4174, DOI:10.1038/srep04174.

[32] R. Burioni, P. Contucci, M. Fedele, C. Vernia and A. Vezzani, 2015, *Enhancing participation to health screening campaigns by group interactions.* Nature Scientific Reports, 5, Article number: 9904, DOI:10.1038/srep09904.

[33] M. Fedele, C. Vernia and P. Contucci, 2013, *Inverse problem robustness for multi-species mean field spin models.* J. Phys. A: Math. Theor., 46 065001.

[34] R.A. Fisher, 1925, *Theory of statistical estimation.* Mathematical Proceedings of the Cambridge Philosophical Society, 22, pp. 700-725.

[35] E.T. Jaynes, 1957, *Information theory and statistical mechanics.* Physical review, 106, 620.

[36] A. Rodriguez and A. Laio, 2014, *Clustering by fast search and find of density peaks.* Science, vol. 344, no. 6191, pp. 1492-1496.

[37] D.J. MacKay, 2003, *Information theory, inference and learning algorithms.* Citeseer, vol. 7.

[38] A. Decelle and F. Ricci Tersenghi, 2016, *Solving the inverse Ising problem by mean-field methods in a clustered phase space with many states.* Physical Review E, vol. 94, DOI:10.1103/PhysRevE.94.012112.

[39] H. Chau Nguyen and Johannes Berg, 2012, *Mean-Field Theory for the Inverse Ising Problem at Low Temperatures.* Physical Review Letter 109, 050602.

[40] S. Levitus, J. I. Antonov, T. P. Boyer, R. A. Locarnini, H. E. Garcia, A. V. Mishonov, 2009, *Global ocean heat content 1955âĂŞ2008 in light of recently revealed instrumentation problems*, Geophysical Research Letters, vol. 36, L07608, doi:10.1029/2008GL037155.

[41] U. Castiello, 1999, *Mechanisms of selection for the control of hand action*, Trends in Cognitive Sciences, vol. 3, no. 7.

[42] M. C. GonzÃąlez, C. A. Hidalgo, A. BarabÃąsi, 2008, *Understanding individual human mobility patterns*, Nature Publishing Group, https://doi.org/10.1038/nature06958.

[43] M. Nardini, P. Jones, R. Bedford, O. Braddick, 2008, *Development of cue integration in human navigation*, Curr Biol, 18(9):689-93. doi: 10.1016/j.cub.2008.04.021.

[44] J. Hirsh, T. Riemensperger, H. Coulom, M. IchÃľ, J. Coupar, S. Birman, 2010, *Roles of dopamine in circadian rhythmicity and extreme light sensitivity of circadian entrainment*, Curr Biol. 2010 Feb 9;20(3):209-14. doi: 10.1016/j.cub.2009.11.037, PubMed PMID: 20096587; PubMed Central PMCID: PMC2811851.

[45] R. G. Pendleton, A. Rasheed, T. Sardina, T. Tully, R. Hillman, 2002, *Effects of tyrosine hydroxylase mutants on locomotor activity in Drosophila: a study in functional genomics*, Behav Genet., 32(2):89-94. PubMed PMID: 12036114.

[46] J.C. Simon, M. H. Dickinson, 2010, *A New Chamber for Studying the Behavior of Drosophila*, PLoS ONE 5(1): e8793. https://doi.org/10.1371/journal.pone.0008793.

[47] K. Branson, A. A. Robie, J. Bender, P. Perona, M. H. Dickinson, 2009, *High-throughput ethomics in large groups of Drosophila*, Nature Methods, Nature Publishing Group, vol. 6, http://dx.doi.org/10.1038/nmeth.1328, https://www.nature.com/articles/nmeth.1328#supplementary-information

[48] C. Baldassi, C. Borgs, Jennifer Chayes, A. Ingrosso, C. Lucibello, L. Saglietti, R. Zecchina, 2016,*Unreasonable Effectiveness of Learning Neural Networks: From Accessible States and Robust Ensembles to Basic Algorithmic Schemes*, vol. 113 no. 48, E7655âĂŞE7662, doi: 10.1073/pnas.1608103113.

[49] Hinton, E. Geoffrey, 2007, *To recognize shapes, first learn to generate images*, Progress in brain research, 165:535âĂŞ547.

[50] Hinton, E. Geoffrey, R. R. Salakhutdinov, 2006, *Reducing the dimensionality of data with neural networks*, Science, 313(5786):504âĂŞ507.

[51] J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, Q. V Le, A. Y Ng., 2011, *On optimization methods for deep learning*, Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 265-272.

[52] S. Zhang, A. E. Choromanska, Y. LeCun., 2015, *Deep learning with elastic averaging sgd*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems 28, Curran Associates, Inc., pp. 685âĂŞ693.

[53] S. Kirkpatrick, M. P. Vecchi, et al., 1983, *Optimization by simmulated annealing*, Science, 220(4598):671-680.

[54] M. Mezard, G. Parisi, R. Zecchina, 2002, *Analytic and algorithmic solution of random satisfiability problems*, Science, 297(5582):812âĂŞ815.

[55] F. Krzakala, A. Montanari, F. Ricci-Tersenghi, G. Semerjian, L. Zdeborova, 2007,*Gibbs states and the set of solutions of random constraint satisfaction problems*, Proceedings of the National Academy of Sciences, 104(25):10318âĂŞ10323.

[56] L. Zdeborova, M. Mezard, 2008,*Locked constraint satisfaction problems*, Phys. Rev. Lett., 101:078702.

[57] C. Baldassi, A. Braunstein, 2015, *A Max-Sum algorithm for training discrete neural networks*, Journal of Statistical Mechanics: Theory and Experiment, no. 8, P08008, http://stacks.iop.org/1742-5468/2015/i=8/a=P08008.

[58] C. Baldassi, A. Ingrosso, C. Lucibello, L. Saglietti, R. Zecchina, 2015, *Subdominant Dense Clusters Allow for Simple Learning and High Computational Performance in Neural Networks with Discrete Synapses*, Physical Review Letters, 115(12):128101.

[59] H. Huang, Y. Kabashima, 2015, *Origin of the computational hardness for learning with binary synapses*, Physical Review E, 90(5):052813x.

[60] S. Franza, G. Parisi, 1998, *Exective potential in glassy systems: theory and simulations* , Physica A 261 317âĂŞ339.

[61] A. Braunstein, M. MÃľzard, R. Zecchina, 2005, *Survey Propagation: An Algorithm for Satisfiability*, Wiley InterScience (www.interscience.wiley.com), DOI 10.1002/rsa.20057.

[62] M. Mezard and A. Montanari, 2009, *Information, Physics, and Computation*, Oxford University Press.

[63] M. Mezard, G. Parisi, 2003, *Journal of Statistical Physics*, 111: 1. https://doi.org/10.1023/A:1022221005097.

[64] Yedidia, J.S.; Freeman, W.T.; Weiss, Y., 2009, *Constructing free-energy approximations and generalized belief propagation algorithms.* IEEE Transactions on Information Theory. 51 (7): 2282âĂŞ2312. doi:10.1109/TIT.2005.850085.

[65] A. Braunstein, R. Zecchina, 2006, *Learning by message-passing in neural networks with material synapses*, Phys. Rev. Lett., 96:030201.

[66] A. Bucchi, C. Sangiorgi, V. Vignali, 2012, *Traffic Psychology and Driver Behavior*, Science Direct, Procedia - Social and Behavioral Sciences, vol. 53, pp. 972-979, https://doi.org/10.1016/j.sbspro.2012.09.946.

[67] Al-Sayed Ahmed Al-Sobkya, Ragab M.Mousab, 2016, *Traffic density determination and its applications using smartphone*Alexandria Engineering Journal, vol. 55, issue 1, pp. 513-523, https://doi.org/10.1016/j.aej.2015.12.010.

[68] M. Treibera, A. Kestingba, 2017,*The Intelligent Driver Model with Stochasticity -New Insights Into Traffic Flow Oscillations*, Transportation Research Procedia, vol. 23, pp. 174-187, https://doi.org/10.1016/j.trpro.2017.05.011.

[69] B. Chen, Z. Yang, S. Huang, et al., 2017, *Cyber-physical system enabled nearby traffic flow modelling for autonomous vehicles*, Performance Computing and Communications Conference (IPCCC) 2017 IEEE 36th International, pp. 1-6, ISSN 2374-9628.

[70] L. A.Pipes, 1965, *Wave theories of traffic flow*, Journal of the Franklin Institute, vol. 280, issue 1, pp. 23-41, https://doi.org/10.1016/0016-0032(65)90025-6.

[71] E. Vlahogianni, M. Karlaftis, J. Golias, 2014, *Short-term traffic forecasting: Where we are and where we're going*, Transportation Research Part C, vol. 43, Part 1, pp. 3-19.

[72] H. Hashemi, K. F. Abdelghany, 2015, *Real-time traffic network state prediction for proactive traffic management*, Transportation Research Record, Journal of the Transportation Research Board., vol. 2491, pp. 22âĂŞ31.

[73] M. Elhenawy and H. A. Rakha, 2015, *Automatic congestion identification with two-component mixture models*, Transportation Research Record, Journal of the Transportation Research Board, vol. 2489, pp. 11âĂŞ19.

[74] C. Dong, Z. Xiong, C. Shao, H. Zhang, 2015, *A spatial temporal-based state space approach for freeway network traffic flow modelling and prediction*, Transportmetrica A: Transport Science. Taylor and Francis, vol. 11, no. 7.

[75] Y. Yuan, A. Duret, H. van Lint, 2015, *Mesoscopic traffic state estimation based on a variational formulation of the lwr model in lagrangian-space coordinates and kalman filter*, Transportation Research Procedia, vol. 10, pp. 82âĂŞ92.

[76] T. Cheng, J. Haworth, J. Wang, 2011, *Spatio-temporal autocorrelation of road network data*, Journal of Geographical Systems, vol. 14, no. 4, pp. 1âĂŞ25.

[77] J. W. C. V. Lint, Y. Yuan, S. P. Hoogendoorn, J. L. M. Vrancken, and T. Schreiter, *Freeway traffic state estimation using extended kalman*

filter for first-order traffic model in lagrangian coordinates, Proceedings of the IEEE International Conference on Networking, Sensing and Control, ICNSC 2011, Delft, The Netherlands., 2011.

[78] M. Papageorgiou, Y. Wang, and A. Messmer, *Real-time freeway traffic state estimation based on extended kalman filter: adaptive capabilities and real data testing*, Transport. Research Part A: Policy Pract., vol. 42, no. 10, pp. 1340âĂŞ1358, 2008.

[79] Y. Zheng, L. Capra, O. Wolfson, et al., 2014, *Urban computing: concepts, methodologies, and applications*, ACM Trans. Intell. Syst. Technol. 5 (3) 38.

[80] M.B. Younes, A. Boukerche, , 2015, *A performance evaluation of an efficient traffic congestion detection protocol (ECODE) for intelligent transportation systems*, Ad Hoc Netw. 24 317âĂŞ336.

[81] V. R. TomÃąs, L. A. Garcia, 2006, *A Cooperative Multiagent System for Traffic Management and Control*, The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-06).

[82] T. Nakata, J. Takeuchi, 2004, *Mining Traffic Data from Probe-Car System for Travel Time Prediction*, KDDâĂŹ04, pp. 22-25.

[83] X.-H. Yu, W. W. Recker, 2006, *Stochastic adaptive control model for traffic signal systems*, Transportation Research Part C 14, pp. 263-282, Elsevier.

[84] K. Tufte, J. Li, D. Maier, V. Papadimos, R. L. Bertini, J. Rucker, 2007, *Travel Time Estimation Using NiagaraST and latte*, SIGMODâĂŹ07, pp. 1091-1093.

[85] J.-D. Schmocker, S. Ahuja, M. G. H. Bell, 2008, *Multi-objective signal control of urban junctions - Framework and a London case study*, Transportation Research Part C 16, pp. 454-470.

[86] E. Chung, 2003, *Classification of traffic pattern*, Proceedings of the 11th World Congress on ITS.

[87] B. Kerner, 2004, *The Physics of Traffic, Empirical Freeway Pattern Features*, Engineering Applications and Theory Springer.

[88] S. Kurihara, 2013, *Traffic-Congestion Forecasting Algorithm Based on Pheromone Communication Model*, Ant Colony Optimization -

Techniques and Applications, Dr. Helio Barbosa (Ed.), InTech, DOI: 10.5772/52563.

[89] C. Baldassi, A. Ingrosso, C. Lucibello, L. Saglietti, R. Zecchina, 2015, *Subdominant Dense Clusters Allow for Simple Learning and High Computational Performance in Neural Networks with Discrete Synapses*, Phys. Rev. Lett., pub. American Physical Society, vol. 115, issue 12, doi 10.1103/PhysRevLett.115.128101.

[90] B.S.Kerner, 1997, *Experimental Characteristics of Traffic Flow for Evaluation of Traffic Modelling*, IFAC Proceedings, vol. 30, issue 8, pp. 765-770, https://doi.org/10.1016/S1474-6670(17)43914-0.

[91] M. Saeedmanesha, N. Geroliminis, 2017, *Dynamic clustering and propagation of congestion in heterogeneously congested urban traffic networks*, Transportation Research Procedia, vol. 23, pp. 962-979, https://doi.org/10.1016/j.trpro.2017.05.053.

[92] The car data are available at http://www.mts.com/it/index.html.

[93] M. Bando, K. Hasebe, K. Nakanishi, A. Nakayama, 1998, *Analysis of optimal velocity model with explicit delay*, Phys. Rev. E 58, 5429.

[94] G. Fair1965, *Bethe-Peierls-Weiss approximation and a model ferromagnetic thin films*, Lewis Research Center Cleveland, Ohio, National areonautics and space administration, Washington, D.C.

[95] Little, Max A et al., 2018, *Using and Understanding Cross-Validation Strategies*, Perspectives on Saeb et al. GigaScience, 6(5), 1âĂŞ6. http://doi.org/10.1093/gigascience/gix020.

[96] M. Mezard, G. Parisi, 1991, *Replica field theory for random manifolds*, Journal de Physique I, EDP Sciences, 1 (6), pp.809-836.

[97] G. Parisi, N. Sourlas, 1979, *Random Magnetic Fields, Supersymmetry, and Negative Dimensions*, Phys. Rev. Lett. 43, 744.