UNIVERSITAT POMPEU FABRA

This thesis entitled

Algorithms and complex phenomena in networks:
neural ensembles, statistical inference and online communities

Written by

Vicenç Gómez i Cerdà

And directed by

Prof. Dr. Vicente López Martínez

Has been approved by the Department of Information and Communication Technologies

Barcelona, May 2008

A thesis submitted in partial fulfillment of the requirements for the Degree of

Doctor per la Universitat Pompeu Fabra

Mare, pare i germana,
per vosaltres

Agraïments

Vull agrair especialment a Vicente López la seva confiança durant tots aquests anys i el ser una font inesgotable d'idees i inspiració. Malgrat ser una persona molt ocupada, sempre ha trobat temps per atendre els meus dubtes i aconsellar-me quan més ho he necessitat.

Moltes gràcies també a Bert Kappen per donar-me l'oportunitat de treballar amb ell i per compartir amb mi els seus amplis coneixements.

Agraeixo a tots els que d'una manera o altra han col·laborat en la realització d'aquesta tesi: a Andreas Kaltenbrunner, per les seves lliçons i per ser company i un gran amic. Gràcies a Alberto Suárez, Joris Mooij i a Bastian Wemmenhove, a Adan Garriga, Ayman Moghnieh, Rodrigo Meza i a Josep Blat.

Gràcies també a Hector Geffner per ser referent en el meu curt recorregut docent i en molts aspectes de la meva recerca. Gràcies a Hector Palacios per la seva amistat, i a la resta de membres del grup d'Intel·ligència Artificial de la UPF.

Agraeixo a Gustavo Deco haber-me aconsellat durant el primer període de tesi, a Ralph Andrzejak i Anders Ledberg els fructífers journal clubs setmanals. Gràcies a Dani Martí, Ernest Montbrió, i a la resta de membres del grup de Neurociència Computacional de la UPF.

Vull agrair també a Misha Chertkov per la seva invitació a Santa Fe, i a la gent de Nijmegen per ser tant amables durant la meva estància.

Finalment, seria un desagraït sino m'enrecordés d'aquells amb qui he compartit grans moments durant aquest anys: Pau Arumí, Òscar Civit, Ernic Meinhardt, Juan Cardelino, Lucas Vallejos, Stelios Kourakis, Gabriele Facciolo, Carles Castellví i Sergi Masip.

Gràcies Laura, per haver estat al meu costat.

De la mar al precepto,
del precepto al concepto,
del concepto a la idea
—¡oh la linda tarea!—,
de la idea a la mar.
¡Y otra vez a empezar!

Antonio Machado

Abstract

This thesis is about algorithms and complex phenomena in networks. In the first part we study a network model of stochastic spiking neurons. We propose a modeling technique based on a mesoscopic description level and show the presence of a phase transition around a critical coupling strength. We derive a local plasticity rule which drives the network towards the critical point.

We then deal with approximate inference in probabilistic networks. We develop an algorithm which corrects the belief propagation solution for loopy graphs based on a loop series expansion. By adding correction terms, one for each "generalized loop" in the network, the exact result is recovered. We introduce and analyze numerically a particular way of truncating the series.

Finally, we analyze the social interaction of an Internet online community by characterizing the structure of the network of users, their discussion threads in form of comments and the temporal patterns of reaction times to a new post.

Resum

Aquesta tesi tracta d'algorismes i fenòmens complexos en xarxes. En la primera part s'estudia un model de xarxes de neurones estocàstiques inter-comunicades mitjançant potencials d'acció. Proposem una tècnica de modelització a escala mesocòpica i estudiem una transició de fase per un acoblament crític entre les neurones. Derivem una regla de pasticitat sinàptica local que fa que la xarxa s'auto-organitzi en el punt crític.

Seguidament tractem el problema d'inferència aproximada en xarxes probabilístiques mitjançant un algorisme que corregeix la solució obtinguda via *belief propagation* en grafs cíclics basada en una expansió en sèries. Afegint termes de correcció que corresponen a cicles generals en la xarxa, s'obté el resultat exacte. Introduïm i analitzem numèricament maneres de truncar aquesta sèrie.

Finalment analizem la interacció social en una comunitat d'Internet caracteritzant l'estructura de la xarxa d'usuaris, els fluxes de discussió en forma de comentaris i els patrons temporals de temps de reacció davant una nova notícia.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Networks can be found everywhere in nature and their study is a fundamental topic in science. Many systems can be represented as a graph where nodes indicate some magnitude and edges represent certain type of interaction between nodes. At very small scales, cells in the brain are wired forming a network and communicate using stereotyped signals which consist of short electrical pulses which propagate across the synaptic junctions. On a larger scale, people communicate and are linked with others by means of friendship relations or common interests, thereby forming a social network.

Last decades have witnessed a substantial new interest in network research. Largely, this is explained by the increasing wide availability of empirical data and computational power. In consequence, questions related to the emergence of collective phenomena within large networked systems are becoming crucial. How weak and local interactions between nodes give rise to macroscopic behavior? How a single element is affected by the interaction of thousands of nodes? Further, can we find common structural patterns taking place at rather different scales of description, such as the microscopic neuronal scale, and the human scale?

With the goal to answer these questions, scientists build models that capture a necessarily constrained description of the reality. Often, this description is too detailed to account for fundamental mechanisms. For instance, although huge experimental advances can explain with surprising accuracy the microscopic physical mechanisms taking place on an isolated neuron, less is known about the collective phenomena emerging from the interaction of thousands of neurons, where behavior is qualitatively more complex than the simple sum of the behavior of each neuron alone. Statistical physics has been concerned about the study of these questions. An example of such complex phenomena occurs in the vicinity of a phase transition, a phenomenon first discovered in materials that change abruptly from one phase to another as a result of a small change in a control parameter. Here, long-range correlations typically appear between distant particles, and the application of traditional methods is limited.

In computer science, when a problem has been expressed using a network representation, not only better algorithms have been devised, but also qualitative progress concerning the unification of different approaches has been produced. Message passing algorithms are a clear example of this fact, being the *belief propagation* (BP) algorithm the most influential of this family of algorithms. The roots of this algorithm have been rediscovered several times in many areas of science: in artificial intelligence for probabilistic inference (Pearl, 1988), in information theory for error correction (Gallagher, 1963) and in statistical physics (Bethe, 1935; Peierls, 1936).

There exists an old cross-disciplinary field between the statistical physics community and the computer science community, which has resulted in a fruitful interchange of ideas and progress. Methods borrowed from statistical physics have helped to understand and improve existing algorithms. For instance, the fundamental result that of the solution to a combinatorial optimization problem is formally equivalent to the *ground state*, or lowest-energy state, of a physical system in thermal equilibrium Kirkpatrick et al. (1983). This old result prompted broader connections between these disciplines. We can now understand the nature of the BP algorithm as a minimization of an associated free energy (Yedidia et al., 2005). By studying the complexity of random instances of major computational network problems such as satisfiability it has been found that the boundary separating different problem instances is sharp. Moreover, the hardest problem instances are those located near to this sharp boundary, and that threshold becomes increasingly sharp as the problem size increases (Selman et al., 1992; Kirkpatrick and Selman, 1994).

We have mentioned computational problems and models. But, what can be said about real-world networks? Several studies show that real-world networks differ substantially from traditional random networks (Newman, 2003b; Dorogovtsev and Mendes, 2002). An example of a real-world network is the social network composed of users interacting in an online virtual space. As first glance, we are in front of a complex system produced by the interaction of thousands of users expressing their subjective opinions about a wide range of topics. However, we can try to understand such a system by ignoring all semantic information and considering each individual user just as a unit linked to a population who sends messages just as neurons emit action potentials. Do this network differ from other real-world networks? Can we design algorithms to improve the performance of such a system?

The work in this thesis is an exploration of this very generic topic, namely network research, from three very different points of view. This fact inevitably implies the use of different tools and methods. One approach analyzes a neural network model for which analytical results can be derived. The second approach proposes an algorithm to improve the BP algorithm based on a theoretical tool originated in the statistical physics community. Finally, we are faced with a huge dataset produced by humans interchanging messages from which we can build a social network and study its structure and temporal patterns applying concepts of complex network theory.

1.1 Outline of this thesis

We first study a simplified neural network model where noise plays a fundamental role. As biological neurons, the neurons in this model communicate by means action potentials or spikes. Spike trains represent the elementary units of signal transmission in neural systems. Despite of the simplicity of the model, it presents rich and complex dynamics such as a phase transition and hysteresis.

After a first introductory chapter, in **Chapter 3** we develop a modeling technique which is focused on a mesoscopic level of description, since the microscopic states of each unit in the ensemble is not considered. The model retains the same network dynamics and quantities relevant for information processing of the original microscopic model. In computational terms, due to its event-driven nature, the technique results in a more efficient approach than the traditional way of simulate this type of networks.

In **Chapter 4** we study the dynamics of this network model. In a regime with weak interactions among the units, the effect of the message exchange is to reduce the dispersion of the firing period of the individual neurons. In a critical interaction regime, a number of activity clusters emerge in the ensemble. Neurons in each cluster fire periodically and in synchrony with each other. The number of these self-sustained firing states characterized by distinct patterns towards which the network can evolve is very large. Because of their stability with respect to intrinsic fluctuations in the dynamics of the stochastic neurons, these states could, in principle, be used to encode and process large amounts of information. The first part of this chapter is devoted to the analysis of the dynamics of this model. In the second, we derive a local synaptic rule which drives the system to the critical point.

The second part is devoted to approximate inference in probabilistic networks. In **Chapter 5** we introduce an original scheme based on a theoretical tool named loop calculus derived in (Chertkov and Chernyak, 2006b). Loop calculus allows to express a crucial magnitude, the partition function or normalization factor of a probability distribution via a series expansion where each term is expressed in terms of the solution of the BP algorithm and corresponds to a generalized loop in the original graph. This theoretical result motivates the development of new inference algorithms to improve the BP solution in cases where exact inference is not tractable. Our proposed algorithm allows exhaustive classification of all the generalized loops contributing to the series. We suggest several truncation parameters and experiment with three models of statistical inference: the square Ising lattice, Random graphs and a Bayesian network for medical diagnosis. We test the truncation scheme analyzing approximations for the partition sum and single marginal probabilities of variables against the exact results of tractable instances, amenable for exact inference using the junction tree algorithm. We test the dependence of the number of loops necessary to achieve convergence of the truncated series to the exact result on auxiliary parameters controlling complexity of the problem. While for relatively

easy problems truncation of the series works well with only few terms accounted for, the number of terms grows significantly with complexity.

Finally, in the last part of the thesis, we mine a huge online social space using tools from complex network theory. In **Chapter 6** we introduce some concepts of complex networks and in **Chapter 7** we present the popular bulletin board website named Slashdot and describe our process used to collect the empirical data. We analyze the social network extracting relationships between users from their commenting activity. We then consider a representation of the nested dialogues in a form of radial tree and study their statistical properties. These results are used to introduce a simple measure to evaluate the degree of controversy provoked by a published post.

The temporal patterns originated from the posts and comments are also analyzed, and remarkable regularities are found in the site. For instance, the reaction times of users comments triggered when a new post is published are very stereotypical, and can be described with high accuracy using a mixture of two log-normal probability distributions. This result is explained considering the interplay of two waves of activity, one generated when the post is published, and the other caused by the circadian rhythm.

Part I

Neural Ensembles

Chapter 2

Information processing in neural networks

Over the last decades an enormous amount of detailed knowledge about the structure and the function of brain has been accumulated. It is widely accepted that neurons are the main processing elements in the brain. They interchange messages of pulsed nature by means of synaptic junctions, forming a network. To understand the information processing mechanisms which take place in these neural systems and to build artificial ones inspired in these mechanisms, a theoretical model is required. Two main scientific approaches can be differentiated concerning neural modeling:

One has its origins in the artificial intelligence field, in the connectionist community, and has its motivation as an alternative view to the traditional computation. The roots underlying this approach were that natural systems perform extremely parallelized operations using a lot of simple processing units. Robustness and error tolerance to failures in some units are also essential characteristics of the biological systems not structurally found in traditional computer architectures, which operate in a totally different way. The task here was to build a system capable of showing intelligence in a biologically realistic way. The work of McCulloch and Pitts (1943) represented a very important step in that direction. They proposed to model a neuron just as a bistable on/off switch and showed that a network of these units, if properly configured, is able to perform any logical computation. Although this idea of a brain acting as a logical system has to be taken with caution, this simple model represents the basis of artificial neural networks and inspired many of the current models.

Nowadays, we can use a statistical framework to formally describe most of the tools used by that community. The concept of learning is understood in a statistical sense: given a model which can be interpreted as a nonlinear function which performs an input-output mapping and has some free parameters, an optimization procedure is used to set their optimal values. The learning algorithm is the particular rule to update the parameters. Once the model is trained, it can be used to predict responses if unknown data is presented as input. Generalization, or the ability to predict well the new outputs, is the main task, and the measure in which the performance of those systems is compared. Multilayer

perceptrons, Radial Basis Functions and Support Vector Machines are the most used examples of that tools.

The other main approach is less engineering, and takes profit from the models to explain and understand the biological mechanisms governing neural systems. Models are by definition constraint to cover a certain range of biological level of detail. Compartmental models represent the most detailed descriptions and easily exceed tens of dynamic variables to describe the evolution of just one single neuron. Some steps further, the Hodking-Huxley model (Hodgkin and Huxley, 1952), which describes the membrane potential of a neuron in terms of the dynamic behavior of various ion channels, is perhaps the most representative model able to reproduce neurodynamic data with a high level of precision. However, the computational resources required to simulate large networks of these elements and its high number of free parameters make difficult to investigate analytically its behavior and understand its fundamental nature.

Probably the best compromise between analytical tractability and realism is the integrate-and-fire (IF) neuron model (Lapicque, 1907). This model is simple but reproduces the fundamental features and characteristics of neural systems, namely, their spiking nature: the combination of sub-threshold dynamics which take place on a large temporal scale and the supra-threshold dynamics characterized by a transient event, the emission of a spike, which is propagated to other connected neurons.

This introductory chapter reviews the basic elements of the first part of this thesis. We first introduce some neural network models from an "engineering" motivation. Then we present in more detail the integrate-and-fire (IF) model, with special emphasis on the diffusion approximation and the inter-spike-interval (ISI) distributions.

2.1 Neural networks for pattern recognition

McCulloch and Pitts (1943) introduced the binary threshold unit as a simple neuron model. The state of a neuron j can be either activated 1 (or firing at maximum rate) or resting 0 (not firing) in function of the states of the connected units and an intrinsic threshold parameter. The connections between neurons are specified using a synaptic matrix $\mathbf{W}$ with elements w_{ij}. The output of a unit j is just a transfer function σ applied to a weighted sum:

$$y_j = \sigma\left(\sum_{i=1}^{N} w_{ij}x_i + \theta_j\right). \tag{2.1}$$

The transfer function σ originally takes the form of a heaviside step function $\sigma(a) = \mathrm{H}(a)$. McCulloch and Pitts (1943) were interested on the computational properties of these networks. These devices can perform logical computations if they are properly configured. For instance, a logical OR gate can be easily implemented using two input units $1,2$ and an output unit 3 with weights $w_{13} = w_{23} = 1$ and

threshold $\theta_3 = 0.5$. Since we can build logical gates for the AND and NOT functions as well, any computable binary function can be written in terms of these circuits.

Replacing the transfer function σ of equation (2.1) with a continuous and differentiable function such as the sigmoid function $\sigma(a) = 1/(1+\exp(-2a))$ allows to interpret the output of a neuron as its *firing rate* and also facilitates the mathematical analysis of large networks. If real neurons are analyzed in function of their input, sometimes they show a response pattern similar to the profile of a sigmoid function: the output rate of a neuron is very small for small inputs and saturates when it is very excited.

This is the basis of the perceptron (Rosenblatt, 1962), the first neural network model. In this first generation of neural networks, the connectivity between neurons is feed-forward. There exists an input layer of neurons connected to an output layer, but there are no feedback loops between the neurons of the output layer and the input layer. These models are able to *learn* a specific input-output mapping given some training examples. Learning is just the search for optimal synaptic weights and is performed minimizing some error function. The perceptron has inspired the current standard methods for addressing pattern recognition tasks such as classification and regression.

Another example of model inspired in the McCulloch and Pitts neurons with special interest for this thesis is the **Boltzmann machine** (Hinton and Sejnowski, 1983; Ackley et al., 1988). Contrary to the previous models, it is not constrained to have feed-forward connectivity, and the dynamics are stochastic and asynchronous. The global state of a Boltzmann Machine is a vector $\mathbf{x}$ with N components corresponding to the units in the network. For convenience, we assume they take ± 1 values. The state of a randomly selected unit j is $+1$ with the probability given by the sigmoid function:

$$p(x_j = +1) = \sigma\left(\sum_{i=1}^{N} w_{ij}x_i + \theta_j\right) \qquad \sigma(h_j) = \frac{1}{1+\exp(-2\beta h_j)}, \tag{2.2}$$

where a noise parameter β is introduced, which in statistical physics terms can be interpreted as the inverse of the temperature $1/T$. This factor scales the weights and thresholds.

In this model, one is interested in the behavior of the network state given a synaptic matrix $\mathbf{W}$ and thresholds Θ. If weights are symmetric, i.e. $w_{ij} = w_{ji}$, it is guaranteed that after many applications of Equation (2.2), the system reaches an attractor state, with a certain configuration of the units, corresponding to a local minima of the energy function:

$$E(\mathbf{x}; \mathbf{W}, \Theta) = -\frac{1}{2}\sum_{i,j} w_{ji}x_i x_j + \sum_i \theta_i x_i, \tag{2.3}$$

The stationary probability distribution for the network states is called the Boltzmann-Gibbs distribution, and is determined by the energy of a state vector relative to the energies of all possible binary

state vectors:

$$P(\mathbf{x}|\beta,\mathbf{W},\Theta) = \frac{1}{Z(\beta,W,\Theta)} \exp\left(-\beta E(\mathbf{x};\mathbf{W},\Theta)\right). \tag{2.4}$$

where:

$$Z(\beta,\mathbf{W},\Theta) = \sum_{\mathbf{x}} \exp\left(-\beta E(\mathbf{x};\mathbf{W},\Theta)\right) \tag{2.5}$$

is a normalization constant, called the partition function.

The Boltzmann machine is a model for an associative memory. An attractor state represents a stored pattern in the network. The task to find the most probable state of the network given a fixed connectivity matrix $\mathbf{W}$ would correspond to recover the pattern, and can be formulated as an inference problem. But the connectivity matrix $\mathbf{W}$ can also be *learned* in such a way that the Boltzmann-Gibbs distribution which 'best' describes an specified number of patterns, or training data, is represented by the network.

Hopfield networks (Hopfield, 1982) are very related to Boltzmann machines. Their dynamics are deterministic and all units are visible. Boltzmann machines are also a particular case of Markov random fields, a model we will find again in the second part of this thesis.

2.2 Networks of Integrate-and-fire neurons

A more realistic model of neurons considers the microstate of a neuron characterized by the membrane potential $v(t)$. Although all spatial structure is still neglected, the evolution of $v(t)$ is more complex. The membrane potential receives excitatory and inhibitory contributions from inputs that arrive from other neurons by their associated synapses. Each input is integrated (summed) according to its weighted synaptic strength and when $v(t)$ reaches a certain threshold V_{th}, a spike event (or action potential) is generated and propagated through the connected target neurons. Immediately, $v(t)$ returns to its resting level V_0 and the cell remains inactive during a small refractory period τ^{refr}. In the absence of any input, the leaky IF model is equivalent to an electric circuit formed by a resistor and capacitor in parallel and $v(t)$ decays with a characteristic time constant. Formally, $v(t)$ evolves according to the following equation:

$$C_m \frac{\partial v(t)}{\partial t} = -\frac{C_m}{\tau_m}\left[v(t) - V_0\right] + I_s(t) + I_{inj}(t). \tag{2.6}$$

Here, V_0 is the resting potential, τ_m is the passive membrane time constant, which is related by $\tau_m = R_m C_m$ to the capacitance and the leak resistance R_m. The terms $I_s(t)$ and $I_{ext}(t)$ correspond to injected current from the incoming synaptic inputs or from external current using an intracellular electrode, for instance.

The synaptic current I_s can be modeled independently from the membrane potential as a summation of the received spikes (current synapses) or can include a $v(t)$-dependent term (conductance synapses). In that case, changes in $v(t)$ are nonlinear (Burkitt, 2006). We restrict this exposition to current synapses only:

$$I_s = C_m \left(\sum_{k=1}^{N_E} \varepsilon_{E,k} S_{E,k}(t) + \sum_{k=1}^{N_I} \varepsilon_{I,k} S_{I,k}(t) \right). \tag{2.7}$$

In this case the synaptic efficacy or the change in $v(t)$ is assumed identical for all N_E excitatory synapses ($\varepsilon_{E,k} > 0$) and all the N_I inhibitory synapses ($\varepsilon_{E,k} < 0$) The input spike trains enter in (2.7) as sums of δ-Kronecker functions:

$$S_E(t) = \sum_{t_{E,k}} \delta(t - t_{E,k}) \qquad\qquad S_I(t) = \sum_{t_{I,k}} \delta(t - t_{I,k}), \tag{2.8}$$

where $t_{E,k}$ and $t_{I,k}$ denote the exact spike time of an excitatory and inhibitory input neuron respectively. In reality, there exists also an axonal delay so that the spike generation and micro-state update do not occur simultaneously. Although this is a very relevant feature of biological systems, this transmission delay introduces many mathematical difficulties and is often neglected.

Usually, only subthreshold dynamics is considered, and the originated spike event after threshold crossing of $v(t)$ is not explicitly modelled. It is treated as a δ-pulse. This nonlinearity, together with the step discontinuities in $v(t)$ due to the coupling between neurons, makes the model difficult to study analytically. However, these features are also the main source of its interesting and complex phenomenology.

Noise is an essential characteristic of neural systems. In the leaky IF, intrinsic sources of noise are often neglected, and fluctuations are incorporated by means of external mechanisms, for instance, in the form of stochastic synaptic input. It is common, for instance, to model synaptic input as a Poisson processes. In this case, $I_s(t)$ is characterized by an aggregation excitatory and inhibitory Poisson processes (pool approximation), with spiking rates λ_E and λ_I, and $\varepsilon_{E,k} = \varepsilon_E, \varepsilon_{I,k} = \varepsilon_I$, see Equation (2.7). The evolution of $v(t)$ can be regarded as a one-dimensional Markov process, and the theory of stochastic processes has proved very useful. This approach considers the diffusion approximation of the micro-state variable.

Diffusion approximation

A useful way to approximate the evolution of the micro-state variable $v(t)$ is to assume that between two spike events a neuron receives many incoming impulses whose individual contributions are infinitesimally small (Tuckwell, 1988). We will consider the two most used stochastic processes to describe the noisy evolution of $v(t)$ toward the threshold: the Wiener process, and the Ornstein-Uhlenbeck process (OUP).

The **Wiener process** is the continuous limit of a simple random walk and was studied in (Gerstein and Mandelbrot, 1964). In this approximation, the micro-state acts as a perfect integrator and the leaky term is neglected.

A standard Wiener process is defined as a stochastic process $\{W(t), t \geq 0\}$ which satisfies the following two conditions (Stirzaker, 2005):

1. $W(0) = 0$,
2. $W(t)$ has independent increments $(W(t+s) - W(s)) \sim \mathcal{N}(0,t) \quad s,t \geq 0$.

The membrane potential is then described by multiplying the standard Wiener process by an infinitesimal variance parameter σ_W^2 and a linear drift $\mu_W t$:

$$v(t) = V_0 + \mu_W t + \sigma_W W(t), \qquad t > 0, \tag{2.9}$$

This approximation is valid only when the neuron has an exceedingly large time constant and the rate of incoming excitation is very fast. Two examples of trajectories of this approximation are shown in Figure 2.1.

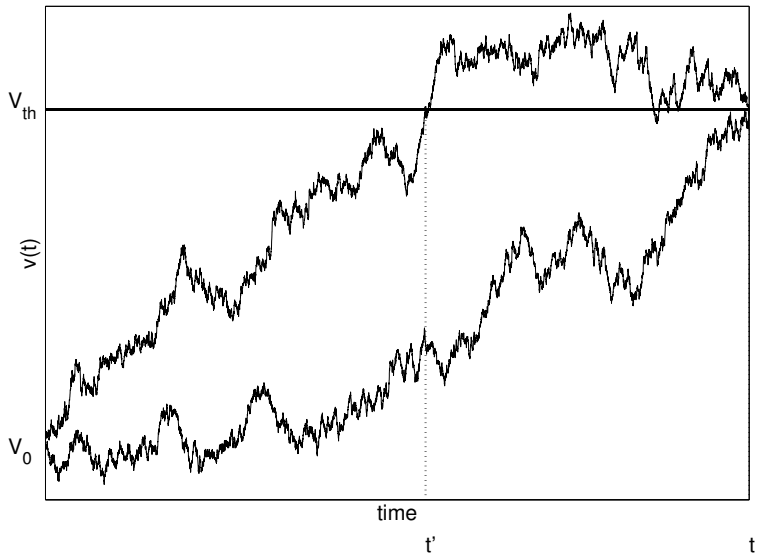

Figure 2.1: Two examples of trajectories of the Wiener process. The figure also illustrates the renewal equation (2.15): the conditional probability $p(V_{th}, t|V_0, t_0)$ of crossing the threshold V_{th} at time t given that $v(t_0) = V_0$ is the integration of all realizations from $v(t_0) = V_0$ to $v(t) = V_{th}$. The renewal condition allows to split each path into the first threshold crossing at t' and the return to V_{th} at time t (inspired in (Burkitt, 2006)).

The **Ornstein-Uhlenbeck process** (Uhlenbeck and Ornstein, 1930) incorporates the leaky term, and is defined by the following stochastic differential equation:

$$\tau \frac{\partial v(t)}{\partial t} = -[v(t) - V_0] + \mu(t) + \sigma(t)\sqrt{2\tau}\xi(t), \tag{2.10}$$

where $\mu(t)$ is the mean input. Noise enters in (2.10) as $\xi(t)$, which is Gaussian white noise, i.e. $\langle\xi(t)\rangle = 0$ and $\langle\xi(t)\xi(t')\rangle = \delta(t-t')$. Under homogeneous input, the mean input and noise intensity are time independent $\mu(t) = \mu, \sigma(t) = \sigma$ [1].

The OUP process is actually a diffusion process constructed with the same first and second infinitesimal moments as the membrane potential of the Stein's model (Stein, 1967). In Stein's original model the membrane potential is described as a stochastic process with discontinuities (jumps). The parameters τ, μ and σ^2 of the OUP process for the case of homogeneous and poisson are (see Tuckwell, 1988, for more details):

$$\begin{aligned} \tau &= \tau_m, \\ \mu &= \tau_m \left(\varepsilon_E \lambda_E - \varepsilon_I \lambda_I\right), \\ \sigma^2 &= \frac{\tau_m}{2}\left(\varepsilon_E^2 \lambda_E + \varepsilon_I^2 \lambda_I\right). \end{aligned} \tag{2.11}$$

First-Passage Times and Inter-Spike Interval Distributions

The time at which $v(t)$ reaches the threshold V_{th} for the first time assuming an initial value of $v(0) = V_0$ arises as a quantity of interest:

$$T_{ISI} = \inf\{t > 0; v(t) \geq V_{th} | v(0) = V_0 < V_{th}\}. \tag{2.12}$$

The probability distribution of this variable leads to the first-passage-time (FPT) density $f_{th}(t)$ that can be regarded also as the inter-spike interval (ISI) distribution. From this distribution, the rate of a neuron can be obtained, as well as other important statistics. The mean firing rate of a neuron ν_{out}, defined as the reciprocal of the average ISI $\langle T_{ISI}\rangle$, is the most common characterization of the experimental data on spontaneous neuronal activity:

$$\nu_{out} = \left[\tau^{refr} + \langle T_{ISI}\rangle\right]^{-1}, \tag{2.13}$$

$$\langle T_{ISI}\rangle = E\left[T_{ISI}\right] = \int_0^\infty t f_{th}(t) dt, \tag{2.14}$$

where we have summed the refractory period τ^{refr}.

The *conditional* probability density $p(v,t|v_0,t_0)$ is a useful distribution to obtain the ISI distribution. In the diffusion approximation, the following relation is true and takes the name of **renewal equation** (see Figure 2.1):

$$p(V_{th},t|V_0,t_0) = \int_{t_0}^{t} dt' f_{th}(t') p(V_{th,t}|V_{th},t'). \tag{2.15}$$

We now describe in detail the case of a perfect integrator with homogeneous input because of its relevance in the next chapter. Consider that V_{th} is the number of excitatory synaptic inputs of uniform

[1] Note that the increments of the Wiener process are precisely Gaussian white noise, so $\xi(t) = \frac{\partial W(t)}{\partial t}$.

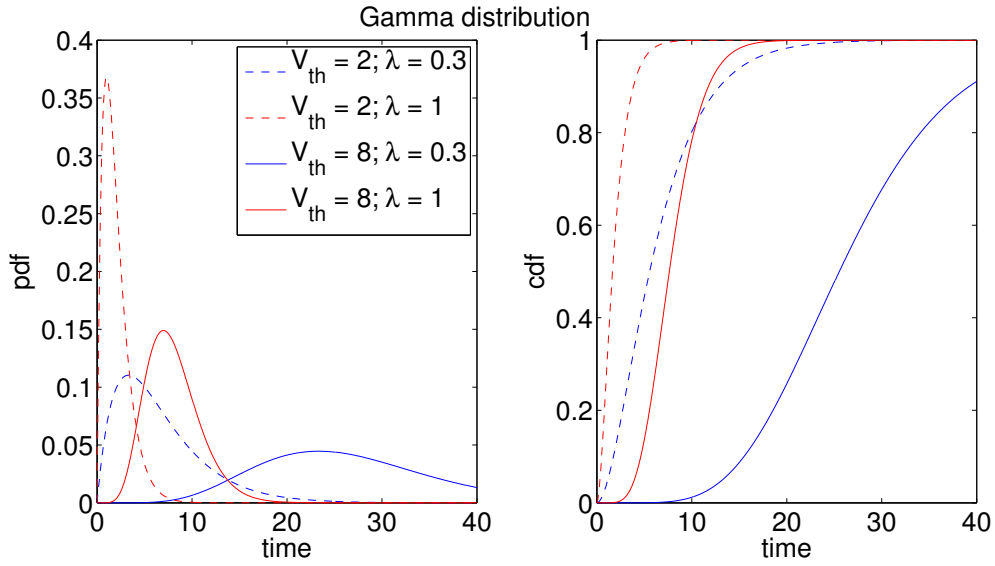

Figure 2.2: Examples of the gamma distribution with rate $\lambda = 1$ and various values of the shape parameter.

amplitude that are required to reach the threshold, and that they occur according to an homogeneous Poisson process of rate λ. The ISI distribution is then a sum of V_{th} exponentials, which is a gamma distribution with *shape* parameter V_{th} and *rate* parameter λ. Its probability density function (pdf) is:

$$f_{th}(t;V_{th},\lambda) = t^{V_{th}-1}\frac{\lambda_{th}^{V}}{\Gamma(V_{th})}e^{-\lambda t} \qquad t,V_{th},\lambda > 0, \tag{2.16}$$

where $\Gamma(\cdot)$ is the gamma function, the continuous extension of the factorial function to real and complex numbers: $\Gamma(V_{th}) = \int_0^\infty t^{V_{th}-1}e^{-t}dt$. For integer numbers, $\Gamma(V_{th}) = (V_{th}-1)!$. Figure 2.2 shows three examples of pdfs for this distribution. This distribution is one of the most frequent used to characterize ISIs (Levine, 1991; McKeegan, 2002; Hentall, 2000). Its mean, variance and coefficient of variation are:

$$\langle T_{ISI}\rangle^{\Gamma} = \frac{V_{th}}{\lambda} \qquad Var^{\Gamma} = \frac{V_{th}}{\lambda^2} \qquad C_v^{\Gamma} = \sqrt{\frac{1}{V_{th}}} \tag{2.17}$$

Alternatively, one can approximate the random arrival times using a Wiener process. In this case, the integral in (2.15) can be solved using the convolution theorem of Laplace transforms and the first passage time is the *inverse Gaussian* (IG) distribution (see Tuckwell, 1988, for the derivation):

$$f_{th}(t;\mu_W,\sigma_W) = \frac{V_{th}}{\sqrt{2\pi\sigma_W^2 t^3}}\exp\left(-\frac{(V_{th}-\mu_W t)^2}{2\sigma_W^2 t}\right), \tag{2.18}$$

which has the following mean, variance and coefficient of variation:

$$\langle T_{ISI}\rangle^{IG} = \frac{V_{th}}{\mu_W}, \qquad Var^{IG} = V_{th}\frac{\sigma_W^2}{\mu_W^3}, \qquad C_V^{IG} = \frac{\sigma_W}{\sqrt{V_{th}\mu_W}}. \tag{2.19}$$

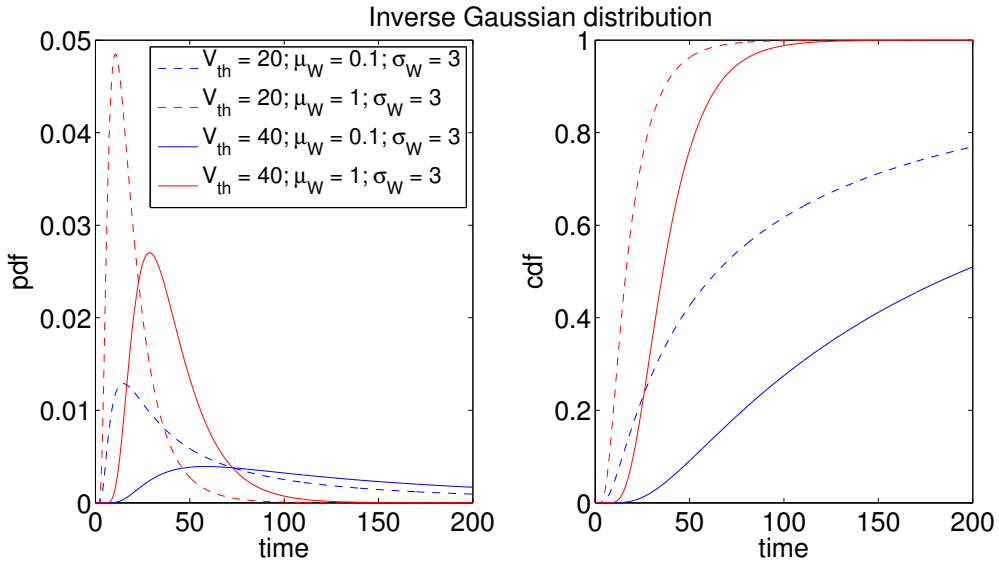

Figure 2.3: Examples of the IG distribution with different values of the mean and scaling parameters (compare with Figure 2.2).

The IG is usually described using a mean parameter μ and scaling parameter λ. This parameterization is recovered under the following substitution: $\mu = V_{th}/\mu_W$ and $\lambda = (V_{th}/\sigma_W)^2$. The IG distribution is symmetric and moderate tailed for small λ. It is highly skewed and long tailed for large λ. It approaches normality as λ approaches zero. Compared to the gamma, the IG distribution lacks very short ISIs and can present larger tails.

For the general case of the leaky IF model, there is no closed form of the ISI distribution. In this case, the Fokker-Planck formalism has proved succesful. It is based on the following stochastic differential equation for the probability density of the membrane potential (Chandrasekhar, 1943):

$$\frac{\partial}{\partial t}p(v,t) = \left(-\frac{\partial}{\partial v}A(v)\frac{1}{2}\frac{\partial^2}{\partial v^2}B(v)\right)p(v,t), \tag{2.20}$$

where $A(v)$ is called the drift function and $B(v)$ is the diffusion function, which are the first two moments of the distribution of the independent jumps in $v(t)$ due to the stochastic input. For the OUP process with homogeneous Poisson input, Equation (2.10), these functions are (Brunel and Hakim, 1999):

$$A(v) = -\frac{1}{\tau}(v - V_0 - \mu) \qquad B(v) = \frac{2\sigma^2}{\tau}. \tag{2.21}$$

and the corresponding mean firing rate is:

$$\nu_{out} = \left(\tau^{refr} + \frac{\tau}{\sigma}\sqrt{\frac{\pi}{2}}\int_{V_0}^{V_th} \exp\left(\frac{(u-\mu)^2}{2\sigma^2}\right) \times \left(1 + \left(\frac{u-\mu}{\sigma\sqrt{2}}\right)\right)\right)^{-1} \tag{2.22}$$

where τ, μ and σ^2 correspond to the OUP parameters of (2.11).

Chapter 3

Mesoscopic event-driven modeling

In the previous chapter we have seen that a model is necessarily a constrained description of the reality. For this reason, it is of relevance importance to build simple models that preserve essential properties of more detailed models. In this chapter we will show that it is possible to reformulate a model of stochastic IF neurons in such a way that the temporal dynamics, considered the dominant mode of communication and information processing in neural systems, is reproduced with perfect accuracy without the necessity to determine the microstates of each unit in the network. For this reason, we call this technique *mesoscopic* modeling, since it relies on a temporal scale between the microscopic evolution of each neuron and the macroscopic description of a population. The resulting model also reduces the computational resources required for its numerical simulation under certain realistic conditions and allows to extend the original model which formulated in the discrete time domain to the continuous time domain. Our strategy is based on the event-driven framework, where the spikes are considered as the events which drive the system dynamics.

This chapter is structured as follows. In the next section we present the neural network model considered in this thesis. In Section 3.2 we review the traditional event-driven framework applied to general populations of spiking neurons. Our proposed technique is presented as a modification of the original framework and applied to our model in Section 3.3. Finally, in Section 3.4 we show results from numerical simulations and conclude this chapter.

3.1 Discrete and stochastic neural networks

The model used in this thesis is based in Rodríguez et al. (2001); Rodríguez et al. (2002) and can be considered as an extension of the one proposed in (Vreeswijk and Abbott, 1993) and (Gerstein and Mandelbrot, 1964). The microstate of a neuron i represents its activation level a_i and takes the form of a discrete random walk with positive drift towards an absorbing barrier L. When the threshold L

is reached, unit i emits a message to the rest of the units of the network which is propagated during a transmission delay δ. After the firing of unit i, a_i is reset to the initial value 1, and the unit enters a refractory period t_{ref} where it remains insensitive to incoming spikes. Thus, according to the notation followed in the previous chapter, the activation level a_i^t and threshold L correspond to $v(t)$ and V_{th} respectively. We prefer to maintain the original terminology of Rodríguez et al. (2001). Formally, the stochastic growth of a_i is composed of two types of dynamics:

- **Stochastic state transitions** modeled by the following rule, associated to a Bernoulli process with parameter p. From now on this dynamics will be denoted *spontaneous* evolution:

$$\begin{aligned} a_i^{t+1} &= \begin{cases} a_i^t + 1 & \text{with probability } p \\ a_i^t & \text{with probability } (1-p) \end{cases} && \text{if } a_i^t < L\,, \\ a_i^{t+t_{ref}} &= 1 \text{ with probability } 1 && \text{if } a_i^t \geq L\,. \end{aligned} \tag{3.1}$$

- **State transitions induced by the rest of the population**. The strength of the possibly received messages from units which fired at time-step $t-\delta$ is integrated. Considering this deterministic growth, the state a_i changes according to this rule:

$$a_i^{t+\delta} = \begin{cases} a_i^t + \sum_{j \neq i} \varepsilon_{ji} H_L(a_j^t) & \text{if } a_i^t < L \\ 1 + \sum_{j \neq i} \varepsilon_{ji} H_L(a_j^t) & \text{if } a_i^t \geq L \text{ and } t^\delta \geq t_{ref}\,, \\ 1 & \text{if } a_i^t \geq L \text{ and } t^\delta < t_{ref} \end{cases} \tag{3.2}$$

  where $H_L(x)$ is the heavy-side step function, i.e. $H_L(x) = 1$ if $x \geq L$ and 0 otherwise, and ε_{ji} represents the synaptic efficacy between unit j and i.

For simplicity, we use here $\varepsilon_{ji} = \varepsilon = 1$, so that ε can be referred to as an homogeneous global coupling term. Contrary to the original model in Rodríguez et al. (2001), we consider ε as a continuous variable. In addition, to simplify this first analysis we set $t_{ref} = \delta = 1$. According to these assumptions, the final evolution of the microstate of i can be summarized in one single equation:

$$\begin{aligned} a_i^{t+1} &= \begin{cases} a_i^t + \sum_{j \neq i} \varepsilon_{ji} H_L(a_j^t) + 1 & \text{with probability } p \\ a_i^t + \sum_{j \neq i} \varepsilon_{ji} H_L(a_j^t) & \text{with probability } 1-p \end{cases}, && \text{if } a_i^t < L \\ a_i^{t+1} &= 1 + \sum_{j \neq i} \varepsilon_{ji} H_L(a_j^t) && \text{if } a_i^t \geq L \end{aligned} \tag{3.3}$$

Figure 3.1 illustrates Equation (3.2) for a certain values of parameters L, p and ε.

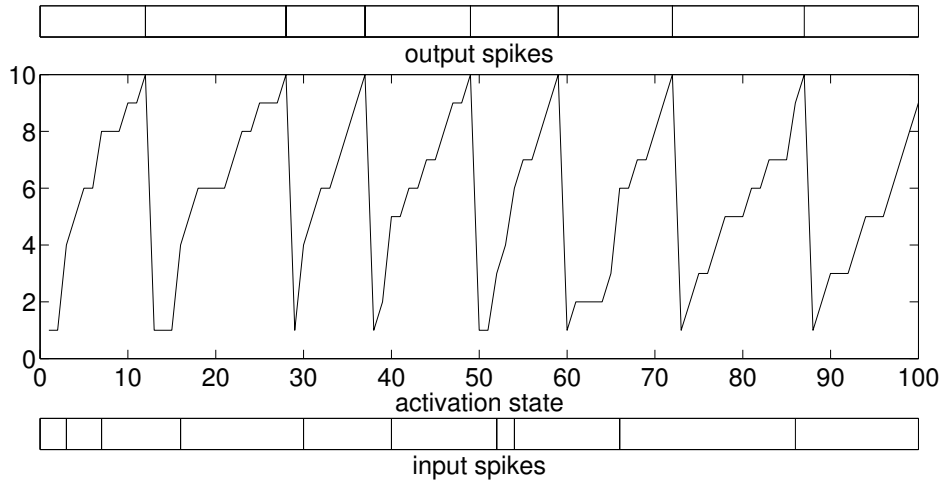

Figure 3.1: Example of temporal trace (100 time-steps) of a unit under the dynamics defined by the rule (3.2). The values of the parameters are $p = 0.4, L = 10$ and $\varepsilon = 2$. Bottom panel indicates incoming spikes from other units in the network. They increase the activation state in ε units, shown in the middle panel. Top panel indicates the output spike train of the neuron.

A characteristic parameter used to describe the degree of interaction between the units is:

$$\eta = \frac{L-1}{(N-1)\varepsilon} \tag{3.4}$$

Parameter η indicates whether the spontaneous dynamics or the message interchange mechanism dominate the behavior of the system. For $\eta \gg 1$, where interactions are small, the system is basically driven by the stochastic, spontaneous activity, and all units behave as independent oscillators. This weak coupling regime is mostly referred as the "Noise-dominated", or spontaneous activity regime, which is recognized as a representative of typical cortical conditions. For values of η near 1 the behavior of the system is dominated by the pulse-exchange mechanism and synchronous behavior starts to emerge. This corresponds to the "driven" activity regime or also called "Drift-dominated" regime. The transition between these two regimes will be analyzed in the next chapter.

3.2 Event-driven simulation of spiking neurons

We consider a generic model composed of N coupled spiking units which evolve spontaneously in time. The ensemble can be considered as a directed graph with arbitrary connectivity in which nodes emit and receive messages of pulsed nature through the connections. According to the framework described by Rochel and Martinez (2003), to rewrite a particular model using the event-driven approach, four components need to be defined:

- The state variable a_i, which usually represents the membrane potential of neuron i.

- Two functions, r_i and s_i, which describe how a_i changes after the reception and emission of a message respectively.
- The function $\tilde{t}_i$, which gives the predicted time of the next firing, given the present state variable, assuming that no messages will affect i until then.

Provided an instance of these four components $\langle a_i, r_i, s_i, \tilde{t}_i \rangle$, an event-driven engine drives the simulation without the necessity to integrate the microstate variable within the interval between two relevant events. This event-driven simulation is also called *asynchronous*, instead of the original *synchronous* approach which updates simultaneously all microstates at each time-step Δt. Since Δt has to be chosen very small in order to reproduce with enough accuracy the system dynamics, avoiding this computational demand can decrease dramatically the time required to simulate numerically such type of systems. The event-driven engine essentially performs these four steps:

1. Select the next event to be processed.
2. Process the event.
3. Schedule all possible new events generated in the previous step.
4. Return to the first step or end the simulation.

Since messages can be delayed after their emission, two types of events have to be distinguished: the *firing* events and the *reception* events. Step 1) selects the event with minimum time label and Step 2) is essentially a disjunction which evaluates if the selected event is a firing or a reception event, and applies r_i or s_i respectively. In the case of a firing event, new *reception* events should be scheduled as well. This is indicated in Step 3). Once the value of the microstate a_i is determined, the function $\tilde{t}_i$ is used to predict the next firing time. The previous prediction is then discarded and not used anymore.

Concerning the implementation of this framework, a crucial question concerns the data structure used to schedule the different types of events. The optimal one depends on the characteristics of particular model under consideration. In the general case, a priority queue is required, which can be implemented using, for instance, skip-lists (Reutimann et al., 2003).

The firsts studies which relate the event-driven paradigm with networks of spiking neurons were made by Pratt (1989); Watts (1994), although their formulation was quite different from the presented above. The simulator in Watts (1994) was restricted to a specific class of neuron models which use piecewise linear microstate trajectories in time, and it could only be applied to small networks. One of the papers which triggered more interest in this context is due to Mattia and Giudice (2000). Their approach allowed large-scale simulations reducing the number of *reception* events scheduled in what they called synaptic matrix (the event list). Many simulators have extended their software

frameworks to include this type of dynamics. For instance, NEURON (Hines and Carnevale, 2000), SpikeNET (Delorme and Thorpe, 2003) or SpikeNNs (Marian et al., 2002) to cite a few of them. The main limitation in the approach of Mattia and Giudice (2000) was the way to handle stochastic dynamics. Noise was included by means of additional events according to a Poisson process in the priority queue, causing the saturation of the data structure when the frequency of the external stochastic input was high. This was addressed in Reutimann et al. (2003), who proposed to use a stochastic stationary process in such a way that only N "extra" events were required. These N events (the predictions obtained via the function $\tilde{t}$) where computed using the Fokker-Planck equation to obtain numerically the required density functions under stationary conditions. This extension needs the use of large lookup tables where the required density functions are stored after an offline pre-calculation task. The use of lookup tables to characterize synaptic dynamics as well has been proposed in Ros et al. (2006).

It is important to note that this approach is not *always* better than the original synchronous approach. In particular, when very dense connectivity, high firing rates or abrupt changes in the ISI distributions occur, it is better to keep using the approach which integrates numerically the variables on a grid with certain resolution Δt. In Morrison et al. (2005) a distributed framework was presented which combined both synchronous and asynchronous updates. Currently, improving numerical simulations via the event-driven strategy is an active area of research, and the approach has been extended, for instance, to include synaptic conductances (Rudolph and Destexhe, 2006; Brette, 2006, 2007). For an extensive review of the different strategies used to simulate networks of spiking neurons see (Brette et al., 2007).

3.3 Mesoscopic modeling

Contrary to the previous approach we adopt a different strategy: we define the state of a neuron i as its next predicted firing time, or belief b_i which is reset to an initial value according to a predefined initial ISI probability distribution every time the neuron i fires. If a relevant event occurs, instead of discarding the prediction and determining the value of the microstate a_i, we use b_i as an evidence to obtain its new value b_i' using an inference procedure. Every time a neuron receives an pulse from other unit we update the belief according to Bayes' rule:

$$P(b_i'|b_i) = \frac{P(b_i|b_i')P(b_i')}{P(b_i)}. \tag{3.5}$$

As a consequence, to update the belief of a unit we need to define two probability distributions. An *unconditional* ISI probability distribution $P(b_i)$, used when a unit fires and the next spiking time is independent of previous predictions, and a *conditional* ISI probability distribution $P(b_i'|b_i)$, used

when an event influences the unit before its predicted next firing time. A specific model of spiking neurons can be defined using this technique, given that the two following mechanisms are provided: the two probability distributions or *belief updating rules*, which describe how the mesoscopic state b_i changes when i fires or receives a message, and the scheduler of the reception events.

We emphasize the main difference between the previous approach, (see Reutimann et al., 2003, for instance) and our technique. In their approach, every time a relevant event occurs the value of the prediction is discarded and two pseudorandom numbers are extracted: the first is required for fixing the microstate a_i, and the second for determining the next possible firing time $\tilde{t}_i$. Our approach does not fix the microstate and uses only one pseudorandom number extraction, resulting in a more efficient approach. The main caveat concerns the implementation of the ISI probability distributions. For the models considered here no lookup tables are required to simulate large networks.

In our strategy the statistical description of the microstate a_i is not retained. We just obtain the sequence of firing times. This is not a limitation, since it is widely assumed that the dynamics at the mesoscopic level (spike trains) concern the main aspect of neural information processing (Rieke et al., 1997), whatever the neural coding mechanism adopted (firing rates or exact timing). Moreover, the technique is not limited if dynamics at the level of the synapse is included, for instance spike-time-dependent plasticity (Bi and Poo, 1998), since this dynamics only considers the spike timing.

We now illustrate the proposed strategy starting with a simple microscopic model and derive the mesoscopic reformulation which reproduces the same dynamics in the temporal domain.

3.3.1 Mesoscopic reformulation

We provide a reformulation of the model presented in Section 3.1 in terms of the framework explained above, which reproduces identically the dynamics at the mesoscopic level. We replace the microstate a_i by $m_i = \langle b_i, L_i \rangle$, where b_i is the belief and L_i summarizes all the mesoscopic activity since the last firing time T_i^e. L_i can be viewed as an effective threshold which decreases deterministically every time an incoming pulse is received. It can be calculated from the temporal sequences of spikes $\langle T_k^j \rangle$ but we propagate it as a component of the mesoscopic state m_i for computational efficiency. We now describe how the reception events are handled and which are the rules of belief updating.

The model is defined with global connectivity and homogeneous coupling so each generated message affects equally all the units. Thus N iterations are sufficient to process one event even in the worst case that all units spike in unison. Moreover, since we set delay δ and refractory period t_{ref} homogeneously and equal to one, all units can be updated consistently at the firing time, avoiding the necessity of a priority queue in the implementation. Rather than being a limitation, this allows us to focus our analysis on the belief updating mechanism instead of the implementation engine. In the

continuous time model presented in Section 3.3.2 the priority queue will be required. We now specify the two ISI probability distributions and the model will be fully redefined.

Unconditional ISI probability distribution

This probability distribution is used when a unit fires and thus the next spiking time is independent of previous predictions. Consider a unit i which fires at time t, possibly simultaneously with n other units. During the transition from t to $t+1$, i is in its refractory period, insensitive to incoming events. At $t+1$, i receives an initial message of strength $\phi = n\varepsilon$ from those units which fired simultaneously with i. A new value of b_i is drawn assuming that evolution is only due to the spontaneous activity. Since the path towards the absorbing barrier L_i which an isolated unit takes can be interpreted as a random walk specified by a Bernoulli process with probability p, the new value of b_i is obtained according to the negative binomial distribution Feller (1968):

$$P_{NB}(b_i') = \binom{b_i - 1}{L_i - 2} p^{L_i - 1}(1-p)^{b_i - L_i + 1} \qquad \text{for } b_i = 1, 2, \ldots \tag{3.6}$$

where for clarity we have taken temporal values relative to $T_i^e + t_{ref}$, the last firing time of i plus the refractory period. Note that the quantity $(b_i - L_i + 1)$ is the number of failures, which in our case indicates the number of time-steps that i will delay its firing from $(T_i^e + t_{ref} + L_i - 1)$. This probability distribution has an infinite support. Thus, the microscopic state has a nonzero probability of crossing the threshold L_i at any time-step greater than $L_i - 1$ in the future. The first two moments of this probability distribution are:

$$\mu_{NB} = \frac{L_i - 1}{p} \qquad \sigma^2_{NB} = \frac{(L_i - 1)(1-p)}{p^2} \tag{3.7}$$

Conditional ISI probability distribution

In the case of a reception event, the belief b_i of unit i, which was last updated at time-step t, is affected by a subsequent reception event of strength ϕ at time-step t'. Omitting time indices for clarity, we are interested in the rule which describes the update from $m_i = \langle b_i, L_i \rangle$ to $m_i' = \langle b_i', L_i' \rangle$.

The strength of the message ϕ is the number of units which fired at time-step $t' - 1$ multiplied by ε. We set $L_i' = L_i - \phi$ and apply Bayes' rule to obtain the new value b_i' using the previous prediction b_i as evidence. For the model under study the three probability distributions involved in the rule are negative binomial distributions. Figure 3.2 illustrates the procedure. The light grey region covering all possible trajectories of the microscopic state a_i within the interval $[T_i^e + t_{ref}, b_i]$ can be decomposed in two disjoint sets of possible realizations of a_i. The dark gray area contains the trajectories where unit i fires at time b_i' with effective threshold L_i', and the black one contains the trajectories where

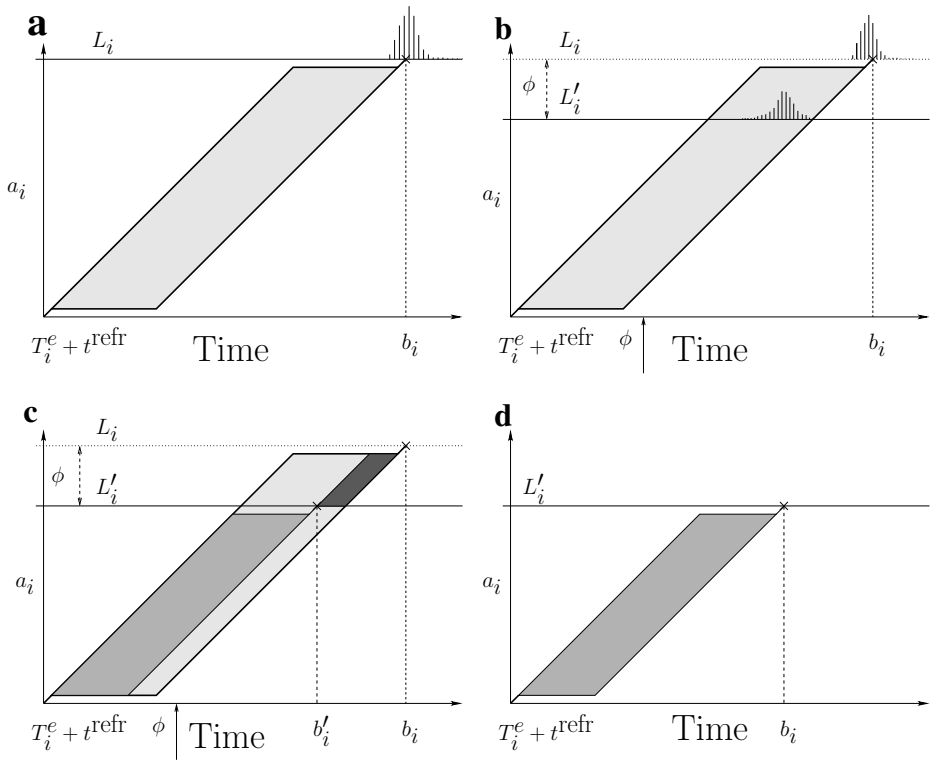

Figure 3.2: Illustration of the process involved in the belief updating of a given unit i. Horizontal axis denotes time and vertical axis the microstate variable. (a) After i has fired, the belief is obtained according to the unconditional ISI probability distribution $P(b_i)$. (b) When subsequent events affect i a conditional ISI probability distribution $P(b'_i|b_i)$ is defined. (c) The previous prediction (first packet of trajectories) can be decomposed in two disjoint regions ($P(b'_i)$, $P(b_i|b'_i)$, see text for details). (d) The belief updating is repeated with the new value b'_i.

a_i starts at L'_i and reaches L_i at time b_i. We are now interested in the probability distribution of the waiting time to reach L'_i state transitions (successes), where the total number of successes L_i and failures $(b_i - L_i + 1)$ are known from the previous prediction. Using the ISI probability distribution of (3.6) in Bayes' rule (3.5) we get for the numerator:

$$P_{NB}(b_i|b'_i)P_{NB}(b'_i) = \binom{b_i - b'_i - 1}{L_i - L'_i - 1}\binom{b'_i - 1}{L'_i - 2} p^{L_i - 1} q^{b_i - L_i + 1}. \tag{3.8}$$

The denominator can be obtained summing over all possible values of b'_i:

$$\begin{aligned} P_{NB}(b_i) &= \sum_{b'_i = L'_i - 1}^{b_i - (L_i - L'_i)} \binom{b_i - b'_i - 1}{L_i - L'_i - 1}\binom{b'_i - 1}{L'_i - 2} p^{L_i - 1} q^{b_i - L_i + 1} \\ &= \sum_{b'_i = 0}^{b_i - L_i + 1} \binom{b_i - b'_i - L'_i}{L_i - L'_i - 1}\binom{b'_i + L'_i - 2}{L'_i - 2} p^{L_i - 1} q^{b_i - L_i + 1}. \end{aligned}$$

Decomposing the sum:

$$P_{NB}(b_i) = \left(\sum_{b'_i=0}^{b_i-L'_i} \binom{b_i - b'_i - L'_i}{L_i - L'_i - 1} \binom{b'_i + L'_i - 2}{L'_i - 2} \right.$$
$$\left. - \sum_{b'_i=b_i-L'_i+1}^{b_i-L'_i} \binom{b_i - b'_i - L'_i}{L_i - L'_i - 1} \binom{b'_i + L'_i - 2}{L'_i - 2} \right) p^{L_i-1} q^{b_i-L_i+1}.$$

The second term of the sum is always zero, since $b_i - (b_i - L'_i + 1) - L'_i < L_i - L'_i - 1$ and $L_i - L'_i = \phi > 0$. Applying the following equality (Feller, 1968):

$$\sum_{k=0}^{r} \binom{r-k}{m} \binom{s+k}{n} = \binom{r+s+1}{m+n+1},$$

we get:

$$P_{NB}(b_i) = \binom{b_i - L'_i + L'_i - 2 + 1}{L_i - L'_i - 1 + L'_i - 2 + 1} p^{L_i-1} q^{b_i-L_i+1} = \binom{b_i - 1}{L_i - 2} p^{L_i-1} q^{b_i-L_i+1} \tag{3.9}$$

Using (3.8) and (3.9) we obtain the **negative hypergeometric distribution** (Johnson and Kotz, 1969):

$$P_{NH}(b'_i | b_i) = \frac{\binom{b_i - b'_i - 1}{L_i - L'_i - 1} \binom{b'_i - 1}{L'_i - 2}}{\binom{b_i - 1}{L_i - 2}},$$

for $b'_i = L'_i - 1, \ldots, b_i - \phi$, the interval of values restricted by the previous belief b_i. This probability distribution, unlike the one of Eq.(3.6) has finite support, showing that the number of possible firing times are constrained from previous predictions. The first two moments of this distribution applied to this particular model are Johnson and Kotz (1969):

$$\mu_{NH} = \frac{b_i(L'_i - 1)}{L_i - 1}, \qquad \sigma^2_{NH} = \frac{b_i(b_i + L_i - 1)(L'_i - 1)\phi}{L_i(L_i - 1)^2}. \tag{3.10}$$

Figure 3.3 shows several examples of this probability distribution. For small values of the event strength ϕ, the distribution is peaked at the right part of the domain, indicating that the original belief b_i will only be advanced a few time steps. This case is typical when neurons are weakly coupled and behave approximately as independent oscillators. For intermediate values of ϕ, the distribution takes a symmetric flat shape in the middle of the domain. Finally, when the incoming pulse is very strong, the next predicted firing is advanced considerably, and the unit is very likely to fire at the next time step. This is a characteristic situation when units are strongly coupled and receive many incoming pulses simultaneously.

Note that once the first prediction has been made according to Equation (3.6), the rate p, which represents the noise in this model, is no longer relevant in subsequent updates, where only a number counting method is applied until i fires again.

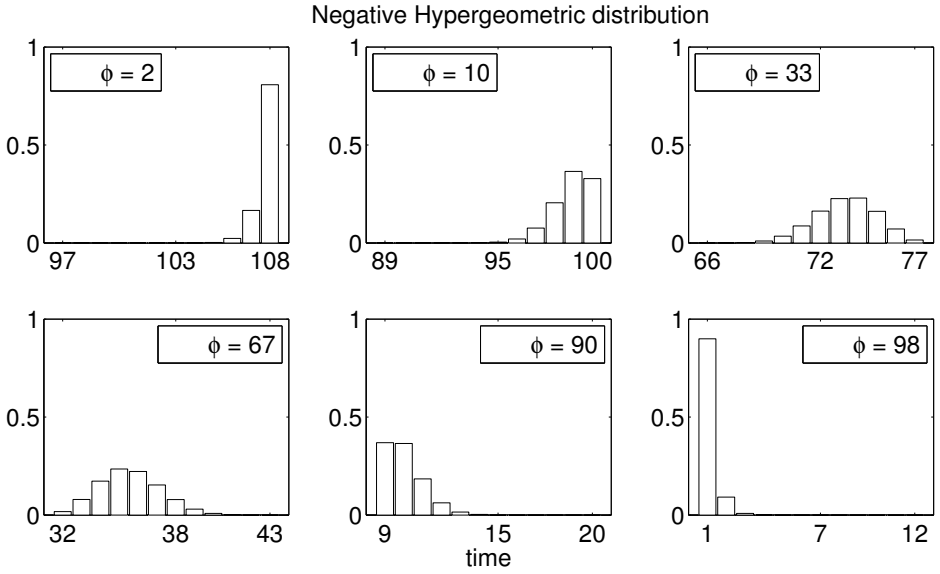

Figure 3.3: Examples of negative hypergeometric distribution. Threshold was fixed to $L_i = 100$ and belief $b_i = 110$. The panels show the resulting probability distribution for several values of the strength of the received event $\phi = \{2, 10, 33, 67, 90, 98\}$.

3.3.2 Extension to continuous time domain models

We now extend the previous discrete model by replacing the Bernoulli process governing the evolution of the microstate variable by its continuous counterpart, the Poisson process. The resulting model corresponds to the Poisson excitation model (Tuckwell, 1988), where excitatory inputs occur at random according to a simple Poisson process and cause the microstate to increase. As in the discrete case, this stochastic evolution can also be interpreted as an intrinsic source of noise.

We analyze two possible continuous extensions: the Poisson process of rate p and a version obtained by equating the moments of the gamma density function to those of the negative binomial distribution (method of moments).

To accommodate the dynamics of the discrete case, we choose the same values for the transmission delay and the refractory period, i.e. $\delta = t_{ref} = 1$ and full connectivity as well. The fact that events may occur at non-integer times prevents a synchronous update of all the beliefs at firing times as in the discrete case. Therefore, a priority queue where firing and reception events coexist is required as explained in the general framework of Section 3.3.

Continuous version I: Poisson process approximation

We take the limit in which the size of a time-step Δt tends to zero but the number of state transitions to reach the threshold L remains constant. In this case the probability p of a state transition

tends to zero and the amount of time-steps before firing tends to infinity. Taking again values of the belief relative to $T_i^e + t_{ref}$ we obtain in the limit of $\Delta t \to 0$ the following gamma distribution: [1]

$$p_\Gamma(b_i) = \frac{\beta^\alpha}{\Gamma(\alpha)} b_i^{\alpha-1} e^{-\beta t} \qquad \text{for } b_i \geq 0 \,. \tag{3.11}$$

where $\alpha = L-1$ and $\beta = p$. This ISI probability distribution can be understood as the sum of $L-1$ exponentially distributed variables with $\lambda = p$. Each of these variables represents the waiting time between two state transitions, which are now Poisson events with rate p. The first two moments of the gamma density applied to the model are:

$$\mu_\Gamma = \frac{L-1}{p} \,, \qquad\qquad \sigma_\Gamma^2 = \frac{L-1}{p^2} \,. \tag{3.12}$$

Note that the mean μ_Γ is the same as μ_{NB} of the discrete case, Equation (3.7), but the variance is overestimated, and only coincides with the discrete case in the limit of $p \to 0$.

Following a similar procedure as in section 3.3, we apply Bayes' rule to obtain the conditional ISI density function. An equivalent picture as Figure 3.2 can be used to illustrate the different terms involved in the rule. Bayes' rule (3.5) can be written in this case as:

$$g'(b_i'|b_i) = \frac{g(b_i|b_i')h'(b_i')}{h(b_i)} \,. \tag{3.13}$$

In this case $h'(b_i')$ is the density function of the waiting time until the threshold lowered by the incoming messages $L_i' = L_i - \phi$ is reached. We use the gamma density function with $\alpha = L_i' - 1$ and $\gamma = p$:

$$h'(b_i') = \frac{\gamma^\alpha}{\Gamma(\alpha)} {b_i'}^{\alpha-1} e^{-\gamma b_i'} \qquad\qquad \text{for } b_i' \geq 0. \tag{3.14}$$

Similarly $g(b_i|b_i')$ is the density function of the waiting time until an effective threshold $(L_i - 1)$ is reached at time b_i, assuming that unit i starts with activation state $(L_i' - 1)$ at time b_i'. It represents another gamma density, in this case with $\beta = \phi$ and $\gamma = p$:

$$g(b_i|b_i') = \frac{\gamma^\beta}{\Gamma(\beta)} (b_i - b_i')^{\beta-1} e^{-\gamma(b_i - b_i')} \qquad\qquad \text{for } (b_i - b_i') \geq 0. \tag{3.15}$$

The normalization factor is $h(b_i)$, which is the marginal probability of b_i:

$$h(b_i) = \int_{-\infty}^{\infty} \frac{\gamma^\beta}{\Gamma(\beta)} (b_i - b_i')^{\beta-1} e^{-\gamma(b_i - b_i')} \frac{\gamma^\alpha}{\Gamma(\alpha)} {b_i'}^{\alpha-1} e^{-\gamma b_i'} db_i'.$$

Using the substitution $t = b_i'/b_i$ and eliminating the regions where the integral is zero we get:

$$h(b_i) = \frac{\gamma^{\beta+\alpha} e^{-\gamma b_i} {b_i}^{\beta-1}}{\Gamma(\alpha)\Gamma(\beta)} \int_0^1 (1-t)^{\beta-1} (t b_i)^{\alpha-1} b_i dt = \frac{\gamma^{\alpha+\beta} e^{-\gamma b_i} {b_i}^{\alpha+\beta-1}}{\Gamma(\alpha+\beta)}. \tag{3.16}$$

[1] Note that unlike the discrete case, beliefs are not restricted to have values greater than the threshold L. Units in the continuous time domain can advance and fire before an interval of time L.

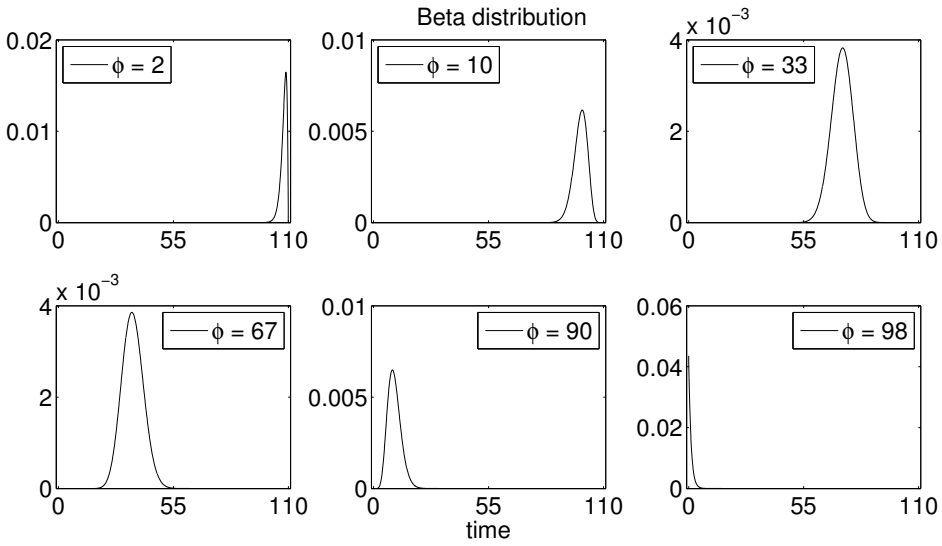

Figure 3.4: Examples of Beta distribution. Threshold was fixed to $L_i = 100$ and belief $b_i = 110$. The panels show the resulting probability distribution for several values of the strength of the received event $\phi = \{2, 10, 33, 67, 90, 98\}$.

We use Eqs. (3.14), (3.15) and (3.16) to find the density function:

$$g'(b_i'|b_i) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \left(1 - \frac{b_i'}{b_i}\right)^{\beta-1} \left(\frac{b_i'}{b_i}\right)^{\alpha-1} \frac{1}{b_i} \qquad \text{for } b_i' \in [0, b_i],$$

which is a **beta distribution** with parameters $\alpha = L_i' - 1$ and $\beta = \phi$ and rescaled by the inverse of the previous value of the belief, $1/b_i$.

The beta distribution is thus the analogous counterpart of the negative hypergeometric distribution in the continuous time domain. Figure 3.4 show several examples of density functions of this distribution using the same values as in the equivalent Figure 3.3.

Again, the parameter p is no longer relevant in subsequent updates of the belief. Note also that the previous value of the belief b_i defines the bounded support of the beta distribution, so the new belief is constrained by the previous prediction. The first two moments of this distribution applied to the model are:

$$\mu_\beta = b_i \frac{L_i' - 1}{L_i - 1}, \qquad \sigma_\beta^2 = b_i^2 \frac{(L_i' - 1)\phi}{(L_i - 1)^2 L_i}. \tag{3.17}$$

We compare them with the moments of the negative hypergeometric distribution in the discrete case, see Eq. (3.10). The means coincide, but the variance is overestimated by a factor $(b_i + L_i - 1)$ instead of b_i.

Table 3.1: Moments of the related ISI probability distributions.

Discrete (waiting time)	Continuous (Poisson)
$\mu_{NB} = \frac{L_i - 1}{p}$ $\sigma^2_{NB} = \frac{(L_i-1)(1-p)}{p^2}$	$\mu_{\Gamma} = \frac{L-1}{p}$ $\sigma^2_{\Gamma} = \frac{L-1}{p^2}$
$\mu_{NH} = \frac{b_i(L'_i-1)}{L_i-1}$ $\sigma^2_{NH} = \frac{b_i(b_i+L_i-1)(L'_i-1)\phi}{L_i(L_i-1)^2}$	$\mu_{\beta} = \frac{b_i(L'_i-1)}{L_i-1}$ $\sigma^2_{\beta} = \frac{b_i^2(L'_i-1)\phi}{L_i(L_i-1)^2}$
Discrete (failures)	**Continuous (Fitting moments)**
$\mu_{NB} = \frac{(L_i-1)(1-p)}{p}$ $\sigma^2_{NB} = \frac{(L_i-1)(1-p)}{p^2}$	$\mu_{\Gamma'} = \frac{(L_i-1)(1-p)}{p}$ $\sigma^2_{\Gamma'} = \frac{(L_i-1)(1-p)}{p^2}$
$\mu_{NH} = \frac{(L_i-1)(1-p)}{p}$ $\sigma^2_{NH} = \frac{b_i(b_i+L_i-1)(L'_i-1)\phi}{L_i(L_i-1)^2}$	$\mu_{\beta'} = \frac{(L_i-1)(1-p)}{p}$ $\sigma^2_{\beta'} = \frac{b_i^2(L'_i-1)\phi}{((1-p)(L_i-1)+1)(L_i-1)^2}$

Continuous version II: method of moments

Other extensions can be obtained depending on how the continuous density functions are chosen, or which limiting process is taken. For example, we get another continuous model if we consider the value of the beliefs relative to $T_i^e + L_i$, instead of T_i^e. This situation is equivalent to considering in the discrete case the belief b_i as the number of *failures*, or delayed time-steps from $T_i^e + L_i$, instead of the waiting time from T_i^e. The discrete dynamics is not modified by this new counting mechanism (the negative binomial is just shifted), but in the continuous case the beliefs are prevented from taking values smaller than the number of required state transitions to reach the threshold. Then, with the following suitable choice of the gamma density parameters: $\alpha = (L_i - 1)(1 - p)$ and $\beta = p$, both continuous and discrete probability distributions will have the same first two moments.

$$\mu_{\Gamma'} = \frac{(L_i - 1)(1 - p)}{p} \qquad \sigma^2_{\Gamma'} = \frac{(L_i - 1)(1 - p)}{p^2} \ . \tag{3.18}$$

Again, applying Bayes' rule leads to another rescaled beta distribution, but now with parameters $\alpha = (L'_i - 1)(1 - p)$ and $\beta = \phi(1 - p)$ and rescaled in the time interval $[L_i - \phi, b_i - \phi]$, as the discrete version. The moments of this distribution are:

$$\mu_{\beta'} = b_i \frac{L'_i - 1}{L_i - 1} \qquad \sigma^2_{\beta'} = \frac{b_i^2 (L'_i - 1)\phi}{(L_i - 1)^2 \left((1 - p)(L_i - 1) + 1\right)} \ . \tag{3.19}$$

As before there is only one difference with the discrete case in the standard deviation, with the factor $b_i + L_i - 1$ instead b_i, and the factor $((1-p)(L_i-1)+1)$ instead of L_i which coincide when $p \to 0$. To facilitate the comparison between the three different versions of the model we group in Table 3.1 all the moments of the used probability distributions and density functions. The negative binomial (NB) and negative hypergeometric (NH) distributions are used in the discrete model (left side of the table). Regardless of whether the waiting time (which is equivalent to take the belief with respect to $T_i^e + t_{ref}$) or the number of failures (which is equivalent to taking the belief with respect to $T_i^e + t_{ref} + L - 1$) is considered as the belief variable, both use the same parameters. In the continuous version (right side of the table), the Poisson approximation uses a gamma density function with parameters $\alpha = (L-1)$ and $\beta = p$ and a beta density function with parameters $\alpha = L_i' - 1$ and $\beta = \phi$, whereas the approximation which fits the moments uses a gamma density function with parameters $\alpha = (L-1)(1-p)$ and $\beta = p$ and a beta density function with parameters $\alpha = (L_i'-1)(1-p)$ and $\beta = \phi(1-p)$.

3.4 Simulation results

We performed simulations to evaluate the proposed technique. First, we compare the computational complexity of the mesoscopic version of the model against the original microscopic model and

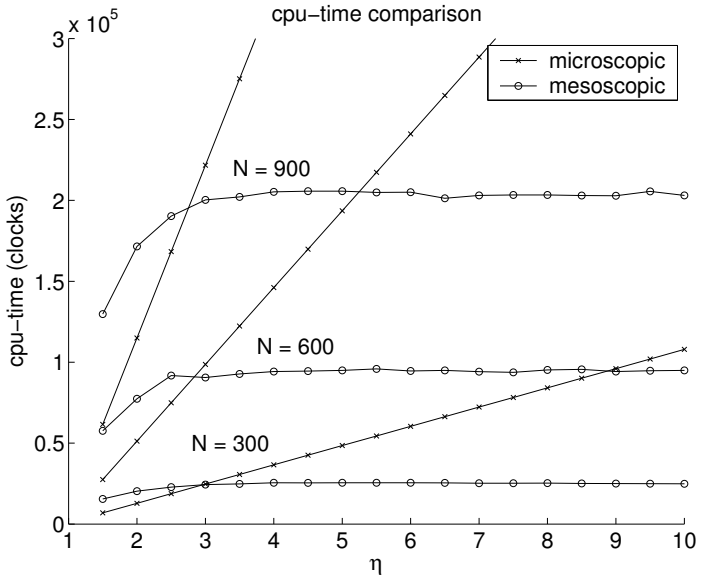

Figure 3.5: Comparison of the computational cost between the original microscopic model and the mesoscopic event-based reformulation in function of the degree of interaction for different sizes of the ensemble. We use a full-connected network and the noise rate is $p = 0.9$. No lookup tables were computed a priori. For $\eta > \eta^* \sim 2.5$ the mesoscopic version is more efficient.

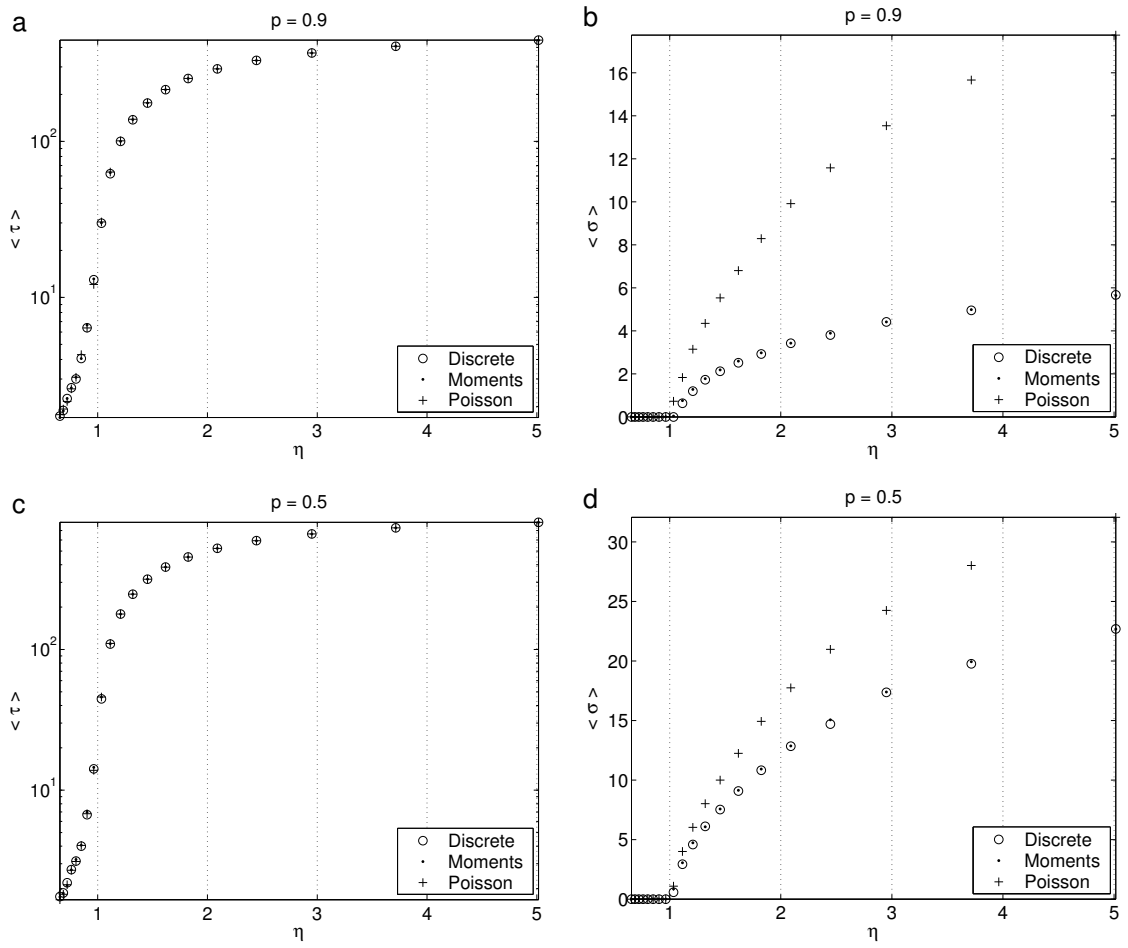

Figure 3.6: Simulation results of the event-driven model for the discrete and two possible continuous extensions. The extension which considers the spontaneous evolution as a Poisson process with rate p is indicated by **Poisson** whereas **Moments** indicates the model which uses the method of moments (*see the text*). Upper figures show **(a)** the mean of the ISI in logarithmic scale and **(b)** the respective standard deviation over 500 simulations with $p = 0.9$. Lower figures **(c, d)** show the same statistics but for a noise rate of $p = 0.5$. Simulations were performed over an ensemble of size $N = 200$ and threshold $L = 500$. The coupling strength ε was varied around the critical value of $\eta = 1$. Both continuous models fit well the mean ISI. The standard deviation is well fitted by the **Moments** version while the **Poisson** version significantly overestimates it for higher values of the rate p.

then we compare the accuracy of the continuous versions with the discrete one.

Since the model does not assume any particular time units, we quantify the performance of a given model in terms of the number of updates (in cpu-time) within an ISI of length τ. This measure allows us to easily compare both models. Figure 3.5 shows this quantity for different values of the characteristic parameter η which indicates the degree of interaction between the units of the ensemble. Higher values of η correspond to small interaction between the units, and vice versa. Clearly, the mesoscopic event-based version of the model is suitable when spikes are rare events in time, since

time-steps where neurons do not fire are not simulated. It turns out that for regimes of $\eta < 1$ the event-based version does not perform better than the microscopic version, since spikes occur at every time-step (Rodríguez et al., 2002) and hence, the number of updates is the same in both approaches.

The computational complexity of the original microscopic model is proportional to the ISI $O(\tau N)$, since in an ISI of size τ all N states need to be updated at each time step. The event-based strategy, however, requires the update of N predictions every time an event is produced because we are using a full connected network. The homogeneous coupling allows to update a unit in constant time. Thus an average of N^2 updates will be required within an ISI leading to an $O(N^2)$ computational complexity. Without the requirement of simulating the evolution of the microstate variable, the cost does not depend on the size of the ISI τ unlike the microscopic approach. As can be observed in Fig.(3.5) the number of updates of the mesoscopic approach is constant and the event-driven approach performs better than the original approach the higher the value of η (the lower the coupling). A decrease is observed for $\eta \sim 2$ and below, which is caused because in this regime simultaneous spikes occur often and can be grouped together into one single pulse. The time independence is a common feature of all event-driven approaches.

Figure 3.6 shows simulation results of the discrete and both continuous versions of the model, where L and N are fixed and the coupling strength is varied to cover a significant parameter space of η. We simulate the model starting at a regime of small values of ε (weak coupling) and progressively increase it until all units emit spikes synchronized in unison. As previously explained, the event-based strategy allows the model to be simulated exactly, without precision errors. The mean ISI $\langle\tau\rangle$ and the dispersion $\langle\sigma\rangle$ for a unit are averaged over all the ensemble and over 500 initial conditions. We report exact agreement between mesoscopic and microscopic versions as expected from the analytical results (not shown in the figure for clarity) and also the mean of the ISIs is reproduced with accuracy in the continuous versions. As for the standard deviation, the approach based on the method of moments reproduces much better the discrete dynamics than the Poisson approximation, which is more similar if p is low, as theory suggests (note the differences between the upper-right plot and the lower-right plot). Figures 3.6.b and 3.6.d show a rapid decrease in the standard deviation which vanishes at $\eta = 1$, where the coupling strength reaches a critical value (Kaltenbrunner et al., 2007b).

3.5 Conclusions

The presented framework is useful and convenient for modeling large-scale neural populations when the detailed evolution of the microscopic variables is not the main aspect of analysis and efficient simulations are needed.

In general two critical points arise regarding the computational efficiency of all event-based ap-

proaches. On the one hand, the necessity of an ordered data structure for storing the events for models with heterogeneous δ and t_{ref}. On the other hand, for a general case with heterogeneous efficacies, and high connectivity the cost of processing one event is $O(N^2)$. These are inherent limitations which can only be alleviated using parallel and distributive computing (Morrison et al., 2005) for which the approach can also be applied.

Despite the simplicity of the model presented, the related ISI probability distributions have been successfully used many times for modeling neural data. See, for example (Bishop et al., 1964), (Bair et al., 1994) where the gamma density function is used. This is not surprising, since empirical ISI densities under stationary background activity often have three basic forms (Tuckwell, 1988): for very large excitatory pulses, neurons fire at the reception of one spike and the ISI density is exponential-like; if few pulses occurring in a short enough interval are sufficient to trigger a spike, the ISI density is the gamma density analyzed here. Finally, if the threshold is large compared to the average strength of the pulses, the ISI density has Gaussian appearance. We notice the great similarity between the first-passage-time density function of the Ornstein-Uhlenbeck Model and the gamma density function except for the long tail of the first case.

Indeed, the presented technique is not restricted to the particular model analyzed here. Inhibition in the form of negative pulses can easily be incorporated given that no reflecting barrier is imposed at the zero value of the microstate as in the case of Stein's model (Stein, 1967). For example, the Wiener process (Brownian motion) with positive drift governing the microstate evolution can also be modeled using our approach. In that case, the *inverse Gaussian* density function would be the unconditional ISI density function to be used. In general the method is applicable provided the ISI density function has a closed analytical form and when the underlying evolution of the microstate is such that the update is independent of the time at which an event affects the neuron. It is worth emphasizing that any relevant magnitude which can be obtained from the spike history can be captured by the presented approach, given that the required computations can be performed in an event-driven basis. An example in this direction would be to incorporate the dynamics of slow variables whose time constants may span several ISIs.

The approach also has potential applications in the context of pulse-coupled oscillators when a linear (or a piecewise linear) evolution is considered in the phase variable. For example, it can be used for improving the numerical method used in deterministic models (Strang and Ostborn, 2005) as well as in the linear pulse-coupled oscillator where noise is applied only every time an event occurs (Ernst et al., 1998).

Summarizing, under simplifying assumptions on the equation governing the microstate variable the high demand placed on computational resources for simulating stochastic populations of neurons can be alleviated using the proposed technique.

Chapter 4

Phase transition and self-organization

In this chapter we study the dynamics of the neural network model presented in the previous chapter. We give evidence of the presence of a phase transition accompanied by an hysteresis effect. Phase transitions are mostly studied in statistical physics and can be identified by abrupt changes in one or more physical properties of a system. They occur in the limit of a large system size, in our case, when the number of units N tends to infinity, which is also denoted as the thermodynamic limit.

In the model under consideration, if we start from a weakly coupled network (small ε) and progressively increase the coupling strength ε between the units, the network shows an abrupt change from a noisy state where it is dominated by the stochastic, irregular dynamics and a robust state where activity is self-sustained, and the network is partitioned into several clusters showing a periodic firing pattern. If the coupling strength ε is further increased from this critical value, then some clusters are merged. Eventually, for a very large coupling strength, a state of fully synchronization with all neurons spiking in unison within one giant cluster is reached.

We derive a plasticity rule which modifies the individual coupling strengths between the units using only local information accessible at the level of a single synapse. With this synaptic dynamics, the network self-organizes into the critical state and stays stable in an effective synchronization state.

In the self-sustained regime, we also report hysteresis. Starting from above the critical coupling strength the network produces a periodic firing pattern. If the coupling is then progressively decreased, the pattern remains constant and 'remembered' by the network until the critical ε is reached again. The transmission delay is essential to obtain the clustering and hysteresis effects.

4.1 Introduction

One of the simplest models which show phase transitions is a system of interacting Ising spins, similar to the Boltzmann network we have briefly presented in the first chapter. The heat capacity

of a system of N spins, which corresponds to the second derivative of the partition function Z, see Eq. (2.5), also represents the fluctuations of the energy. For a certain value of the temperature parameter $1/\beta$, the heat capacity diverges in the limit of infinite size, meaning that the system can absorb or release energy without any change in the temperature.

Interesting examples of phase transitions can be also found not only in physical models, but also in other areas of science, such as artificial intelligence. A major example is the sharp transition between solvable and unsolvable random instances of the classical $k-$SAT problem (Selman et al., 1992; Kirkpatrick and Selman, 1994). The $k-$SAT problem is the problem of deciding whether one can satisfy simultaneously a set of M constraints between N boolean variables, where each constraint is a clause built as the logical OR involving K variables (or their negations). Depending on the ratio of number of clauses to number of variables $\alpha = M/N$, problems are solvable for $\alpha < \alpha_c$, and problems are unsolvable for $\alpha > \alpha_c$. Most computationally difficult problems correspond to those within the critical range $\alpha = \alpha_c$. The survey propagation algorithm (Braunstein et al., 2005; Mézard et al., 2002) is inspired in those ideas, and represents the best method for solving large instances of random problems.

It has been postulated that optimal information processing in complex systems occurs near a transition between an ordered and an unordered regime of dynamics (Packard, 1988; Langton, 1990). Although this idea is very popular, only a few realizations of this claim has been observed. The work of Bertschinger and Natschlager (2004) in the context of the liquid state machine (Maass et al., 2002; Jaeger, 2001) could represent an example. They use a recurrent network of binary threshold units with external stochastic input streams. By changing parameters such as the variance of the synaptic weights or the average number of connections of a given unit, the system changes between an ordered and chaotic dynamics. This network is connected to a layer of readout neurons which perform an online (real-time) linear combination of the states of the units of the recurrent network. They propose as a measure for computational power the *separation property*. Roughly speaking, this property is satisfied by a network if, given two different input streams that should produce different outputs, the recurrent network is driven into two significantly different states which can be easily classified using the linear readout neurons. A similar study was proposed in (Legenstein and Maass, 2007) using microcircuits which consisted of a recurrent network of leaky-integrate-and-neurons instead of binary recurrent networks. For a review of these ideas, see Haykin et al. (2007).

More recently, Kinouchi and Copelli (2006), using a network of oscillators similar to the one we analyze here, showed that the sensitivity and dynamic range of the network are maximized at the critical point of a non-equilibrium phase transition. The dynamic range is defined as the stimulus interval where variations in intensity can be robustly coded by variations in the activity of the network. The network used for that study is a regular random graph where a unit j is connected with K_j

neighbors. As in our model, a unit j can fire because of the interplay of two processes: on one hand, a stochastic Poisson process and on the other hand, the coupling with adjacent units. Instead of modifying the coupling strength as in our case, they vary the average branching ratio $\sigma = \langle \sigma_j \rangle$, where σ_j is defined as the average number of units excited by the jth unit, $\sigma_j = \sum_i^{K_j} p_{ij}$. There exists a critical value of $\sigma = \sigma_c = 1$ for which the dynamic range reaches a maximum. For sub-critical networks $\sigma < \sigma_c$, the network activity is not self-sustained and dies after some time-steps. Conversely, for super-critical networks $\sigma > \sigma_c$, the spontaneous activity of the network masks the stimuli.

The Self-Organized Criticality (SOC) property has been proposed by Bak et al. (1987) as a mechanism for some dynamical systems which spontaneously evolve to a "critical state" that lacks a characteristic length, i.e. presents structural and/or temporal scale-invariance of one or more macroscopic magnitudes. Remarkably, the critical state is reached independently of the initial conditions and without fine tuning of any external control parameter. Many dynamical systems have been linked to SOC, including earthquakes (Bak et al., 2002), forest fires (Malamud et al., 1998), piling of granular media (Frette et al., 1996), financial markets (Lux and Marchesi, 1999) or stick-slip processes (Feder and Feder, 1991). It is therefore of great interest to see whether a population of stochastic neurons like the one we have presented can exhibit the properties related to SOC, and which are the main consequences derived from this fact.

Synchronization is intimately related with phase transitions. In populations of weakly coupled oscillators, the onset of synchronization represents a phase transition (Winfree, 1967; Kuramoto, 2003). Interacting chaotic oscillators also exhibit a special kind of phase transition which closely resembles that seen in spin glasses (Kaneko, 1990). Recent work analyzes the existence of phase transitions for chains and lattices (Östborn, 2002; Östborn et al., 2003) of pulse-coupled oscillators with a particular, biologically inspired phase response curve.

In the context of pulse coupled oscillators, most studies consider only instantaneous coupling (i.e no delay in the message exchange) between the units which simplifies the analysis of the resulting dynamics. Under this restriction Mirollo and Strogatz (1990) demonstrated that certain types of identical leaky oscillators with global coupling synchronize for almost all initial conditions. Their result has been extended by Senn and Urbanczik (2000) allowing non-identical oscillators whose intrinsic frequencies, thresholds and couplings are heterogeneous within a certain range. They showed that non-leaky linear integrate-and-fire neurons synchronize for any initial condition for almost all parameter values of the system and speculate that, using perturbative arguments, their results might be still valid in the presence of a small leakiness. The influence of an absolute refractory period on the Mirollo-Strogatz model has been analyzed by Chen (1994) and Kirk and Stone (1997). The authors of these papers showed that the system approaches synchrony for almost all initial conditions if the

refractory period is below a critical value.

If a delay for the message exchange is added more complex forms of synchronization are observed. Ernst et al. (1998) showed empirically that for both, excitatory and inhibitory coupling, the neurons tend to cluster their activities. All neurons within a cluster are synchronized and fire in unison whereas the clusters are phase-locked with constant phase differences. The number of clusters of the system was inversely proportional to the length of the delay for inhibitory coupling. The stability of these clusters was analyzed in Timme et al. (2002) and similar phenomena were observed by Vreeswijk (1996) for coupling with α functions and have been proposed as a possible mechanism for neural information coding and processing in the form of synfire chains (Abeles, 1991; Bienenstock, 1995; Diesmann et al., 1999; Ikegaya et al., 2004). The importance of the delay for neural modeling has recently been addressed by Izhikevich et al. (2004); Izhikevich (2006), claiming that it allows an unprecedented information capacity, which translates into an increase of stable firing patterns in more realistic neural populations due to heterogeneous delays.

The rest of the chapter is structured as follows. In the next section we analyze numerical simulations of our simplified model and derive an analytic bound for the mean ISI $\langle\tau\rangle_{min}$ of the ensemble which is valid for all regions of the coupling strength. A detailed analysis of the phase transition is outside of the scope of this thesis and is presented in Kaltenbrunner et al. (2007b). In Section 4.3 we present the learning mechanism which drives the system toward an effective synchronization state, and derive a formula to calculate the time required for convergence to the critical state starting from any arbitrary initial state. We conclude this chapter in Section 4.4.

4.2 Dynamics of the model

The simplified model of neural ensembles under study was presented in Section 3.1. Its parameters are the number of units N, the threshold L, the noise rate p, and a global coupling term ε. The transmission delay of a message is denoted by δ and the refractory period as t_{ref}. In the following analysis, if otherwise stated, results are consistent with arbitrary δ and t_{ref}.

A characteristic parameter $\eta = \frac{L-1}{(N-1)\varepsilon}$ gives the ratio between the total change in activation needed for a neuron to fire and the one provided by the coupling with the rest of the population. For $\eta \gg 1$ the effect of the messages from the population into a given unit can be replaced with a linear decay of its *effective* threshold (Rodríguez et al., 2001) where fluctuations are averaged out, i.e. $L_{eff}(t) \approx L - \alpha t$. The fluctuations in this progressive decay are small for $\eta \gg 1$ and can be neglected. This can be regarded as a mean field approximation and Rodríguez et al. (2001) derived the following

formulas for approximating the mean and standard deviation of the ISI of a neuron in the ensemble:

$$\tau_{mf} = t_{ref} + \frac{L-(N-1)\varepsilon-1}{p}\ , \qquad \sigma_{mf} = \frac{\eta-1}{\eta}\frac{\sqrt{(L-(N-1)\varepsilon-1)(1-p)}}{p}\ . \tag{4.1}$$

This approach fails to describe the behavior of the system in regions of high coupling, were the feedback of the population introduces correlations between neurons that are no longer negligible. At $\eta = 1$ they predict one giant cluster with all units synchronized and firing at each time-step. This would only be true for a system without delay and refractory period. Equations (4.1) thus represent a lower bound for the mean period. In our system, however, the more important correlations due to the delayed message exchange become (i.e. the smaller becomes η), the bigger is the difference between τ_{mf} and the ISI of the units. This was first reported by Rodríguez et al. (2002), who found that for $\eta = 1$, after an initial transient, the system reaches one of a large number of periodic firing patterns, composed of several clusters. The same is true for $\eta < 1$, as can be observed in Figure 4.1, where raster plots of spikes of a system consisting of 100 neurons are shown. The irregular behavior of the system at $\eta = 1.2$ (Figure 4.1a) changes into a regular repetitive spiking pattern at $\eta = 1$ (Figure 4.1b) if the coupling is increased. If increased further, some clusters merge but the system continues with the phase-locked clustered firing as shown in Figure 4.1c for $\eta = 0.9$.

4.2.1 Numerical Simulations

To analyze the system dynamics, we observe how it system reacts to adiabatic [1] changes of the coupling strength. A simple analogy is useful to understand the procedure. Consider a cloud of particles that is slowly concentrated or diluted by increasing or decreasing the volume. We begin with a very dispersed cloud with little interaction between the particles and start to concentrate it in a stepwise manner. At every concentration step the interaction among the particles increases. At some point the process is inversed and the cloud is diluted again until reaching the original state. We therefore distinguish two different processes in our experiments, to which we refer as *concentration process* and *dilution process*. The particles are in our case the spiking units and the interaction can be measured via the relation of the threshold L and the coupling strength ε multiplied by the number of units N. This relation is reflected in the parameter η, which in our analogy represents the volume of the system.

In our experiments we choose a fixed set of N neurons with fixed threshold L. The only parameter allowed to change is the global interaction strength ε. We start with units at random initial states and at regions of high η (usually $\eta = 2$) where the system can be described with high accuracy by equations (4.1) and is ergodic in the sense that all accessible network states are visited over a long period of time. As explained before, the units in these regions can be viewed as nearly independent oscillators with

[1] We use the term *adiabatic* as it is used in quantum mechanics, meaning a "sufficiently slow" change of the system.

a threshold lowered by the mean activity induced by messages received from other units. The only difference to real independence is a period focusing effect described by Rodríguez et al. (2001). This results in a slightly lower (by a factor $(\eta - 1)/\eta$) standard deviation than the one of an independent unit with lowered threshold.

Once an experiment is started, we let the system evolve enough time-steps to avoid dependence on unnatural initial conditions (i.e. conditions that are not typical of the system) and let the ISI stabilize. Now we can start the concentration process by decreasing η in a stepwise manner. We achieve this via adapting ε. Notice that, although we change η by a constant $\Delta\eta$, the changes of ε are not constant due to the inverse relation of η and ε. After every decrease of η we let the system evolve enough time-steps until the ISI stabilizes again. This procedure is repeated until a value of η in the range between 1 and 0.5 is reached. Then we reverse the procedure and start the dilution process. We increase η in a stepwise manner until we reach again the starting value of η.

Results were analyzed using the same two statistics of the ISIs of the units as in the previous chapter for every value of η. We average the ISIs τ over all the ensemble and over all realizations to get the mean ISI $\langle\tau\rangle$. Similarly, we take the mean of the standard deviation σ of the ISIs over all the ensemble and all realizations. This average $\langle\sigma\rangle$ gives us an idea of the likelihood to end up firing

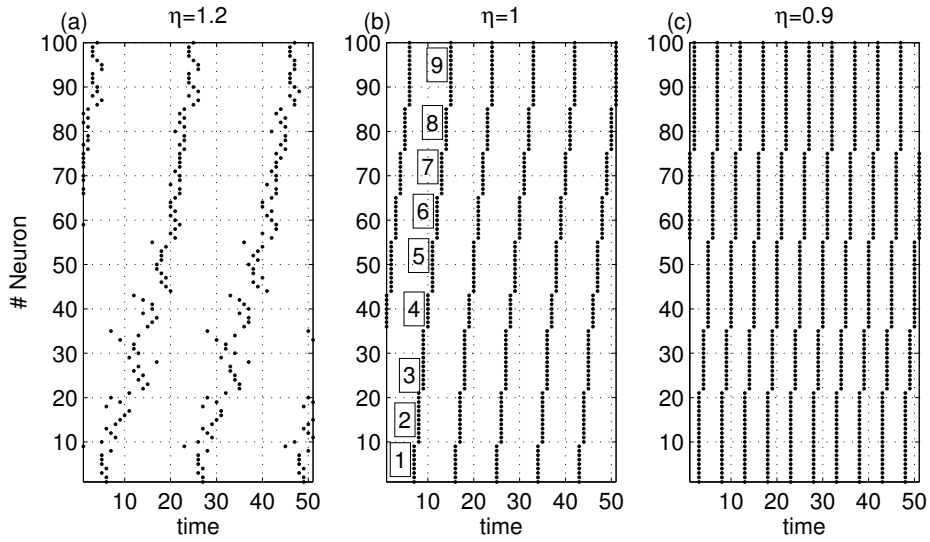

Figure 4.1: Raster plot of spikes (firing patterns) of 100 neurons (noise rate $p = 0.9$ and threshold $L = 100$) for different values of η. Simulation started with a random initial state for every neuron. Coupling strength is slowly increased every 100 time-steps and time was set to 0 after a transient of 50 time-steps. For clarity in the visualization the neurons are re-labeled according to their spike-time at $\eta = 1$. **(a)** We observe irregular firing at $\eta = 1.2$. **(b)** At $\eta = 1$ the neurons organize into 9 phase-locked clusters (labeled by the boxed numbers). **(c)** At $\eta = 0.9$ the number of phase-locked clusters is reduced to 5 since clusters number $1,2$; $4,5$; $6,7$ and $8,9$ merged into new bigger clusters. Note that the length of the ISI coincides with the number of clusters for $\eta = 1$ and $\eta = 0.9$.

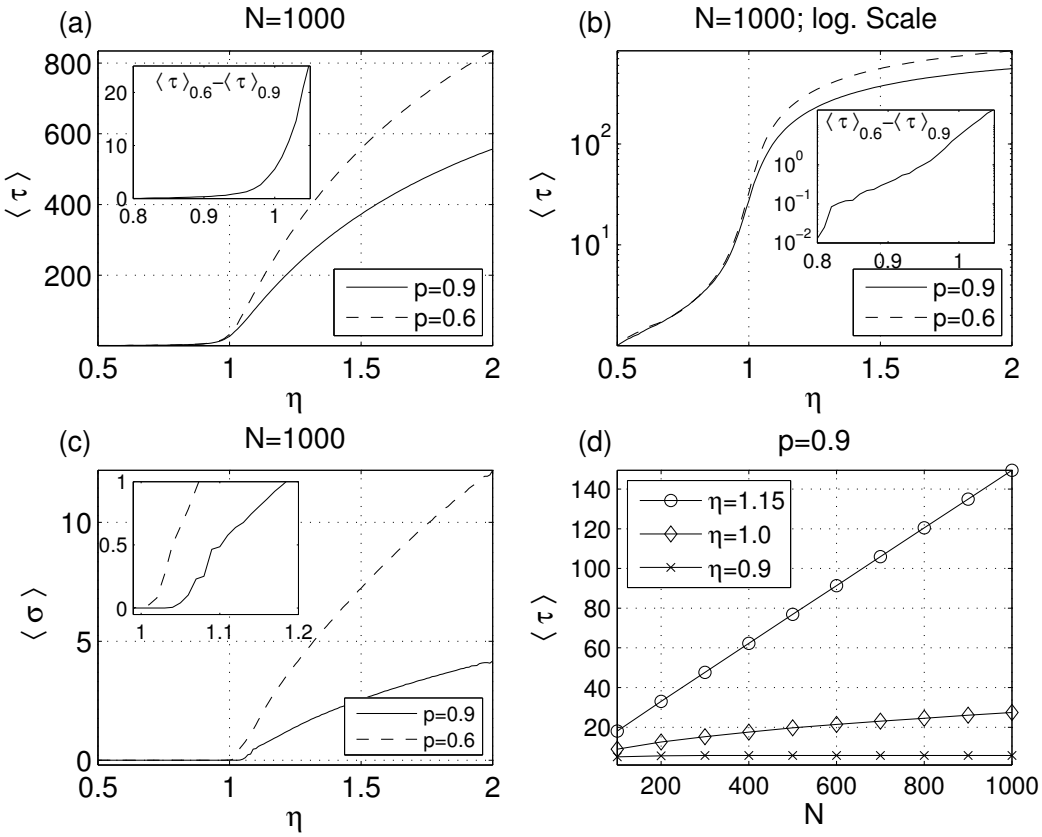

Figure 4.2: Numerical results of 1000 concentration experiments by incresing ε (i.e. decreasing η) on a network of $N = 1000$ neurons. Threshold L is fixed to N in all cases. **(a)** Dependence of $\langle\tau\rangle$ on η for two different noise levels p. Inset shows the difference of both curves in the interesting region around $\eta = 1$. **(b)** Same as (a) but in logarithmic scale. **(c)** Dependence of $\langle\sigma\rangle$ on η for the same values of p. Inset shows the region around $\eta = 1$. **(d)** Dependence of $\langle\tau\rangle$ on N for different values of η and $p = 0.9$. We can see a linear dependence of $\langle\tau\rangle$ on N for $\eta = 1.15$, a square root dependence for $\eta = 1$ and nearly no dependence for $\eta = 0.9$.

with phase-locked clusters for the given parameter values. The closer $\langle\sigma\rangle$ is to 0 the bigger is this likelihood. If the units in one experiment have all the same ISI, their standard deviation σ equals 0. The approximation σ_{mf} of equation (4.1) estimates $\langle\sigma\rangle$.

First we analyze only the concentration process, where we slowly increase the amount of coupling between the neurons. Figures 4.2a and 4.2b show the results for 1000 concentration experiments for noise rates of $p = 0.9$ (solid line) and $p = 0.6$ (dashed line). We used an ensemble size of $N = 1000$ neurons and can observe how the mean ISI $\langle\tau\rangle$ decreases as we increase the coupling. Initially, at high values for η, there is a clear dependence on the noise rate p which seems to disappear as we reach $\eta = 1$. A closer examination of the mean ISIs of both experiments in this region reveals that their difference for η close to 1 and below decays exponentially to 0 with decreasing η. (Shown in the

insets of Figures 4.2a and 4.2b).

When we analyze the deviation of the ISIs we also notice a change in the behavior of the system if we approach $\eta = 1$. The mean deviation $\langle\sigma\rangle$ of several experiments drops to 0 when the critical value of η is reached, indicating that the units organize into clusters and fire phase-locked, all with the same ISI. Figure 4.2c shows this effect for the concentration experiments with the two different noise rates analyzed before. In the inset we notice that for a noise level of $p = 0.9$ (solid line) the onset of phase-locking already happens at $\eta \approx 1.04$, which can be explained by the finite size N of the network. One would expect that the greater N the faster is this decay and, at the thermodynamic limit, the mean ISI is independent of p for all values of $\eta < 1$.

The curves for $p = 0.9$ and $p = 0.6$ at $\eta < 1$ are practically identical for both statistics if the system is further concentrated, meaning that the noise looses all its influence for $\eta < 1$. Therefore, the activity of the network in this case is self-sustained. Even if we remove all the stochastic activity and set $p = 0$, the units continue firing in its periodic orbit.

The strange shape of the curves in the logarithmic scale of Figure 4.2b suggests that apart from the elimination of the dependency on the noise rate p something else is going on around the value of $\eta = 1$. To investigate this point further we observed the dependence of the ISI $\langle\tau\rangle$ on the ensemble size N. Figure 4.2d reveals a quite different kind of dependence for different values of the coupling parameter η. For $\eta = 1.15$ (line with circles) we observe a linear dependence of $\langle\tau\rangle$ on N, whereas for $\eta = 0.9$ (line with crosses) the value of the ISI stabilizes once a certain number of neurons is in the ensemble ($N \geq 300$) and does not show any dependence on the ensemble size. At $\eta = 1$ (line with diamonds) a relationship of type $\sqrt{N} \sim \langle\tau\rangle$ is observed. The higher the value of N the big is the difference in $\langle\tau\rangle$ for the three values of η, indicating that a phase transition occurs around the critical value of $\eta = 1$.

Hysteresis effect

After having analyzed the concentration process experimentally we are interested in what happens if we invert the process. Instead of increasing the coupling strength, we decrease it in a stepwise manner. As described previously, we call this type of experiment dilution process. If we combine concentrations and dilutions to obtain a cyclic process we notice a hysteresis effect comparing the mean ISIs $\langle\tau\rangle$ of both processes for values of η close to 1 and below.

Figure 4.3 shows this effect for 3 different starting points of the dilution process. The dotted line represents the mean ISI $\langle\tau\rangle$ of the concentration process of 1000 experiments. When the concentration process stops and the dilution process is started, τ and therefore also $\langle\tau\rangle$ remain constant until a dilution of $\eta > 1$ is reached. The solid line with circles represents a dilution process stating at $\eta = 0.5$ and the dashed line with $+$ markers one starting at $\eta = 0.9$. In both cases the ISI remains unchanged

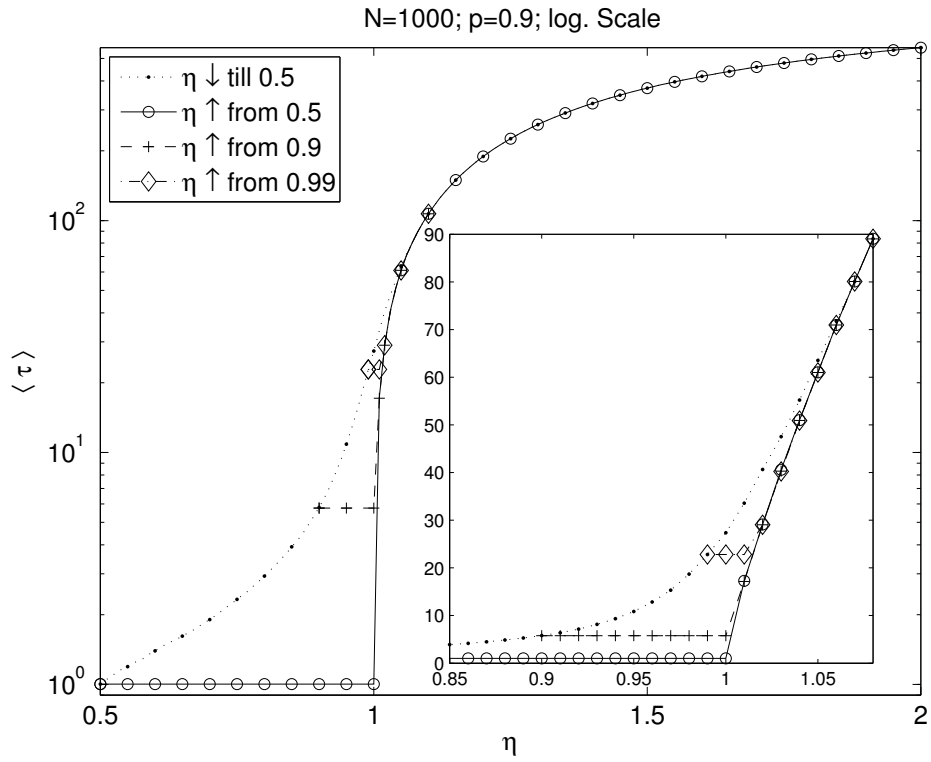

Figure 4.3: Hysteresis effect in the comparison of the dependence of $\langle\tau\rangle$ on η between the concentration and dilution process of the experiments. The y-axis shows $\langle\tau\rangle$ of 1000 experiments in logarithmic scale. Four curves are shown: $\eta\downarrow$ till 0.5: shows the results of the concentration process till $\eta = 0.5$. $\eta\uparrow$ from 0.5 is the corresponding part of 1000 dilution processes starting from $\eta = 0.5$. $\eta\downarrow$ from 0.9 shows the result for the dilution process starting after a concentration till $\eta = 0.9$. And $\eta\downarrow$ from 0.99 the same for a concentration until $\eta = 0.99$. The inset shows a zoom on the interesting region around $\eta = 1$ in linear scale. The number of neurons $N = 1000$ equals threshold L and $p = 0.9$ in all cases.

until $\eta = 1$ where it jumps then to a value slightly higher than the one predicted by the formulas (4.1) for this case. If we start the dilution process already at $\eta = 0.99$ we observe that $\langle\tau\rangle$ remains constant even for $\eta = 1.01$. Only if we dilute further $\langle\tau\rangle$ increases and starts to coincide with $\langle\tau\rangle$ of the other two dilution processes. Approximately at $\eta = 1.08$ the ISI of the dilution process coincides with the one of the concentration process.

This hysteresis effect may relevant to build a simple memory. By stimulating the system with a strong input, the system is driven to a value below $\eta = 1$. If the input is then substituted by a smaller one, which is just big enough to maintain the system below the critical couping strength and could represent the will to remember the first input, the ISI and the firing pattern of the ensemble will still be the same as if the strong input were still present. Once the system receives a short erase signal (e.g. in form of a negative input, or the absence of the small input), which allows it to reach a state corresponding to $\eta > 1$, the firing pattern produced by the strong input will disappear (i.e.

the memory will be deleted). Such a mechanism might be a novel way to represent working memory functions (Wang, 2001), which are often modeled in the form of bistable dynamical attractor networks (Durstewitz et al., 2000). In our case it seems that we have multi-stability for coupling greater than the critical coupling strength, but further analysis is needed to verify this claim.

4.2.2 A tighter bound for $\langle\tau\rangle$

Once identified the phenomena occurring in experiments we make some theoretical observations to gain further insight. We base these observations on a deterministic approximation of the model where the stochastic evolution (3.1) of a neuron i is simplified into the following deterministic iterative rule:

$$a_i(t+1) = a_i(t) + p \qquad \text{if } a_i(t) < L, \tag{4.2}$$

$$a_i(t+t_{ref}) = 1 \qquad \text{if } a_i(t) \geq L. \tag{4.3}$$

The random walk performed by the units is replaced by their average behavior: a deterministic motion with constant homogeneous velocity p. An equivalent continuous time system but with heterogeneous velocities (frequencies) and without delay and refractory period has been studied by Senn and Urbanczik (2000). In the following we will restrict our analysis to the case where the delay δ of the message exchange is greater than or equal to the refractory period t_{ref}.

After an initial transient the deterministic system shows a periodic pattern of spikes. The period of a pattern is the ISI of the ensemble (Figure 4.1b and 4.1c illustrate such patterns). If we take an arbitrary neuron and start the pattern at a spike of this unit, all units will spike exactly once until the next spike of the same unit. The sequence of these spikes will then start again with exactly the same time differences between the spikes, and this pattern will repeat itself forevermore if the parameters of the system are not changed. One can derive the following condition the system fulfills if it shows a periodic firing pattern.

Theorem 1 *A system that consists of κ clusters, where every cluster i consists of k_i elements, shows a periodic firing pattern with ISI τ if for every $i \in \{1,\ldots,\kappa\}$*

$$k_i > k_{min}(\tau) = \begin{cases} (N-1)(1-\eta) + \frac{p(\tau-1-t_{ref})}{\varepsilon} & \text{if } \tau \geq 1+t_{ref} \\ (N-1)(1-\eta) & \text{if } \tau < 1+t_{ref} \end{cases} \tag{4.4}$$

is fulfilled.

To prove this condition we use the deterministic rule (4.2) instead of the stochastic state transitions (3.1) and restrict ourself to the case of a delay higher than the refractory period, $\delta \geq t_{ref}$. A unit with ISI τ can make $\tau - 1 - t_{ref}$ transitions due to this rule before reaching the threshold. The

term t_{ref} corresponds to the refractory period where no increase of the state of an unit is allowed and the -1 to the last time-step when the threshold is reached. We have therefore a contribution of $p(\tau-1-t_{ref})H(\tau-1-t_{ref})$ because of rule (4.2) to the total evolution of a neuron before its threshold is reached[2].

With this result we can calculate the mean minimum cluster size of a system of κ clusters. We call the clusters K_i ($i \in \{1,\dots,\kappa\}$) and set the number of elements of every cluster $|K_i| = k_i$. The clusters are ordered according to their spiking time. At every time-step one cluster reaches threshold starting with cluster K_1. After cluster K_κ spikes the cycle starts again with cluster K_1. When cluster K_i reaches threshold, the elements of cluster K_{i+1} (or of K_1 in case of $i = \kappa$) have received the inputs of all clusters except cluster K_i and an increase of $p(\tau-1-t_{ref})$ due to rule (4.2). This leads to the following condition:

$$1+\Big(-1+\sum_{\substack{j=1\\ j\neq i}}^{\kappa} k_j\Big)\varepsilon + p(\tau-1-t_{ref})H(\tau-1-t_{ref}) < L. \tag{4.5}$$

Which due to $\sum_{j=1}^{\kappa} k_j = N$ is equivalent to

$$k_i > k_{min} = N-1+\frac{1+p(\tau-1-t_{ref})H(\tau-1-t_{ref})-L}{\varepsilon} \qquad \text{for all } i \in \{1,\dots,\kappa\}. \tag{4.6}$$

Written in terms of η this is equal to

$$k_i > k_{min}(\tau) = (N-1)(1-\eta)+\frac{p(\tau-1-t_{ref})H(\tau-1-t_{ref})}{\varepsilon}\,. \tag{4.7}$$

□

Condition (4.4) simply tells us that every cluster (i.e. units that reach the threshold at the same time-step) has to be greater than a certain minimum cluster size which depends on the system's parameter and its ISI.

Before we continue our analysis we make some comments on the validity of this rule for the stochastic system. The firing patterns of the deterministic system may also occur in the stochastic system as can be seen in Figure 4.1b and 4.1c. According to the robustness of these patterns against variations in the stochastic evolution we can distinguish between three types of patterns.

Robust firing patterns are totally insensitive to variations of stochastic state transitions in the sense that, no matter how they evolve, even if they are totally suppressed, the system cannot change its periodic pattern.

Semi-robust firing patterns remain unchanged if one or more units evolve slower than with their mean velocity p, but if they evolve much faster the spiking pattern may change. Since such

[2]Note the use of the Heaviside step function $H(\cdot)$ to have a valid formula also for the case of $\tau < t_{ref}+1$.

changes are rare, as will be explained subsequently, and the patterns are robust against at least half of the possible stochastic events, we choose the name semi-robust.

Variable firing patterns may change due to very slow or very fast evolution of one or more units.

At $\eta \leq 1$ we can find only robust and semi-robust patterns. Condition (4.4) gives us the rule for a semi-robust pattern in the stochastic system. To get the condition of a robust pattern in this case we would have to replace p with 1.

A change of a semi-robust pattern of phase-locked clusters implies that one or more units change from one cluster to the one firing directly before it. This increases the robustness of the resulting new firing pattern since the smallest cluster has the highest probability of receiving a neuron and leads to a certain balancing of the sizes of the clusters. Only if we have two small clusters spiking one directly after the other a merge of these two clusters might occur, which would imply a decrease of the ISI. Every decrease of the ISI enhances the robustness of the resulting firing pattern, since it implies a decrease of the minimum cluster size $k_{min}(\tau)$. Such events however are rare especially for large populations and their influence is far below the standard deviation of the experiments. Since we are mainly interested in the ISIs of our system we can neglect them in our analysis. Although we have to state that for an infinite simulation time all semi-robust patterns would transform into robust ones.

For $\eta > 1$ we only find variable firing patterns (Figure 4.1a). In the stochastic system, the higher the value of η the lower is the probability to observe between three consecutive spikes of a certain single unit the same two spiking patterns of the rest of the neurons. It is not even granted that the two ISIs of this unit have the same length. But now, unlike the case of $\eta \leq 1$, the fluctuations of the noise cannot create irreversible effects on the ISI of the system. The mean ISI of the system coincides therefore with the one of the deterministic system, which makes the following analysis valid in this region as well.

Every time the coupling strength is increased, after a short transient, the deterministic system fulfills again condition (4.4). This allows us to make some observations on the ISI of the system. Since the cluster sizes have to be integer numbers we define

$$\hat{k}_{min}(\tau) = \lfloor k_{min}(\tau) + 1 \rfloor \tag{4.8}$$

and have the condition $k_i \geq \hat{k}_{min}(\tau)$. This definition guarantees that $\hat{k}_{min}(\tau) \geq 0$ and is necessary for technical reasons in the derivation of the lower bound we will see later. A system with ISI τ consists of $\kappa = \tau/\delta$ clusters since a spiking cluster at time t provokes the spiking of the next one at time $t + \delta$. With this we calculate another quantity we need to describe our system, the mean cluster size given a certain ISI τ:

$$\bar{k}(\tau) = \frac{N\delta}{\tau} \, . \tag{4.9}$$

We then introduce a new function $g(\tau)$ which gives us the ratio between $\bar{k}(\tau)$ and $\hat{k}_{min}(\tau)$.

$$g(\tau) = \frac{\bar{k}(\tau)}{\hat{k}_{min}(\tau)} \,. \tag{4.10}$$

Intuitively this quantity can be seen as the frequency a system consisting only of clusters with the minimum cluster size $\hat{k}_{min}(\tau)$ has to fire with, to achieve the same ISI as the system with cluster-size $\bar{k}(\tau)$. Note that $g(\tau) > 0$ since $\hat{k}_{min}(\tau) \geq 0$.

Combining (4.9), (4.10) and (4.7) we arrive to:

$$\begin{aligned} \frac{N\delta}{\tau} &= g(\tau)\,\lfloor k_{min}(\tau)+1 \rfloor \\ &\leq g(\tau)\left((N-1)(1-\eta)+\frac{p(\tau-1-t_{ref})H(\tau-1-t_{ref})}{\varepsilon}+1\right). \end{aligned} \tag{4.11}$$

We calculate the mean of the left and right hand side according to the probabilities $P(\tau)$, which denominate the probability of the system ending up in a system with ISI τ.

$$\sum_{i=1}^{\tau_{max}} \frac{N\delta}{i} P(i) \leq \sum_{i=1}^{\tau_{max}} g(i)\left((N-1)(1-\eta)+\frac{p(i-1-t_{ref})H(i-1-t_{ref})}{\varepsilon}+1\right)P(i) \,. \tag{4.12}$$

We set $1/\tau = f$ where f is the spiking-frequency, use $P(\tau) = P(f) = P(g(\tau))$, eliminate the terms of the summation which equal 0 and get

$$\begin{aligned} N\langle f\rangle\delta &\leq \langle g\rangle\left((N-1)(1-\eta)+1-\frac{(1+t_{ref})p}{\varepsilon}\right)+ \\ &\quad +\frac{p}{\varepsilon}\left(\sum_{i=1}^{1+t_{ref}} g(i)P(i)+\sum_{i=2+t_{ref}}^{\tau_{max}} g(i)iP(i)\right). \end{aligned} \tag{4.13}$$

Using the following identity $\langle XY\rangle = \langle X\rangle\langle Y\rangle + \mathrm{Cov}(X,Y)$ for two random variables X and Y we achieve

$$\begin{aligned} N\langle f\rangle\delta &\leq \langle g\rangle\left((N-1)(1-\eta)+1-\frac{(1+t_{ref})p}{\varepsilon}\right)+ \\ &\quad +\frac{p}{\varepsilon}\left(\langle g\rangle\langle\tau\rangle+\mathrm{Cov}(g(\tau),\tau)-\sum_{i=2}^{1+t_{ref}} g(i)(i-1)P(i)\right). \end{aligned} \tag{4.14}$$

From (4.11) it is easy to see that $\mathrm{Cov}(g(\tau),\tau) \leq 0$ since an increase of τ translates into a decrease of $g(\tau)$ and vice versa. Because of this fact and $g(\tau) \geq 0$ we can eliminate the two leftmost terms of inequality (4.14) by weakening the inequality.

$$N\langle f\rangle\delta \leq \langle g\rangle\left((N-1)(1-\eta)+1+\frac{p(\langle\tau\rangle-1-t_{ref})}{\varepsilon}\right). \tag{4.15}$$

The mean frequency $\langle f\rangle$ is equal to the inverse of the harmonic mean $h(\tau)$. Since for a set of positive numbers its harmonic mean is never greater than its arithmetic mean Bullen (2003) we have $\frac{1}{\langle\tau\rangle} \leq \frac{1}{h(\tau)} = \langle f\rangle$. Applying this on inequality (4.15) leads to

$$\frac{N\delta}{\langle\tau\rangle} \leq \langle g\rangle\left((N-1)(1-\eta)+1+\frac{p(\langle\tau\rangle-1-t_{ref})}{\varepsilon}\right). \tag{4.16}$$

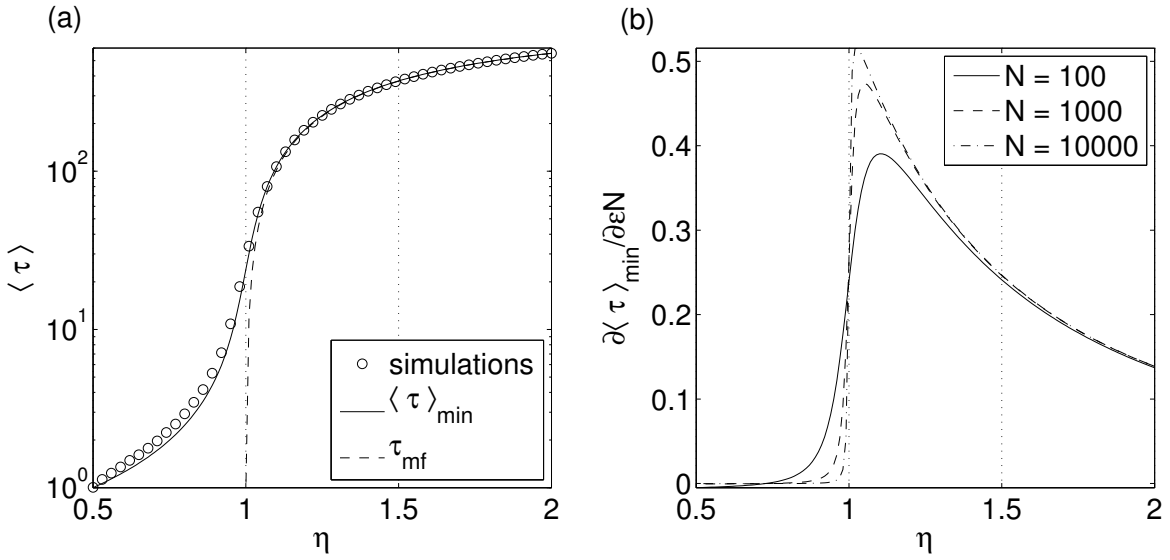

Figure 4.4: **(a)** The empirical mean ISI $\langle\tau\rangle$ of 1000 concentration processes compared to the mean field bound $\langle\tau\rangle_{mf}$ of equation (4.1) and the bound $\langle\tau\rangle_{min}$ derived in equation (4.18). **(b)** the rate of change of the period τ using the theoretical approximation $\langle\tau\rangle_{min}$ properly scaled for increasing values of N. A phase transition occurs for $\eta = 1$, where the derivative experiments a sharp change which becomes more pronounced as we approach to the thermodynamic limit $N \to \infty$.

We can transform this into a quadratic inequality of $\langle\tau\rangle$ since $\langle\tau\rangle > 0$. It has the only positive solution

$$\langle\tau\rangle \geq \langle\tau\rangle_{min} = \frac{(N-1)\varepsilon(\eta-1)-\varepsilon}{2p} + \frac{1+t_{ref}}{2} + \\ + \sqrt{\left(\frac{(N-1)\varepsilon(\eta-1)-\varepsilon}{2p} + \frac{1+t_{ref}}{2}\right)^2 + \frac{N\varepsilon\delta}{p\langle g\rangle}}\,. \tag{4.17}$$

The new lower bound for the mean ISI distribution is:

$$\langle\tau\rangle_{min} = \frac{(N-1)\varepsilon(\eta-1)-\varepsilon}{2p} + \frac{1+t_{ref}}{2} + \\ \sqrt{\left(\frac{(N-1)\varepsilon(\eta-1)-\varepsilon}{2p} + \frac{1+t_{ref}}{2}\right)^2 + \frac{N\varepsilon\delta}{p\langle g\rangle}} \tag{4.18}$$

Figure 4.4a shows the accuracy of the proposed theoretical bound. We plot the results of 1000 concentration processes in circles. As can be seen, $\langle\tau\rangle_{min}$, denoted using a continuous line, represents a very accurate approximation of the mean period $\langle\tau\rangle$ of the units, even within the region corresponding to $\eta < 1$, where the mean field approximation, denoted in dashed line, predicts the full synchronized state with period one.

The existence of a phase transition is illustrated in Figure 4.4b, where we plot the derivative of the theoretical bound $\langle\tau\rangle_{min}$ with respect to the coupling strength. There exists an abrupt change at $\eta = 1$, which becomes a discontinuity as we let the size of the population tend to ∞.

4.3 Self-organization using dynamical synapses

In this section we introduce synaptic plasticity in the proposed model. As before, we are not focused on particular details occurring at very small scales of description. For us, each synaptic junction between neuron j and neuron i is represented as a continuous variable ε_{ij}.

According to the definition of the dynamics of the model in Section 3.1, the total evolution of the state of a unit can be described by its individual stochastic evolution and the evolution due to the interaction between other neurons in the network. In the discrete model, the individual stochastic evolution is described as a random walk toward the threshold barrier L in the form of a Bernoulli process. We denoted this evolution as the *spontaneous* evolution. For values of $\eta > 1$ above the critical point, i.e. weak interactions, most of the spontaneous evolution is required by the units in the network to reach the threshold. In this regime, the system dynamics is irregular and fluctuates excessively. On the other hand, for decreasing values of $\eta < 1$, although the activity is robust and self-sustained, the mean ISI also decreases and consequently, the size of the robust firing pattern [3] as well. Only for $\eta = 1$, when the onset of self-sustained activity occurs, the periodic firing pattern produced by the network has the largest mean period and the network breaks down in the maximum number of clusters. At this point, the system fulfills both conditions, namely, robustness with respect to intrinsic fluctuations and a large number of self-sustained periodic patterns.

We describe this phenomenon considering a magnitude of interest: the dissipated spontaneous evolution, which is defined in the next Section. This magnitude is also maximized at $\eta = 1$, where the phase transition occurs. The gradient of this magnitude allows to derive a plasticity rule that can be expressed using only local information encoded in the post-synaptic unit. Finally, we present simulation results and an analytical expression to calculate the time required for a network to self-organize at the critical state starting at any arbitrary configuration, i.e. for any initial value of η.

4.3.1 The dissipated spontaneous evolution

During one ISI, we distinguish between the spontaneous evolution carried out by a unit and the actual spontaneous evolution needed for a unit to reach the threshold L. Note that the second is always smaller than the first. The subtraction of both quantities can be regarded as a surplus of spontaneous evolution, which is the fraction of the spontaneous evolution that is dissipated during one ISI.

First, we calculate the spontaneous evolution of a given unit during one ISI. It is just the number of stochastic state transitions the unit experiments in one period τ. These state transitions occur with probability p through the elapsed time during which the unit is allowed to evolve spontaneously. If we take average values of the period τ over many ISIs and over all the units in the network we obtain

[3] A robust firing pattern is defined in Section 4.2.2.

the mean total spontaneous evolution:

$$E_{total} = (\langle\tau\rangle - t_{ref})p. \tag{4.19}$$

Since the state of a given unit can exceed the threshold because of the received messages from the rest of the population, a fraction of (4.19) is actually not required to induce a spike in that unit, and therefore is dissipated. We can obtain this fraction by subtracting to (4.19) the actual number of state transitions that was required to reach the threshold L. The latter quantity can be referred as an *effective* spontaneous evolution and is just $L-1$ minus the evolution caused by the messages received during the ISI[4], which is on average $(N-1)\varepsilon$ since all units receive on average one message from the rest of the units of the network during one ISI. For values of $\eta < 1$, the activity is self-sustained and just the received messages from other units are enough to drive the state of a unit to reach the threshold. In this case, all the spontaneous evolution is dissipated and thus the effective spontaneous evolution is zero. Summarizing, we have that:

$$E_{eff} = \begin{cases} L-1-(N-1)\varepsilon & \text{for } \eta \geq 1 \\ 0 & \text{for } \eta < 1. \end{cases} \tag{4.20}$$

or, equivalently:

$$E_{eff} = \max\{0, L-1-(N-1)\varepsilon\} \tag{4.21}$$

If we subtract (4.21) from the mean spontaneous evolution (4.19), we obtain the mean dissipated spontaneous evolution, which for simplicity we will denote it from now on as the dissipated spontaneous evolution:

$$E_{diss} = E_{total} - E_{eff} = (\langle\tau\rangle - 1)p - \max\{0, L-1-(N-1)\varepsilon\}. \tag{4.22}$$

We can obtain an analytic expression for this quantity using the lower bound of the mean ISI $\langle\tau\rangle_{min}$ derived in the previous chapter (4.18) as an accurate approximation of the mean period $\langle\tau\rangle$ of a unit. For clarity and without loss of generality, we assume the same values for transmission delay and the refractory period in all the units, $t_{ref} = \delta = 1$. Then E_{diss} in our model is:

$$E_{diss} = \left(\frac{(N-1)\varepsilon(\eta-1)-\varepsilon}{2p} + \sqrt{\left(\frac{(N-1)\varepsilon(\eta-1)-\varepsilon}{2p} + 1 \right)^2 + \frac{N\varepsilon}{2p}} \right) p - \max\{0, L-1-(N-1)\varepsilon\}. \tag{4.23}$$

[4]Note that units start their random walk at the state 1.

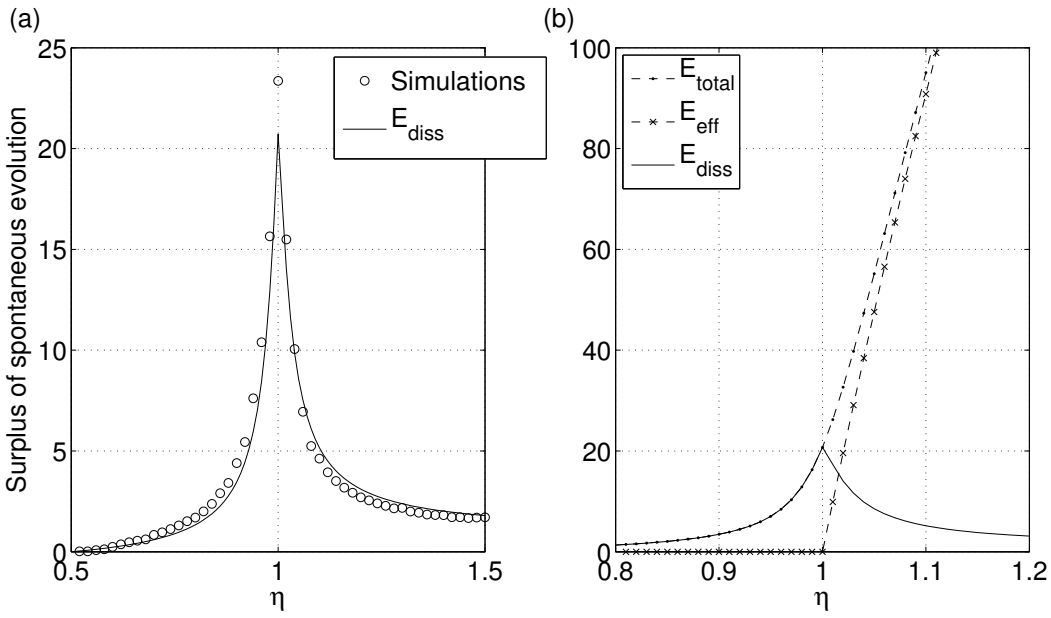

Figure 4.5: The spontaneous evolution which is dissipated E_{diss} is maximized at $\eta = 1$. **(a)** Theoretical curve of E_{diss} versus empirical results. **(b)** The three different evolutions involved in the analysis. Parameters are $N = 1000, L = 1000$ and $p = 0.9$. For the mean period we have used the bound for $\langle\tau\rangle_{min}$ derived in the previous chapter.

Figure 4.5a shows the analytic curve E_{diss} in function of η together with the empirical outcome of a simulation consisting in several concentration processes (see Section 4.2.1). The analytical formula provides an acceptable approximation of the simulation results. As we can see, E_{diss} is maximized at $\eta = 1$, where the phase transition occurs.

To understand why this quantity is maximized at the critical point, we consider how both E_{total} and E_{eff} change as we increase the coupling strength ε in a concentration process. We need to analyze three scenarios. First $\eta \gg 1$, where the effect of the interaction between neurons is negligible. Second, as η approaches 1 from the right. And third, for values of $\eta < 1$.

The first case is the simplest one. At $\eta \gg 1$, in a very noise-dominated regime, all the units reach the threshold L only because of their spontaneous evolution. Hence, $E_{total} \approx E_{eff}$ and $E_{diss} \approx 0$.

As the system approaches to $\eta = 1$, both E_{total} and E_{eff} decrease. The reason why E_{diss} grows can be explained comparing both curves as we show in Figure 4.5b. On one hand, E_{eff} is reduced because we increase ε, so the number of required state transitions to reach the threshold decreases because units receive more messages from the population during one ISI and becomes zero when the network reaches a self-sustained firing state. The rate of change of E_{eff} is almost constant for values of η near 1. On the other hand, E_{total} is determined essentially by the mean period $\langle\tau\rangle$ and thus it is always above zero. As we can see in Figure 4.5b, since E_{total} remains positive, it differs

more and more from E_{eff} as we decrease η. Unlike the rate of change of E_{eff} which remains almost constant, the rate of change of E_{total} decreases abruptly at η near 1. Since E_{diss} is the difference of both quantities it is increased as the system approaches to $\eta = 1$.

Finally, for values of $\eta < 1$, we have that $E_{eff} = 0$. Hence, all the dissipated spontaneous evolution can be explained by means of the total spontaneous evolution E_{total}, which in turn is determined mainly by the mean period $\langle\tau\rangle$. The decrease of E_{diss} can thus be understood entirely because $\langle\tau\rangle$ decreases as well.

4.3.2 Synaptic dynamics

After having introduced our magnitude of interest we now derive a plasticity rule in the model. Our approach essentially assumes that updates of the individual synapses ε_{ij} are made in the direction of the gradient of the surplus spontaneous evolution. The analytical results turn out to be rather simple and allow a clear interpretation of the underlying mechanism governing the dynamics of the network under the proposed synaptic rule.

For such a rule to be biologically plausible it must be local, so only the states of the presynaptic j and postsynaptic i neurons should be accessible in order to define how an individual ε_{ij} changes. Because now each synapse ε_{ij} is updated independently of the rest of synapses, the network can become heterogeneous, in the sense that a global coupling strength ε is no longer considered. Now, an individual synapse ε_{ij} encode local information of the neurons at its both ends.

In order to obtain our desired local rule which modifies independently each synapse ε_{ij} between a presynaptic neuron j and postsynaptic neuron we first rewrite E_{diss} in terms of the global coupling ε by replacing η by its definition (3.4):

$$E_{diss} = \left(\frac{L-1-(N-1)\varepsilon-\varepsilon}{2p} + \sqrt{\left(\frac{L-1-(N-1)\varepsilon-\varepsilon}{2p} + 1 \right)^2 + \frac{N\varepsilon}{2p}} \right) p - \max\{0, L-1-(N-1)\varepsilon\}. \quad (4.24)$$

Now we can see that the terms appearing in the numerator of (4.24) are close to the effective spontaneous evolution with one additional negative ε term. For simplicity, let us neglect this additional negative ε term and assume that a unit receives $N\varepsilon$ messages on average during one ISI. This assumption can be made in the limit of a large size N. We can then approximate the term $(N-1)\varepsilon+\varepsilon$ with the sum of the received messages during one ISI:

$$(N-1)\varepsilon+\varepsilon \approx (N-1)\varepsilon = \sum_{k\neq i} \varepsilon_{ik}. \quad (4.25)$$

The dissipated spontaneous evolution is now defined in terms of each individual neuron i in the network, $i = 1,\ldots,N$, and represents an average over many ISIs of neuron i:

$$E^i_{diss} = \left(\frac{L-1-\sum_{k\neq i}\varepsilon_{ik}}{2p} + \sqrt{\left(\frac{L-1-\sum_{k\neq i}\varepsilon_{ik}}{2p}+1\right)^2 + \frac{\sum_{k\neq i}\varepsilon_{ik}}{2p}} \right) p - \max\{0, L-1-\sum_{k\neq i}\varepsilon_{ik}\}. \quad (4.26)$$

We now derive our synaptic plasticity rule, which can be applied at every time-step when a spike from the presynaptic unit j induces a spike in a postsynaptic unit i. This rule modifies the synapses of the presynaptic firing units ε_{ij} in the direction of the gradient of (4.26):

$$\Delta\varepsilon_{ij} = \kappa \frac{\partial E^i_{diss}}{\partial \varepsilon_{ij}}, \quad (4.27)$$

where the constant κ scales the amount of change in the synapse. In the next section we will analyze the role of a global κ in the system.

For the moment, let us calculate the derivative E^i_{diss} with respect to an arbitrary synapse ε_{ij}. On one hand, differentiating E^i_{total} yields to:

$$\begin{aligned}
\frac{\partial E^i_{total}}{\partial \varepsilon_{ij}} &= \left(-\frac{1}{2p} + \frac{2\left(\frac{L-1-\sum_{k\neq i}\varepsilon_{ik}}{2p}+1\right)\left(-\frac{1}{2p}\right)+\frac{1}{2p}}{2\sqrt{\left(\frac{L-1-\sum_{k\neq i}\varepsilon_{ik}}{2p}+1\right)^2+\frac{\sum_{k\neq i}\varepsilon_{ik}}{2p}}} \right) p \\
&= \frac{1}{2}\left(-1 + \frac{-2\left(\frac{L-1-\sum_{k\neq i}\varepsilon_{ik}}{2p}+1\right)+1}{2\sqrt{\left(\frac{L-1-\sum_{k\neq i}\varepsilon_{ik}}{2p}+1\right)^2+\frac{\sum_{k\neq i}\varepsilon_{ik}}{2p}}} \right) \\
&= -\frac{1}{2}\left(\frac{\frac{L-1-\sum_{k\neq i}\varepsilon_{ik}}{2p}+\frac{1}{2}}{\sqrt{\left(\frac{L-1-\sum_{k\neq i}\varepsilon_{ik}}{2p}+1\right)^2+\frac{\sum_{k\neq i}\varepsilon_{ik}}{2p}}} + 1 \right),
\end{aligned}$$

where $1 = 1$ if $k = j$ and zero otherwise. On the other hand, the derivative of E^i_{eff} with respect to ε_{ij} is:

$$\frac{\partial E^i_{eff}}{\partial \varepsilon_{ij}} = \begin{cases} 0 & \text{if } (L-1-\sum_{k\neq i}\varepsilon_{ik}) < 0 \\ \text{indef} & \text{if } (L-1-\sum_{k\neq i}\varepsilon_{ik}) = 0 \\ -1 & \text{if } (L-1-\sum_{k\neq i}\varepsilon_{ik}) > 0. \end{cases}$$

The previous two expressions include two terms not locally accessible at the level of the single synapse ε_{ij}. One is the term $\sum_{k\neq i}\varepsilon_{ik}$, which requires that all the values of the synaptic efficacies of the presynaptic units k are accessible to the single synapse ε_{ij}. However, the efficacies of presynaptic units

can be propagated to the state of the postsynaptic unit i by considering again an effective threshold L^i for every unit i which decreases deterministically every time an incoming pulse is received, as we did in the mesoscopic reformulation of the model, see Section 3.3.1. At the end of an ISI, the value L^i is just $(L-1-\sum_{k\neq i}\varepsilon_{ik})$. Therefore the values of all presynaptic neurons of i are implicitly encoded for any individual synapse ε_{ij} in the state L^i of the postsynaptic unit i at the precise spiking time.

The other term which involves non-local information accessible to ε_{ij} includes the noise term, and is $2p$. For the moment, we just replace p with a constant c and postpone the analysis of the dependency of the synaptic rule on this parameter.

These modifications allow to write the final derivative of E^i_{diss} in a more compact form, in function of terms directly accessible to the synapse ε_{ij}:

$$\begin{aligned}\frac{\partial E^i_{diss}}{\partial \varepsilon_{ij}} &= -\frac{1}{2}\left(\frac{\frac{L^i}{c}+\frac{1}{2}}{\sqrt{\left(\frac{L^i}{c}+1\right)^2+\frac{L-L^i}{c}}}+1\right)-\begin{cases}0 & \text{if } L^i<0\\ \text{indef} & \text{if } L^i=0\\ -1 & \text{if } L^i>0.\end{cases} \\ &= \frac{-L^i-c}{2\sqrt{(L^i+2c)^2+2c(L-L^i)}}+\frac{\operatorname{sgn}(L^i)}{2}.\end{aligned} \tag{4.28}$$

We can understand the mechanism involved in a particular synaptic update by analyzing in detail expression (4.28). First, we can see that it differentiates between three scenarios:

1. The corresponding to a negative effective threshold $L^i<0$ at the end of one ISI. In this case, the unit received more strength from the rest of the units than the required to spike, and occurs when $\eta<1$. The resulting update weakens the synapse.

2. The case of $L^i=0$. This occurs when the unit received the same activity from the rest of the population as the distance to the initial threshold L. This scenario is precisely the optimal one and corresponds to $\eta=1$. Here, no synaptic update is defined, although we will consider a 0 update for practical purposes.

3. Finally, if $L^i>0$, some spontaneous evolution was required for the unit to fire, and occurs when $\eta>1$. The gradient is positive and therefore the synapse is strengthened.

Figure 4.6a shows the mechanism. We plot the values of Equation (4.28) in bold lines together with $\frac{\partial E^i_{total}}{\partial \varepsilon_{ij}}$, which corresponds to the case of $\eta>1$, and $\frac{\partial E^i_{total}}{\partial \varepsilon_{ij}}+1$, which corresponds to $\eta<1$, for different values of the effective threshold L^i of a given unit at the end on an ISI. As can be seen, the term corresponding to the total spontaneous evolution indicates the amount of change in the synapses and the term corresponding to the effective spontaneous evolution indicates whether the synapses must be strengthened or weakened.

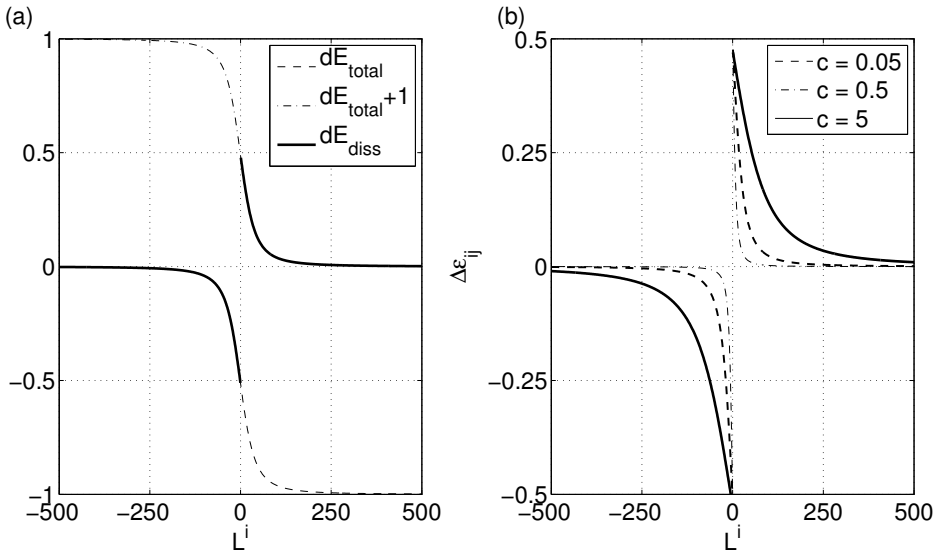

Figure 4.6: Plasticity rule. **(a)** First derivative of the dissipated spontaneous evolution E_{diss} for $\kappa = 1, L = 1000$ and $c = 0.9$. **(b)** The same rule for different values of c.

The largest updates occur in the transition from a positive L^i to a negative L^i and become zero for large absolute values of L^i. This implies that significant updates correspond to those synapses with postsynaptic neurons which during the last ISI have received a similar amount of activity from the network that the one required to fire. Consequently, we deduce that a network in a state far away from the critical point will require more steps to reach it than a network near the critical point.

We notice a similarity in the shapes between the obtained function and the one used in Spike-Timing dependent plasticity (STDP), a widely used plasticity rule in experimental and theoretical studies (Bi and Poo, 1998; Song et al., 2000). However, it is important to note that the domains of both functions are different. While in STDP the change in the synaptic conductances is determined by the relative spike timing of the presynaptic neuron and the postsynaptic neuron, in our case it is determined by the effective threshold of the postsynaptic unit at its spiking time. If we replace the domains we get very similar functions. On one hand, STDP long-term strengthening of synapses occurs if presynaptic action potentials precede postsynaptic firing by no more than about few milliseconds. The analog situation in our model corresponds to the case where $L^i > 0$ and the postsynaptic unit has not received enough spikes to reach the threshold in the last period. On the other hand, presynaptic action potentials that follow postsynaptic spikes produce long-term weakening of synapses, which in our case corresponds to the situation where $L^i < 0$ and the unit receives an excess of activity from the units in the network in the last period.

As in our case, the largest changes in synaptic efficacies in the STDP framework occur in a abrupt transition from strengthening to weakening (corresponding to $L^i = 0$ in Figure 4.6a). In STDP,

however, the transition is produced when the time differences between pre and postsynaptic action potentials passes through zero.

Figure 4.6b illustrates the role of c in the plasticity rule. If a small value is used, significant updates are only defined in a tiny range of values of L^i near zero. For higher values of c, the range of values of L^i for which significant updates is widened. The shape of the rule, however, is preserved, and the role of c is just to scale the change in the synapse. For the rest of this chapter, we use $c = 1$.

4.3.3 Simulations

In this section we show empirical results of the application of the proposed plasticity rule. We focus our analysis on the time τ_{conv} required for the system to converge toward the critical point. In particular, we analyze how τ_{conv} depends on the starting initial configuration and on the value of the constant κ. For simplicity, we consider a global κ, but note that this is not a restriction, since different κ_{ij} can be applied to each particular synapse ε_{ij}.

The network used for the experiments is composed of $N = 500$ units with homogeneous $L = 500$ and $p = 0.9$. Synapses are initialized homogeneously and random initial states are chosen for all the units in each trial. Every time a unit i fires, we only update its afferent synapses ε_{ij}, for all $j \neq i$. This fact breaks the homogeneity in the coupling strengths. The global characteristic parameter η must then be redefined by replacing the term $(N-1)\varepsilon$ with the sum of all synaptic efficacies:

$$\eta' = \frac{L-1}{(N-1)\langle\varepsilon\rangle}. \tag{4.29}$$

Initially, the network is on a certain starting condition η'_0. We let it evolve applying its original discrete dynamics together with the plasticity rule based on (4.27) and (4.28), and measure the time τ_{conv} when it reaches a value close to $\eta' = 1$ for the first time. In order to calculate τ_{conv}, we select one unit i randomly and compute the value η' in ISI units, that is, at every time the same unit i fires. We assume convergence from a practical point of view when η' falls in the interval $(1-\nu, 1+\nu)$ for the first time. In these initial experiment, ν is set to 10^{-2}.

First, we study the time required for a network initialized in the noise-dominated regime. We performed 50 random experiments using different initial configurations. For all cases, after an initial transient, the network settles in a value of η' around $\eta' = 1$ presenting some fluctuations. These fluctuations did not grow even after 10^6 periods in all realizations. For clarity, we only show in Figure 4.7 three examples for values of $\eta'_0 = \{1.1, 1.3, 1.7\}$. Only the periods corresponding to the first 1000 time-steps after convergence are shown.

As expected, the system converges fast for an initial value η'_0 near the critical point $\eta' = 1$. This can be explained because the gradient (4.26) in this region is higher in absolute value than for η' far away from $\eta' = 1$.

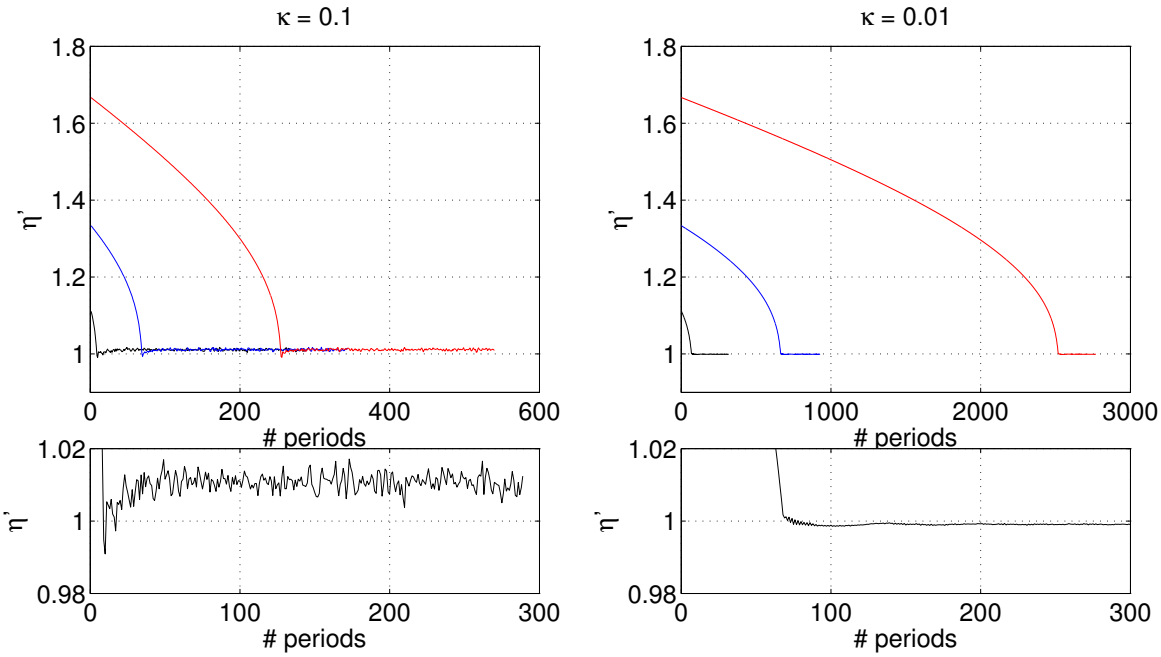

Figure 4.7: Empirical results of convergence toward $\eta' = 1$ starting from three different states above the critical point. Horizontal axis denotes number of periods of the same random unit during all the simulation. On the left, results using the constant $\kappa = 0.1$. Top panel shows the full trajectory until 10^3 time-steps after convergence. Bottom panel is a zoom of the first trajectory $\eta'_0 = 1.1$. Right panels show the same type of results but using a smaller constant $\kappa = 0.01$.

Comparing left and right panels, we confirm that for larger updates of the synapses ($\kappa = 0.1$) the network converges faster than for $\kappa = 0.01$. However, as illustrated in the bottom panels, where we zoom the trajectory corresponding to $\eta'_0 = 1.1$, for $\kappa = 0.01$ the convergence state is closer to $\eta' = 1$ and fluctuations around the reached state are approximately one order of magnitude smaller than for $\kappa = 0.1$. We therefore can conclude that κ plays the role of determining the speed of convergence and the quality and stability of the network at the critical state. There is a trade-off because high values of κ speed the convergence but turn the dynamics of the network less stable at the critical state.

A similar picture is obtained if the network is initialized below the critical point, shown in Figure 4.8. As in the previous case, after a transient, the system settles near the critical point. On one hand, the number of periods of the transient again depends on how far is the initial state from $\eta' = 1$. On the other hand, using $\kappa = 0.1$ leads to faster convergence but larger fluctuations around a value slightly above $\eta' = 1$, whereas using $\kappa = 0.01$ results in slower convergence but more stability.

An important difference with the previous scenario $\eta'_0 > 1$ is the existence of the hysteresis effect (see Section 4.2.1). For $\eta'_0 < 1$, the dynamics of the network stays robust during the route toward the critical state. For instance, if we initialize the network in $\eta'_0 < 0.5$, all neurons become fully

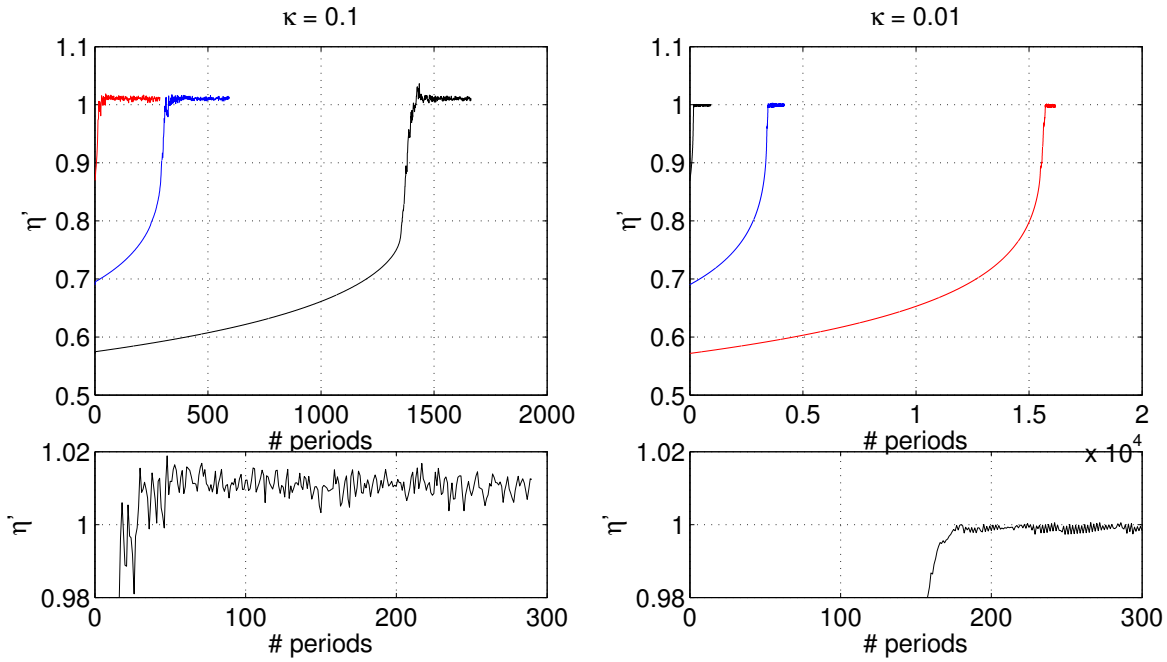

Figure 4.8: Empirical results of convergence toward $\eta' = 1$ starting from three different states below the critical point. Horizontal axis denotes number of periods of the same random unit during all the simulation. On the left, results using the constant $\kappa = 0.1$. Top panel shows the full trajectory until 10^3 time-steps after convergence. Bottom panel is a zoom of the first trajectory $\eta'_0 = 1.1$. Right panels show the same type of results but using a smaller constant $\kappa = 0.01$.

synchronized and spike in unison, and this pattern remains stable until the critical state is reached. Because of this, the total synaptic change in one period can be very large, and consequently, it may happen that the network exceeds the critical point and stays above it for a small number of periods. The black trajectory of Figure 4.8 for $\kappa = 0.1$ represents an example of this situation which is not observed for $\eta'_0 > 1$. This effect may remain for some periods but always disappears afterwards. It is directly related with the initialization η'_0 and the value of κ, and is more likely to occur for small values of η'_0 and large values of κ.

Comparing figures 4.7 and 4.8 one can be tempted to deduce that a network initialized at $\eta'_0 < 1$ needs more iterations to converge. Note, however, that this is not generally the case. The reason is that we use the period as the time unit. In general, the number of elapsed time-steps until convergence depends on the number of spikes of the network at each time-step. Hence, the system typically takes less time-steps to reach $\eta' = 1$ starting from an initial $\eta'_0 < 1$, where units are phase-locked and spikes are produced at every time-step, than starting from an initial $\eta'_0 > 1$, where periods are larger.

To end this subsection, we study how τ_{conv} depends on η'_0 in more detail using a simple analytical expression for the number of periods (or time-steps) required for convergence in function of an

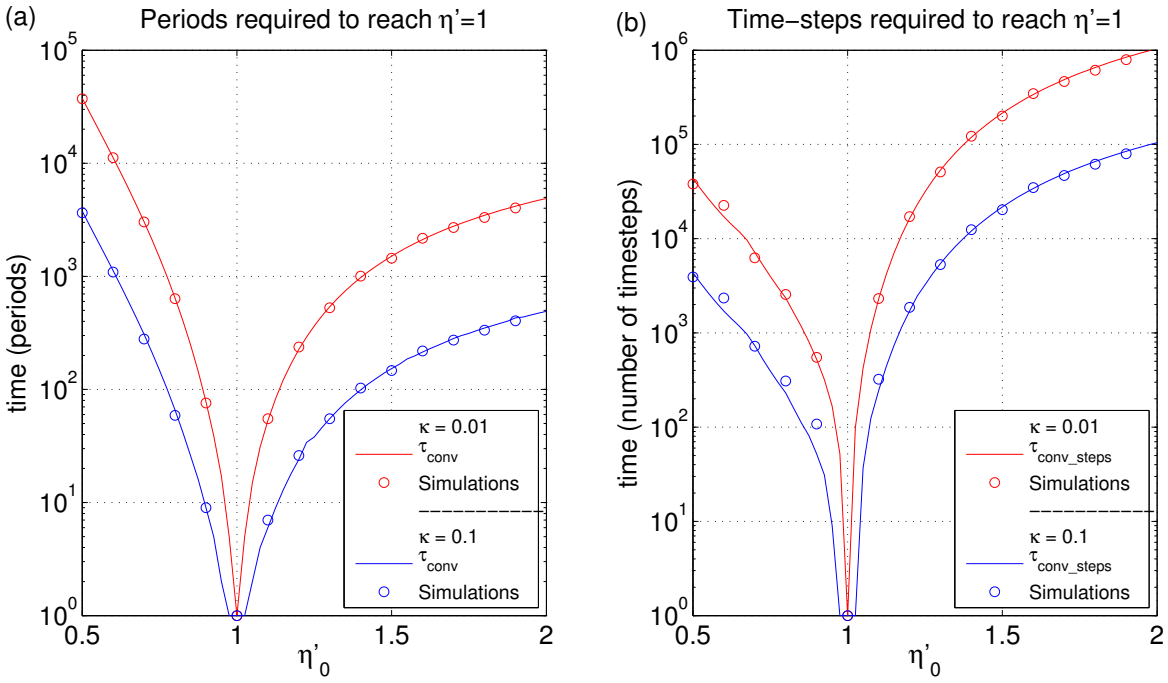

Figure 4.9: Number of periods **(a)** and time-steps **(b)** required to reach the critical value of η' in function of the initial configuration η'_0. Rounded dots indicate empirical results as averages over 10 different realizations starting from the same η'_0. Continuous curves correspond to the numerical solution of the approximations (4.30) and (4.31). Parameter values are $N = 500, L = 500, p = 0.9, c = 2$. Since κ sets the size of the fluctuations around the critical point, the tolerance value is set to $\nu = \kappa$.

arbitrary η'. Given the values of the parameters N, L, c and κ fixed, we can approximate the global change in η' after one entire period of a random unit assuming that all neurons change its afferent synapses uniformly:

$$\Delta(\eta') = \kappa(N-1)\left(-\frac{1}{2}\left(\frac{\frac{L_{eff}(\eta')}{c}+\frac{1}{2}}{\sqrt{\left(\frac{L_{eff}(\eta')}{c}+1\right)^2+\frac{L-L_{eff}(\eta)}{c}}}+1\right)-\begin{cases}0 & \text{if } \eta'<1\\ \text{indef} & \text{if } \eta'=1\\ -1 & \text{if } \eta'>1\end{cases}\right),$$

$$\text{where } L_{eff}(\eta') = L-1-\frac{(N-1)}{\eta'}.$$

The number of periods can then be obtained by means of the following recursive equation:

$$\tau_{conv}(\eta') = \begin{cases}1 & \text{if } \eta' = 1 \pm \nu\\ 1+\tau_c(\eta'+\Delta(\eta')) & \text{otherwise.}\end{cases} \tag{4.30}$$

Similarly, the number of time-steps can be approximated as well replacing the 1 in the recursive case

by the lower bound approximation of the mean period τ:

$$\tau_{conv\_steps}(\eta') = \begin{cases} 1 & \text{if } \eta' = 1 \pm \nu \\ \langle\tau\rangle_{min} + \tau_{conv\_steps}(\eta' + \Delta(\eta')) & \text{otherwise.} \end{cases} \tag{4.31}$$

Figure 4.9 shows empirical values of τ_{conv} and τ_{conv_steps} for several values of η'_0 together with the analytical approximations (4.30) and (4.31). As we have seen in the previous examples, for $\eta'_0 < 1$, typically more periods are required for convergence than for $\eta'_0 > 1$. However, the opposite occurs if we consider time-steps as time units. If we consider time-steps (right), despite the heterogeneity of the coupling strengths, the analytical approximation is quite accurate.

4.4 Conclusions and future work

In this chapter we have characterized the complex phenomena taking place in a simplified neural network model, namely, the existence of a phase transition and hysteresis induced by the transmission delay of the spikes. Using a deterministic approximation of the model, we have derived an analytical lower bound for the mean period which is more accurate than the existing mean field approach (Rodríguez et al., 2001). The results has been illustrated by means of a globally coupled homogeneous network in which all neuron states are updated in parallel at each time-step. As described in more detail in (Kaltenbrunner et al., 2007b), these results can be extended to heterogeneous networks and sequential dynamics.

We also have considered synaptic dynamics in the model. Implicit in our analysis is that synaptic plasticity acts as a homeostatic mechanism which drives and maintains the system into the critical regime. When the network has learned this dynamic state, external perturbations such as stimuli presented to the network would cause a transient activity. During this transient, synapses could be modified to encode the proper values which drive the whole network to attain a characteristic synchronized pattern according to the external stimuli presented. We conjecture that the hysteresis effect shown in the regime of $\eta < 1$ may be suitable for such purposes, since the network here is able to keep the same pattern of activity until the critical state is reached again. We have developed a local learning rule which successfully makes the network to self-organize near the critical state where the phase transition occurs. The exploration of its use and applications is left for future studies.

In the presented experiments, the plasticity rule has been applied to networks initialized with homogeneous synaptic efficacies. Heterogeneity is dynamically introduced because when a neuron produces a spike, only its afferent synapses are updated. This heterogeneity, however, cancels out when all units have fired the same number of times. Consequently, the coupling strengths remain homogeneous if the period is considered as the time unit. We expect that results still hold for small or intermediate deviations in the distributions of the synapses.

Our approach is based on what we called dissipated or surplus of spontaneous evolution. Note that an extremely similar approach has been suggested in (Kinouchi and Copelli, 2006), where the analog quantity to our dissipated spontaneous evolution corresponds to the sensitivity and dynamic range of the network. Certainly, the latter magnitude leads to a more suitable interpretation in computational terms of sensory processing, but as far as we know, no local rule exists which makes the system self-organize into the critical state except the one proposed here.

When the network is operating at the critical regime, the dynamics can be seen as balancing between a predictable pattern of activity and uncorrelated random behavior typically present in SOC (Bak, 1996). One would expect as well to find macroscopic magnitudes distributed according to scale-free distributions. Preliminary results indicate that if the stochastic evolution is reset to zero ($p = 0$) at the critical state, inducing an artificial spike on a randomly selected unit causes neuronal avalanches of sizes and lengths which span several orders of magnitude and follow heavy tailed distributions. These results are in concordance with what is usually found for SOC.

We emphasize that two other quantities directly related to information processing may be also maximized at the critical point in the model under consideration. One is directly related with the range of sensitivity, which has already been observed to have an maximum at $\eta = 1$. The work developed in (Kaltenbrunner et al., 2007b) provides both upper τ_{max} and lower bounds τ_{min} of the period τ and it has been observed that the difference of both bounds is maximal at the critical point. This implies that the number of frequencies is also maximal, in harmony with the model considered in (Kinouchi and Copelli, 2006).

It is interesting to see the differences between our approach and the one taken in (Kinouchi and Copelli, 2006). There the average branching ratio σ of the network is used to define criticality. Briefly, σ is defined as the average number of excitations created in the next time step by a spike of a given neuron. The inverse of η seems to play the role of the branching ratio σ in our model. If we initialize the units uniformly in $[1, L]$, we have approximately one unit in every subinterval of length $\eta\varepsilon$, and in consequence, the closest unit to the threshold spikes in $1/\eta$ cases if it receives a spike. For $\eta > 1$, a spike of a neuron rarely induces another neuron to spike, so $\sigma < 1$. Conversely, for $\eta < 1$, the spike of a single neuron triggers more than one neuron to spike ($\sigma > 1$). Only for $\eta = 1$ the spike of a neuron elicits the order of one spike ($\sigma = 1$). Our study thus represents a realization of a local synaptic mechanism which induces global homeostasis towards an optimal branching factor.

Furthermore, since for $\eta \leq 1$ units are partitioned in several clusters, the number of accessible clusterings for the network at different values of $\eta \leq 1$ can be mapped directly to the problem of counting of the number of partitions of N into τ elements with constrained size, where the constraints can be directly deduced from the minimum cluster size $k_{min}(\tau)$ of Equation 4.4. Apparently, since τ lies in the interval $[\tau_{min}, \tau_{max}]$, the number of possible robust firing patterns should be also maximized

at $\eta = 1$. These patterns could, in principle, be used to encode and process large amounts of information. In this chapter, we have used the dissipated spontaneous evolution because of its simplicity to lead naturally to a local plasticity rule. The analysis of the latter two magnitudes is a topic of future research.

Part II

Statistical Inference

Chapter 5

Approximate inference in graphical models

In the second part of this thesis we address the problem of statistical inference in probabilistic graphical models. In a probabilistic network, each variable is represented as a node and edges indicate influence between variables. They represent powerful tools because they provide a compact representation of the joint distribution of the underlying probabilistic model where structural independence assumptions are captured.

Bayesian networks (BN) and Markov random fields (MRF) are canonical "examples" of graphical models. In a BN, edges are directed and represent causal relationships between variables. In contrast, MRF have undirected edges and there is no notion of causality between the variables. Factor graphs provide a unifying framework for representing and develop algorithms for both types of graphical models since any BN or MRF can be expressed in terms of a factor graph (Kschischang et al., 2001).

The most common inference task in a graphical model concerns the computation of single-variable marginals or *belief updating* task. Given an evidence, i.e. a set of observed variables, we want to compute the posterior probability of each variable alone. Another task of significant interest is the calculation of the partition sum Z, or normalization constant. In this work, we focus on both tasks. It is well known that they can be solved in polynomial time in the case of single-connected (or tree-like) networks applying the belief propagation algorithm (BP) Pearl (1988). For general networks, however, their complexity is NP-hard. This fact makes exact inference unfeasible for large instances and one has to look for approximate solutions. One way to get an approximate solution is precisely to use the same BP algorithm. Although the convergence of BP is not guaranteed in graphs with cycles, if it converges it can provide very accurate approximations.

Recently, Chertkov and Chernyak (2006b) developed a mathematical tool named loop calculus. They derived an exact expression for the partition sum (normalization constant) corresponding to a graphical model, which is an expansion around the BP solution. By adding correction terms to the BP

free energy, one for each "generalized loop" in the factor graph, the exact partition sum is obtained. However, the usually enormous number of generalized loops generally prohibits summation over *all* correction terms. In this chapter, we introduce truncated loop series BP (TLSBP), a particular way of truncating the loop series of Chertkov & Chernyak by considering generalized loops as compositions of simple loops. We analyze the performance of TLSBP in different scenarios, including the Ising model on square grids and regular random graphs, and on PROMEDAS, a large probabilistic medical diagnostic system. We show that TLSBP often improves upon the accuracy of the BP solution, at the expense of increased computation time. We also show that the performance of TLSBP strongly depends on the degree of interaction between the variables. For weak interactions, truncating the series leads to significant improvements, whereas for strong interactions it can be ineffective, even if a high number of terms is considered.

5.1 Introduction

Belief propagation (BP) (Pearl, 1988; Murphy et al., 1999) is a popular inference method that yields exact marginal probabilities on graphs without loops and can yield surprisingly accurate results on graphs with loops. BP has been shown to outperform other methods in rather diverse and competitive application areas, such as error correcting codes (Gallagher, 1963; McEliece et al., 1998), low level vision (Freeman et al., 2000), combinatorial optimization (Mézard et al., 2002) and stereo vision (Sun et al., 2005).

Associated to a probabilistic model is the partition sum, or normalization constant, from which marginal probabilities can be obtained. Exact calculation of the partition function is only feasible for small problems, and there is considerable statistical physics literature devoted to the approximation of this quantity. Existing methods include stochastic Monte Carlo techniques (Potamianos and Goutsias, 1997) or deterministic algorithms which provide lower bounds (Jordan et al., 1999; Leisink and Kappen, 2001), upper bounds (Wainwright et al., 2005), or approximations (Yedidia et al., 2005).

Recently, Chertkov and Chernyak (2006b) have presented a loop series expansion formula that computes correction terms to the belief propagation approximation of the partition sum. The series consists of a sum over all so-called generalized loops in the graph. When all loops are taken into account, Chertkov & Chernyak show that the exact result is recovered. Since the number of generalized loops in a graphical model easily exceeds the number of configurations of the model, one could argue that the method is of little practical value. However, if one could truncate the expansion in some principled way, the method could provide an efficient improvement to BP.[1]

[1]Note that the number of generalized loops in a finite graph is finite, and strictly speaking, the term *series* denotes an *infinite* sequence of terms. For clarity, we prefer to use the original terminology.

Most inference algorithms on loopy graphs can be viewed as generalizations of BP, where messages are propagated between regions of variables. For instance, the junction-tree algorithm (Lauritzen and Spiegelhalter, 1988) which transforms the original graph in a region tree such that the influence of all loops in the original graph is implicitly captured, and the exact result is obtained. However, the complexity of this algorithm is exponential in time and space on the size of the largest clique of the resulting join tree, or equivalently, on the tree-width of the original graph, a parameter which measures the network complexity. Therefore, for graphs with high tree-width one is resorted to approximate methods such as Monte Carlo sampling or generalized belief propagation (GBP) (Yedidia et al., 2005), which captures the influence of short loops using regions which contain them. One way to select valid regions is the cluster variation method (CVM) (Pelizzola, 2005). In Dechter et al. (2002) a heuristic to select regions which avoid cycles as much as possible was proposed. In general, selecting a good set of regions is not an easy task, as described by Welling et al. (2005). Alternatively, double-loop methods can be used (Heskes et al., 2003; Yuille, 2002) which are guaranteed to converge, often at the cost of more computation time.

In this chapter we propose TLSBP, an algorithm to compute generalized loops in a graph which are then used for the approximate computation of the partition sum and the single-node marginals. The proposed algorithm is parametrized by two arguments which are used to prune the search for generalized loops. For large enough values of these parameters, all generalized loops present in a graph are retrieved and the exact result is obtained. One can then study how the error is progressively corrected as more terms are considered in the series. For cases were exhaustive computation of all loops is not feasible, the search can be pruned, and the result is a truncated approximation of the exact solution. We focus mainly on problems where BP converges easily, without the need of damping or double loop alternatives (Heskes et al., 2003; Yuille, 2002) to force convergence. It is known that accuracy of the BP solution and convergence rate are negatively correlated. Throughout the chapter we show evidence that for those cases where BP has difficulties to converge, loop corrections are of little use, since loops of all lengths tend to have contributions of similar order of magnitude.

The rest of this chapter is organized as follows. In Section 5.2 we briefly summarize the series expansion method of Chertkov and Chernyak (2006b). In Section 5.3 we provide a formal characterization of the different types of generalized loops that can be present in an arbitrary graph. This description is relevant to understand the proposed algorithm described in Section 5.5. We present experimental results in Section 5.6 for the Ising model on grids, regular random graphs and medical diagnosis. Concerning grids and regular graphs, we show that the success of restricting the loop series expansion to a reduced quantity of loops depends on the type of interactions between the variables in the network. For weak interactions, the largest correction terms come from the small elementary loops and therefore truncation of the series at some maximal loop length can be effective. For

strong interactions, loops of all lengths contribute significantly and truncation is of limited use. We numerically show that when more loops are taken into account, the error of the partition sum decreases and when all loops are taken into account the method is correct up to machine precision. We also apply the truncated loop expansion to a large probabilistic medical diagnostic decision support system (Wiegerinck et al., 1999). The network has 2000 diagnoses and about 1000 findings and is intractable for computation. However, for each patient case unobserved findings and irrelevant diagnoses can be pruned from the network. This leaves a much smaller network that may or may not be tractable depending on the set of clamped findings. For a number of patient cases, we compare the BP approximation and the truncated loop correction. We show results and characterize when the loop corrections significantly improve the accuracy of the BP solution. Finally, in Section 5.7 we provide some concluding remarks.

5.2 Belief Propagation and the Loop Series Expansion

Consider a probability model on a set of binary variables $x_i = \pm 1, i = 1, \ldots, n$:

$$P(x) = \frac{1}{Z} \prod_{\alpha=1}^{m} f_\alpha(x_\alpha), \qquad Z = \sum_x \prod_{\alpha=1}^{m} f_\alpha(x_\alpha), \tag{5.1}$$

where $\alpha = 1, \ldots, m$ labels interactions (factors) on subsets of variables x_α, and Z is the partition function, which sums over all possible states or variable configurations. Note that the only restriction here is that variables are binary, since arbitrary factor nodes are allowed, as in Chertkov and Chernyak (2006b).

The probability distribution in (5.1) can be directly expressed by means of a factor graph (Kschischang et al., 2001), a bipartite graph where variable nodes i are connected to factor nodes α if and only if x_i is an argument of f_α. Figure 5.5 (left) on page 84 shows an example of a graph where variable and factor nodes are indicated by circles and squares respectively.

For completeness, we now briefly summarize Pearl's belief propagation (BP) (Pearl, 1988) and define the Bethe free energy. If the graph is acyclic, BP iterates the following message update equations, until a fixed point is reached:

variable i to factor α:
$$\mu_{i\to\alpha}(x_i) = \prod_{\beta \ni i \setminus \{\alpha\}} \mu_{\beta\to i}(x_i), \tag{5.2}$$

factor α to variable i:
$$\mu_{\alpha\to i}(x_i) = \sum_{x_{\alpha\setminus\{i\}}} f_\alpha(x_\alpha) \prod_{j\in\alpha\setminus\{i\}} \mu_{j\to\alpha}(x_j), \tag{5.3}$$

where $i \in \alpha$ denotes variables included in factor α, and $\alpha \ni i$ denotes factor indices α which have i as argument. After the fixed point is reached, exact marginals and correlations associated with the

factors ("beliefs") can be computed using:

$$b_i(x_i) \propto \prod_{\alpha \ni i} \mu_{\alpha \to i}(x_i), \tag{5.4}$$

$$b_\alpha(x_\alpha) \propto f_\alpha(x_\alpha) \prod_{i \in \alpha} \mu_{i \to \alpha}(x_i), \tag{5.5}$$

where $\propto$ indicates normalization so that beliefs sum to one.

For graphs with cycles the same update equations can be iterated (the algorithm is then called loopy, or iterative, belief propagation), and one can still obtain very accurate approximations of the beliefs. However, convergence is not guaranteed in these cases. For example, BP can get stuck in limit cycles. An important step towards the understanding and characterization of the convergence properties of BP came from the observation that fixed points of this algorithm correspond to stationary points of a particular function of the beliefs, known as the Bethe free energy (Yedidia et al., 2000), which is defined as:

$$F_{BP} = U_{BP} - H_{BP}, \tag{5.6}$$

where U_{BP} is the Bethe average energy:

$$U_{BP} = -\sum_{\alpha=1}^{m} \sum_{x_\alpha} b_\alpha(x_\alpha) \log f_\alpha(x_\alpha),$$

and H_{BP} is the Bethe approximate entropy:

$$H_{BP} = -\sum_{\alpha=1}^{m} \sum_{x_\alpha} b_\alpha(x_\alpha) \log b_\alpha(x_\alpha) + \sum_{i=1}^{n} (d_i - 1) \sum_{x_i} b_i(x_i) \log b_i(x_i), \tag{5.7}$$

where d_i is the number of neighboring factor nodes of variable node i. The second term in (5.7) ensures that every node in the graph is counted once (see Yedidia et al., 2005, for details). The BP algorithm tries to minimize (5.6) and, for trees, the exact partition function can be obtained after the fixed point has been reached, $Z = \exp(-F_{BP})$. However, for graphs with loops, F_{BP} provides just an approximation.

If one can calculate the exact partition function Z defined in Equation (5.1), one can also calculate any marginal in the network. For instance, the marginal

$$P_i(x_i) = \left. \frac{\partial \log Z(\theta_i)}{\partial \theta_i(x_i)} \right|_{\theta_i \to 0}, \qquad \text{where} \qquad Z(\theta_i) := \sum_x e^{\theta_i x_i} \prod_{\alpha=1}^{m} f_\alpha(x_\alpha)$$

is the partition sum of the network, perturbed by an additional local field potential θ_i on variable x_i.

Alternatively, one can compute different partition functions for different settings of the variables, and derive the marginals from ratios of them:

$$P_i(x_i) = \frac{Z^{x_i}}{\sum_{x_i'} Z^{x_i'}}, \tag{5.8}$$

where Z^{x_i} indicates the partition function calculated from the same model conditioning on variable i, that is, with variable i fixed (clamped) to value x_i. Therefore, approximation errors in the computation of any marginal can be related to approximation errors in the computation of Z. We will thus focus on the approximation of Z mainly, although marginal probabilities will be computed as well.

Of central interest in this work is the concept of generalized loop, which is defined in the following way:

Definition 1 *A **generalized loop** in a graph $G = \langle V, E \rangle$ is any subgraph $C = \langle V', E' \rangle$, $V' \subseteq V, E' \subseteq (V' \times V') \cap E$ such that each node in V' has degree two or larger. The length (size) of a generalized loop is its number of edges.*

For the rest of the chapter, the terms loop and generalized loop are used interchangeably. The main result of Chertkov and Chernyak (2006b) is the following. Let $b_\alpha(x_\alpha), b_i(x_i)$ denote the beliefs after the BP algorithm has been converged, and let $Z_{\mathrm{BP}} = \exp(-F_{\mathrm{BP}})$ denote the corresponding approximation to the partition sum, with F_{BP} the value of the Bethe free energy evaluated at the BP solution. Then Z_{BP} is related to the exact partition sum Z as:

$$Z = Z_{\mathrm{BP}} \left(1 + \sum_{C \in \mathcal{C}} r(C) \right), \qquad r(C) = \prod_{i \in C} \mu_i(C) \prod_{\alpha \in C} \mu_\alpha(C), \tag{5.9}$$

where summation is over the set $\mathcal{C}$ of all generalized loops in the factor graph.

Any term $r(C)$ in the series corresponds to a product with as many factors as nodes present in the loop. Each factor is related to the beliefs at each variable node or factor node according to the following formulas:

$$\mu_i(C) = \frac{(1-m_i)^{q_i(C)-1} + (-1)^{q_i(C)} (1+m_i)^{q_i(C)-1}}{2(1-m_i^2)^{q_i(C)-1}}, \qquad q_i(C) = \sum_{\alpha \in C, \alpha \ni i} 1, \tag{5.10}$$

$$\mu_\alpha(C) = \sum_{x_\alpha} b_\alpha(x_\alpha) \prod_{i \in C, i \in \alpha} (x_i - m_i), \tag{5.11}$$

where $m_i = \sum_{x_i} b_i(x_i) x_i = b_i(+) - b_i(-)$ is the expected value of x_i computed in the BP approximation.

In this thesis we do not give details of the derivation of this result. The interested reader is addressed to Chertkov and Chernyak (2006b).

Generally, terms $r(C)$ can take positive or negative values. Even the same variable i may have positive or negative subterms μ_i depending on the structure of the particular loop.

Expression (5.9) represents an exact and finite decomposition of the partition function with the first term of the series being exactly represented by the BP solution. Note that, although the series is finite, the number of generalized loops in the factor graph can be enormous and easily exceed the number of configurations 2^n. In these cases the loop series is less efficient than the most naive way to compute Z exactly, namely by summing the contributions of all 2^n configurations one by one.

On the other hand, it may be that restricting the sum in (5.9) to a subset of the total generalized loops captures the most important corrections and may yield a significant improvement in comparison to the BP estimate. We therefore define the truncated form of the loop corrected partition function as:

$$Z_{TLSBP} = Z_{\mathrm{BP}} \left(1 + \sum_{C \in \mathcal{C}'} r(C) \right), \tag{5.12}$$

where summation is over the subset $\mathcal{C}' \subseteq \mathcal{C}$ obtained by Algorithm 2, which we will discuss in Section 5.5. Approximations for the single-node marginals can then be obtained from (5.12), using the method proposed in Equation (5.8):

$$b_i'(x_i) = \frac{Z_{TLSBP}^{x_i}}{\sum_{x_i'} Z_{TLSBP}^{x_i'}}. \tag{5.13}$$

Because the terms $r(C)$ can have different signs, the approximation Z_{TLSBP} is in general not a bound of the exact Z, but just an approximation.

5.3 Loop Characterization

In this section we characterize different types of generalized loops that can be present in a graph. This classification is the basis of the algorithm described in the next section and also exemplifies the different shapes a generalized loop can take. For clarity, we illustrate them by means of a factor graph arranged in a square lattice with only pairwise interactions. However, definitions are not restricted to this particular model and can be applied generally to any factor graph.

Definition 2 *A simple (elementary) generalized loop (from now on **simple loop**) is defined as a connected subgraph of the original graph where all nodes have exactly degree two.*

This type of generalized loop coincides with the concept of simple circuit or simple cycle in graph theory: a path which starts and ends at the same node with no repeated vertices except for the start and end vertex. Figure 5.1a shows an example of a simple loop of size 8. On the contrary, in Figure 5.1b we show an example of generalized loop which is not a simple loop, because three nodes have degree larger than two.

We now define the union of two generalized loops, $l_1 = \langle V_1, E_1 \rangle$ and $l_2 = \langle V_2, E_2 \rangle$, as the generalized loop which results from taking the union of the vertices and the edges of l_1 and l_2, that is, $l' = l_1 \cup l_2 = \langle V_1 \cup V_2, E_1 \cup E_2 \rangle$. Note that the union of two simple loops is never a simple loop except for the trivial case in which both loops are equal. Figure 5.1b shows an example of a generalized loop which can be described as the union of three simple loops, each of size 8. The same example can be also defined as the union of two overlapping simple loops, each of size 12.

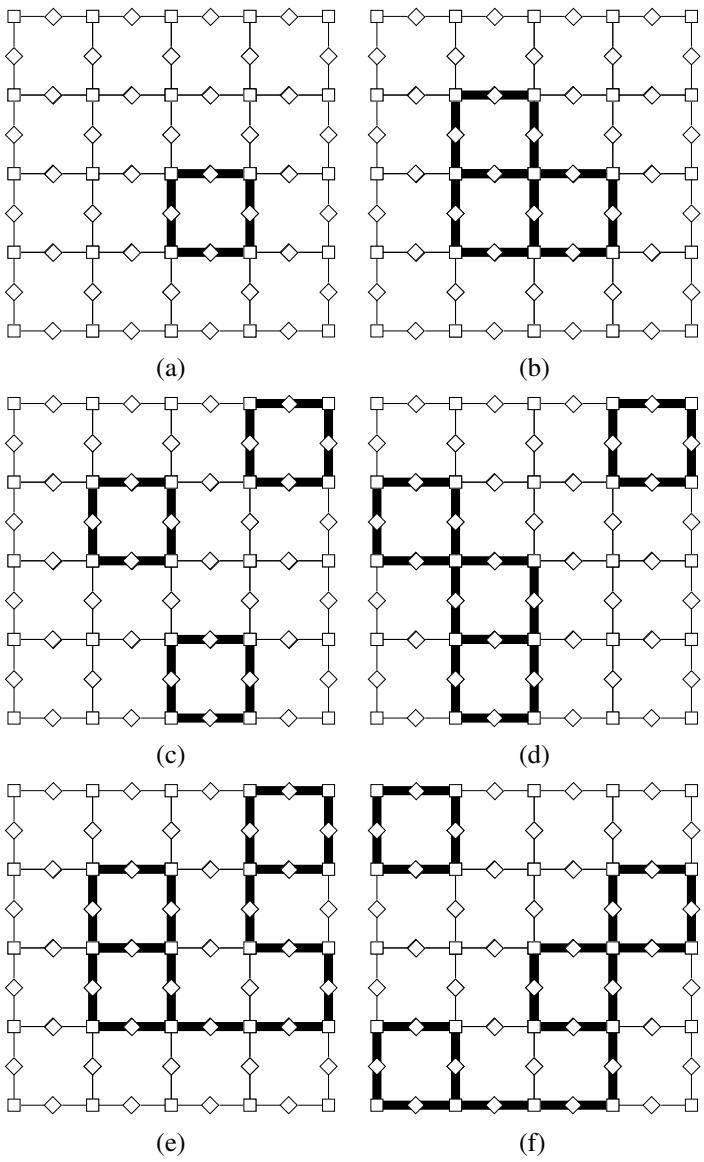

Figure 5.1: Examples of generalized loops in a factor graph with lattice structure. Variable nodes and factor nodes are represented as squares and rhombus respectively. Generalized loops are indicated using bold edges underlying the factor graph. **(a)** A simple loop. **(b)** A non-simple loop which is neither a disconnected loop nor a complex loop. **(c)** A disconnected loop of three components, each a simple loop. **(d)** A disconnected loop of two components, the left one a non-simple loop. **(e)** A complex loop which is not a disconnected loop. **(f)** A complex loop which is also a disconnected loop. *(See text for definitions).*

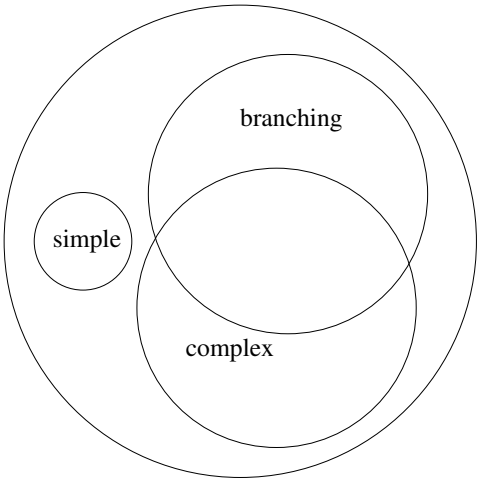

Figure 5.2: Diagrammatic representation of the different types of generalized loops present in any graph. Sizes of the sets are just indicative and depend on the particular instance.

Definition 3 *A disconnected generalized loop,* ***disconnected loop****, is defined as a generalized loop with more than one connected component.*

Figure 5.1c shows an example of a disconnected loop composed of three simple loops. Note that components are not restricted to be simple loops. Figure 5.1d illustrates this fact using an example where one connected component (the left one) is not a simple loop.

Definition 4 *A complex generalized loop,* ***complex loop****, is defined as a generalized loop which cannot be expressed as the union of two or more different simple loops.*

Figures 5.1e and 5.1f are examples of complex loops. Intuitively, they result after the connection of two or more connected components of a disconnected loop.

Any generalized loop can be categorized according to these three different categories: a simple loop cannot be a disconnected loop, neither a complex loop. On the other hand, since Definitions 3 and 4 are not mutually exclusive, a disconnected loop can be a complex loop and vice-versa, and also there are generalized loops which are neither disconnected nor complex, for instance the example of Figure 5.1b. An example of a disconnected loop which is not a complex loop is shown in Figure 5.1c. An example of a complex loop which is not a disconnected loop is shown in Figure 5.1e. Finally, an example of a complex loop which is also a disconnected loop is shown in Figure 5.1f.

We finish this characterization using a diagrammatic representation in Figure 5.2 which illustrates the definitions. Usually, the smallest subset contains the simple loops and both disconnected loops and complex loops have nonempty intersection. There is another subset of all generalized loops which are neither simple, disconnected, nor complex.

5.4 A small example: SAT problem

We now illustrate all these concepts considering an example in the context of the SAT problem. The SAT problem is the problem of deciding whether a formula is satisfiable or not. A formula is given in conjunctive normal form (CNF), a conjunction of m constraints or clauses between n boolean variables A formula is satisfiable if there exists a satisfying assignment, that is, an assignment of truth values to the variables such that the formula is evaluated as true. Such an assignment is called a *model* for the formula. In our case, to keep the same notation as before, we interpret that $+1$ correspond to true and -1 to false for the individual variables, and clauses evaluate to 1 if are satisfied and 0 otherwise.

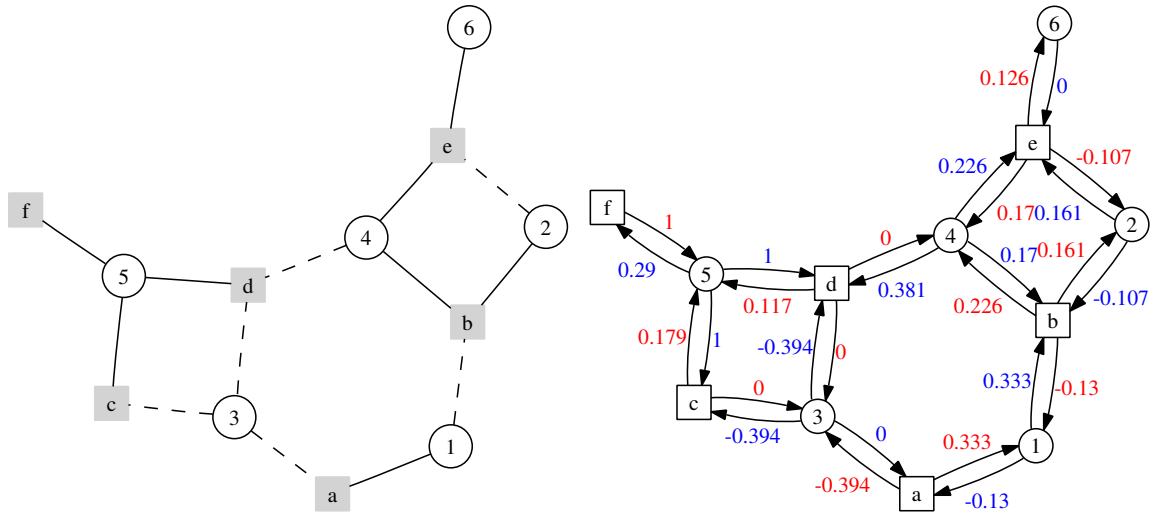

Figure 5.3: (left) An example of a CNF represented as a factor graph. The formula contains 6 variables and 6 clauses indicated by round and squared nodes respectively. We use $a, b, c, \ldots$ to denote clauses and $1, 2, 3, \ldots$ to denote variables. The corresponding CNF is the formula (5.14). (right) The BP messages after convergence. Red coloured messages correspond to factor-to-variable $\gamma_{\alpha \to i}$ messages and blue ones to variable-to-factor $\gamma_{i \to \alpha}$ messages.

Figure 5.3 (left) shows the factor graph representation of a small CNF formula which consists of six variables and six clauses [2]. In this representation each boolean variable i is denoted as a round node and each clause α as a squared node. A factor node α is connected to a variable node i whenever i (or its negation) appears in α. Although the distinction between positive and negative literals in a clause is not necessary according to our previous notation, we differentiate between both types of interactions using solid and dashed lines respectively, so that the entire CNF can be read from its

[2] This simple example has been taken from Braunstein et al. (2005)

factor graph representation. For the example of Figure 5.3 the corresponding CNF is:

$$F = (x_1 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee x_4) \wedge (\bar{x}_3 \vee x_5) \wedge (\bar{x}_3 \vee \bar{x}_4 \vee x_5) \wedge (\bar{x}_2 \vee x_4 \vee x_6) \wedge (x_5). \tag{5.14}$$

A given factor $f_\alpha(x_\alpha)$ takes value 1 if the configuration x_α satisfies clause α or 0 otherwise. In the case that all factors are 1, the corresponding assignment is a model. If we consider the probability space built by all models taken with uniform probability, the partition function Z indicates their total number. Estimation of Z is therefore of interest for the model counting task. The single-variable marginal of a variable i represents the proportion of satisfying assignments in which x_i is true, $x_i = +1$. Taking again our previous example, we can enumerate all the possible $2^6 = 64$ existing assignments and compute by brute force all the models and the marginal probabilities for each variable. It turns out that there exist 17 satisfying assignments. The single variable marginals are indicated in the second column of Table 5.1.

var	$P(+)$	$b(+)$
1	$10/17 = 0.5882$	0.6064
2	$9/17 = 0.5294$	0.5272
3	$5/17 = 0.2941$	0.3031
4	$12/17 = 0.7059$	0.6907
5	$17/17 = 1$	1
6	$10/17 = 0.5882$	0.5632

Table 5.1: Probability (up to four decimals of precision) of each variable of being true in a satisfying assignment. The second column indicates the exact probability and the third column the BP approximation.

Since the factor graph is not a tree we can only expect a more or less accurate approximation if we iterate the BP equations (5.2). We start with random messages and update them using a parallel scheduling until changes in all BP messages between two successive updates where smaller than the machine precision threshold 10^{-16} in absolute value. For this example, convergence was reached after 8 iterations. We show in Figure 5.3 (right) the BP messages after the fixed point was reached. Note that, since variables are binary, all BP messages are vectors with two components. To summarize them in one single number we first normalized both variable-to-factor and factor-to-variable messages such that they sum up one, and subsequently we used the following parameterization: $\gamma_{\alpha\to i} = \mu_{\alpha\to i}(1) - \mu_{\alpha\to i}(2)$, and the same for $\gamma_{i\to\alpha}$.

Consider, for instance, the computation of the belief of variable 1. According to Equations (5.4) we multiply the messages $\mu_{a\to 1}\mu_{b\to 1}$ which correspond to $\left(\frac{\gamma_{a\to 1}+1}{2}, \frac{1-\gamma_{a\to 1}}{2}\right) \times \left(\frac{\gamma_{b\to 1}+1}{2}, \frac{1-\gamma_{b\to 1}}{2}\right) = (0.6667, 0.3333)\cdot(0.4351, 0.5649) = (0.2901, 0.1883)$. After proper normalization, $b_1 = (0.6064, 0.3936)$. The rest of beliefs are shown in the third column of Table 5.1. As can be seen, for most of the marginals the error is of the order of 0.01. Note that the fact that variable 5 is true in all assignments is exactly found by BP.

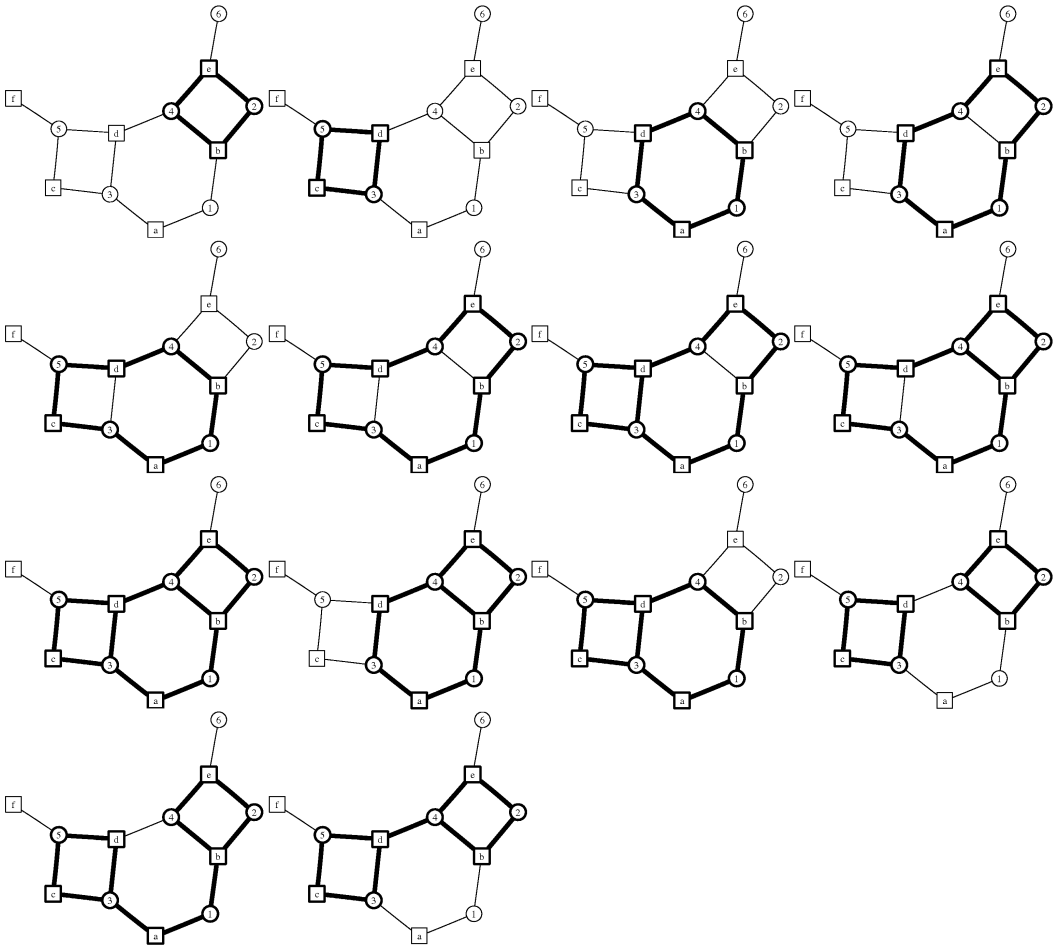

Figure 5.4: The 14 generalized loops of the SAT problem example. We can differentiate, in this order, 6 simple loops, 5 connected non-complex loops, 1 disconnected non-complex loop and 2 connected complex loops.

Using equations (5.6) and (5.7) we can compute the BP estimation of the partition function Z. In this context, the Bethe average energy U_{BP} is always zero [3] and for this example the entropy is $H_{BP} = -2.8724$. Therefore, the BP approximation of the number of satisfying assignments is $Z_{BP} = \exp(2.8724) = 17.6789$. The difference with the exact partition sum Z is 0.6789. Note that rounding Z_{BP} to the nearest integer would have overestimated the number of models in one unit.

To correct the BP result we use the loop series approach and apply the formulas (5.9), (5.10) and (5.11). The number of generalized loops in this small graph is 14. They are shown in Figure 5.4. According to our previous characterization, we can differentiate 6 simple loops, 5 connected non-complex loops, 1 disconnected non-complex loop and 2 connected complex loops. We only sketch the calculation of the loop term $r(1)$ corresponding to the first simple loop of Figure 5.4, which consists

[3] Note that factors can take 1 or 0 values only. If we assume that $(0 \log 0 = 0)$ then all the terms in (5.6) are zero.

of four nodes $e-2-b-4$. Since all nodes within the loop have two neighbours, we have $q_i = 2$ for $i = \{2,4\}$. The magnetizations are $m_2 = b_2(+) - b_2(-) = 0.0544$ and $m_4 = b_4(+) - b_4(-) = 0.3813$. The term is:

$$\begin{aligned} r(1) &= \mu_2(1)\mu_4(1)\mu_b(1)\mu_e(1) \\ &= \frac{1-m_2^2}{2-2m_2^2}\frac{1-m_4^2}{2-2m_4^2}\left(\sum_{x_b} b_b(x_b)(x_2-m_2)(x_4-m_4)\right)\left(\sum_{x_e} b_e(x_e)(x_2-m_2)(x_4-m_4)\right) \\ &= 1.0030 \cdot 1.1701 \cdot (-0.2232) \cdot 0.1466 = -0.0384. \end{aligned}$$

The 13 remaining terms can be obtained in a similar way. Surprisingly, for this random example no other loops with nonzero values exist, i.e. $r(i) = 0$ for $i = 2 \ldots 14$. This means that just using the correction corresponding to the first simple loop, we already recover the exact result and calculation of the remaining 13 terms is not needed. Expression (5.9) becomes:

$$Z = Z_{\text{BP}}\,(1 + r(1)) = 17.6789 \cdot (1 - 0.0384) = 17.$$

The reason behind this fact is that variable 5 is true in all assignments, and the BP approximation is already exact for this variable (see 5th row of Table 5.1). This statement implies that the original probabilistic model can be replaced by an equivalent model in which variable 5 is "clamped" to value 1. In terms of the factor graph representation, this is equivalent to remove the node 5, thus all generalized loops in which variable 5 appears do not exist in the equivalent model. There are 10 loops which contain variable 5 that do not exist in the clamped network. Note that the remaining 3 loops which do not contain variable 5 have zero contribution because of the factor d.

The former argument is related with the computation of the partition sum Z. To obtain the single-node marginal corresponding to variable i according to the method proposed in Section 5.2, we have to clamp i to its two possible values in the original model, and proceed as before to calculate both $Z_{TLSBP}^{x_i=+1}$ and $Z_{TLSBP}^{x_i=-1}$. The desired marginal is finally the ratio expressed in (5.8). The computation of all single variable marginals requires 12 runs of the BP algorithm in this small example.

5.5 The Truncated Loop Series Algorithm

In this section we describe the TLSBP algorithm to compute generalized loops in a factor graph, and use them to correct the BP solution. The algorithm is based on the principle that every generalized loop can be decomposed in smaller loops. The general idea is to search first for a subset of the simple loops and, after that, merge all of them iteratively until no new loops are produced. As expected, a brute force search algorithm will only work for small instances. We therefore prune the search using two different bounds as input arguments. Eventually, a high number of generalized loops which

presumably will account for the major contributions in the loops series expansion will be obtained. We show that the algorithm is complete, or equivalently, that all generalized loops are obtained by the proposed approach when the constraints expressed by the two arguments are relaxed. Although exhaustive enumeration is of little interest for complex instances, it allows to check the validity of (5.9) and to study the loop series expansion for simpler instances. The algorithm is composed of three steps:

1. First, we remove recursively all the leaves of the original graph, until its 2-core is obtained. This initial step has two main advantages. On one hand, since some nodes are deleted, the complexity of the problem is reduced. On the other hand, we can use the resulting graph as a test for any possible improvement to the BP solution. Indeed, if the original graph did not contain any loop then the null graph is obtained, the BP solution is exact on the original graph, and the series expansion has only one term. On the other hand, if a nonempty graph remains after this preprocessing, it will have loops and the BP solution can be improved using the proposed approach.

2. After the graph is preprocessed, the second step searches for simple loops. The result of this search will be the initial set of loops for the next step and will also provide a bound b which will be used to truncate the search for new generalized loops. Finding all circuits in an undirected graph is a problem addressed for long (Gotlieb and Corneil, 1967; Honkanen, 1978) whose computational complexity grows exponentially with the length of the cycle in a general graph. Nevertheless, we do not count all the simple loops but only a subset. Actually, to avoid dependence on particular instances, we parametrize this search by a size S, which limits the number of shortest simple loops to be considered. Once S simple loops have been found in order of increasing length, the length of the largest simple loop is used as the bound b for the remaining steps.

3. The third step of the algorithm consists of obtaining all non-simple loops that the set of S simple loops can"generate".

According to definition 4, complex loops can not be expressed as union of simple loops. To develop a complete method, in the sense that all existing loops can be obtained, we define the operation *merge loops*, which extends the simple union in such a way that complex loops are retrieved as well. Given two generalized loops, l_1, l_2, *merge loops* returns a *set* of generalized loops. One can observe that for each disconnected loop, a set of complex loops can be generated by connecting two (or more) components of the disconnected loop. In other words, complex loops can be expressed as the union of disjoint loops with a path connecting two vertices of different components. Therefore the set

computed by *merge loops* will have only one element $l' = \{l_1 \cup l_2\}$ if $l_1 \cup l_2$ is not disconnected. Otherwise, all the possible complex loops in which $l_1 \cup l_2$ appears are included in the resulting set.

We use the following procedure to compute all complex loops associated to the disconnected loop l': we start at a vertex of a connected component of l' and perform depth-first-search (DFS) until a vertex of a different component has been reached. At this point, the connecting path and the reached component are added to the first component. Now the generalized loop has one less connected component. This procedure is repeated again until the resulting generalized loop is not disconnected, or equivalently, until all its vertices are members of the first connected component. Iterating this search for each vertex every time two components are connected, and also for each initial connected component, one obtains all the required complex loops.

Algorithm 1 merge loops

Require:
$l_1 = \langle V_1, E_1 \rangle$ loop,
$l_2 = \langle V_2, E_2 \rangle$ loop,
b maximal length of a loop,
M maximal depth of complex loops search,
G preprocessed factor graph

1: $newloops \leftarrow \emptyset$
2: **if** $(|E_1 \cup E_2| \leq b)$ **then**
3: $C \leftarrow$ Find connected components$(l_1 \cup l_2)$
4: $newloops \leftarrow \{l_1 \cup l_2\}$
5: **for all** $(c_i \in C)$ **do**
6: **for all** $(v_i \in c_i)$ **do**
7: $newloops \leftarrow newloops \cup$ Find complex loopsDFS(v_i, c_i, C, M, b, G)
8: **end for**
9: **end for**
10: **end if**
11: **RETURN** *newloops*

Note that deciding whether $l_1 \cup l_2$ is disconnected or not requires finding all connected components of the resulting loop. Moreover, given a disconnected loop, the number of associated complex loops can be enormous. In practice, the bound b obtained previously is used to reduce the number of calculations. First, testing if the length of $l_1 \cup l_2$ is larger than b can be done without computing the connected components. Second, the DFS search for complex loops is limited using b, so very large complex loops will not be retrieved.

However, restricting the DFS search for complex loops using the bound b could result in too deep searches. Consider the worst case of merging the two shortest, non-overlapping, simple loops

which have size L_s. The maximum depth of the DFS search for complex loops is $d = b - 2L_s$. Then the computational complexity of the merge loops operation depends exponentially on d. This dependence is especially relevant when $b >> L_s$, for instance in cases where loops of many different lengths exist. To overcome this problem we define another parameter M, the maximum depth of the DFS search in the merge loops operation. For small values of M, the operation *merge loops* will be fast but a few (if any) complex loops will be obtained. Conversely, for higher values of M the operation *merge loops* will find more complex loops at the cost of more time.

Algorithm 1 in the previous page describes briefly the operation *merge loops*. It receives two loops l_1 and l_2, and bounds b and M as arguments, and returns the set *newloops* which contains the loop resulting of the union of l_1 and l_2 plus all complex loops obtained in the DFS search bounded by b and M.

Algorithm 2 Algorithm TLSBP

Require:
S maximal number of simple loops,
M maximal depth of complex loops search,
G original factor graph

1: Run belief propagation algorithm over G
2: $G' \leftarrow$ Obtain the 2-core(G)
3: $\mathcal{C}' \leftarrow \emptyset$
4: **if** ($\neg$empty(G')) **then**
5: $\langle sloops, b \rangle \leftarrow$ Compute first S simple loops(G')
6: $\langle oldloops, newloops \rangle \leftarrow \langle sloops, \emptyset \rangle$
7: $\mathcal{C}' \leftarrow sloops$
8: **while** ($\neg$empty($oldloops$)) **do**
9: **for all** ($l_1 \in sloops$) **do**
10: **for all** ($l_2 \in oldloops$) **do**
11: $newloops \leftarrow newloops \cup$ mergeLoops(l_1, l_2, b, M, G')
12: **end for**
13: **end for**
14: $oldloops \leftarrow newloops$
15: $\mathcal{C}' \leftarrow \mathcal{C}' \cup newloops$
16: **end while**
17: **end if**
18: **RETURN** the result of expression (5.12) using $\mathcal{C}'$

Once the problem of expressing all generalized loops as compositions of simple loops has been solved using the *merge loops* operation, we need to define an efficient procedure to merge them. Note that, given S simple loops, a brute force approach tries all combinations of two, three, ... up to $S-1$ simple loops. Hence the total number is:

$$\binom{S}{2} + \binom{S}{3} + \ldots + \binom{S}{S-1} = O(2^S),$$

which is prohibitive. Nevertheless, we can avoid redundant combinations by merging pairs of loops iteratively: in a first iteration, all pairs of simple loops are merged, which produces new generalized loops. In a next iteration i, instead of performing all $\binom{S}{i}$ mergings, only the new generalized loops obtained in iteration $i-1$ are merged with the initial set of simple loops. The process ends when no new loops are found. Using this merging procedure, although the asymptotic cost is still exponential in S, many redundant mergings are not considered.

Summarizing, the third step applies iteratively the *merge loops* operation until no new generalized loops are obtained. After this step has finished, the final step computes the truncated loop corrected partition function defined in Equation (5.12) using all the obtained generalized loops. We describe the full procedure in Algorithm 2. Lines 2 and 4 correspond to the first and second steps and lines $5-13$ correspond to the third step.

To show that this process produces all the generalized loops we first assume that S is sufficiently large to account for all the simple loops in the graph, and that M is larger or equal than the number of edges of the graph. Now let C be a generalized loop. According to the definitions of Section 5.3, either C can be expressed as a union of s simple loops, or C is a complex loop. In the first case, C is clearly produced in the sth iteration. In the second case, let s' denote the number of simple loops which appear in C. Then C is produced in iteration s', during the DFS for complex loops within the merging of one of the simple loops contained in C.

The obtained collection of loops can be used for the approximation of the singe node marginals as well, as described in Equation (5.13). The method consists of clamping one variable i to all its possible values (± 1) and computing the corresponding approximations of the partition functions: $Z_{TLSBP}^{x_i=+1}$ and $Z_{TLSBP}^{x_i=-1}$. This requires to run BP in each clamped network, and reuse the set of loops replacing with zero those terms where the clamped variable appears. The computational complexity of approximating all marginals using this approach is in general $O(N \cdot L \cdot d \cdot T_{BP})$, where L is the number of found loops, d is the cardinality of the variables (two in our case), and T_{BP} the average time of BP to converge after clamping one variable. Usually, this task requires less computation time than the search for loops.

As a final remark, we want to stress a more technical aspect related to the implementation. Note that generalized loops can be expressed as the composition of other loops in many different ways.

In consequence, they all must be stored incrementally and the operation of checking if a loop has been previously counted or not should be done efficiently. An appropriate way to implement this fast look-up/insertion is to encode all loops in a string composed by the edge identifiers in some order with a separator character between them. This identifier is used as a key to index an ordered tree, or hash structure. In practice, a hash structure is only necessary if large amounts of loops need to be stored. For the cases analyzed here, choosing a balanced tree instead of a hash table resulted in a more efficient data structure.

5.6 Experiments

In this section we show the performance of TLSBP in three different scenarios. First, we focus on square lattices and study how loop corrections improve the BP solution as a function of the interaction between variables and the size of the problem. Second, we study the performance of the method in random regular graphs as a function of the degree between the nodes. Finally, we apply the algorithm on a medical diagnosis bayesian network.

In all the experiments we show results for tractable instances, where the exact solution using the junction tree (Lauritzen and Spiegelhalter, 1988) can be computed. Performance is evaluated comparing the TLSBP error against the BP solution, and also against the cluster variation method (CVM). Instead of using a generalized belief propagation algorithm (GBP) which usually requires several trials to find the proper damping factor to converge, we use a double-loop implementation which has convergence guarantees (Heskes et al., 2003). For this study we select as outer regions of the CVM method all maximal factors together with all loops that consist up to four different variables. This choice represents a good trade-off between computation time required for convergence and accuracy of the solution.

We report two different error measures. Concerning the partition function Z we compute:

$$\text{Error}_{Z'} = \left| \frac{\log Z'}{\log Z} \right|, \tag{5.15}$$

where Z' is the partition function corresponding to the method used: BP, TLSBP, or CVM. Error of single-node marginals is measured using the maximum ℓ_∞ error, which is a reasonable quantity if one is interested in worst-case scenarios:

$$\text{Error}_b = \max_{\substack{i=1,\dots,n \\ x_i=\pm 1}} |P_i(x_i) - b_i(x_i)|, \tag{5.16}$$

were again $b_i(x_i)$ are the single-node marginal approximations corresponding to the method used.

We use four different schemas for belief-updating of BP: *(i)* fixed and *(ii)* random sequential updates, *(iii)* parallel (or synchronous) updates, and *(iv)* residual belief propagation (RBP), a recent

method proposed by Elidan et al. (2006). The latter method schedules the updates of the BP messages heuristically by selecting the next message to be updated which has maximum *residual*, a quantity defined as an upper bound on the distance of the current messages from the fixed point. In general, we experienced that for some instances where the RBP method converged, the other update schemas (fixed, random sequential and parallel updates) failed to converge.

In all schemas we interpret that a fixed point is reached at iteration t when the maximum absolute value of the updates of all the messages from iteration $t-1$ to t is smaller than a threshold ϑ. We notice a large correlation between the order of magnitude of ϑ and the ratio between the BP and the TLSBP errors. For this reason we used a very small value of the threshold, $\vartheta = 10^{-15}$.

5.6.1 Ising Grids

This model is defined on a grid where each variable, also called spin, takes binary values $x_i = \pm 1$. A spin is coupled with its direct neighbors only, so that pairwise interactions $f_{ij}(x_i, x_j) = \exp(\theta_{ij} x_i x_j)$ are considered, parametrized by θ_{ij}. Every spin can be exposed to an external field $f_i(x_i) = \exp(\theta_i x_i)$, or single-node potential, parametrized by θ_i. Figure 5.5 (left) shows the factor graph associated to the 4x4 Ising grid, composed of 16 variables. The Ising grid model is often used as a test-bed for inference algorithms. It is of great relevance in statistical physics, and has applications in different areas such as image processing. In our context it also represents a challenge since it has many loops. Good results in this model will likely translate into good results for less loopy graphs.

Usually, two cases are differentiated according to the sign of the θ_{ij} parameters. For $\theta_{ij} > 0$

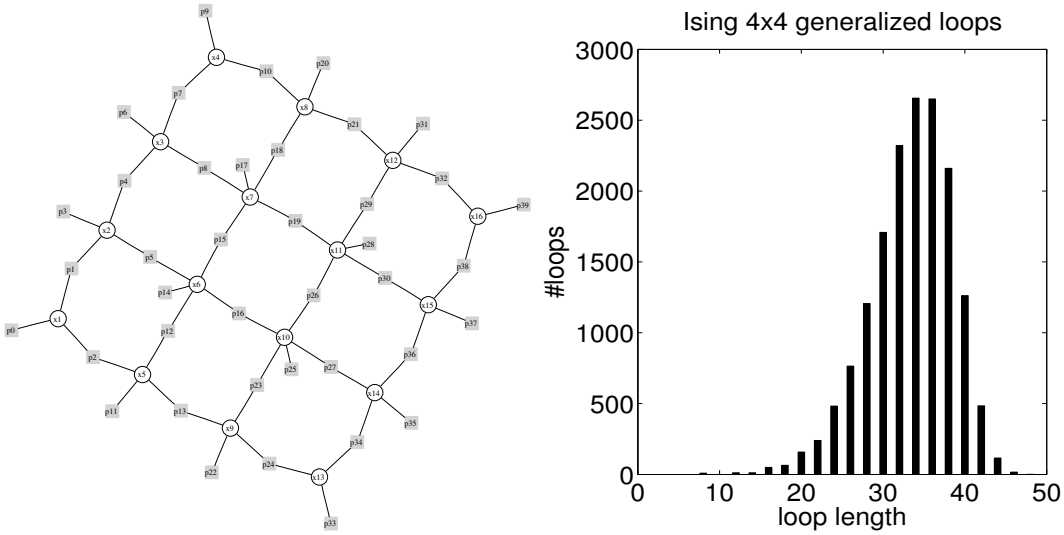

Figure 5.5: **(left)** A factor graph representing the 4x4 Ising grid. **(right)** Number of generalized loops as a function of the length using the factor graph representation.

coupled spins tend to be in the same state. This is known as the attractive, or "ferromagnetic" setting. On the other hand, for mixed interactions, θ_{ij} can be either positive or negative, and this setting is called "spin-glass" configuration. Concerning the external field, one can distinguish two cases. For the case of nonzero fields, larger values of θ_i imply easier inference problems in general. On the other hand, for $\theta_i = 0$, there exist two phase transitions from easy inference problems (small θ_{ij}) to more difficult ones (large θ_{ij}) depending on the type of pairwise couplings (see Mooij and Kappen, 2005, for more details).

This experimental subsection is structured in three parts: First, we study a small 4x4 grid. We then study the performance of the algorithm in a 10x10 grid, where complete enumeration of all generalized loops is not feasible. Finally, we analyze the scalability of the method with problem size.

The 4x4 Ising grid is complex enough to account for all types of generalized loops. It is the smallest size where complex loops are present. At the same time, the problem is still tractable and exhaustive enumeration of all the loops can be done.

We ran the TLSBP algorithm in this model with arguments S and M large enough to retrieve all the loops. Also, the maximum length b was constrained to be 48, the total number of edges for this model. After 4 iterations all generalized loops were obtained. The total number is 16371 from which 213 are simple loops. The rest of generalized loops are classified as follows: 174 complex and disconnected loops, 1646 complex but non-disconnected loops, 604 non-complex but disconnected loops, and 13734 neither complex nor disconnected loops.

Figure 5.5 (right) shows the histogram of all generalized loops for this small grid. Since we use the factor graph representation the smallest loop has length 8. The largest generalized loop includes all nodes and all edges of the *preprocessed* graph, and has length 48. The Poisson-like shape of the histogram is a characteristic of this model and for larger instances we observed the same tendency. Thus the analysis for this small model can be extrapolated to some extent to grids with more variables.

To analyze how the error changes as more loops are considered it is useful to sort all the terms $r(C)$ by their absolute value in descending order such that $|r(C_i)| \geq |r(C_{i+1})|$. We then compute, for each number of loops $l = 1 \ldots 16371$, the approximated partition function which accounts for the l most important loops:

$$Z_{TLSBP}(l) = Z_{BP}\left(1 + \sum_{i=1\ldots l} r(C_i)\right). \tag{5.17}$$

From these values of the partition function we calculate the error measure indicated in Equation (5.16). Estimations of the single-node marginals were obtained using the clamping method, and their corresponding error was calculated using Equation (5.15).

We now study how loop contributions change as a function of the coupling strength between the variables. We ran several experiments using mixed interactions with $\theta_{ij} \sim \mathcal{N}(0, \sigma^2)$ independently

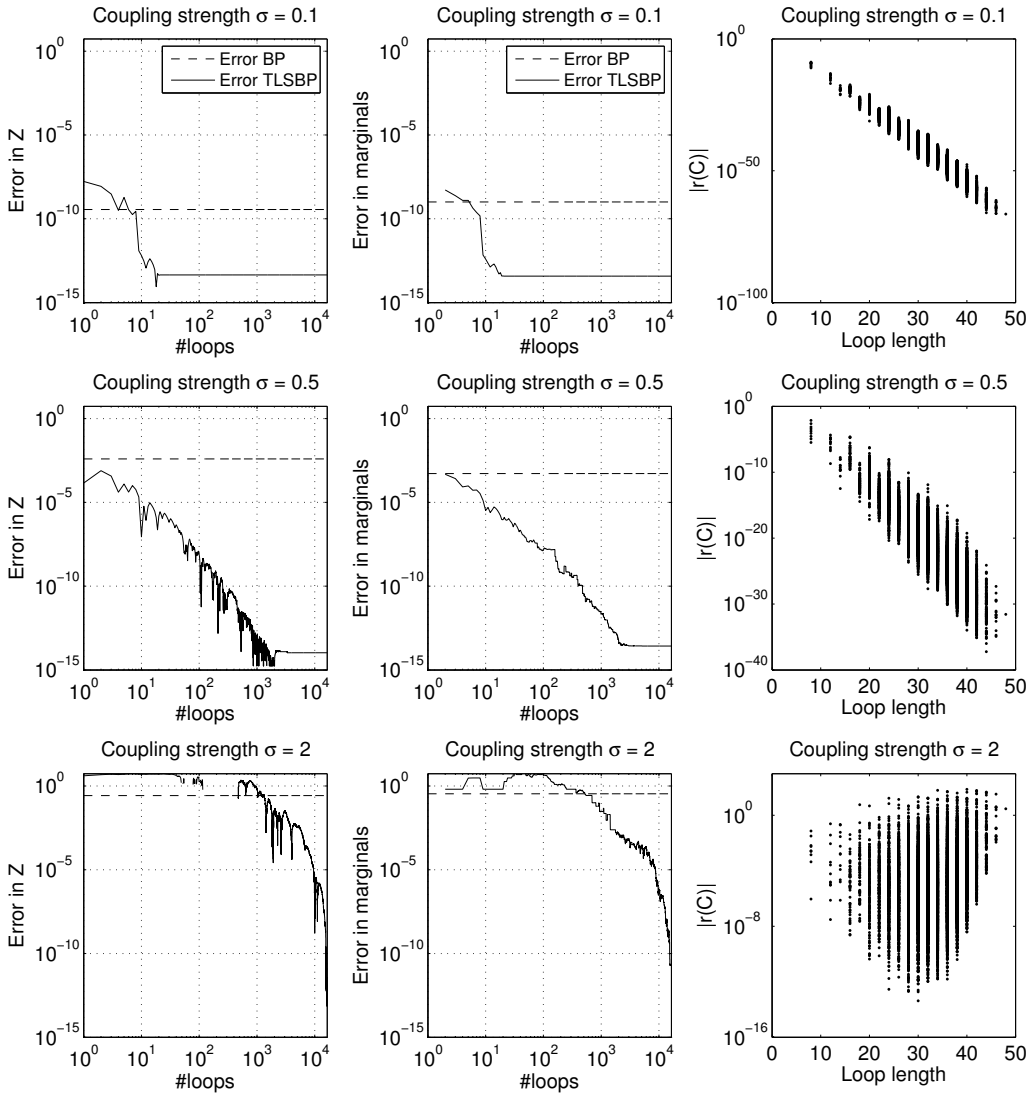

Figure 5.6: Cumulative error for the spin-glass 4x4 Ising grid for different interaction strengths, see Equation (5.17). **(left column)** Error of Z. **(middle column)** Error of single-node marginals. Dashed lines correspond to the BP error, and solid lines correspond to the loop-corrected (TLSBP) error. **(right column)** Absolute values of all loop terms as a function of the length of the corresponding loop.

for each factor node, and σ varying between 0.1 and 2. Single-node potentials were drawn according to $\theta_i \sim \mathcal{N}(0, 0.05^2)$. For small values of σ, interactions are weak and BP converges easily, whereas for high values of σ variables are strongly coupled and BP has more difficulties, or does not converge at all.

Figure 5.6 shows results of representative instances of three different interaction strengths. For each instance we plot the partition function error (left column) together with errors of the single-node marginals (middle column) and loop contributions as a function of the length (right column).

First, we can see that improvements of the partition sum correspond to improvements of the estimates of marginal probabilities as well. Second, for weak couplings ($\sigma = 0.1$, first row) we can see that truncating the series until a small number of loops (around 10) is enough to achieve machine precision. In this case the errors of BP are most prominently due to small simple loops. As the right column illustrates, loop contributions decrease exponentially with the size, and loops with the same length correspond to very similar contributions. Larger loops give negligible contributions and can thus be ignored by truncating the series. As interactions are strengthened, however, more loops have to be considered to achieve maximum accuracy, and contributions show more variability for a given length (see middle row). Also, oscillations of the error due to the different signs in loop terms (caused by the mixed interactions) of the same order of magnitude become more frequent. Eventually, for large couplings ($\sigma \geq 2$), where BP often fails to converge, loops of all lengths give significant contributions. In the bottom panels of Figure 5.6 we show an example of a 'difficult' case for which the BP result is not improved until more than 10^3 loop terms are summed. The observed discontinuities in the error of the partition sum are caused by the fact that oscillations become more pronounced, and corrections composed of negative terms $r_i(C_i)$ can result in negative values of the partially corrected partition function, see Equation (5.17). This occurs for very strong interactions only, and when a small fraction of the total number of loops is considered. In addition, as the right column indicates, there is a shift of the main contributions towards the largest loops.

After analyzing a small grid, we now address the case of the 10x10 Ising grid, where exhaustive enumeration of all the loops is not computationally feasible. We test the algorithm in two scenarios: for attractive interactions (ferromagnetic model) where pairwise interactions are parametrized as $\theta_{ij} = |\theta'_{ij}|, \theta'_{ij} \sim \mathcal{N}(0, \sigma^2)$, and also for the previous case of mixed interactions (spin-glass model). Single-node potentials were chosen $\theta_i \sim \mathcal{N}(0.1, 0.05^2)$ in both cases.

We show results in Figures 5.7 and 5.8 for three values of the parameter $S = \{10, 100, 1000\}$ and a fixed value of $M = 10$. For $S = 10$ and $S = 100$, only simple loops were obtained whereas for $S = 1000$, a total of 44590 generalized loops was used to compute the truncated partition sum. Results are averaged errors over 50 random instances. The selected loops were the same in all instances. Although in both types of interactions the BP error (solid line with dots) is progressively reduced as more loops are considered, the picture differs significantly between the two cases.

For the ferromagnetic case shown in Figure 5.7, we noticed that all loops have positive contributions, $r(C) > 0$. This is a consequence of this particular type of interactions, since all magnetizations have the same sign at the BP fixed point, and also all nodes have an even number of neighbors. Consequently, improvements in the BP result are monotonic as more loops are considered, and in this case, the TLSBP can be considered as a lower bound of the exact solution. For the case of $S = 1000$, the BP error is reduced substantially at nonzero σ, but around $\sigma \sim 0.5$, where the BP error reaches

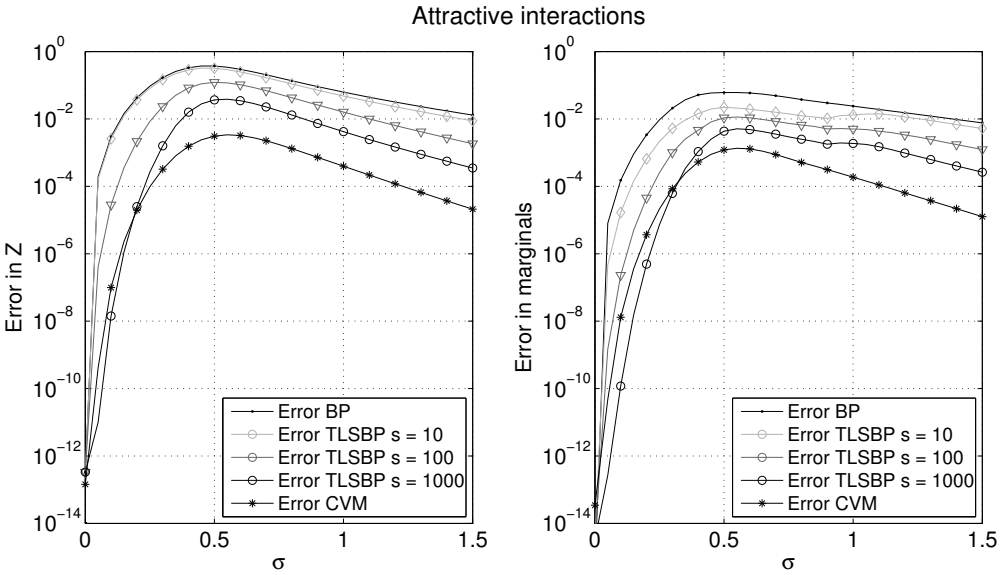

Figure 5.7: TLSBP error for the 10x10 Ising grid with attractive interactions for different values of the parameter S. **(left)** Error of the partition function. **(right)** Error of single-node marginals.

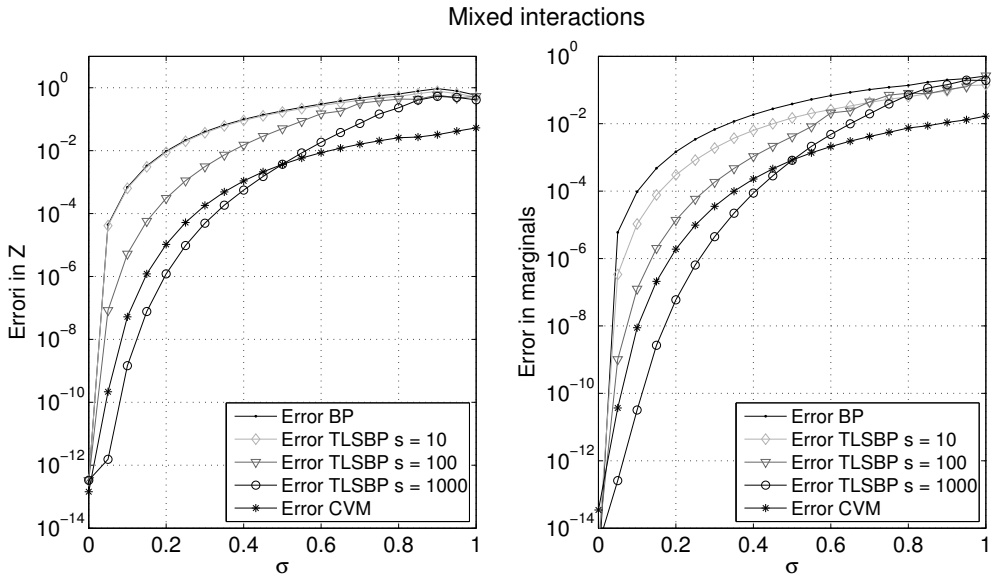

Figure 5.8: TLSBP error of the 10x10 Ising grid with mixed interactions for different values of the parameter S. **(left)** Error of the partition function. **(right)** Error of single-node marginals.

a maximum, also the TLSBP improvement is minimal. From this maximum, the BP error decreases again, and loop corrections tend to improve progressively the BP solution again as the coupling is strengthened. We remark that improvements were obtained for all instances in the three cases.

Comparing with CVM, TLSBP is better for weak couplings and for $S = 1000$ only. This indicates

that for intermediate and strong couplings one would need more than the selected 44590 generalized loops to improve on the CVM result.

For the case of spin-glass interactions we report different behavior. From Figure 5.8 we see again that for weak couplings the BP error is corrected substantially, but the improvement decreases as the coupling strength is increased. Eventually, for $\sigma \sim 1$ BP fails to converge in most of the cases and also gives poor results. In these cases loop corrections are of little use, and there is no actual difference in considering $S = 1000$ or $S = 10$. Moreover, because loop terms $r(C)$ now can have different signs, truncating the series can lead to worse results for $S = 1000$ than for $S = 10$. Interestingly, the range where TLSBP performs better than CVM is slightly larger in this type of interactions, TLSBP being better for $\sigma < 0.5$.

To end this subsection, we study how loop corrections scale with the number of nodes in the graph. We only use spin-glass interactions, since it is a more difficult configuration than the ferromagnetic case, as previous experiments suggest. We compare the performance for weak couplings ($\sigma = 0.1$), and strong couplings ($\sigma = 0.5$), where BP has difficulties to converge in large instances. The number of variables N^2 is increased for grids of size $N \times N$ until exact computation using the junction tree algorithm is not feasible anymore.

Since the number of generalized loops grows very fast with the size of the grid, we choose increasing values of S as well. We use values of S proportional to the number of variable nodes N^2 such that $S = 10N^2$. This simple linear increment in S means that as N is increased, the proportion of simple loops captured by TLSBP over the total existing number of simple loops decreases. It is interesting to see how this affects the performance of TLSBP in terms of time complexity and accuracy of the solution. For simplicity, M is fixed to zero, so no complex loops are considered. Moreover, to facilitate the computational cost comparison, we only compute mergings of pairs of simple loops. Actually, for large instances the latter choice does not modify the final set of loops, since generalized loops which can only be expressed as compositions of three or more simple loops are pruned using the bound b.

In Figure 5.9, the top panels show averaged results of the computational cost. The left plot indicates the relation between the number of loops computed by TLSBP and the time required to compute them. This relation can be fit accurately using a line which means that for this choice of parameters S and M, and considering only mergings of simple loops, the computational complexity of the algorithm grows just linearly with the found loops. One has to keep in mind that the number of loops obtained using the TLSBP algorithm grows much faster, but much less than the total number of existing loops in the model.

Figure 5.9b shows the CPU time consumed by CVM, TLSBP, and the junction tree algorithm. In this case, since we only compute the partition function Z, the CPU time of TLSBP is constant

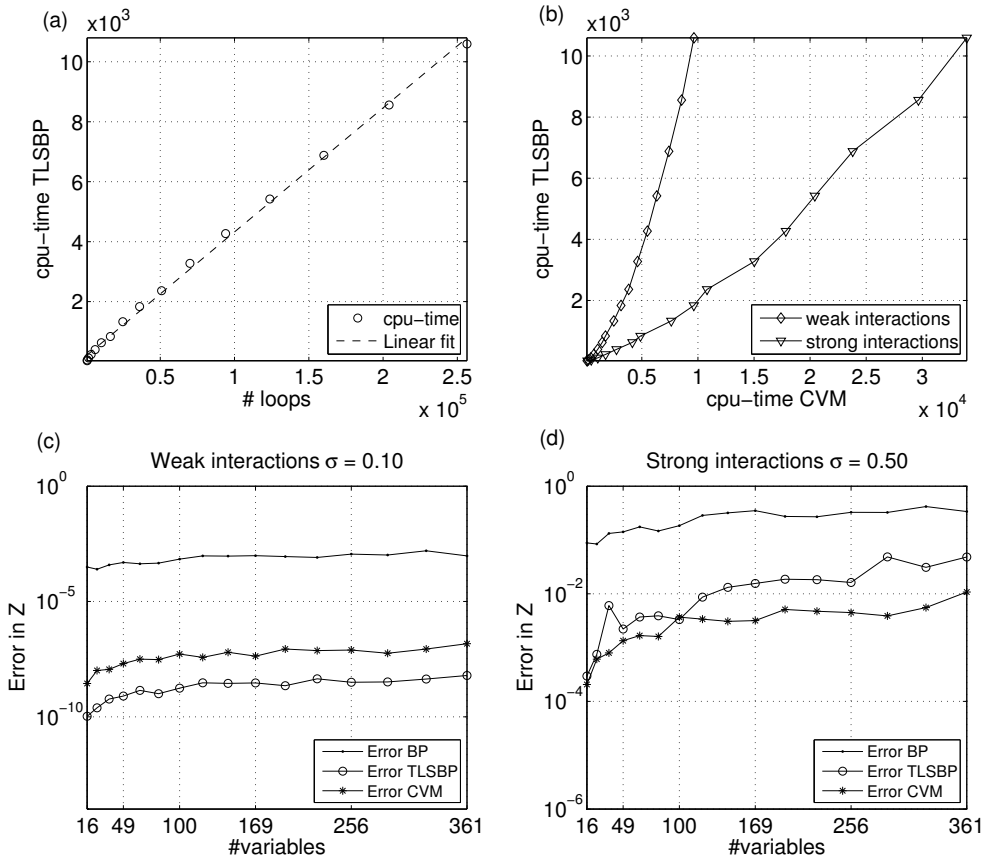

Figure 5.9: Scalability of the method in the Ising model. **(a)** Time complexity as a function of the produced number of generalized loops. **(b)** Relation between the time complexity of TLSBP and CVM. Comparison of the error of the partition function between BP, TLSBP and CVM as a function of the graph size for **(c)** weak interactions and **(d)** strong interactions.

for both weak and strong couplings. On the contrary, CVM depends on the type of interactions. For weak interactions, TLSBP is in general more efficient than CVM, although the scaling trend is slightly better for CVM. For $N = 19$, CVM starts to be more efficient than TLSBP. For strong interactions, CVM needs significantly more time to converge in all cases. If we compare the computational cost of the exact method against TLSBP, we can see that the junction tree is very efficient for networks with small N, and the best option in those cases. However, for $N > 17$, the junction tree needs more computation time, and for $N > 19$, the tree-width of the resulting grids is too large. TLSBP memory requirements were considerably less in these cases, since loops can be stored efficiently using sets of chars. Also, we can see that the TLSBP scaling is better for this choice of parameters than that of the exact method.

Bottom panels show the accuracy of the TLSBP solution. For weak couplings (bottom-left) the BP error is always decreased significantly for this choice of parameters and the improvement remains

almost constant as N increases, meaning that, in this case the number of loops which contribute most to the series expansion does not grow significantly with N. Interestingly, results are always better than CVM for this regime.

For strong couplings (bottom-right) the picture changes. First, results differ more between instances causing a less smooth curve. Second, the TLSBP error also increases with the problem size, so improvements tend to decrease with N, even faster than the BP error increase. Eventually, for the largest tractable instance the TLSBP improvement is still significant, about one order of magnitude. Comparing against CVM, unlike in the weak coupling scenario, the TLSBP method does not seem to perform better, and only for some cases TLSBP error is comparable to the CVM error on average. The accuracy of the TLSBP solution for these instances can be increased by considering larger values of S and M, at the cost of more time.

5.6.2 Random Graphs

The previous experimental results were focused on the Ising grid which only considers pairwise and singleton interactions in such a way that each node in the graph is at most linked with four neighbors. Here we briefly analyze the performance of TLSBP applied on a more general case, where interactions are less restricted.

We perform experiments on random graphs with regular topology, where each variable is coupled randomly with d other variables using pairwise interactions parametrized by $\theta_{ij} \sim \mathcal{N}(0, \sigma^2)$. Single-node potentials were parametrized in this case by $\theta_i \sim \mathcal{N}(0, 0.05^2)$. We study how loop corrections improve the BP solution as a function of the degree d, and compare improvements against the CVM. As in the previous subsection, for CVM we select the loops of four variables and all maximal factors as outer clusters.

Note that the rate of increase in the number of loops with the degree d is even higher than with the number of variables in the Ising model. Adding one more link to all the variables means adding N more factor nodes to the factor graph. This raises the number of loops dramatically.

For this scenario, we use $N = 20$ variables and also increase S every time d is increased. We simply start with $S = 10$ and use increments of 250 for each new d. M was set to 10, and all possible mergings were computed. We analyze two scenarios, weak ($\sigma = 0.1$) and strong couplings ($\sigma = 0.5$), and report averages over 60 random instances for each configuration. As Figure 5.11 (right) indicates, for $\sigma = 0.1$ BP converged in all instances, whereas for $\sigma = 0.5$ BP convergence becomes more difficult as we increase d.

Figure 5.10 (top) shows results for weak interactions. The TLSBP algorithm always corrects the BP error, although as d increases, the improvement is progressively reduced. We also notice that in all cases and methods the approximation of the partition function (left) is less accurate than the

approximation of the marginals (right). For $d = 15$, TLSBP improvements are still about one order of magnitude for the partition function, and even better for the marginals. As in previous experiments with square lattices, the TLSBP approach is generally better than CVM in the weak coupling regime. Here, it is also more stable, since for some dense networks the CVM error can be very large, as we can see for $d = 13$ and $d = 15$.

For strong interactions (bottom panels), we see that differences between approximations of the partition function and single-node marginals are more remarkable than in the previous case. The BP partition function is corrected by TLSBP in more than half of the instances for all degrees (see inset of Figure 5.10c, where we plot the fraction of instances where BP was corrected in those cases that converged), although for higher degrees, the TLSBP corrections are small using this choice of parameters. On the other hand, single-node BP marginals are corrected in almost all cases. In contrast,

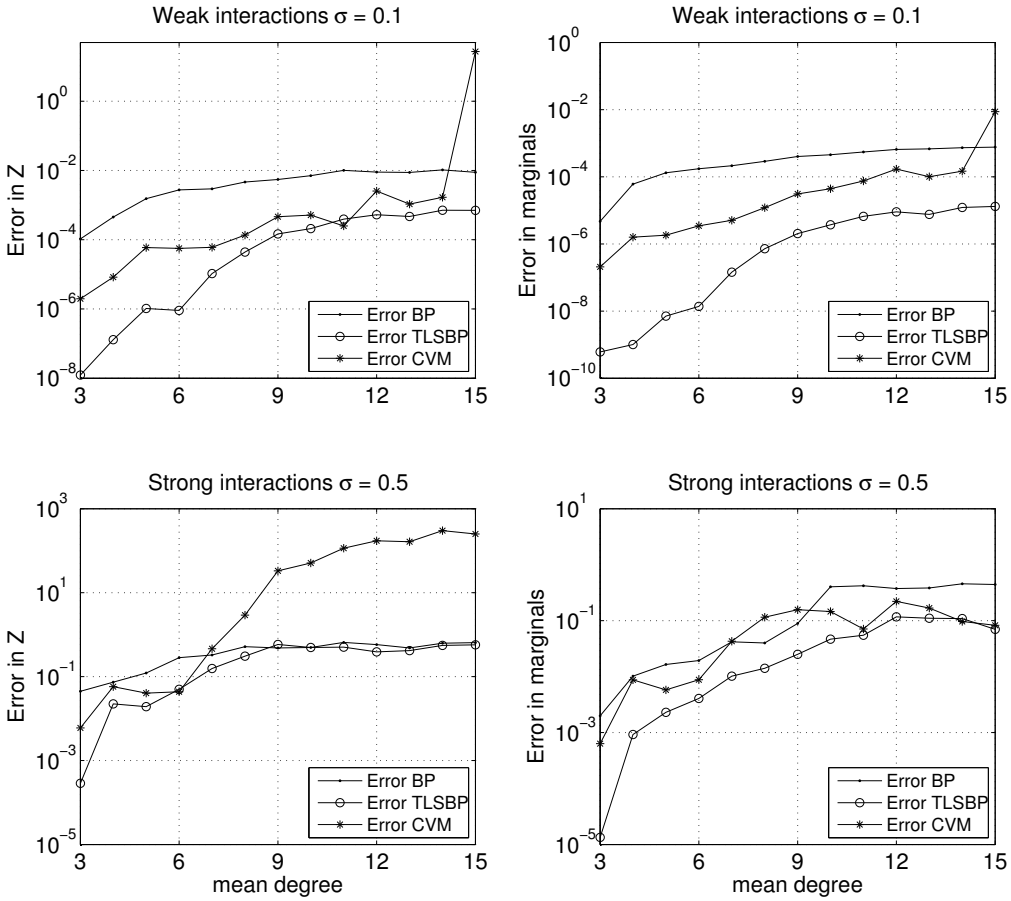

Figure 5.10: Results on random regular graphs. TLSBP and CVM errors as a function of the degree d. Results are averages over 60 random instances. Errors in the partition function for weak interactions **(a)**, marginals for weak interactions **(b)**, partition function for strong interactions **(c)**, and marginals for strong interactions **(d)**. Insets show percentage of instances where the BP error was corrected.

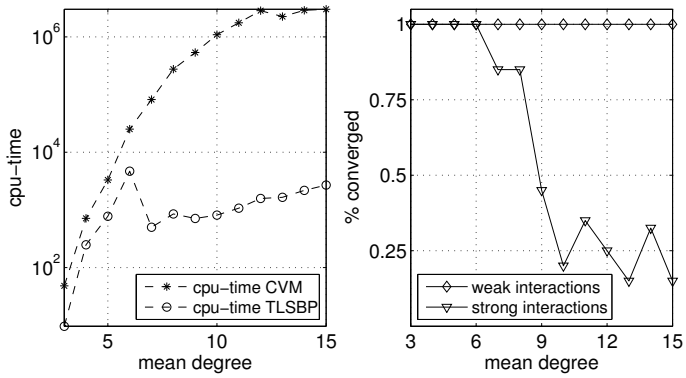

Figure 5.11: Results on random regular graphs. **(a)** Computation time of TLSBP and CVM. In this case, we averaged also all instances over all 60 weak and 60 strong interactions, since costs were very similar in both cases. **(b)** Fraction of the instances were BP converged. No convergence is reported when none of the four proposed schedules converged.

the CVM approach with our selection of outer clusters does not perform better than TLSBP in general. In particular, we see that CVM estimates of the partition function are very degraded as networks become more dense. This unsatisfactory performance of CVM in the estimation of the partition function is not as noticeable in the marginal estimates, where BP results are often improved, although with much more variability than the TLSBP method. Interestingly, for those few instances of dense networks for which BP converged, CVM estimates of the marginals were very similar to TLSBP.

Finally, we compare computational costs in Figure 5.11 (left). CVM requires significantly more time to converge than the time required by TLSBP searching for loops and calculating marginals. If we analyze in detail how the TLSBP cost changes, we can notice different types of growth for $d < 7$ and for $d \geq 7$. The reason behind these two scaling tendencies can be explained by the choice of TLSBP parameters, and the bound b (the size of the largest simple loop). For $d < 7$, many simple loops of different lengths are obtained. Consequently, the cost of the merging step grows fast, since many loops with length smaller than b are produced. On the other hand, for $d \geq 7$ simple loops have similar lengths and, therefore, less combinations result in additional loops with length larger than the bound b. Without bounding the length of the loops in the merging step, we would expect the first scaling tendency ($d < 7$) also for values of $d \geq 7$.

From these experiments we can conclude that TLSBP performance is generally better than CVM in this domain. We should mention that alternative choices of regions would have lead to different CVM results, but will probably not change this conclusion.

5.6.3 Medical Diagnosis

We now study the performance of TLSBP on a "real-world" example, the PROMEDAS medical diagnostic network. The diagnostic model in PROMEDAS is based on a bayesian network. The global architecture of this network is similar to QMR-DT (Shwe et al., 1991). It consists of a diagnosis layer that is connected to a layer with findings. In addition, there is a layer of variables, such as age and gender, that may affect the prior probabilities of the diagnoses. Since these variables are always clamped for each patient case, they merely change the prior disease probabilities and are irrelevant for our current considerations. Diagnoses (diseases) are modeled as a priori independent binary variables causing a set of symptoms (findings) which constitute the bottom layer. The PROMEDAS network currently consists of approximately 2000 diagnoses and 1000 findings.

The interaction between diagnoses and findings is modeled with a noisy-OR structure. The conditional probability of the finding given the parents is modeled by $n+1$ numbers, n of which represent the probabilities that the finding is caused by one of the diseases and one that the finding is not caused by any of the parents.

The noisy-OR conditional probability tables with n parents can be naively stored in a table of size 2^n. This is problematic for the PROMEDAS networks since findings that are affected by more than 30 diseases are not uncommon. We use efficient implementation of noisy-OR relations as proposed by Takinawa and D'Ambrosio (1999) to reduce the size of these tables. The trick is to introduce dummy variables s and to make use of the property

$$\mathrm{OR}(x|y_1,y_2,y_3) = \sum_s \mathrm{OR}(x|y_1,s)\mathrm{OR}(s|y_2,y_3).$$

The interaction potentials on the right hand side involve at most three variables instead of the initial four (left). Repeated application of this formula reduces all tables to three interactions maximally.

When a patient case is presented to PROMEDAS, a subset of the findings will be clamped and the rest will be unclamped. If our goal is to compute the marginal probabilities of the diagnostic variables only, the unclamped findings and the diagnoses that are not related to any of the clamped findings can be summed out of the network as a preprocessing step. The clamped findings cause an effective interaction between their parents. However, the noisy-OR structure is such that when the finding is clamped to a negative value, the effective interaction factorizes over its parents. Thus, findings can be clamped to negative values without additional computation cost (Jaakkola and Jordan, 1999).

The complexity of the problem now depends on the set of findings that is given as input. The more findings are clamped to a positive value, the larger the remaining network of disease variables and the more complex the inference task. Especially in cases where findings share more than one common possible diagnosis, and consequently loops occur, the model can become complex. We

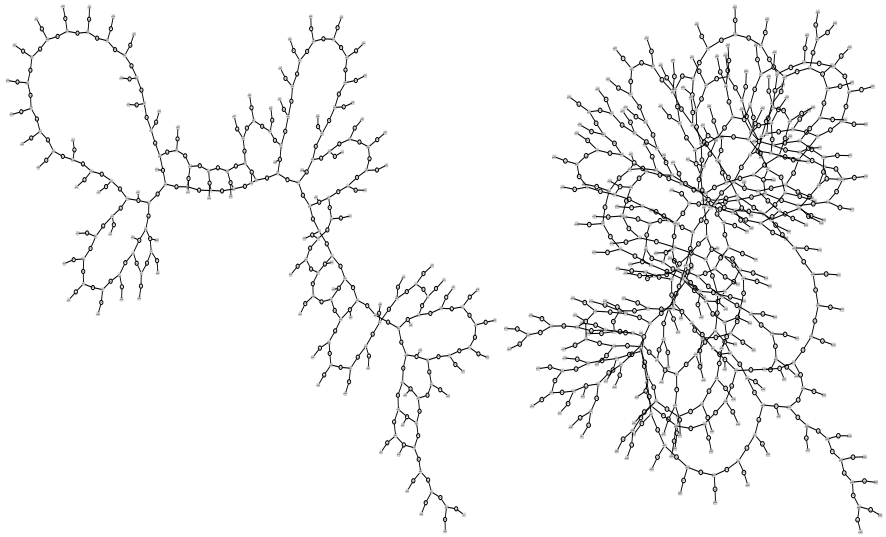

Figure 5.12: Examples of graph structures, corresponding to patient cases generated with one disease, after removal of unclamped findings and irrelevant disease variables and the introduction of dummy variables. Left and right graphs corresponds to an "easy" and a "difficult" case respectively.

illustrate some of the graphs that result after pruning of unclamped findings and irrelevant diseases and the introduction of dummy variables for some patient cases in Figure 5.12.

We use the PROMEDAS model to generate virtual patient data by first clamping one disease variable to a positive value and then clamping a finding to its positive value with probability equal to the conditional distribution of the findings given this positive disease. The union of all positive findings thus obtained constitute one patient case. For each patient case, the corresponding truncated graphical model is generated. Note that the number of disease nodes in this graph can be large and hence loops can be present.

In this subsection, as well as comparing errors of single-node marginals obtained using TLSBP against CVM, we also use another loop correction approach, loop corrected belief propagation (LCBP) (Mooij and Kappen, 2007), which is based on the cavity method and also improves over BP estimates. We use the following parameters for TLSBP: $S = 100$, $M = 5$, and no bound b. Again, we apply the junction tree method to obtain exact marginals and compare the different errors. Figure 5.13 shows results for 146 different random instances.

We first analyze the TLSBP results compared with BP (Figure 5.13a). The region in light gray color indicates TLSBP improvement over BP. The observed results vary strongly because of the wide diversity of the particular instances, but we can basically differentiate two scenarios. The first set of results include those instances where the BP error is corrected almost up to machine precision. These patient cases correspond to graphs where exhaustive enumeration is tractable, and TLSBP found almost all the generalized loops. These are the dots appearing in the bottom part of Figure 5.13a,

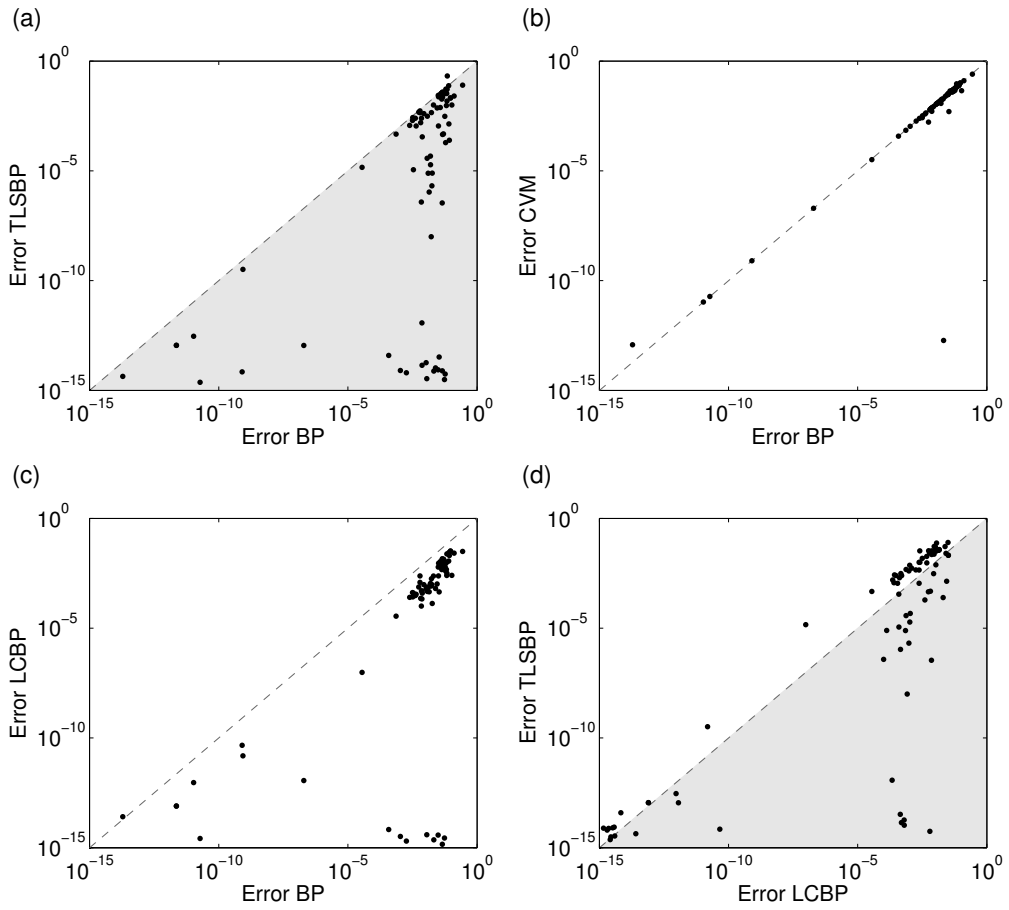

Figure 5.13: Results of 146 random patient cases with one disease. **(a)** TLSBP error versus BP error. **(b)** CVM error versus BP error. **(c)** LCBP error versus BP error. and **(d)** LCBP error versus TLSBP error.

approximately 14% of the patient cases. Note that even for errors of the order of 10^{-2} the error was completely corrected. Apart from these results, we observe another group of instances where the BP error was not completely corrected. These cases correspond to the upper dots of Figure 5.13a. The results in these patient cases vary from no significant improvements to improvements of four orders of magnitude.

Figure 5.13b shows the performance of CVM considering all maximal factors together with all loops that consist up to four different variables as outer regions. We observe that, contrary to TLSBP, CVM in this domain performs poorly. For only one instance the CVM result is significantly better than BP. Moreover, the computation time required by CVM was much larger than TLSBP in all instances (data not shown). These results can be complemented with the study developed by Mooij and Kappen (2007), where it is shown that CVM does not perform significantly better for other choices of regions.

Figure 5.13c shows results of LCBP, the approach presented in Mooij and Kappen (2007) on the

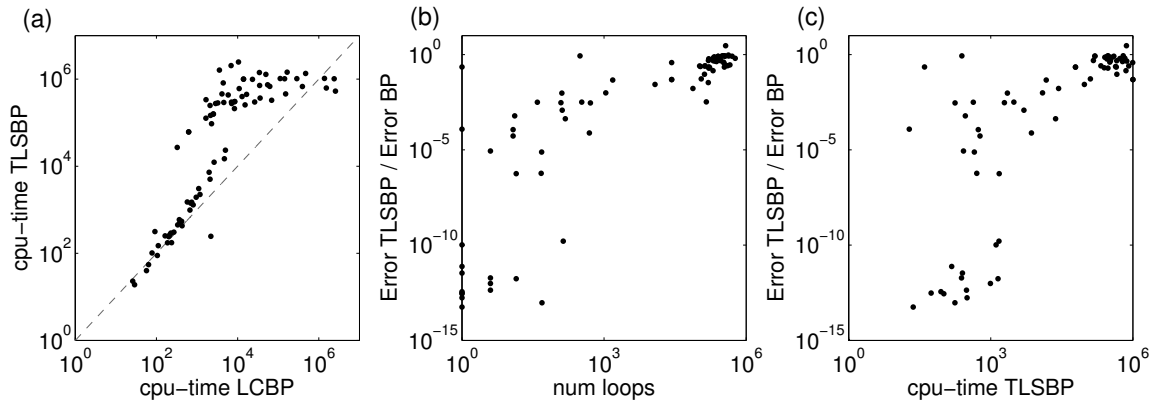

Figure 5.14: Results of applying TLSBP to 146 patient cases with one disease. **(a)** Relation between computational cost needed by LCBP and TLSBP. **(b)** Ratio between TLSBP and BP errors versus number of loops found. **(c)** Ratio between TLSBP and BP errors versus computation time.

same set of instances. As in the case of TLSBP, LCBP significantly improves over BP. A comparison between both approaches is illustrated in Figure 5.13d, where those instances where TLSBP is better are marked in light gray color. For 41% of the cases TLSBP improves the LCBP results, sometimes notably. TLSBP enhancements were made at the cost of more time, as Figure 5.14a suggest, where in 85% of the instances TLSBP needs more time.

To analyze the TLSBP results in more depth we plot the ratio between the error obtained by TLSBP and the BP error versus the number of generalized loops found and the CPU time. From Figure 5.14b we can deduce that cases where the BP error was most improved, often correspond to graphs with a small number of generalized loops found, whereas instances with highest number of loops considered have minor improvements. This is explained by the fact that some instances which contained a few loops were easy to solve and thus the BP error was significantly reduced. An example of one of those instances corresponds to the Figure 5.12 (left). On the contrary, there exist very loopy instances where computing some terms was not useful, even if a large number of them (more than one million) where considered. A typical instance of this type is shown in Figure 5.12 (right). The same argument is suggested by Figure 5.14c where CPU time is shown, which is often proportional to the number of loops found.

In general, we can conclude that although the BP error was corrected in most of the instances, there were some cases in which TLSBP did not give significant improvements. Considering all patient cases, the BP error was corrected in more than one order of magnitude for more than 30% of the cases.

5.7 Discussion

We have presented TLSBP, an algorithm to compute corrections to the BP solution based on the loop series expansion proposed by Chertkov and Chernyak (2006b). [4] In general, for cases where all loops can be enumerated the method computes the exact solution efficiently. In contrast, if exhaustive enumeration is not tractable, the BP error can be reduced significantly. The performance of the algorithm does not depend directly on the size of the problem, but on how loopy the original graph is, although for larger instances it is more likely that more loops are present.

We have also shown that the performance of TLSBP strongly depends on the degree of coupling between the variables. For weak couplings, errors of BP are most prominently caused by small simple loops and truncating the series is very useful, whereas for strongly coupled variables, loop terms of different sizes tend to be of the same order of magnitude, and truncating the series can be useless. For those difficult cases, BP convergence is also difficult, and magnetizations at the fixed point tend to be close to extreme values, causing numerical difficulties in the calculation of the loop expansion formulas. In general, we can conclude that the proposed approach is useful in an intermediate regime, where BP results are not very accurate, but BP is still converging.

We have confirmed empirically that there is a correlation between the BP result and the potential improvements using TLSBP. Those cases where the BP estimate is most corrected correspond often to cases where the BP estimate is already accurate. Whether a given BP error is acceptable or not depends on the inference task and the specific domain.

The proposed approach has been compared with CVM selecting loops of four variables and maximal factors as outer clusters. For highly regular domains with translation invariance such as square grids, CVM performs better than TLSBP in difficult instances (strong interactions). This is not surprising, since CVM exploits the symmetries on the original graph. However, for other domains such as random graphs, or medical diagnosis, TLSBP show comparable, or even better results than CVM with our choice of clusters.

The TLSBP algorithm searches the graph without considering information accessible from the BP solution, which is used to compute the loop corrections only as a final step. Therefore, it can be regarded as a blind search procedure. We have also experimented with a more "heuristic" algorithm where the search is guided in some principled way. Two modifications of the algorithm have been done in that direction:

1. One approach consisted in modifying the third step in a way that, instead of applying blind mergings, generalized loops which have larger contributions (largest $|r(C)|$) were merged preferentially. In practice, this approach tended to check all combinations of small loops which

[4] The source code of the algorithm and a subset of the data sets used in the experimental section can be downloaded from: `http://www.cns.upf.edu/vicent/`.

produced the same generalized loop, causing many redundant mergings. Moreover, the cost of maintaining sorted the "best" generalized loops caused a significant increase in the computational complexity. This approach did not produce more accurate results neither was a more efficient approach.

2. Also, instead of pruning the DFS search for complex loops using the parameter M, we have used the following strategy: we computed iteratively the *partial term* of the loop that is being searched, such that at each DFS step one new term using Equations (5.10) and (5.11) is multiplied with the current *partial term*. If at some point, the *partial term* was smaller than a certain threshold λ, the DFS was pruned. This new parameter λ was then used instead of M and result in an appropriate strategy for graphs with weak interactions. For cases where many terms of the same order existed, a small change of λ caused very different execution times, and often too deep searches. We concluded that using parameter M is a more suitable choice in general.

TLSBP can be easily extended in other ways. For instance, as an anytime algorithm. In this context, the partition sum or marginals can be computed incrementally as more generalized loops are being produced. This allows to stop the algorithm at any step and presumably, the more time used, the better the solution. The "improvement if allowed more time" can be a desirable property for applications in approximate reasoning (Zilberstein, 1996). Another way to extend the approach is to consider the search for loops as a *compilation* stage of the inference task, which can be done offline. Once all loops are retrieved and stored, the inference task would require much less computational cost to be performed.

During the development of this work another way of selecting generalized loops has been proposed (Chertkov and Chernyak, 2006a) in the context of Low Density Parity Check codes. Their approach tries to find only a few *critical* simple loops, related with dangerous noise configurations that lead to Linear Programming decoding failure, and use them to modify the standard BP equations. Their method shows promising results for the LDPC domain, and can be applied to any general graphical model as well, so it would be interesting to compare both approaches.

There exists another type of loop correction methods that improves BP, which is quite different from the approach discussed here (Montanari and Rizzo, 2005; Parisi and Slanina, 2006; Mooij et al., 2007; Mooij and Kappen, 2007). Their argument is based on the cavity method. BP assumes that in the absence of variable i, the distribution of its Markov blanket factorizes over the individual variables. In fact, this assumption is only approximately true, due to the loops in the graph. The first loop correction is obtained by considering the network with variable i removed and estimating the correlations in the Markov blanket. This argument can be applied recursively, yielding the higher order loop corrections. Whereas TLSBP computes exactly the corrections of a limited number of

loops, the cavity based approach computes approximately the corrections due to all loops. An in-depth comparison of the efficiency and accuracy of these approaches should be made.

As a final remark, we mention the relation of the loop series expansion with a similar method originated in statistical physics, namely, the high-temperature expansion for Ising models. This expansion of the partition function is similar to the one proposed by Chertkov and Chernyak (2006b), in the sense that every term has also a direct diagrammatic representation on the graph, although not in terms of generalized loops. Note however, that the loop expansion is relative to the BP result, contrary to the high-temperature expansion. Another difference is that the high temperature expansion is an expansion in a small parameter (the inverse temperature), whereas the loop expansion has no such small parameter. Finally, another related approach is the walk-sum framework for inference in certain Gaussian Markov Models (Malioutov et al., 2006), where means and covariances between any two nodes of the graph have an interpretation in terms of an expansion of walks in the graph. They also show that Gaussian loopy BP has a walk-sum interpretation, computing all walks for the means but only a subset of walks for the variances.

Part III

Online communities

Chapter 6

Preliminaries

In the previous parts of this thesis our study about networks has been focused on theoretical models of neural populations and on inference algorithms on probabilistic networks. In this part, our motivation is different. Given a set of empirical measurements originated from the activity of an online social network, we want to extract useful information from them to characterize the patterns governing the underlying network, and if possible, build useful algorithms which take profit from these results. For this purpose, the data we analyze here corresponds to a gathering of one year of human activity in a public bulletin board named Slashdot [1]. What follows is an introductory preamble, which settles the context for the last three chapters of this thesis.

The last decades have witnessed a significant change in human communication behaviour. Nowadays, the use of email, chats or discussion forums is essential for many people. Associated with this qualitative change is the generation of an invaluable experimental source of empirical data that can be used to investigate the patterns of human activity and communication at nearly no cost on a scale and dimension which would have been impossible some decades ago. Internet has growth democratically in a distributed way, and reached very large sizes with proportions unfeasible for exact analysis one decade ago. At present, it comprises millions of individual end nodes connected to tens of thousands of Internet service providers whose relationships are continually in flux and only partially observable.

The development of the theory of complex networks has clearly benefit from this expansion. Internet relies on a graph structure that can be studied at different levels. Not only at the router level, as a technological network which considers different devices as nodes and wires as links, but also as an information network, considering webpages and hyperlinks. More recently, hyperlink structures present in large online communities define social networks with high interest in the social sciences too.

One of the firsts attempts to study the topological aspects of Internet considered the distributions

[1] www.slashdot.org

of the in-links and out-links of the World Wide Web (Kleinberg et al., 1999; Broder et al., 2000), which turn to be governed by heavy tailed distributions. Similar distributions at the hardware and interdomain levels using traceroute-based sampling were also reported by Faloutsos et al. (1999) and also in file sizes and user event interarrivals by (Crovella and Bestavros, 1997). These types of probability distributions, in particular the power-law distribution, arise much interest because they are indicators of high heterogeneity and self-similarity at different scales. In addition, they deviate from the degree distributions that one would expect to obtain in classical random graphs. In a heavy tailed distribution, only a few extreme values have most part of the probability mass, while the vast majority of values have very little probability. A possible mechanisms to explain power-law distributed degrees in a network is the *preferential attachment* (or *rich-get-richer*) mechanism, with origins in the work of the statistician Yule (1925) and Simon (1955), incorporated subsequently in the celebrated Barabási model (BA) of network growth (Albert et al., 1999) for the World Wide Web.

Another characteristic of real networks which attracted considerable attention is the small-world effect, popularized by Milgram's experiments (Milgram, 1967), which dates back in the early 20th century with the six degrees of separation concept (Karinthy, 1929). Both ideas gave name to the fact that human society is a network characterized by very small path lengths. In addition, real networks also exhibit large degree of cohesiveness. The groundbreaking paper of Watts and Strogatz (1998) presented a first model which interpolates between regular lattice graphs to fully random networks and show a mechanism were both small average path lengths and large clustering coefficient coexist. This seemed to be also a universal property not only in social networks, but also in technological and information networks such as Internet (Adamic, 1999).

Prompted by those aforementioned studies, an enormous catalogue of network case-studies which are fully or partially related with Internet and human communication, have been developed during the last years. Some of these social analysis include the scientific collaboration network (Newman, 2001), email networks (Ebel et al., 2002; Newman et al., 2002; Drineas et al., 2004), the world trade web (Serrano and Boguñá, 2003), several academic webs (Thelwall and Wilkinson, 2003), a web-based tourism system (Baggio, 2007), a dating online community (Holme et al., 2004), the Spanish web (Baeza-Yates et al., 2005), the bulleting boarding system (BBS) community (Goh et al., 2006), and two online communities (Kumar et al., 2006). Typically, these studies address properties of the network with significant interest such as the size, degree distributions, average distance, clustering coefficient, connected components, the existence of a giant component in the community structure, degree correlations, reciprocity coefficients, etc. These measures will be described in the next chapter.

As more empirical data became available and more rigorous statistical techniques are used, some of the established results in the theory of complex networks have been questioned, or even invalidated. Huberman and Adamic (1999a,b) proposed a stochastic model to explain the power-law

distribution alternative BA model. They argued that the fluctuation effect arising in the process of connecting and disconnecting links between sites represents a key feature of the growth dynamics, and considered the growth of the number of pages per site as a multiplicative process, leading to a mixture of log-normally distributed variables, which result in a power-law distribution in the site's size. The original BA model has also been modified in many ways to accommodate the discrepancies found in empirical data, always at the cost of reducing its original simplicity. For instance Dorogovtsev et al. (2000) proposed to introduce a parameter called the initial attractiveness which governs the probability for young sites to get new links. The resulting power-law distribution had a tunable scaling exponent. Another generalization of the basic BA model, which incorporated deletion of individual links and gave rise also to a power-law distribution was made in (Fenner et al., 2006). Other modeling efforts have shifted from an original *degree driven* motivation towards another network properties, for instance, by considering weighted networks (Yook et al., 2001; Barrat et al., 2004), or by incorporating specific properties of the domain under study, such as reciprocity in social networks (Schnegg, 2006).

Even the widely assumed power-law behavior has been questioned several times. Lakhina et al. (2003) showed that the traceroute-based methodology to sample the connectivity between domains is biased because tends to correlate the observed degree of a vertex and the proximity of the source of the traceroute query. Therefore, the obtained degree distributions can differ sharply from that of the underlying graph. Furthermore, they show that power-law distributions can arise even in classical random graphs if a traceroute-based methodology is used. Chen et al. (2002) claimed that the power-law distribution is only valid for the tail values, leaving the most probability mass without explanation and emphasized other important properties not considered in the *degree driven* models. In their study, they propose the stretched exponential or Weibull distribution because of its flexibility (the tail behavior is power-law but the rest of the distribution can be arbitrary). The justification from the point of view of statistical physics is that the power-law is only asymptotically true, in the limit of an infinity large system, and that the observed deviations are caused by strong size effects and insufficient statistics. This is the explanation to justify why an exponentially truncated power-law (a power-law with an exponential cutoff) is often a better fit of the data. Studies such as the *Winners do not take all* model (Pennock et al., 2002), which combines a mixture of preferential and uniform attachment, are able to explain more diversity of empirical data. A real fact is that the use of rigorous statistical tests (Goldstein et al., 2004) is not sufficiently extended. Many studies still use a highly biased testing procedure based on a log-log linear fit of the data to conclude that the power-law hypothesis is correct (Siganos et al., 2003), when other distributions, such as the Weibull or the log-normal distribution may provide a better explanation (Lawrence, 1988). This discrepancy, new in the computer science community, has occurred many times in other areas of science, as noted

by Mitzenmacher (2006).

In addition to the network topology, another topic addressed in the last part of this thesis is concerned with the temporal patterns of communication occurring in a social network. In a pioneering work, Leland et al. (1994) determined that the Local Area Network traffic is self-similar, showing burstiness and long-range dependences across a wide range of scales. Similar results were shown in (Paxson and Floyd, 1994) for Wide Area traffic. Therefore, heavy-tailed distributions are ubiquitous not only in structural properties of complex networks, but also in the temporal domain as well, in contrast to the probability distributions found in traditional Poisson-like processes. Subsequent studies reported the same phenomena, giving more causes and analyzing its consequences, for instance, in the World Wide Web traffic (Crovella and Bestavros, 1997), inter-arrival times of job submissions (Kleban and Clearwater, 2003), internet chat systems (Dewes et al., 2003), email (Johansen, 2004), web accesses (Dezsö et al., 2006), or arrival times of requests to print (Harder and Paczuski, 2006).

The former studies propose an initial attempt to model such activity. For instance, using the immigration death process of $M/G/\infty$ queuing based model (Paxson and Floyd, 1994; Cox, 1984). Self-similar traffic can be also produced by multiplexing ON/OFF sources that have a fixed rate in the ON periods and ON/OFF period lengths that are heavy-tailed (Willinger et al., 1997), or using the approach approach taken by Kleinberg (2003) based on an infinite automaton. However, the most popular (and controversial) model is again due to Barabási (Barabási, 2005) subsequently extended by Vázquez et al. (2006). According to this simple model, humans maintain a queuing process in mind, and whenever an individual is presented with multiple tasks, the election among them is made based on some perceived priority parameter. The resulting waiting time of the various tasks is Pareto distributed with an exponent. Empirical email data has been satisfactorily explained by this queuing model (Vázquez et al., 2006). A key feature of this model is that the waiting time distribution is completely independent of the individual priorities assigned to the tasks. However, disregarding semantic content and social context has been considered objectionable (Kentsis, 2006). Interestingly, the question whether heavy-tailed probability distribution best explains the email temporal data used for the analysis arose an intense debate too (see the comments and replies by Stouffer et al., 2005; Barabási et al., 2005; Johansen, 2005; Stouffer et al., 2006, for details).

Currently, the latest form of online communication which is rapidly becoming mainstream is the blogging phenomenon (Kumar et al., 2004). Blogs (or weblogs) are essentially online diaries or personal newsletters that can be easily managed using online software packages. They are not only the best way for an ordinary person to reach a large audience, but they can also be alternative sources of news and public opinion, or environments for knowledge sharing (Herring et al., 2004). Several papers have appeared which analyze the social networks underlying this form of communication

using techniques similar to the previously described. The work in (Herring et al., 2005; Bachnik et al., 2005; Fu et al., 2006) represents initial steps in this direction, where a network of *bloggers* is constructed considering hyper-links appearing within the comments. The traffic characteristics have been also addressed (Kumar et al., 2003; Duarte et al., 2007).

A particular type of weblogs are message boards or web-based forums. They are online areas where discussions are held by many users on a variety of topics. Some users post articles and other users can comment on these posts, forming a discussion thread or nested dialogue. Examples of systems which follow this architecture are for instance the USENET, the bulletin based system (BBS), and more recently Slashdot, Digg, or Fark. Unlike personal weblogs which receive a few number of replies (Mishne and Glance, 2006), message board blogs can receive thousands of messages during a day and determine almost the entire content of the website. Although the first message boards, USENET and the bulletin board system (BBS), date back to 1979 only recently the social networks emerging from the comment interaction between their users and the temporal patterns taking place in these websites have been studied. A few articles appeared during the development of this research which analyze the BBS. Goh et al. (2006) builds a social network using the comments in a way very similar to us. Naruse and Kubo (2006) found log-normal distributions in frequency plots of number of users and comments they write, and propose a model. Other papers also studied the BBS (Matsumura et al., 2003, 2005a,b) from different points of view.

In addition to this form of networks, message boards can show rich complexity in the structure of their discussion threads. Previous studies of USENET have been focused mainly on visualization techniques to facilitate understanding of the social and semantic structure (Sack, 2000). The Netscan system (Smith, 2002), a powerful interface to track discussion threads and authors, has proven to be a valuable tool to understand different roles appearing in these newsgroups (Fisher et al., 2006; Brush et al., 2005). It is therefore of interest to analyze the statistics governing the structure of threads in order to understand the underlying patterns of communication existing in these large online spaces, and to develop efficient techniques which improve the system performance.

The last part of this thesis is focused on this context. We analyze an example of public online web-based community: Slashdot. The next section basically covers the fundamental aspects of complex network theory, reviews two well known heavy-tailed probability distributions and the archetypical models of network growth. These topics are extensively used when analyzing empirical data originated from a complex network and have not been discussed earlier in this thesis. In chapter 7, we analyze statistically both structural and temporal aspects of the social network of Slashdot.

6.1 Fundamentals of the theory of complex networks

This section briefly reviews basic aspects of complex networks. We first describe some properties widely used in the literature and then some models of networks. Most material contained here can be found in the reviews of Newman (2003b); Albert and Barabási (2002); Dorogovtsev and Mendes (2002).

6.1.1 Structural network properties

A complex network can be defined as a network that has certain non-trivial topological features not present in simple (or random) networks. Since most of the interesting features of real-world networks concern the way in which they differ from classical random graphs, throughout this subsection a comparison with the random network model is provided. Random networks were first studied by Rapoport (1957); Solomonoff and Rapoport (1951), although Erdös and Rény (1959, 1960) gave it the name "random graphs" and developed the theory exhaustively. A random network is defined as the ensemble of all graphs $G = \langle V, E \rangle$ where $N = |V|$ is the number of nodes and $M = |E|$ is the number of edges, in which each edge is chosen randomly with probability p. Equivalently, it is the ensemble of all graphs in which a graph having M edges appears with probability $p^M(1-p)^{M-N}$.

Average path length

The average path length ℓ of a graph is a global measure defined as the mean geodesic (shortest) distance between every pair of nodes. For definition of distances, we consider all lengths of all edges to be one. Given d_{ij} the geodesic distance between vertices i and j:

$$\ell = \frac{2}{N(N-1)} \sum_{i,j \in V} d_{ij}. \tag{6.1}$$

The standard way to compute ℓ is using a *breath-first* traversal for each node, with computational complexity $O(NM)$ in time and $O(M+N)$ in memory. For disconnected graphs, we only include in 6.1 all connected pairs of nodes.

For random graphs, ℓ can be calculated analytically. Let $\langle k \rangle$ denote the mean node degree, which is $p(N-1)$. The average number of nodes at distance d is thus $\langle k \rangle^d$. Since we are interested in the case of N vertices, we have $\langle k \rangle^\ell = N$. Taking logarithms on both sides, we obtain the average path length for a random graph:

$$\ell_{\text{rand}} \sim \frac{\log N}{\log \langle k \rangle}. \tag{6.2}$$

The *small-world effect* describes the situation where only a small number of steps is required to reach a target node starting from an arbitrary initial node. Formally, when the value of ℓ scales

logarithmically (or slower) with N for a fixed $\langle k \rangle$. As Equation (6.2) indicates, random graphs already show this property. [2]

Clustering coefficient

The clustering coefficient C measures the connectivity density between the direct neighbours of a given nodes, and is defined as the probability that two nodes linked to a common node will also be linked to each other. According to the definition of Watts and Strogatz (1998), C is defined as the average over a local measure, the individual clustering coefficient C_i:

$$C_i = \frac{2|E_i|}{k_i(k_i-1)}, \tag{6.3}$$

where E_i is the set of existing edges between direct neighbours of i, and k_i the degree of node i. Therefore, C_i is the ratio of the actual number of edges between neighbours of i and the potential total number of edges between neighbours of i.

There exists another definition of C, which relies in the concept of graph transitivity and has been more used in social sciences. It is the fraction of transitive triples, and measures how likely is for a neighbour of one of the neighbours of node i to be also a neighbour of node i. It is calculated taking the ratio between all the triangles in the graph and all connected triples of vertices (a single vertex with edges running to an unordered pair of nodes):

$$C = \frac{3 \times \#\text{ of triangles}}{\#\text{ of connected triples}}. \tag{6.4}$$

Expression (6.3) is easier to compute, but one has to keep in mind that both measures can differ, since (6.3) is a mean of ratios, and (6.4) is a ratio of means. For random graphs, both expressions scale inversely with N. It is easy to see that $C_{\text{rand}} = \langle k \rangle / N = p$. Typically, real networks show higher values of C.

Degree distribution

The degree distribution is the property that has received most attention in network studies. It is defined as the probability $P(k_i)$ that a vertex i chosen uniformly at random has k neighbours.

For random graphs, since each node is connected with $(N-1)p$ nodes on average, the degree k_i of a node i follows a binomial distribution. Indeed, it is the number of ways in which k edges can be drawn from a certain node. In the limit of large N, it can be replaced with a Poisson distribution:

$$P(k_i = k) = \binom{N-1}{k} p^k (1-p)^{N-1-k} \approx \frac{\langle k \rangle^k e^{-\langle k \rangle}}{k!}. \tag{6.5}$$

[2] Some authors add the condition of high clustering coefficient (defined next), so there is ambiguity in the use of this term (Watts and Strogatz, 1998; Barrat and Weigt, 2000).

Hence the reason why random graphs are also named Poisson graphs. For both distributions appearing in (6.5) there exists a typical value around which degrees are centered (the mean). Real-world networks, however, are mostly found to be very unlike random graphs in their degree distributions, which are highly right-skewed, meaning that they have a long right tail of values that are fall above the mean. Two examples are the power-law or the log-normal distribution, which are described in detail in the next section.

For the moment, it is worth to say that one can obtain power-law distributed degrees from random graphs by *forcing* its degree distribution by construction and keeping other parameters randomized. The procedure looks as follows: first, choose a degree sequence, a set of values k_i for each node $i = 1 \dots N$. Next, connect iteratively pairs of nodes at random according to the predefined sequence. In this way, arbitrary degree distributions can be generated, including the power-law or other heavy-tailed distributions. Actually, this process generates every possible graph with a given degree distribution with equal probability. The ensemble composed of graphs generated using this procedure is named the *configuration model*. In this case, the degree distribution is "imposed" from the beginning, it does not emerge from the growth process.

Mixing patterns and degree correlations

Nodes in real-world networks often have several attributes which can be used to define different classes, such as race or sex in a human network. This fact has been addressed by sociologists to study different patterns of connectivity between the existing classes. For instance, whether nodes of one type show preference to be linked with nodes of the same type. When this occurs, the network is called assortative with respect to the analyzed attribute. Dissassortative mixing denotes the opposite pattern of connectivity.

The degree of assortativity can be measured from the connectivity matrix $\mathbf{E}$ between the elements of different classes, where E_{ij} is the number of edges between nodes of type i and j (not to confuse with the set of edges). The normalized matrix $\mathbf{e}$ is defined by dividing each element by the sum of all elements:

$$\mathbf{e} = \frac{\mathbf{E}}{||\mathbf{E}||}. \tag{6.6}$$

The assortativity coefficient r is then defined according to the following expression (Newman, 2003a):

$$r = \frac{Tr\mathbf{e} - ||\mathbf{e}^2||}{1 - ||\mathbf{e}^2||}, \tag{6.7}$$

and takes values between 1 for a perfect assortative graph when $\mathbf{e}$ is a diagonal matrix, and a negative number in the range $[-1,0)$, given by $1 - ||\mathbf{e}^2||$, for a dissassortative network.

Another type of mixing coefficient often used is the degree correlation, which uses the vertex degree as attribute. In a network with assortative (dissortative) mixing by degree, a vertex with large degree tends to connect to vertices with large (small) degree, while for the network of the neutral mixing, there is no such tendency. Assortative mixing can be found in social networks such as the coauthorship network, the actor network and many others, and the dissortative network in information networks such as the Internet and the World-wide Web, and in biological networks such as protein interaction networks and neural networks.

Newman (2002) proposed the Pearson correlation coefficient to measure this quantity. One has to compute the joint distribution of degrees at either ends of an edges. Let M denote the number of all edges of the graph, the degree correlation is calculated as follows:

$$r = \frac{M^{-1}\sum_{e\in E} i_e j_e - \left(M^{-1}\sum_{e\in E}\frac{1}{2}(i_e + j_e)\right)^2}{M^{-1}\sum_{e\in E}\frac{1}{2}\left(i_e^2 j_e^2\right) - \left(M^{-1}\sum_{e\in E}\frac{1}{2}(i_e + j_e)\right)^2}, \tag{6.8}$$

where i_e, j_e are the degrees of both vertices where edge e ends. For values of $r \approx -1$, degrees are negative correlated (dissassortative mixing), and high degree nodes tend to attach to low degree ones, whereas the opposite (assortative mixing) happens for $r \approx 1$.

Reciprocity

Reciprocity denotes the property of links to be bidirectional. In social networks, for instance, friendship relations are typically reciprocal. Clearly, this property only exists for directed graphs. Moreover, it can be an indicator of the amount of information ignored when the graph is made undirected.

A simple ratio between the number of links pointing in both directions and the total number of links $r = L^{\leftrightarrow}/L$ is a biased measure, since for highly dense networks r is larger and therefore, there is no reference value. To overcome this problem the following correlation coefficient has been proposed (Garlaschelli and Loffredo, 2004). Given A the adjacency matrix, such that $a_{ij} = 1$ iif there is a *directed* link from i to j and zero otherwise, the reciprocity coefficient ρ is defined as:

$$\rho \equiv \frac{\sum_{i\neq j}(a_{ij} - \bar{a})(a_{ji} - \bar{a})}{\sum_{i\neq j}(a_{ij} - \bar{a})^2}, \tag{6.9}$$

where $\bar{a}$ is an average to denote the link density $\bar{a} = \sum_{i\neq j} a_{ij}/(N(N-1))$, which is the ratio between the actual and potential number of directed links.

For very big graphs, direct computation of (6.9) is hard, since requires to traverse all the adjacency matrix of the network. An equivalent expression which only requires to traverse the *existing*

edges can be found also in (Garlaschelli and Loffredo, 2004):

$$\rho \equiv \frac{L^{\leftrightarrow}/L - \hat{a}}{1-\bar{a}} = \frac{r-\hat{a}}{1-\bar{a}}. \tag{6.10}$$

Community structure

Some networks exhibit a natural decomposition in clusters or subsets of densely connected nodes. For instance, in social networks people tend to join groups with similar interests. Although a concise definition of community is not straightforward, it is common to characterize this concept in topological terms only, and denote the community structure as the division of network nodes into groups within which the network connections are dense, but between which they are sparser.

The traditional method for detecting community structure in networks is hierarchical clustering (Everitt, 1993). It can only be applied on weighted networks, so we assume that a weight w_{ij} exists for every pair i, j of vertices. When the network is unweighted, some measures can be used which represent "how closely connected" the vertices are (Wasserman and Faust, 1994). The typical hierarchical clustering is the *agglomerative* clustering. The procedure starts with all N vertices in the network, with no edges between them, and adds edges between pairs one by one in order of their weights, starting with the pair with the strongest weight and progressing to the weakest. As edges are added, the resulting graph shows a nested set of increasingly large components (connected subsets of vertices), which are taken to be the communities. The tree that represents this iterative procedure is called dendrogram. One of the drawbacks of the agglomerative procedure is that core nodes in a community have strong similarity, and hence are connected early in the agglomerative process, but peripheral nodes that have no strong similarity to others tend to get neglected. Alternatively, one can start with all the connected graph and remove edges iteratively until all vertices are disconnected. This is called *divisive* clustering.

A popular algorithm based on divisive clustering was developed in (Girvan and Newman, 2002) and applied to several synthetic and real-world networks. The weights are calculated according to the *edge betweenness*, which is the number of minimum paths connecting pairs of nodes that go through an edge. [3] The algorithm identifies successfully communities in graphs for which their structure is known in advance, and extracts clusters which appear to correspond to plausible communities in other cases. In those unknown cases, a possible measure to select the optimal level of the dendrogram is the *modularity*, proposed in Newman (2004); Newman and Girvan (2004), which is based on the assortative mixing coefficient, Equation (6.7). In principle, the most likely community structure is the one which maximizes this modularity coefficient.

[3] The measure is inspired in the betweenness centrality of a vertex (Freeman, 1977), defined as the number of shortest paths between pairs of other vertices which run through it.

In the recent years, the number of related papers which have developed and improved community related algorithms has increased significantly. An interesting approach which formulates the community detection task as an inference problem for which the belief propagation algorithm can be applied is presented in (Hastings, 2006).

6.1.2 Mathematical models of networks

We now review briefly two canonical models of graphs which use analytic arguments to reproduce some features of real-world networks not found in random graphs.

Watts-Strogatz *small-world* model

The main motivation underlying the Watts-Strogatz (WS) model (Watts and Strogatz, 1998) is to explain why in many real-world networks both high clustering coefficient C and small average path length ℓ coexist.

The model starts with a regular lattice of any dimension such that every node is connected to its first K neighbors. For simplicity, the model is defined for the one-dimensional case (a ring). The model is based on a *rewiring* procedure, where a fraction p of the existing edges is selected, and the end of that edge is replaced from the current node to another node chosen uniformly at random from the lattice, preventing self-connections or duplicated edges.

The completely regular lattice ($p = 0$) has a very large C but a very large ℓ as well, and the completely random graph ($p = 1$) has a very small ℓ but a small C too. In particular, for a ring lattice, $\ell(p = 0) \simeq N/2K \gg 1$ so ℓ scales linearly with N and C is large. For $p \to 1$, we already have seen that $\ell(p = 1) \sim \log N/\log K$ and $C(p = 1) \sim K/N$, that is, ℓ scales logarithmically with N, and C decreases with N.

By changing the probability p one can interpolate between the regular lattice and the full random graph. Interestingly, there exists a range of small p where a small ℓ and a high C coexist. This range where ℓ drops fast and C is still high has a very intuitive explanation: some rewired long-range random edges decrease act as **short-cuts** that connect distant vertices which would have remained disconnected without them. Consequently, the distance between the immediate neighbours, and neighbours of neighbours, etc, of these short-cuts is dramatically decreased, while the cohesiveness is still retained.

Scale-free model

Although the WS model provides a possible explanation of why both small ℓ and high C coexist, it cannot reproduce the emergence of the heavy-tailed degree distributions often seen in real-world

networks. This is the main motivation of Barabási model (BA). The BA model is based on the *preferential attachment* mechanism, which incorporates the time dimension. After many steps the degree distribution of the network converges to a power-law. Similar mechanisms to generate power-law distributions were discovered before (Yule, 1925; Simon, 1955). Price (1976) applied this mechanism in the context of networks and gave it the name of *cumulative advantage processes*.

The BA model (Barabási and Albert, 1999) is defined for an undirected graph which starts with a small quantity m_0 of vertices, and at every step, one new vertex is added to the network. The new node is attached to m other vertices in such a way that a very connected node will likely gain more connections in the future. Let $P(k)$ be the fraction of vertices with degree k in the network. The probability that at any step of the growth process one of the new m edges attaches any existing vertex with degree k is $\frac{kP(k)}{\sum_k kP(k)}$. The denominator ensures that the sum of probabilities for each degree k is one. Note that $\sum_k kP(k)$ is also the mean degree of the network, which is equal to $2m$, since every new node adds m new edges, and each degree is counted twice because the graph is undirected.

The rate-equation approach (Krapivski et al., 2001) is based on an equation for the growth of $N_k(t)$, the average number of nodes with degree k at time t. Between two steps, the number of nodes with degree k increases because of the nodes with degree $k-1$ that receive one edge. Their proportion is $\frac{(k-1)N_{k-1}(t)}{\sum_k kN_k(t)}$. Conversely, the nodes with degree k that also are attached to a new edge turn into nodes with degree $k+1$, so they must be subtracted. Their proportion is $\frac{kN_k(t)}{\sum_k kN_k(t)}$. Since the process of adding a new edge is repeated m times in one step, the resulting rate equation is:

$$\frac{dN_k}{dt} = m\frac{(k-1)N_{k-1}(t) - kN_k(t)}{\sum_k kN_k(t)} + \delta_{k,m}. \tag{6.11}$$

For the special case of $k = m$, note that $N_{k-1}(t)$ is zero since there are no vertices with degree $k < m$, so we only have to subtract those nodes with $k = m$ that become $k = m+1$, and sum the new added node, which corresponds to the term $\delta_{k,m}$.

After many iterations the networks reaches a stationary state where $N_k(t) = tP(k)$ (note that t is the number of nodes added to the network since the beginning) and $\sum_k kN_k(t)$ is just $2mt$, the average degree multiplied by the number of nodes added. After these two substitutions, we have the following recursive equation:

$$P(k) = \frac{(k-1)P(k-1) - kP(k)}{2} + \delta_{k,m}. \tag{6.12}$$

If we separate the two existing cases we have:

$$P(k) = \begin{cases} \frac{1}{2}(k-1)P(k-1) - \frac{1}{2}kP(k) & \text{for } k > m \\ 1 - \frac{1}{2}mP(m) & \text{for } k = m \end{cases} \tag{6.13}$$

with solution:

$$P(k) = \frac{2m(m+1)}{k(k+1)(k+2)} \sim 2m(m+1)k^{-3}, \tag{6.14}$$

a power-law with exponent $\alpha = -3$.

Contrary to the previous model of Price (1976) in which m is not fixed and fluctuates at each iteration (although with mean m), the original BA model only predicts a fixed exponent. Several generalizations of the BA model allow different values of the exponent (Dorogovtsev et al., 2000) or different forms of the distribution for the non power-law region before the tail (Pennock et al., 2002). A concise table of them can be found of in Albert and Barabási (2002).

6.2 Heavy-tailed distributions

We now define and discuss some properties of the power-law and the log-normal distribution. Both probability distributions will appear frequently in our analysis of Slashdot in the next chapter.

6.2.1 The power-law distribution

A **continuous** variable x obeys a power-law if it has the following probability density function (pdf):

$$p(x)dx = Pr(x \leq X < x+dx) = Cx^{-\alpha}dx, \tag{6.15}$$

where X is the observed value and C is a normalization constant. The density diverges as $x \to 0$, so a lower bound $x_{min} > 0$ is used. The constant can be obtained given that the integral must sum one:

$$1 = \int_{x_{min}}^{\infty} p(x)dx = C\int_{x_{min}}^{\infty} x^{-\alpha}dx = \frac{C}{1-\alpha}\left[x^{-\alpha+1}\right]_{x_{min}}^{\infty}. \tag{6.16}$$

For being well defined, $\alpha > 1$ and then $C = (\alpha-1)x_{min}^{\alpha-1}$, and the correct normalized expression is:

$$p(x) = \frac{\alpha-1}{x_{min}}\left(\frac{x}{x_{min}}\right)^{-\alpha}. \tag{6.17}$$

In the **discrete** case, x follows a power-law distribution if it takes integer values according to:

$$P(x) = \frac{x^{-\alpha}}{\zeta(\alpha, x_{min})}, \tag{6.18}$$

where the normalizing constant is in this case the generalized or Hurwitz zeta function $\zeta(\alpha, x_{min}) = \sum_{n=0}^{\infty}(n+x_{min})^{-\alpha}$.

If one takes logarithms on both sides of the pdf:

$$\log p(x) = \alpha \log x, \tag{6.19}$$

it looks like a straight line in both logarithmic scales. This is the scale-free property, which means that the shape of the distribution is the same independently on the scale of measure.

Usually, it is convenient to use the cumulative distribution (cdf) instead of the pdf, because the fluctuations present in the pdf due to finite size sampling are removed. In the continuous case, the cdf is defined as:

$$p(x \geq X) = \int_X^{\infty} p(x')dx' = \frac{1}{\alpha - 1} x^{-(\alpha-1)}. \tag{6.20}$$

Note that formally this is more related to the *complementary* cumulative distribution function $p(x > X)$. However, for simplicity, many authors denote it cdf. As Equation (6.20) indicates, the cdf also follows a power-law, but with a different exponent $\alpha - 1$.

Parameter estimation

Given a dataset containing n observations $X_i \geq x_{min}$, we want to know the value of α for the power-law model that is most likely to have generated the observations.

The fact that a power-law follows a straight line in both logarithmic scales prompted many authors to estimate the parameters of a power-law using a simple histogram. This procedure is based on plotting the histogram on double logarithmic axes and perform **least squares fit**. The slope of the obtained line is then used as an estimator of the exponent α. It is better to fit the cdf, since the cdf is also a straight-line in double logarithmic axes and, moreover, the statistical fluctuations due to finite sample size are alleviated. Although this method is still used at present for many authors, the resulting exponent can be highly biased, as reported in (Newman, 2005; Goldstein et al., 2004).

The best known estimator for the power-law distribution is the **Maximum Likelihood Estimator** (MLE), first derived by Muniruzzaman (1957). Generally, MLE maximizes the likelihood function, or equivalently minimizes the negative of the logarithm of the likelihood function, which in the case of a **continuous** power-law distributed variable is (assuming independence between the data points):

$$\begin{aligned} \mathcal{L}_{\text{cont}} &= -\log p(x|\alpha) \\ &= -\log \prod_{i=1}^{n} \frac{\alpha - 1}{x_{min}} \left(\frac{x_i}{x_{min}} \right)^{-\alpha} \\ &= \sum_{i=1}^{n} \left(\log(\alpha - 1) - \log x_{min} - \alpha \log \frac{x_i}{x_{min}} \right) \\ &= n \log(\alpha - 1) - n \log x_{min} - \alpha \sum_{i=1}^{n} \log \frac{x_i}{x_{min}}. \end{aligned} \tag{6.21}$$

Setting the derivative of $\mathcal{L}$ equal to zero and solving for α, the MLE estimator $\hat{\alpha}$ is:

$$\hat{\alpha} = 1 + n \left(\sum_{i=1}^{n} \log \frac{x_i}{x_{min}} \right)^{-1}. \tag{6.22}$$

For a **discrete** power-law distributed variable the MLE is found minimizing:

$$\mathcal{L}_{\text{discr}} = -\log P(x|\alpha) = -\log \prod_{i=1}^{n} \frac{x_i^{-\alpha}}{\zeta(\alpha, x_{min})} = -n \log \zeta(\alpha, x_{min}) - \alpha \sum_{i=1}^{n} \log x_i. \tag{6.23}$$

Setting (6.23) to zero:

$$-\frac{n}{\zeta(\alpha, x_{min})} \frac{d}{d\alpha} \zeta(\alpha, x_{min}) - \sum_{i=1}^{n} \log x_i = 0. \tag{6.24}$$

Therefore, the MLE $\hat{\alpha}$ is the solution of:

$$\frac{\zeta'(\hat{\alpha}, x_{min})}{\zeta(\hat{\alpha}, x_{min})} = -\frac{1}{n} \sum_{i=1}^{n} \log x_i. \tag{6.25}$$

Detailed proofs concerning convergence properties of the MLE estimator can be found in (Muniruzzaman, 1957).

The previous MLE obtains an estimator of the exponent $\hat{\alpha}$ assuming x_{min} is known. However, x_{min} has to be estimated too. Again, many authors set x_{min} by graphical assessment. Two methods exist which work well in practice:

The first method is known as *Bayesian information criterion*. For discrete data,

$$\log P(x|x_{min}) \approx \mathcal{L} - \frac{(x_{min}+1)\log n}{2}. \tag{6.26}$$

The second method, based on intuition and tested empirically, can be applied both for discrete and continuous distributions (Clauset et al., 2007). According to this method, the optimum value $\hat{x}_{min}$ is found by searching sequentially a sufficiently large set of possible values of x and computing the distance d between the cdf of the data $S(x)$ and the cdf of the fit using MLE $P(x)$:

$$d = \max_{x \geq x_{min}} \{|S(x) - P(x)|\}. \tag{6.27}$$

The chosen value for $\hat{x}_{min}$ is the one which minimizes (6.27). Although there is no rigorous proof for the method, it seems to give good results. Intuitively, for $\hat{x}_{min} < x_{min}$, $P(x)$ will differ from $S(x)$ because the empirical distribution does not follow a power-law in the lower end, whereas for $\hat{x}_{min} > x_{min}$, $P(x)$ will differ from $S(x)$ because a significant number of samples is discarded and therefore, statistical fluctuations are more severe.

6.2.2 The log-normal distribution

A variable x follows a log-normal distribution when its logarithm follows a normal distribution: $y = \log(x) \sim \mathcal{N}(\mu, \sigma^2)$. The pdf of a normal distributed variable y is:

$$p(y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(y-\mu)^2/2\sigma^2}, \qquad \text{for } y \in (-\infty, \infty). \tag{6.28}$$

The pdf for x is therefore:

$$p(x) = p(y)\frac{dy}{dx} = p(y)\frac{d}{dx}\log x = \frac{1}{x}\frac{1}{\sqrt{2\pi}\sigma}e^{-(\log x-\mu)^2/2\sigma^2}, \qquad \text{for } x \in (0,\infty). \tag{6.29}$$

Note the additional $1/x$ outside the exponential term. A log-normally distributed variable x is necessarily positive, so the pdf for $x \in (-\infty, 0]$ is zero. The corresponding cdf is:

$$p(x \leq X) = \int_0^X \frac{1}{x}\frac{1}{\sqrt{2\pi}\sigma}e^{-(\log x-\mu)^2/2\sigma^2}dx = \Phi\left(\frac{\log x-\mu}{\sigma}\right), \tag{6.30}$$

where $\Phi(\cdot)$ is the cdf of the standard normal distribution. A three parameter is sometimes used. The third parameter θ corresponds to a threshold or shift so that $x > \theta$. The pdf is then:

$$p(x;\mu,\sigma,\theta) = \frac{1}{x-\theta}\frac{1}{\sqrt{2\pi}\sigma}e^{-(\log(x-\theta)-\mu)^2/2\sigma^2}, \qquad \text{for } x \in (\theta,\infty). \tag{6.31}$$

For simplicity we use here $\theta = 0$. It is easy to show hat a log-normal distribution is very similar to a power-law distribution. Indeed, a log-log plot of the pdf it can be a straight line is the variance σ^2 is large enough. Taking the logarithm of the pdf:

$$\log p(x) = -\log x - \log\sqrt{2\pi}\sigma - \frac{(\log x-\mu)^2}{2\sigma^2} \tag{6.32}$$

$$= -\frac{\log^2 x}{2\sigma^2} + \left(\frac{\mu}{\sigma^2} - 1\right)\log x - \log\sqrt{2\pi}\sigma - \frac{\mu^2}{2\sigma^2}. \tag{6.33}$$

In the limit of very large σ, the first quadratic term will be small for a large range of x values. Although the quadratic terms implies a curve that in a log-log scale, if we only consider a small portion of the curve it will look like a straight line.

Moments and other characteristics

The rth moment of x is:

$$\mu'_r = E(x^r) = e^{r\mu+\frac{r^2\sigma^2}{2}}. \tag{6.34}$$

The first two moments, median and mode are shown in Table 6.1:

Mean	$e^{\mu+\frac{\sigma^2}{2}}$
Variance	$e^{2\mu+\sigma^2}\left(e^{\sigma^2}-1\right)$
Median	e^{μ}
Mode	$e^{\mu-\sigma^2}$

Table 6.1: Some characteristics of the log-normal distribution.

Mechanisms to generate log-normal distributions

Log-normal distributions are found in multiplicative processes. An example of multiplicative process is the following (Kapteyn, 1903): We start with a positive initial variable x_0. At each step, the th step variable is:

$$x_t - x_{t-1} = \varepsilon_j \phi(x_{t-1}), \tag{6.35}$$

where $\{\varepsilon_j\}$ is a set of mutually independent and identically distributed variables and also statistically independent of $\{x_t\}$ and ϕ is a certain function. For the case of $\phi(x) = x$, the process $\{x_t\}$ is said to obey the lay of proportionate effect and Equation (6.35) becomes:

$$x_t = x_{t-1}(1+\varepsilon_t). \tag{6.36}$$

After many steps, x_0 will be multiplied with a large number of random numbers. At step n, the logarithm of $\{x_n\}$ will be:

$$\log x_n = \log\left(x_0 \prod_{t=1}^{n}(1+\varepsilon_t)\right) = \log x_0 + \sum_{t=1}^{n} \log \varepsilon_t, \tag{6.37}$$

if ε_t is small compared with 1. By the Central Limit Theorem, the sum appearing in the right hand side of (6.37) will asymptotically converge to a normal distribution and therefore, $\{x_n\}$ will be log-normally distributed.

For other historical examples about generating log-normal distributions see (Lawrence, 1988).

6.2.3 Testing hypothesis: Kolmogorov and Smirnov Goodness of Fit test

A rigorous statistical test is required when we want to show whether a given random sample comes from a particular probability distribution or not. This can be formulated by means of the following two hypothesis:

- $H_0 : F(x) = F^*(x)$, for $-\infty < x < \infty$
- H_1 : The hypothesis H_0 is not true,

The null hypothesis H_0 indicates that the cdf from which the sample was drawn $F(x)$ is the same as the testing cdf $F^*(x)$. In practice, since we are only given a set of n observations, $F(x)$ is estimated calculating the empirical cdf distribution $F_n(x)$, which is the proportion of observed values in the sample that are less than or equal to x, which converges to the exact $F(x)$ in the limit of a large sample size.

For continuous data, the Kolmogorov-Smirnov statistic is based on the maximum difference between the sample cdf $F_n(x)$ and the hypothesized cdf $F^*(x)$ and is defined as:

$$D_n^* = \sup_{-\infty<x<\infty} |F_n(x) - F^*(x)|, \tag{6.38}$$

The important result established from Kolmogorov and Smirnov is that the probability distribution of the KS statistic converges in the limit of $n \to \infty$ to a form which does not depend on $F^*(x)$ itself (see DeGroot and Schervish, 2002, for more details). From this distribution, one can calculate the probability of how likely a random sample deviates as much (or more) from $F^*(x)$ as our sample. This probability is known as the p-value. Therefore, the bigger the p-value, the more likely is H_0 to be true. The KS test rejects the null hypothesis H_0 at level of significance α when the p-value is less than α. The -pvalue can be computed using the following formula (Press et al., 1992):

$$\text{pvalue} = 2\sum_{i=1}^{\infty} (-1)^{i-1} \exp\left(-2i^2 \left(\left[\sqrt{n} + 0.12 + 0.11/\sqrt{n}\right] D_n^*\right)^2\right). \tag{6.39}$$

Usually, the values of α are taken to be 0.05 or 0.01. The smaller the value of α, the more conservative is the test.

Note that the KS test may be performed when the full $F^*(x)$ is known *in advance*. When some parameters of $F^*(x)$ are obtained from the sample data, for instance using MLE, the target distribution is just an estimation $\hat{F}^*(x)$ of $F^*(x)$, and therefore the distribution of the KS statistic is biased (Goldstein et al., 2004; Press et al., 1992). The p-value should then be computed using Monte Carlo methods, i.e sampling many times from $\hat{F}^*(x)$, and computing the fraction of experiments in which the KS statistic $\hat{D}_n^*$ is bigger than the initially obtained D_n^*.

Chapter 7

Statistical analysis of Slashdot

A user who enters in Slashdot is immediately exposed to a large body of news and an even larger amount of comments written by other users. The mechanism available to this single user to interact with this public space is essentially sending comments to previous contributions posted in the message board. These comments are then published and become accessible to other users, which in turn can answer to the original comment. We therefore can recognize as in the first part of this thesis an interplay between the collective activity of a network of thousands of elements as a whole population, and on the individual behavior of each element in the network.

Through this chapter we want to shade light into the nature of this type of communication. We are interested in recognizing the factors that induce users to write comments in this interactive space, and in understanding what are the social underlying processes which characterize the collective behavior occurring in Slashdot. With this goal in mind, we may pursue to answer the following questions:

- On a structural dimension, we want to address which aspects characterize the social network of Slashdot. Is the Slashdot network similar to other well-known social networks such as the coauthorship network or the actors network? Which peculiarities can be found in Slashdot that differ from other real-world networks? Also, what is the structure of the discussion threads provoked by a post? Can we design simple methods to measure the degree of controversy of a post?

- On a temporal dimension, we are interested in how activity is generated after a post is published. Is this activity characterized by heterogeneous temporal patterns where each discussion differs substantially from the others, or are there underlying homogeneities which allow us to provide a concise explanation of the dynamic process occurring the publication of a post?

The answers to these questions will help us to understand the user behavior in Slashdot. In particular, which causes mainly induce a user to write a comment on the site. Is the interest of the

topic? Is the relation between other users? Is just the hour of the day?

Our main hypothesis relies on the principle that the act of commenting is not guided by a complicated process which needs a considerable number of variables to be understood. In contrast, we believe that users are driven by a simple mechanism. They look at the online space as a board to express their opinion and the use of complicated mechanisms is not required to explain the essential phenomena taking place in Slashdot. The fundamental assumption implicit in all this study is that neglecting almost all semantic information can reveal the inherent structure of this community. We thus consider only the minimal amount of information embodied in a comment, such as the time-stamp, the source and the destination of a message. Using basically these quantities, we find remarkable regularities and give empirical evidences that the main underlying statistics of this community can be actually explained using simple principles.

First, we introduce Slashdot and describe our process used to gather and analyze the data. We are faced with a huge dataset which contains the activity of thousands of users during one year.

In Section 7.3 we analyze the social network extracting relationships between users from their commenting activity. Here we apply standard methods used in complex networks theory and social systems and discuss similarities and discrepancies with existing social networks.

A particular representation of the posts in a form of radial tree is considered in Section 7.4. Threads show strong heterogeneity and self-similarity throughout the different nesting levels of a conversation. We use these results to propose a simple measure to evaluate the degree of controversy provoked by a post.

The dynamics of Slashdot is analyzed in Section 7.5. Here we will see that activity is highly correlated with the circadian rhythm. Surprisingly, the temporal patterns associated to the response time of users to a new post submission are homogeneous and can be described with high accuracy using a mixture of two log-normal distributions.

We conclude this chapter with a discussion of our study and possible lines of future research.

7.1 Slashdot: A web-based bulletin board

Slashdot was created at the end of 1997 and has ever since metamorphosed into a website that hosts a large interactive community capable of influencing public perceptions and awareness on the topics addressed. It gave name to the "Slashdot effect" (Adler, 1999), a huge influx of traffic to a hosted link during a short period of time, causing it to slow down or even to temporarily collapse. The site's interaction consists of short-story **posts** that often carry fresh news and links to sources of information with more details. These posts incite many readers to **comment** on them and provoke discussions that may trail for days. The discussions take the form of threads similar to those that can

be found in private blogs but with a much more density and richness of participation. We refer to such discussion as threads, and therefore comments take place at some nesting level of the discussion thread. Although Slashdot holds much closer ties to web message boards and newsgroups than to weblogs, we can find increasingly interest on studies about the comments to posts on weblogs (Mishne and Glance, 2006; Duarte et al., 2007).

Commentators are encouraged to register and comment under their nicknames, although some of them participate anonymously. The mechanisms that make Slashdot work are open. All its content and computer code is free and available on the web [1] and detailed explanations about its social norms are provided. The moderation and meta-moderation mechanisms are employed to judge comments and enable readers to filter them by quality. Comments are labeled with a score that can be used to filter the visualization. The score values range from minus one (used to denote very low quality comments, possibly spam) up to five, (comments of mandatory reading). Lampe and Resnick (2004) concluded that this moderation system upholds the quality of discussions by discouraging spam and offensive comments, marking a difference between Slashdot and regular discussion forums. This high quality social interaction has prompted several socio-analytical studies about Slashdot. Poor (2005) and Baoill (2000) have both conducted independent inquiries on the extent to which the site represents an online public sphere as defined by Habermas (1991).

7.2 Data acquisition

The architecture of Slashdot is a web server located in the USA where all articles and comments are stored. This server receives HTTP requests from web-browsers around the world. To retrieve all the required information from the site (this process is known as *crawling*) we use software that generates requests and stores the contents of the HTML code received from the server. The software used for the crawling was *wget*, *perl* scripts, and *tidy* on a GNU/Linux, Ubuntu 6.0.6 OS. The data correspond to posts and comments published between August 26th, 2005 and August 31th, 2006.

We divided the crawling process into two stages. The first stage included crawling the main HTML where posts and first level comments were included, and the second stage covered all additional comment pages. Crawling all the data took 4.5 days and produced approximately 4.54 GB of data. Post-processing caused by the presence of duplicated comments was necessary (due to an error of representation on the website). Although a high amount of information was extracted from the raw HTML (sub-domains, title, topics, hierarchical relations between comments) we use in this study a minimal amount of information only:

Type : whether the contribution is a post or a comment.

[1] Under the project **Slash** at http://sourceforge.net/projects/slashcode/

Identifier : All posts and comments have an identifier.

Author : The author is a username or a nick. Posts can be written by a restricted set of fifteen authors, and the rest users are commentators. The majority of post authors also contribute in the discussions writing comments.

Time : The time-stamp of the contribution. They can be obtained from Slashdot with minute precision and corresponded to the EDT time zone (=GMT-4 hours).

Score : A number between minus one and five.

Subdomain : Each post belongs to a section or subdomain. There are currently fourteen subdomains, although three more existed during the crawling process, which were posteriorly erased. These *obsolete* subdomains are: *Apache*, *BSD* and *Features*.

The selected information was parsed and extracted to XML-files and imported into Matlab where the statistical analysis was performed. Table 7.1 shows a summary of the main quantities of the extracted data.

Period covered	26-8-05 $-$ 31-8-06
Time needed for crawling	4.5 days
Amount of data mined	4.54 GB
Posts	10,016
Comments	2,075,085
Commentators	93,636
Anonymous comments	18.6%

Table 7.1: Main quantities of crawling and retrieved data.

7.3 The Social Network

After explaining the details of the data set, we start to build and analyze the social network of Slashdot considering the comments that users write to the site. We first explain the procedure used to create three different versions of the network. We then describe the values obtained of different indicators with special emphasis on the degree distribution. Finally, we describe briefly the community structure.

7.3.1 Building the Network

In a social network, each person corresponds to a node $i \in V$ in a graph $G = \langle V, E \rangle$ and edges $(i, j) \in E$ indicate social relations between two individuals. In our case, we interpret that links exists

filter	total cmnts	%	step cmnts	%
Post	473,065	22.8	473,065	22.8
Anonymous	385,901	18.6	295,396	14.2
Low score	45,785	2.2	9,691	0.4
Self-replies	56,489	2.7	15,045	0.7

Table 7.2: Comments discarded after proper filtering.

between users because of their comment activity. These social relations may designate not only one, but several forms of interaction, such as common interests on the particular topic under discussion, friendship, disagreement, or just sporadic interaction. Further, it can also be the case that a message is not related at all with the source contribution to which is replying. To improve the quality and the representativity of the resulting graph, a filter process was applied which discards comments according to the following four criteria:

1. The **post**: Under this assumption, no relations exist between the post's author and its direct commentators, unless he also participates later in the discussion.

2. **Anonymous** comments were also discarded.

3. We discard very low quality comments with **score** -1.

4. Finally, we filter out **self-replies**, often motivated by a forgotten aspect or error fix of the original comment.

The second and third column of Table 7.2 show the total number and the percentage of comments which fall in each category. Columns 4 and 5 give number and percentage of comments discarded due to the above explained filter steps. Note that after elimination of direct replies to posts and anonymous comments, low-score comments only represent a small fraction.

After the filtering process, the remaining number of comments is $1,281,888$, approximately 63% of the total. The users are reduced to $80,962$, approximately 87% of the initial set of users.

In many social network studies, it is common to neglect the directed nature of the interactions or to assume only binary edges, i.e. two individual are connected or not. The reason is because sometimes these parameters are not easy to characterize. Although this simplifies the statistical analysis, it can also be the case that some relevant information is discarded. In our case, the directed nature of the comments allow to express the underlying graph as a directed graph. Moreover, since two users can exchange comments several times, a weighted network arises naturally. With the aim of being as much systematic as possible, we propose to compare three different types of networks according to the following criteria. Let n_{ij} be the number of times that user i writes a comment to user j. The three types of proposed networks are:

Undirected dense : An *undirected* edge exists between users i and j if either $n_{ij} > 0$ **or** $n_{ji} > 0$. The weight of that edge w_{ij} is simply the sum $n_{ij} + n_{ji}$.

Undirected sparse : An *undirected* edge exists between users i and j if $n_{ij} > 0$ **and** $n_{ji} > 0$. The weight of an edge w_{ij} in this case is defined as $w_{ij} = \min\{n_{ij}, n_{ji}\}$.

Directed : A *directed* edge exists from user i to user j if $n_{ij} > 0$ regardless of n_{ji}. The weight of that edge w_{ij} is simply n_{ij}.

The undirected dense and undirected sparse express different ways to interpret the existence of a relationship. The dense would correspond as a *weak* interpretation, since only the interchange of one message is required to establish a link between two users. On the contrary, the sparse defines a *strong* interpretation, since a reciprocal interaction is required between two users in order to be linked. Both undirected networks, however, ignore who the source and the destination of the messages are. This feature is captured in the directed case.

Figure 7.1 shows a small example to illustrate the generation of the three different graphs. On the left, there is a tree structure corresponding to a small thread of depth 4. Labels denote the user who writes the contribution and valid comments are shown within the gray region. The post triggers four responses from users A, B, C and D. At the second nesting level, five comments appear (two from the same user E, one from user A (who already commented on the original post), and two more from users F and G. At the third level, there are only two comments from users A and C, and finally, there is one last comment from G.

The small graphs on the right correspond to the three graph versions. In Figure 7.1b, users are linked if they exchange at least one message. In Figure 7.1c, bidirectional edges exists between users when both users replied a comment of the other. These reciprocal relations are the links in the

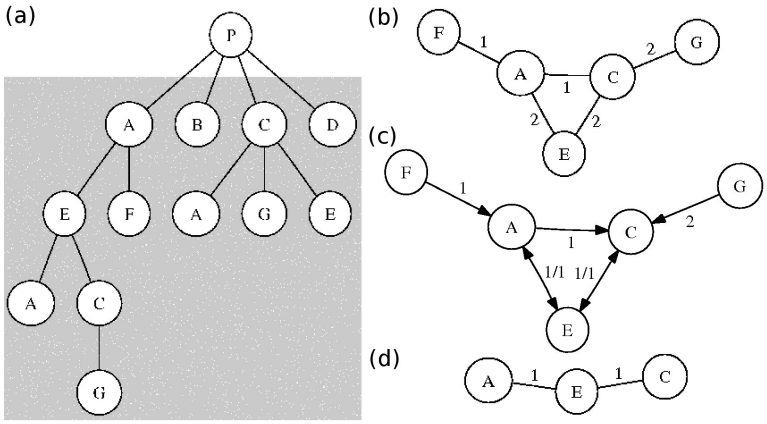

Figure 7.1: Example of graph generation. **(a)** A small thread of comments. **(b)** Undirected dense network. **(c)** Directed network. **(d)** Undir. sparse network.

Indicator	Directed	Und.Dense	Und.Sparse
N	80,962	80,962	37,087
M	1,052,395	905,003	294,784
Max.clust.	73.12%	97.90%	97.15%
$\langle k \rangle$	13(50.1/49.4)	22.36(79.3)	7.95(25.7)
ℓ	3.62(0.7)	3.48(0.7)	4.02(0.8)
ℓ_{rand}	4.38	3.62	5.05
D	10	9	11
C	0.027(0.075)	0.046(0.12)	0.017(0.078)
C^w	0.026(0.074)	0.047(0.12)	0.018(0.080)
C_{rand}	$1.67 \cdot 10^{-4}$	$2.88 \cdot 10^{-4}$	$2.27 \cdot 10^{-4}$
r	-0.016	-0.039	-0.016
ρ	0.28	–	–

Table 7.3: Indicators of the Slashdot social networks.

undirected sparse graph. Note that we do not consider possible relations not associated to the thread structure like mentioning a user within the text of a comment. Semantic analysis would be required to overcome this limitation.

7.3.2 General description

We now characterize the structural properties of the obtained graphs (Newman, 2003b). Table 7.3 shows the values of the indicators considered here for the different networks. If not stated otherwise, indicators are calculated for the unweighted graph.

In the first two rows we show the number of nodes $N = |V|$ and edges $M = |E|$ of the respective networks. In the case of the undirected sparse graph, N is reduced significantly. The number of actual links M is very small compared to the potential number of relationships $O(N^2)$. This would suggest a highly sparse network with many connected components composed of small groups of users. However, as row 3 of Table 7.3 indicates, a vast majority of the population forms a "giant component", leaving only a small proportion of users disconnected from that component. These isolated users are grouped mainly in pairs, or at most, in small clusters of size 4 in all three networks. In both undirected graphs, the "giant component" contains more than 97% of the users and in the directed[2] network almost 75%. These quantities indicate that the social network of Slashdot is characterized by a compact community and a small proportion of isolated users, in concordance with typical social networks.

The average degree $\langle k \rangle$ is shown in row 4 of Table 7.3 (standard deviations between parenthesis).

[2]We consider *weakly* connected components in the directed case, i.e. two vertices i and j belong to the same component if there exists a path between i and j at least in one of the two possible directions. The size of the big cluster for *strongly* connected components is of course, smaller.

The directed network presents an intermediate value between the dense and sparse undirected representations. All cases show high standard deviations, indicating a big level of heterogeneity within the community. This aspect is analyzed in more detail in Section 7.3.3.

The average path length ℓ, measured only for the giant component, takes small values for all three networks, suggesting that the small-world property is present in all of them. The quantities are approximately one unit lower than the corresponding value for a random graph ℓ_{rand}. The maximal distance D between two users is also very small. Even for the undirected sparse case, it only takes a maximum of eleven steps to reach a user starting randomly from any other. These results are also in accordance with similar studies of other traditional social networks.

To study the statistical level of cohesiveness we calculate the clustering coefficient C according to (Watts and Strogatz, 1998), and also its weighted version C^w (Barrat et al., 2004). We notice no significant differences between them. Thus the number of messages interchanged between two users is not relevant to determine the clustering level. The impact of having a weighted network is analyzed in more detail in Section 7.3.5. We can see that for all graphs, C and C^w are about two orders of magnitude higher than their randomized counterpart C_{rand}. This is again in harmony with other analysis of real-world networks, which report similar deviations from the random graph, and enhances the evidence of the small-world property. As before, the directed graph represents an intermediate value between both undirected versions.

Another quantity of special interest in social networks is the degree correlation, or mixing coefficient, which allows to detect whether highly connected users are preferentially linked to other highly connected ones or not. This fact is known as assortative mixing by degree and is present in many social networks (Newman and Park, 2003). Table 7.3 shows the correlation coefficient r for our three networks, which is far from ± 1. Therefore, unlike traditional social networks which present a strong assortative mixing, Slashdot is characterized by neither assortative nor dissortative mixing. Users do not show any preference to write comments in function of the connectivity of the other users. Interestingly, other related studies of online communities show similar (Holme et al., 2004; Goh et al., 2006) coefficients. This seems to be a fundamental difference to social interactions occurring outside these large online spaces.

The last general property we analyze is the reciprocity. High reciprocity is another feature typically present in social networks. In our case, reciprocity occurs when a user i replies the answer of another user j to a previous comment of i, and can be measured by means of a reciprocity coefficient ρ. Using the method proposed in (Garlaschelli and Loffredo, 2004), we quantify how the Slashdot network differs from a random network in the presence of two mutual links (edges in both directions) between pairs of nodes. The small positive value $\rho = 0.28$ suggests that our network is only moderately reciprocal, so that users tend to write slightly more often than expected by chance to other users

who previously wrote them.

From this global characterization we can conclude that the underlying network of Slashdot presents common features of traditional social networks, namely, a giant cluster of connected users, small average path length and high clustering. In contrast to other social networks, Slashdot shows moderate reciprocity and neutral assortativity by degree. We also see that there is significant difference between considering dense and sparse undirected versions, and that the directed version represents an intermediate description between the two. The moderate value of the reciprocity coefficient ρ suggests that studying only the undirected network, one could miss some relevant structural information. Finally, regarding clustering, we see no significant differences between the weighted and the binary network. Despite the strong similarity to a much smaller network of BBS-users (Goh et al., 2006), these two features seem to be exclusive of Slashdot.

7.3.3 Degree Distributions

We now focus on the function describing the number of users in the network with a given number of neighbors. The analysis of this degree distribution describes the level of interaction between users and provides a robust indicator about the grade of heterogeneity in the network.

Figure 7.2 shows in small circles the probability distribution (pdf) and the cumulative distribution (cdf) of the degrees for the directed network. The other two networks present equivalent results. First, we can see that in and out degree distributions are almost identical. Unlike previous studies (Goh et al., 2006), in our case the activities of writing and being replied could be characterized by similar processes.

As expected, the obtained distributions are heavy-tailed, covering in this case more than three orders of magnitude and indicating a high level of heterogeneity between the users. Surprisingly, the users located at the tail of the distribution are not Slashdot authors of posts who also participate actively in the discussions, as one would expect. The first Slashdot author in the list sorted by degree appears at position number 378 (in the case of the in-degree distribution of the directed network). Therefore, the hubs of the social network are not the "affiliated" authors, but regular users who participate actively in the discussions.

To find a functional form which best explains the observed data, we compare two approximations: the "usually assumed" power-law (PL) hypothesis, and a truncated log-normal (LN) hypothesis (Mitzenmacher, 2003). Their corresponding density functions are given by

$$f_{LN}(x;\mu,\sigma,\theta) = \frac{1}{(x-\theta)\sigma\sqrt{2\pi}} \exp\left(\frac{-(\ln(x-\theta)-\mu)^2}{2\sigma^2}\right)$$

$$\text{and} \quad f_{PL}(x;\alpha,x_{min}) = \frac{x^{-\alpha}}{\zeta(\alpha,x_{min})},$$

where $\zeta(\alpha,x) = \sum_{n=0}^{\infty}(n+x)^{-\alpha}$ is the generalized or Hurwitz zeta function. We select the optimal parameter values using maximum likelihood estimation (MLE). The PL distribution has as parameters the scaling exponent α and the minimum degree value x_{min} from which the PL behavior occurs. To find the proper value of x_{min} we apply a recent method proposed in (Clauset et al., 2007). The LN distribution has three parameters: the mean μ, the standard deviation σ and a shift θ, which represents a lower bound of the degree values. Both distributions can be very similar (Mitzenmacher, 2003).

Figure 7.2 shows that for the case of the PL hypothesis (dashed-bold line), the obtained values of x_{min} (represented by the border between gray and white areas) are extremely large, leaving almost all the data samples *outside* the fitted region, which contains only a few users at the tail of the distribution. However, the LN fit provides an explanation of the *entire* dataset, of both, the left-support of the distribution, where most of the probability mass is concentrated, and also in the tail of the distribution, where fluctuations are bigger due to finite sampling effects.

Table 7.4 summarizes the results of the fits of the degree distributions. The first two columns of the upper part of Table 7.4 show the parameters of the PL distribution. The minimum value of x_{min} is 5, corresponding to the undirected sparse version. Even in this case, more than 75% of the users are

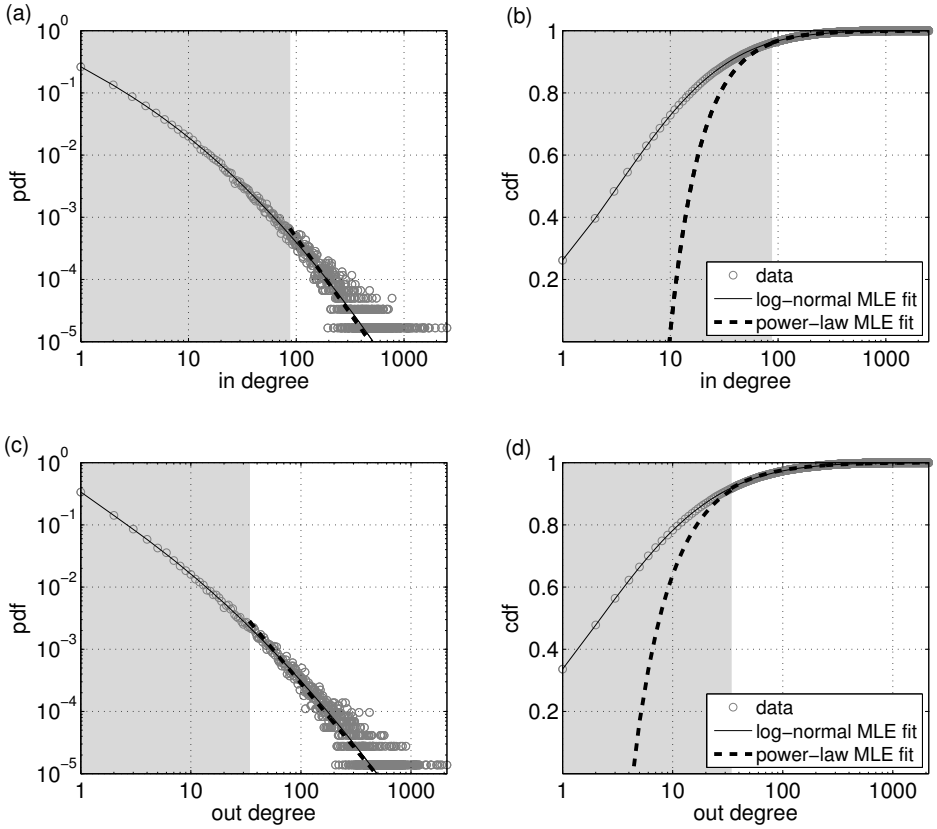

Figure 7.2: In and out degree distributions of the directed Slashdot network and corresponding PL and truncated LN approximations.

power-law				
Network type	x_{min}	α	% disc.	p-value
undir. dense	85	2.27	94.49	0
dir. in-degree	87	2.44	96.05	0
dir. out-degree	34	2.13	91.68	0
undir. sparse	5	1.92	75.85	0
truncated log-normal				
Network type	μ	σ	θ	p-value
undir. dense	1.03	2.04	0.32	**0.43**
dir. in-degree	1.14	1.87	0.47	**0.93**
dir. out-degree	0.45	2.07	0.39	**0.58**
undir. sparse	-0.73	2.14	0.19	**0.99**

Table 7.4: PL and LN fit of degree distribution.

not included in the PL fit. The proportion of discarded samples is indicated in the third column. The lower part of Table 7.4 gives the parameters of the LN approximation, which show more variability than those of the PL.

After selecting the optimal values of the parameters for both hypothesis, we test whether the provided model of the data can be accepted or not. We use the Kolmogorov-Smirnov test (KS), whose p-values are shown in the last column of Table 7.4. In all cases, the PL hypothesis provides a p-value much lower than 0.1 (our choice of the significance level of the KS-test). Hence, we can conclude that, even after discarding most of the data, the PL is not able to explain the tail of the distributions. In contrast, the obtained p-values for the truncated LN model are quite high, all of them bigger than 0.1, so the LN-hypothesis allows to explain the entire distribution.

7.3.4 Mixing patterns

In Section 7.3.2 we have seen that the Slashdot network presents neutral assortativity by degree, therefore neither frequent commentators comment preferentially to sporadic ones, nor vice-versa. It is interesting to see whether Slashdot users show assortative mixing by other attributes. In this subsection, we analyze two possible types of correlations exclusive of the Slashdot community.

First, we associate to each user a *score*, which is calculated by averaging over all the scores of the comments of the same user. In this case we address the question whether users with higher scores tend to comment to other users with higher scores or not.

In addition, we also study if users follow mixing patterns related to the *subdomain* (or sections) to which they write more often. Here we attach an attribute which characterizes the subdomain most addressed by a user.

Mixing by score

The initial score of a comment is generally 1 if it comes from a registered user or 0 if it is anonymous[3]. Moderation can modify the initial score to any integer within the range $[-1,5]$. To ensure a representative subset of the network, we only consider users who wrote at least 10 comments, a total of $18,476$ users, representing approximately 23%.

In Figure 7.3a we plot the histogram and the corresponding cdf of the distribution of the mean scores. Note that the minimum score is 0, since we eliminate -1 comments. The distribution shows an unexpected bimodal profile, with two peaks at mean scores 1.1 and 2.3. This indicates that two different classes of users coexist.

Is the mean score a representative measure of the user's commenting quality? To check its validity we plot in Figure 7.3b the distribution of the standard deviations of the scores. More than $3/4$ of all users show deviations smaller than 1, so the scores a user obtains do not fluctuate significantly. Therefore, their mean seems to be a good candidate to characterize the user.

We now analyze whether users of one class reply preferentially to users of the same class or not. The bimodality suggests a simple characterization using two classes of users. We select the boundary between classes to balance their sizes (the resulting boundary is 1.90). Thus a given user is assigned to class c_1 if its mean score is ≤ 1.90, and otherwise to class c_2. Class c_1 contains $9,254$ users whereas c_2 contains $9,222$. The assortativity matrix $\mathbf{E}$ is calculated counting the number of comments interchanged between classes. Each element E_{ij} indicates the number of comments that

[3] Anonymous users cannot be considered in this analysis.

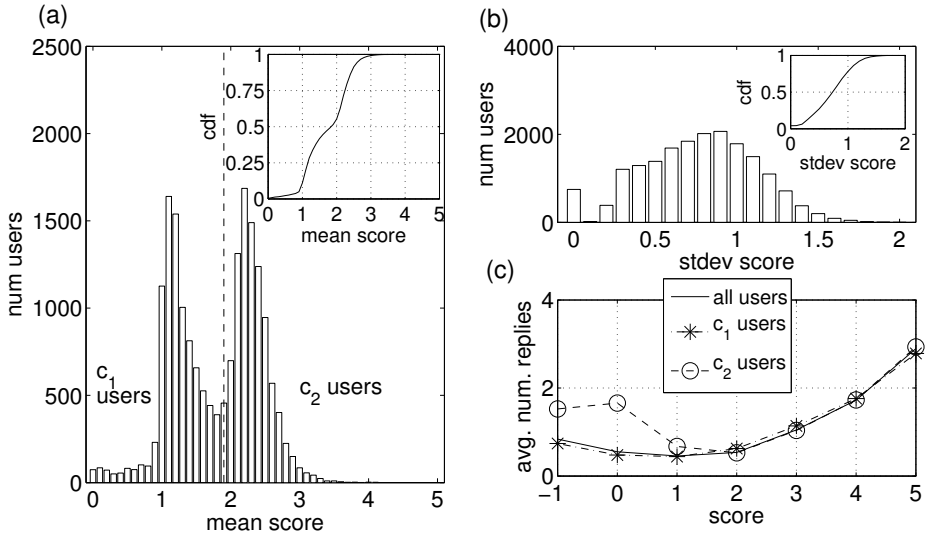

Figure 7.3: **(a)** Mean scores of users who wrote at least 10 comments. **(b)** Standard deviations of the mean scores of the same users. **(c)** Relation between the score of a comment and the average number of received replies for all, c_1 and c_2 users.

users of class i wrote to users of class j. Its normalized version $\mathbf{E}'$ is obtained dividing $\mathbf{E}$ by the total number of comments:

$$\mathbf{E} = \begin{pmatrix} 78{,}341 & 198{,}391 \\ 151{,}013 & 455{,}997 \end{pmatrix} \qquad \mathbf{E}' = \begin{pmatrix} 0.09 & 0.22 \\ 0.17 & 0.52 \end{pmatrix}.$$

The assortativity coefficient is $r_{\text{score}} = 0.036$, so the network is neutrally mixed by mean score. Note, however, that more than half of the comments are written from users of class c_2 to other c_2 users, and that the proportion of comments received by c_2 users is 0.74, so there is a strong bias in favour of good writers.

This can either mean that users tend to reply preferentially to users of the class with higher average score, or simply that high-scored comments tend to receive more reactions than low-scored ones independently of the user. To check this, we compare in Figure 7.3c the average number of replies received by comments in function of their scores for either all users or only the users of classes c_1 and c_2. To get a broader range of scores, we also include negative scores.

It is quite clear that scores ≥ 2 correlate with the average number of reactions and are independent of the user's class, but comments with scores below 2 do not show this correlation and achieve substantially more replies on average if written by users of class c_2.

We can thus conclude that on average, although higher scored comments tend to achieve more replies regardless of the user who wrote it, it is also true that good writers, even when they post low-scored comments, still receive significant more replies than c_1 users.

Mixing by subdomain

We also analyze whether subdomains, or *Slashdot sections*, mostly addressed by the users can be a possible magnitude which helps us to discover possible mixing patterns between the Slashdot users. [4] We extracted all subdomains from the posts and associated them to each comment. As before, we compute the probability distribution of subdomains for each user taking into account the subdomains associated to each comment of the user. In this case, since we deal with discrete distributions with categorical values, the means are not defined. We select the *mode* of the distributions, which is the subdomain most addressed by the user, to characterize its subdomain profile.

To see whether the mode of this distribution is a good measure to characterize users we need a similar quantity as the standard deviation in the previous case of the scores. We define the quantity d as the difference between the probability of the mode s_1 and the probability of the next most frequent

[4] For simplicity, we discard the topic information and use the subdomain field only. We found a total of 164 topics in our crawled data.

subdomain s_2 to which the user has written:

$$d = P(s_1) - P(s_2). \tag{7.1}$$

In this case, high values of d indicate very specialized users in one subdomain, since the number of comments between its most frequent subdomain and its second one differs strongly. On the contrary, small values of this quantity denote users who do not show any preference in writing at least in two subdomains. Equivalently, if the proportion of users with high d is big, we can trust in this measure to characterize the profile of the users.

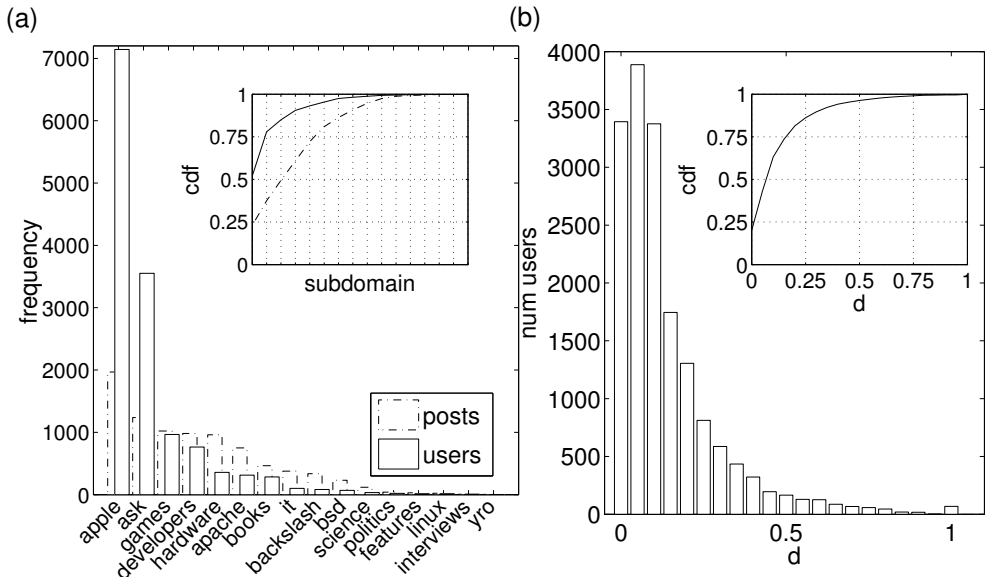

Figure 7.4: **(a)** Histogram of the number of users and posts which fall in each possible subdomain. **(b)** Histogram of the d-values.

Figure 7.4a shows the distribution of subdomains between the users and the posts. [5] Concerning the users, the distribution is highly peaked. The inset indicates that approximately half of the users are enthusiasts on section *Apple*. Note that *Apache*, *BSD* and *Features* where removed at some time between the period covered by our crawling. Unlike the mean score where two classes where clearly differentiated, the peak in this distribution suggests that concerning subdomains, there is not such a natural decomposition in balanced classes.

The distribution of the posts in subdomains (shown in slash-dotted bars in the same Figure 7.4a) is less peaked than the one of the comments. However, both distributions give rise to the same ordering of subdomains in number of contributions. Indeed, the correlation coefficient between both distributions is quite high, $r = 0.8441$. Therefore, the distribution profile of the subdomains for the users does not depend on the user comment activity, and is clearly explained by the post activity only.

[5] See http://slashdot.org/faq/editorial.shtml#ed1000 for a description of all subdomains.

Moreover, as Figure 7.4b indicates, only a few number of users, approximately 15%, show a significant difference ($d > 0.25$) between the mode probability and the next most frequent subdomain probability, so in general, users are not very specialized in an specific subdomain only.

From this results we can conclude that, unlike the mean score, the mode of the subdomains distribution, is not an amenable attribute to analyze the mixing coefficient. Three reasons justify this conclusion. First, the obtained classes are not balanced (most users belong only to two classes). Second, classes are determined mainly by the post activity. Finally, users show a very small d, therefore, the most frequent subdomain addressed by a user does not represent it properly.

7.3.5 Community Structure

To end this characterization of the Slashdot network we analyze its community structure. We take a simple approach based on agglomerative clustering which takes benefit from the weighted nature of the Slashdot network (Newman, 2003b). We choose the dense undirected network and start our procedure with each node as an independent cluster. Let λ denote the number of comments, so that pairs of users (i, j) who interchange a number of comments $w_{ij} \geq \lambda$ are included in the network, and the other connections are discarded. Starting from the biggest value $\lambda = \lambda_{\max}$ and progressively decreasing it, users are connected incrementally and communities can be obtained. This simple procedure is equivalent to building a dendrogram and allows to browse through the community structure at different scales by changing the parameter λ.

Figure 7.5a shows the distribution of the weights w_{ij} of all links in the network. The vast majority of pairs of users only exchanges a small number of comments whereas a few of them really maintain intense dialogues during the year. This seems to be the reason why previous properties such as the clustering coefficient do not show significant differences between the weighted and the unweighted network. The most discussing pair of users exchanged a total of 108 comments. This represents our initial value $\lambda_{\max}$ to start the agglomerative procedure.

Since most of users exchange only a small amount of comments, one would expect that the number of clusters remains quite high for a wide range of λ values. This is indeed the case. As λ is progressively decreased, users are being grouped in small clusters. Simultaneously, a giant cluster is being formed which absorbs the small clusters when they reach a moderate size. In Figure 7.5b we plot the number of clusters in function of λ. It is reduced dramatically in the last step when $\lambda = 1$ is reached. A more detailed analysis can be obtained if we discard isolated users, and focus only on groups of pairs or more users, who at least interchanged one comment. This is shown in Figure 7.5c. We can see that for high values of λ, the number of groups of size two or more is very small. Then it starts to grow significantly around $\lambda \sim 10$, reaches a maximum at $\lambda = 3$, and then again falls to the number of components of the original graph (considering all links). We also plot in Figure 7.5d

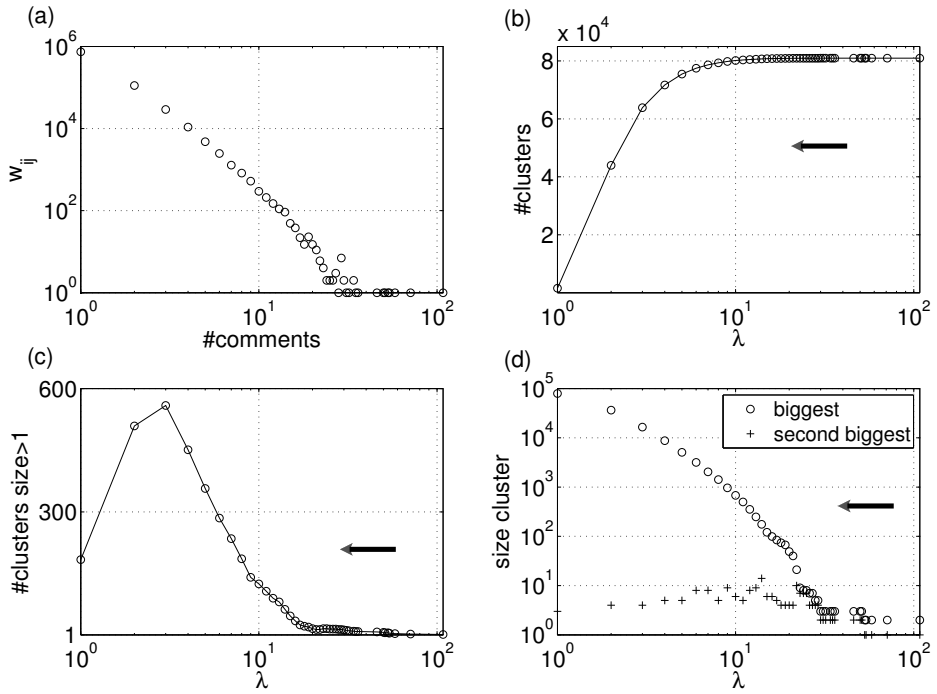

Figure 7.5: Results of the agglomerative clustering. **(a)** Distribution of weights (number of messages between pairs of users). **(b)** Number of clusters in function of λ and **(c)** considering only clusters of size > 1. **(D)** Size of the two biggest clusters.

the sizes of the two biggest components in function of λ. We can see that the biggest component grows very fast and the second biggest remains small, showing evidence of a giant cluster present in all scales.

We can track the resulting communities and show the networks at each λ. This is roughly illustrated in Figure 7.6, where two snapshots of the agglomerative process are shown. Figure 7.6 (top) corresponds to a high value of $\lambda = 20$, where a small backbone of the most connected users is starting to grow. Note that users are colored according to their score attribute (see Section 7.3.4). Users corresponding to the second class (high-quality commentators) are colored in red. Clearly, the backbone of Slashdot users is formed mainly by high-quality commentators. For $\lambda = 15$ (Figure 7.6 bottom), the most connected users receive even more connections and form the giant component. A few clusters of small sizes which still do not have grown sufficiently to be merged with the big community are also present.

7.4 Structure of the Discussions

After analyzing the social network of Slashdot, we focus on the question about how information is structured within a discussion thread. A thread starts with the publication of a post, which in

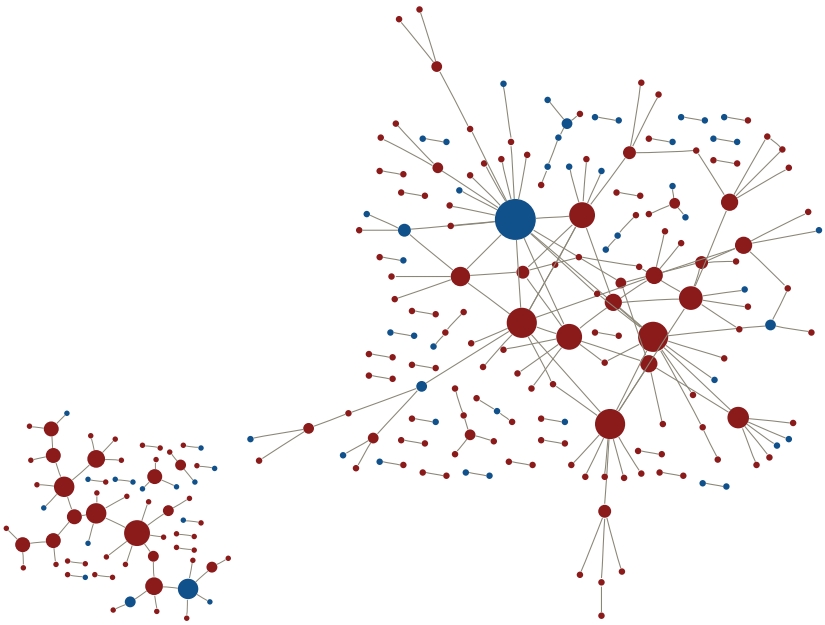

Figure 7.6: Two snapshots of the network for $\lambda = 20, 15$. Nodes are colored according to their score class (red: high quality, blue: medium quality). For clarity, we only show clusters of size > 1.

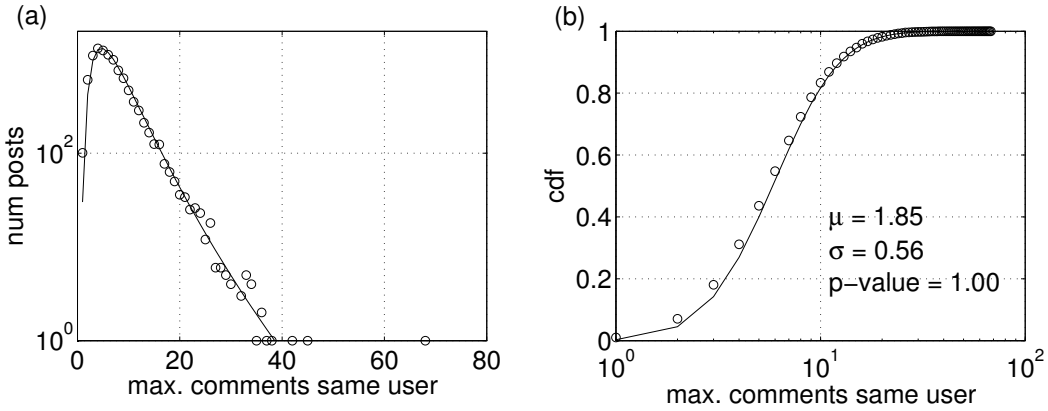

Figure 7.7: **(a)** Histogram of the maximum number of comments of one *single* user per post. Solid line corresponds to the best LN fit **(b)** cdf.

turn triggers an amount of activity in the form of comments. In this section, we present a statistical characterization of the structure of such discussions using a useful and intuitive radial tree representation. This representation leads naturally to a measure which can be useful to evaluate the degree of controversy of a given post.

An initial picture of the activity generated by posts can be found in previous studies (Kaltenbrunner et al., 2007a). Posts receive on average approximately 195 comments and there exists a clear scale in the number of comments a post can originate. Half of them receive less than 160 contributions. A small number of highly discussed ones, however, can trigger more than one thousand contributions.

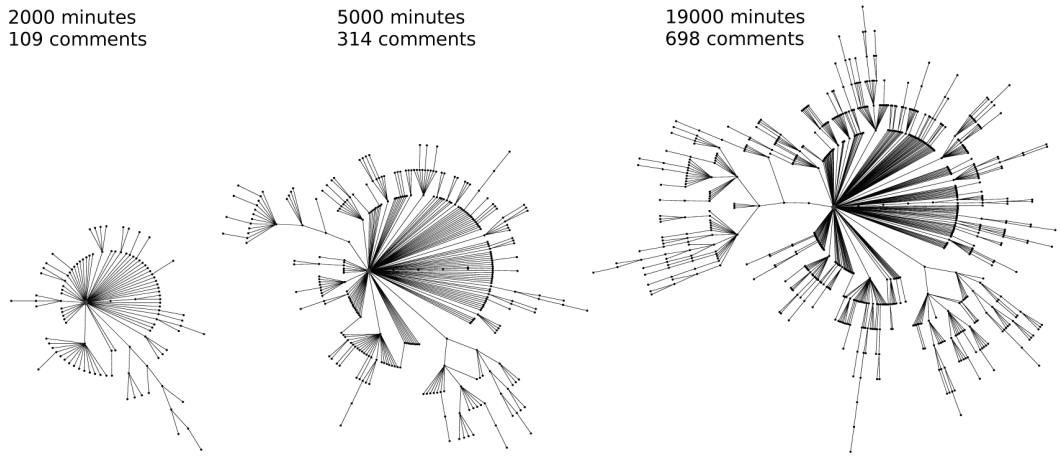

Figure 7.8: An example of radial tree structure corresponding to a controversial post related to Windows and Linux which received a total of 982 comments. The title of the post is *"Can Ordinary PC Users Ditch Windows for Linux?"*. Figures show three snapshots in different times.

The number of comments gives an idea of how the participation is distributed among the different articles, but is not enough to quantify the degree of interaction. For instance, a post may incite many readers to comment, but if the author of a comment does not reply the responses to his comment, there is no reciprocal communication within the thread. In this case, although users can participate significantly, we can hardly interpret that the post has been highly discussed. On the other hand, a post with a small number of contributors but with one long dialogue chain will evidence a high degree of reciprocal interaction (albeit its general interest may be reduced). At the description level of the social network, the reciprocity coefficient ρ and the agglomerative clustering described in the previous section already represent a measure to explain the degree of (reciprocal) interaction. At the description level of the individual post, a possible measure to quantify this type of interaction is the maximum number of comments written by a single user to a particular post.

We show this quantity (excluding the *anonymous* users) versus the number of posts, i.e. how many posts exist with a certain maximum number of comments written by the same user, in Figure 7.7. The obtained distribution has a peak at 4. As the cdf indicates, for approximately half of the posts at least one single user participates 5 times or more in the discussion. The log-normal shape of the distribution suggests a multiplicative process underlying the generation of this quantity. This indicates a strong heterogeneity and level of interaction within discussion threads. Users do not only give an opinion, but also interchange a significant quantity of messages, and the intensity of this interaction varies considerably throughout the different posts. We now study in more detail their intrinsic tree structure.

7.4.1 Radial Tree Representation

The high number of comments elicited by controversial posts makes them difficult to explore and to find relevant contributions within the nested dialogues. The current interface of Slashdot offers a filtering mechanism based on scores. By default, direct comments to a post rated 1 or higher are fully shown. For deeper nesting levels, comments can be fully shown (score 4 or above), abbreviated (score between 1 and 4) or hidden (score below 1).

We propose a natural representation of thread discussions which takes advantage of their structure. Consider a post as a central node. Direct replies to this post are attached in a first nesting level and subsequent comments at increasing nesting levels in a way that the whole thread can be considered as a circular structure which grows radially from a central root during its lifetime, a *radial tree*.

Figure 7.8 shows three snapshots of a radial tree associated to a controversial post which attracted a lot of users. An analog example of a less discussed post can be seen in Figure 7.9. More examples of trees are shown in Figure 7.10. Their profiles are highly heterogeneous. In some examples, only a huge number of contributions without replies appear in the first level, resulting in trees with high widths but small depths. In other examples, however, there are only discussions between two users who comment alternatively giving rise to very deep trees with small widths. Sometimes, the intensity of the discussion is translated to one of the branches because of a controversial comment which triggers even more reactions than the original post (e.g the post in the center of Figure 7.10).

Apart from being a useful tool for browsing and examining the contents of a highly discussed post, radial trees can be used to describe statistically how information is structured in a thread. In Figure 7.11a we plot the distribution of all the extracted comments per nesting level for all posts. This gives an idea about the relation between the width versus the depth of the trees. The first two

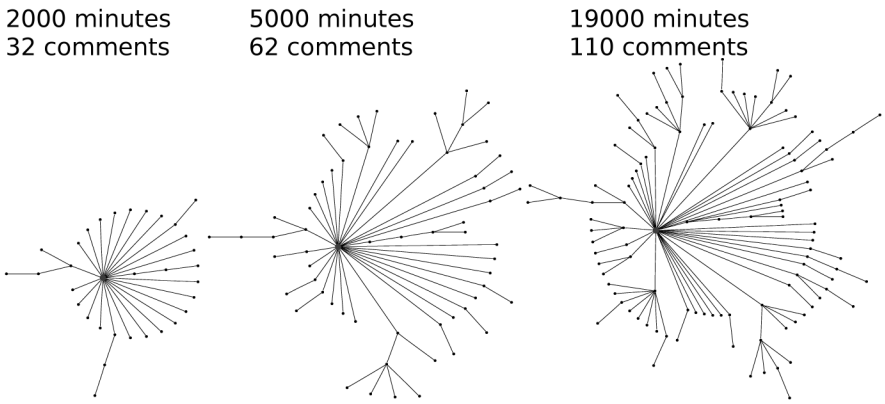

Figure 7.9: Radial tree structure of a little commented post which received 133 comments in total. Its title is *"Amazon One-Click Patent to be Re-Examined"*.

levels contain most of the comments and then their number decays exponentially in function of the depth. The maximum depth was 17. A general pattern which seems to be common to all threads is formed by a broad first nesting level of contributions, followed by a second, even wider set, and finally an exponential decay. This fact is reflected in the peak at depth two of the plot and the decreasing number of comments for deeper nesting levels. This gives evidence of the transient nature of the discussions. The reason behind this pattern is apparently related with an initial growth of interest which is reduced after users may have exposed all their knowledge, or translated to a more recent article. Only those who have engaged a dialogue will keep writing in subsequent levels. This decay could also be explained because of accessibility awkwardness, since the visibility of a comment can be proportional to its depth. The previous result would suggest that the majority of posts does not reach high nesting levels, and one would expect a similar distribution for the maximum depth of the posts. However, as Figure 7.11b indicates, the distribution of maximum depths does not follow the same pattern. It is almost symmetrical, weakly skewed toward smaller maximum depths. Although

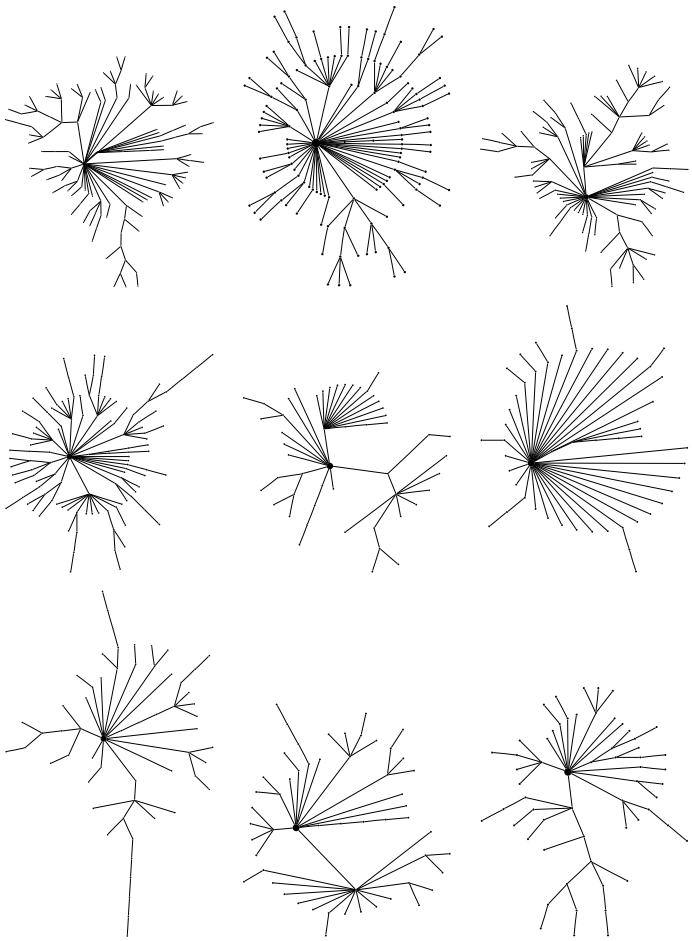

Figure 7.10: Heterogeneity in the radial trees.

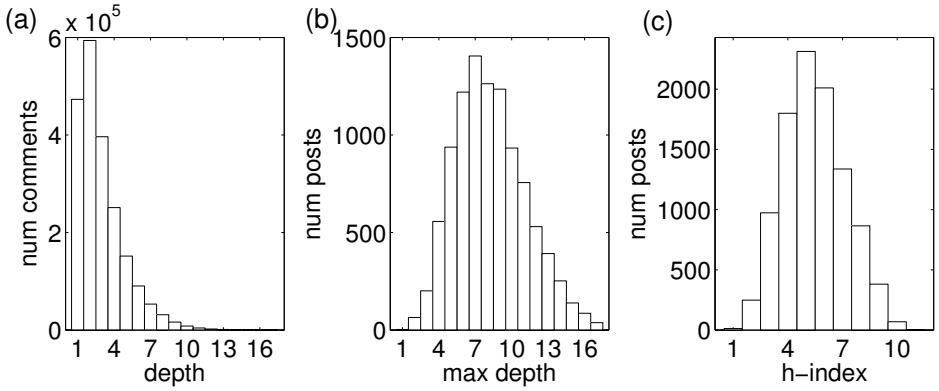

Figure 7.11: Results of **(a)** Number of comments per nesting level. **(b)** Number of posts per *maximum* depth. **(c)** Number of posts per h-index.

comments are concentrated in the first levels, threads typically reach a depth around 7.

Up to now, the quantities analyzed do not capture the apparent heterogeneity of the discussion threads we have reported in previous examples (see Figure 7.10). We now take a look on how the comments are generated within a given nesting level. This analysis can be performed extracting the branching factors b, that is, the number replies provoked by a given comment (or a given post). Figure 7.12 shows in log-log scale the distributions of b for the first five nesting levels. Level 0 corresponds to direct comments to the posts, whereas subsequent plots correspond to replies to comments.

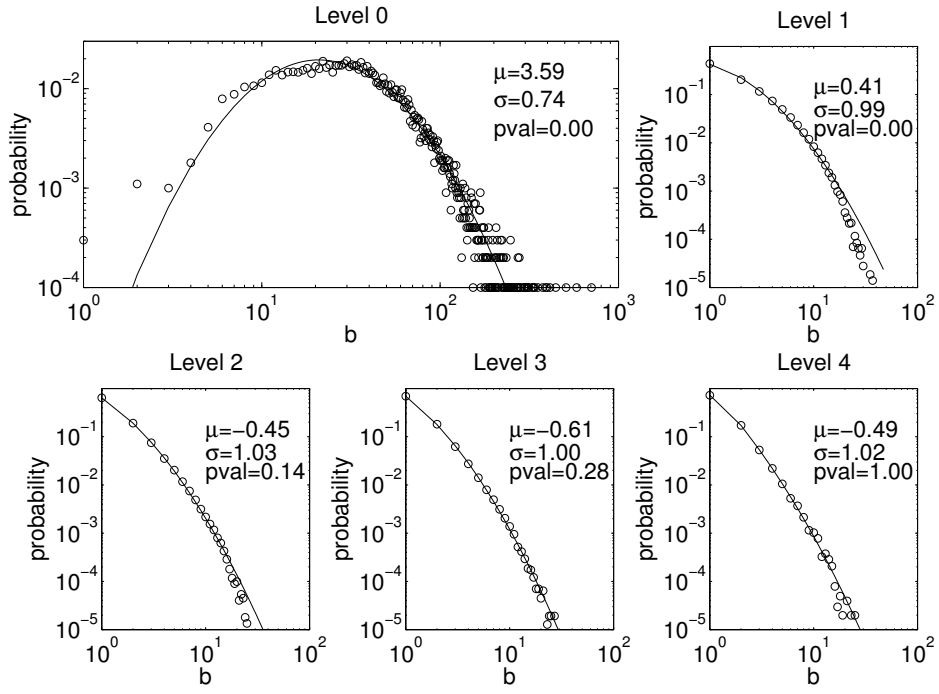

Figure 7.12: Distributions of branching factors b in 5 levels. Level 0 shows direct comments to posts.

First, the range of possible values spans almost three orders of magnitude for direct comments and is considerable for subsequent nesting levels, which gives evidence of the high heterogeneity underlying the discussion threads. Second, there is a clear discrepancy between commenting level 0 and subsequent ones. While the distribution of direct replies follows quite closely a bell-shape in the log-log domain, subsequent levels have an always decreasing probability. This illustrates the different nature of the process underlying the generation of comments to the initial post and the generation of replies to other comments. Interestingly, this variation is not reported in subsequent nesting levels. In addition, no dependency of the score on the nesting level could be found (data not shown). Although the resulting threads can take very different forms as we have previously shown, the same generative process seems to be taking place at all nesting levels. The bell-shaped curve of the first level branching factors, and the curvature in subsequent levels in log-log scales suggest again a good LN fit to explain the observed data, indicated by continuous lines in Figure 7.12. However, according to the p-values of KS-tests, the LN hypothesis is only accepted for levels deeper than 2.

7.4.2 The H-index as a Structural Measure of Controversy

In this subsection we will use the previous results to measure the degree of controversy of a post. As before, our approach does not consider semantic features and only relies on its structural information. It is important to note that a definition of *controversial* is necessarily subjective. However, indicators such as the number of comments received or the maximum depth of the discussions can be, among others, good candidate quantities to evaluate the controversy of a post, but suffer from some drawbacks as we will explain in what follows. We therefore seek for a measure, as simple as possible which incorporates as many of these factors and is able to rank a set of posts properly. The number of comments alone does not tell us much about the structure of the discussion. There might be a lot of comments in the first level but very little real discussions, such as in the post of Figure 7.13a. A better measure for the controversy of a post seems to be the maximum depth of the nesting. But again that measure has some drawbacks. Two users may become entangled in some discussion without participation of the rest of the community, increasing the depth of the thread. The example of Figure 7.13b illustrates this case. We thus want to overcome both types of bias.

We propose to quantify the degree of controversy associated to a post using an adapted version of the h-index (Hirsch, 2005), commonly used to characterize the scientific output of researchers. The papers of a researcher are ordered by their number of citations in descending order and the h-index is then defined as the maximum rank-number, for which the number of citations is greater or equal to the rank number. It represents a fair quantity which considers the number of papers published by the scientist and their visibility, or how often these papers are cited by other scientists. Some extensions of this index have been proposed as an alternative to the impact-factor of journals and

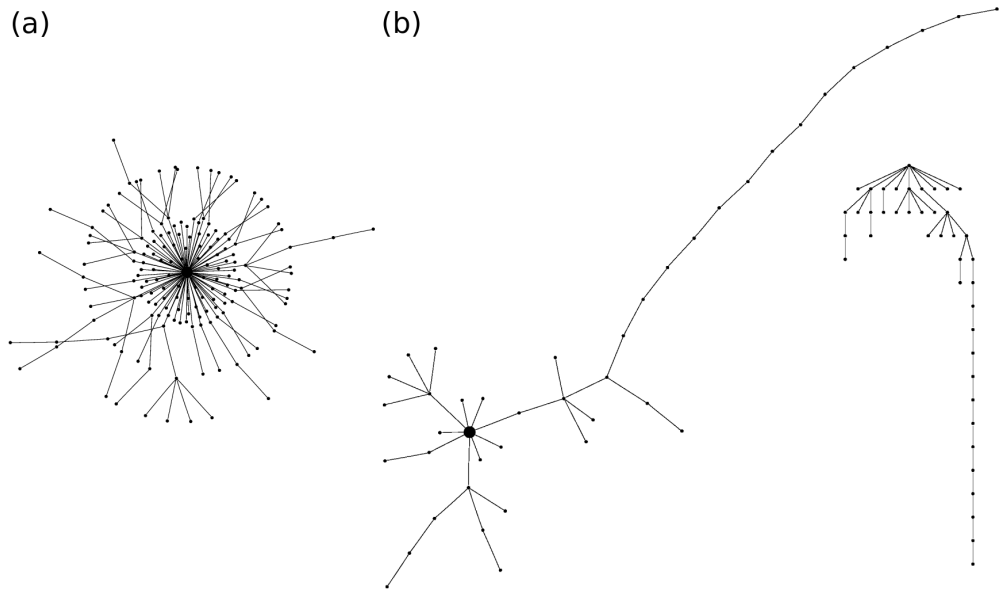

Figure 7.13: **(a)** Thread with many comments in the first level, but a few in subsequent levels. **(b)** Thread with a intense debate between two users.

conferences (Braun et al., 2005; Sidiropoulos and Manolopoulos, 2006). See (Bornmann and Daniel, 2007) for more details and a review on literature about the h-index.

For our purposes, we will define the h-index in the following way: given a radial tree corresponding to a discussion thread and its comments organized in nesting levels, the h-index h of a post is then the maximum nesting level i which has at least $h > i$ comments, or in other words, $h+1$ is the first nesting level i which has less than i comments. Turning back to Figure 7.13b, we can easily calculate the h-index. There are 9 comments in both first and second levels, 6 comments in the third level and 3 comments in the fourth level, which gives an h-index of 3. This post has maximum nesting level of 17, and it is ranked first if only the maximum depth is considered, but drops down to the position $9,239$ using the h-index. Similarly, the post of Figure 7.13a, which received 161 comments, has just an h-index of 3, because most of the comments are located in the upper levels. The post falls $4,412$ positions from a ranking based only on the number of comments to its rank based on the h-index.

Figure 7.11c shows a histogram with the number of posts with a given h-index. This distribution is less skewed than those of the number of comments and the maximum depth (compare with Figures 7.11a and 7.11b). In Figures 7.14 and 7.15 we plot a 3D chart to compare both, number of comments and maximum depth, against the h-index.

Although we observe an evident correlation between both quantities (more pronounced in the maximum depth) there exist posts which receive a lot of comments but interestingly do not have a significant h-index. This is even more evident when comparing h-index and maximum depth.

Since many posts share the same h-index, we need a way to break the ties. In this situation, we

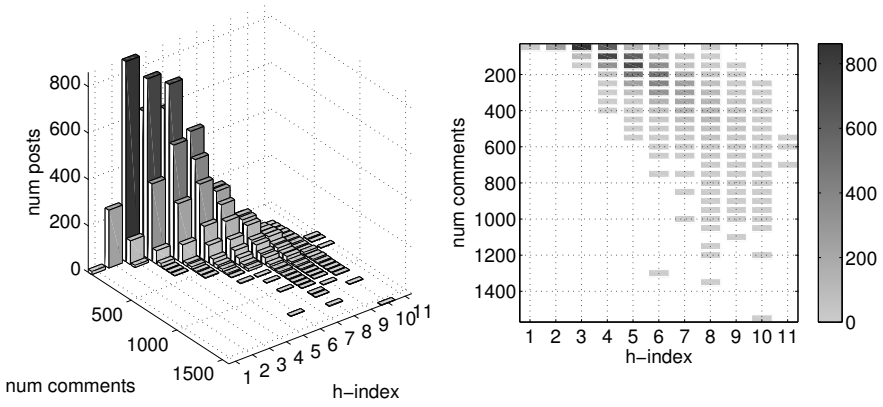

Figure 7.14: H-index versus number of comments.

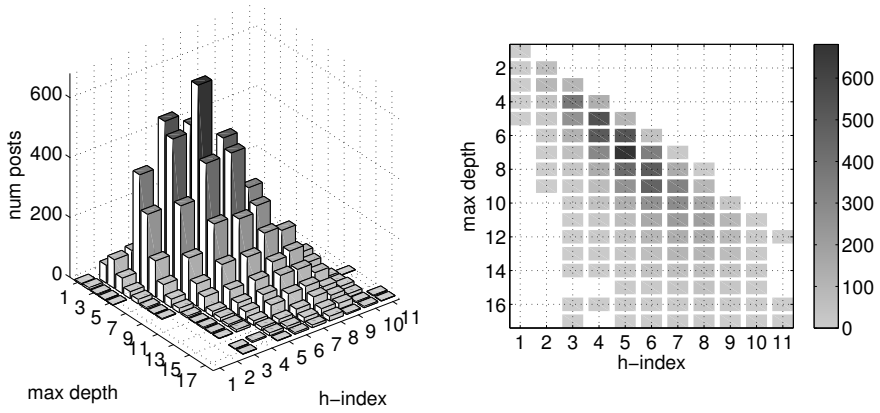

Figure 7.15: H-index versus maximum depth.

prioritize posts which reach a certain h-index with less comments. Thus, our final proposed measure uses as a first ranking criteria the h-index and as second the inverse of the number of comments. For a post i we use the following formula to rank it:

$$r_i = \text{h-index}_i + \frac{k}{\text{num comments}_i}, \tag{7.2}$$

where $k \leq 1$, so the decreasing order on the number of the received comments is preserved. The first 20 posts according to this ranking can be seen in Table 7.5. We also show their ranks if only their number of comments or their maximal discussion depth would have been considered. In the latter

#	H	Num cmnts (#)		Depth (#)		Title
1	11	527	(401)	16	(113)	Violating A Patent As Moral Choice
2	11	529	(390)	12	(1374)	Human Genes Still Evolving
3	11	605	(208)	16	(120)	Powell Aide Says Case for War a 'Hoax'
4	11	693	(96)	17	(34)	US Releasing 9/11 Flight 77 Pentagon Crash Tape
5	10	243	(3287)	15	(159)	Apple Fires Five Employees for Downloading Leopard
6	10	288	(2431)	14	(356)	Linus Speaks Out On GPLv3
7	10	290	(2409)	11	(1774)	New Mammal Species Found in Borneo
8	10	309	(2078)	13	(698)	Biofuel Production to Cause Water Shortages?
9	10	315	(1999)	12	(1168)	Torvalds on the Microkernel Debate
10	10	355	(1511)	17	(17)	Well I'll Be A Monkey's Uncle
11	10	361	(1446)	13	(747)	Windows Vista Delayed Again
12	10	366	(1394)	14	(416)	NSA Had Domestic Call Monitoring Before 9/11?
13	10	367	(1379)	11	(1922)	Unleashing the Power of the Cell Broadband Engine
14	10	380	(1279)	12	(1238)	Making Ice Without Electricity
15	10	384	(1243)	14	(424)	Evidence of the Missing Link Found?
16	10	391	(1178)	15	(198)	U.S. Army To Ramp Up Anthrax Purchasing
17	10	396	(1116)	12	(1263)	RMS Calls to Liberate Cyberspace
18	10	413	(989)	12	(1282)	Sony's Obsession with Proprietary Formats
19	10	420	(937)	13	(787)	Windows vs Mac Security
20	10	422	(919)	14	(437)	Goldfish Smarter Than Dolphins

Table 7.5: Top-20 controversial posts according to our proposed measure and corresponding positions according to the number of comments and maximum depth rankings. Column 1 indicates the rank based on Equation (7.2). Column 2 show the h-index. In columns 3 the number of comments received by the post is indicated. In brackes, we show the corresponding position in a rank based only in the number of comments. The same is done for the maximum depth in column 4.

case, we choose as well the number of comments to break the ties.

7.5 Temporal activity

After analyzing the Slashdot social network and the structure of discussion threads, the rest of this chapter is devoted to the analysis of the user activity in the temporal domain. We first give an overview of the global activity and identify oscillatory trends associated with the circadian cycle of the community. We then focus on the temporal profile of the comment activity induced by a published post, which is characterized by the response time distribution. We find that these temporal patterns are homogeneous in the sense that a big proportion of these patterns can be fit accurately using a mixture of two log-normal distributions with five parameters. The evolution of these parameters depending on the circadian cycle provides a way to understand the underlying process of message production associated to a post. Finally, we briefly study the single user activity and show that burstiness is a common property of this traffic.

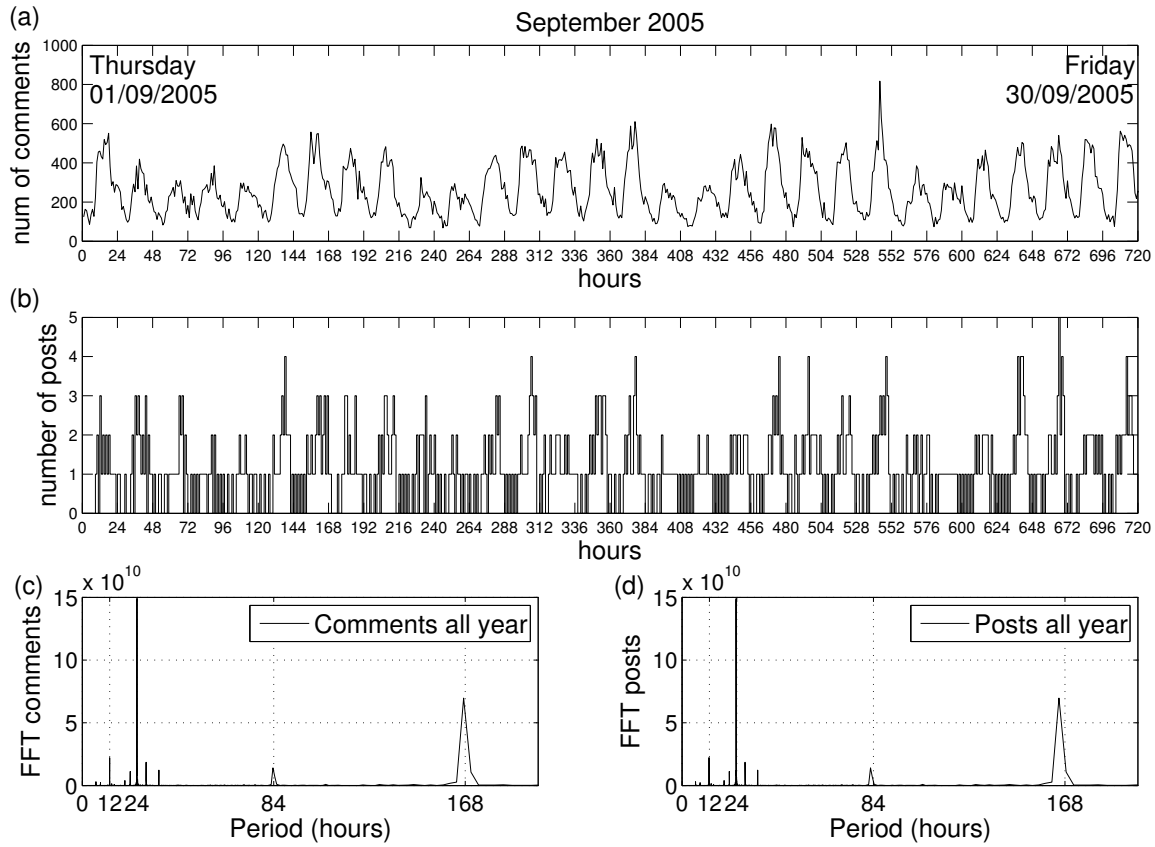

Figure 7.16: Traces of activity corresponding to the month of September, 2005: **(a)** user's comment activity and **(b)** post activity of the site. The FFT is also calculated covering all the crawled period for the **(c)** comments time series and the **(d)** posts time series.

7.5.1 Global cyclic activity

Figures 7.16a and 7.16b show a trace of the comment and post activity during the month of September. A part of small fluctuations, we can see a global cyclic pattern in agreement with the social activity outside the web space. Although the number of comments generated is higher than the number of posts, both seem to be perfectly correlated with the circadian cycle. Another cyclic trend can be observed during the weekends where the activity diminishes. Figures 7.16c and 7.16d corroborate this conclusion showing the fast Fourier transform (FFT) of both time series. We identify the main frequency at 24 hours and the weekly period (168 hours) as the second main frequency.

Figures 7.17a and 7.17b show the (normalized) mean activity and standard deviations of both posts and comments for all the crawled period. Figure 7.17a shows regular, steady activity during working days which slows down during weekends. This weekly cycle is interleaved by daily oscillations illustrated in Figure 7.17b. The daily activity cycle reaches its maximum at 1pm approximately and its minimum during the night between 3am and 4am. Although Slashdot is open to public access

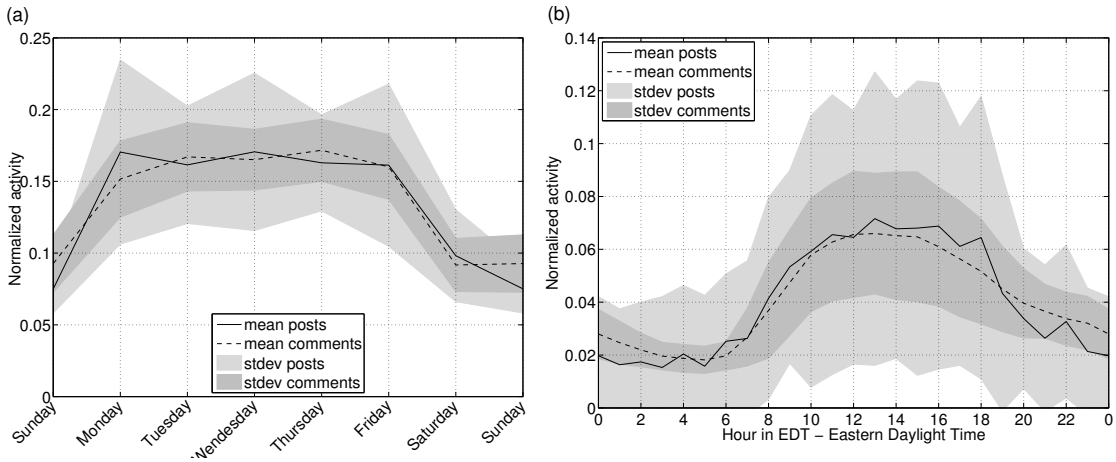

Figure 7.17: **(a)** Weekly and **(b)** daily activity cycles.

around the world, we see that its activity profile is clearly biased towards the American time-schedule.

Interestingly, although post activity shows more fluctuations and higher standard deviations than comment activity, there is little discrepancy between their mean temporal profiles. This difference in the deviations is not surprising given the greater number of comments (see Table 7.1). We notice that the standard deviations of the daily post and commenting activities also show similar cyclic behavior (Figure 7.17b).

7.5.2 Global post-induced activity

We now analyze the comment activity induced by the publication of a post. Given a specific post, we study its post-comment interval (PCI) probability distribution, which we define as the probability distribution for a post to receive a comment t minutes after it has been published.

Analysis of the activity generated by a single post

To start our analysis we choose the same pair of posts presented in Figures 7.8 and 7.9. The first one corresponds to a controversial post which attracted many comments and the other one to a post with less contributions. Figures 7.18a and 7.18b show the PCI histograms with the number of comments received during the first five hours after their publication. The temporal profile clearly shows a brief transient where activity grows very fast and then a long decay. Despite the high heterogeneity present in the structure of the threads reported in Section 7.4.1, both temporal shapes look very similar, regardless the obvious fluctuations and sampling sizes.

Figures 7.18c and 7.18d show the cdfs of the PCIs for both posts in logarithmic time scale. The cdf removes the fluctuations and both posts present characteristic log-normal profiles. Dashed-dotted

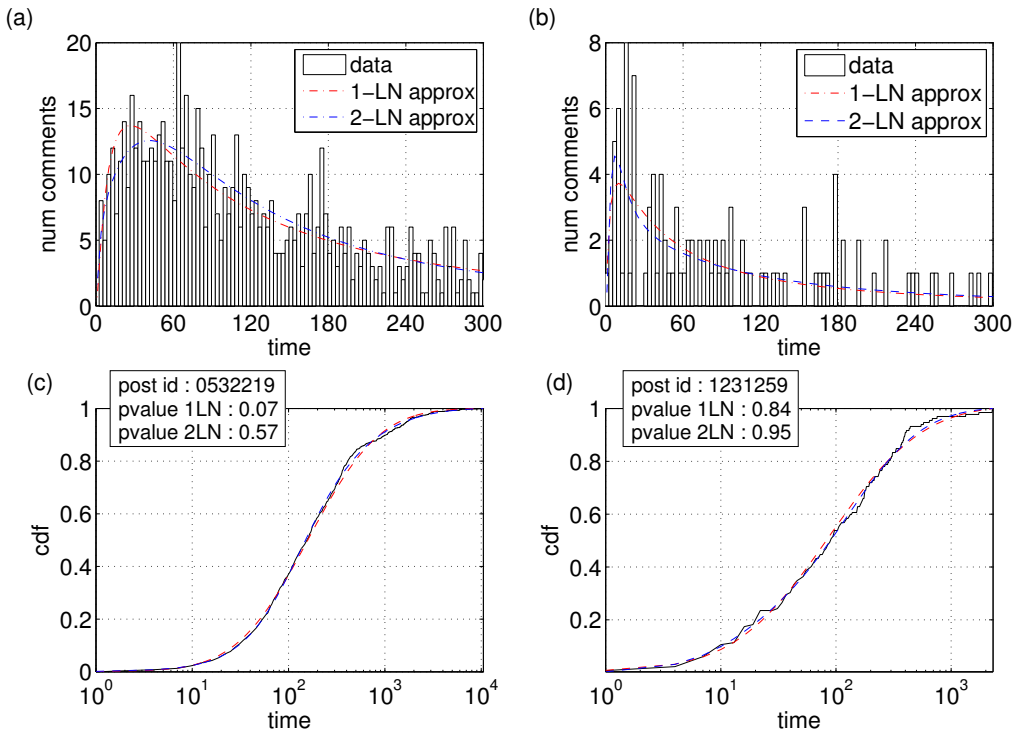

Figure 7.18: Temporal profile of two example posts (left and right). **(a)** Histogram of the number of comments received within the first five hours. Red dashed-dotted line corresponds to a fit using one log-normal distribution and dashed blue line corresponds to a fit using a mixture of two log-normal distributions. **(b)** same as (a) for a post with less comments. **(c)** cdf and approximations in logarithmic temporal scale for the first post. **(d)** Same as (c) for the second post.

red curves indicate the best log-normal fit. For the first post (left), the MLE values are $\mu = 5.07$ and $\sigma = 1.34$ and KS test rejects the log-normal with level of significance $\alpha = 0.1$. However, the second post, with parameters are $\mu = 5.07$ and $\sigma = 1.34$, seems to fit better.

The proposed functional form for the temporal profile seems accurate, but can we improve the log-normal approximation? We also try to fit the PCIs using a mixture of two log-normal distributions (Stouffer et al., 2006), which has the following probability density function:

$$p(x;\theta) = w\frac{1}{x\sqrt{2\pi}\sigma_1}e^{-(\log x-\mu_1)^2/2\sigma_1^2} + (1-w)\frac{1}{x\sqrt{2\pi}\sigma_2}e^{-(\log x-\mu_2)^2/2\sigma_2^2} \tag{7.3}$$

where $\theta = \langle \mu_1, \mu_2, w, \sigma_1, \sigma_2 \rangle$ is the vector of five parameters. w is the mixing coefficient and takes values within the range $[0, 0.5]$. Since a double log-normal has more parameters than the single one, we expect to improve the initial approximations. Optimal values for θ were found applying MLE, this time numerically using the function `fminsearch` of Matlab. The curves corresponding to these fits are shown in dashed blue lines. For both posts their p-values after applying the KS test are higher than before, in agreement with our hypothesis.

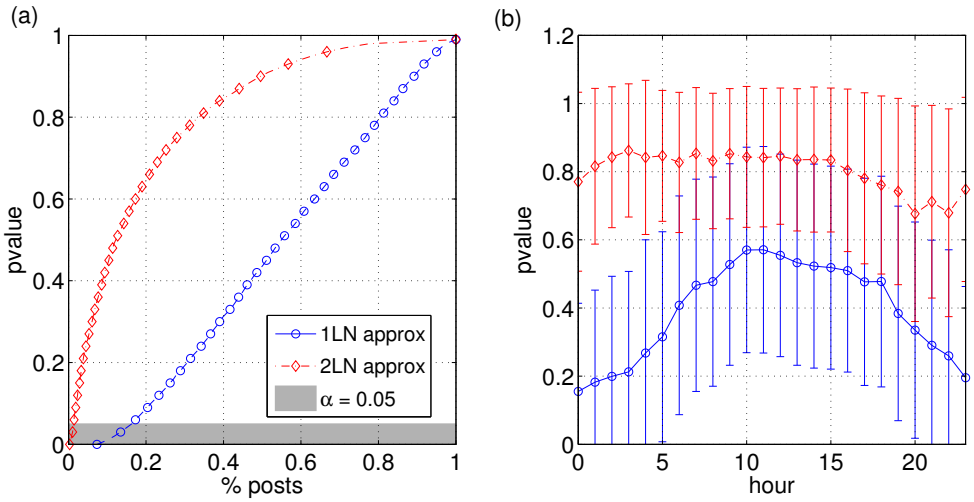

Figure 7.19: Comparison of the approximation quality of the PCIs using a single log-normal distribution against using a mixture of two log-normals. **(a)** Results of the KS-tests for all posts. **(b)** Error-bars of the p-values in function of the publishing hour of the post.

Analysis and approximation of all posts

After accurately approximated the PCI distributions of two single posts we now address the question whether all posts published during our crawling period obey the same probabilistic model.

We thus compute approximations using single and double log-normal models of all the PCIs covered by our crawling, and perform KS tests for each fit. Figure 7.19a shows the cdf of the p-values in function of the proportion of posts. The single log-normal hypothesis is rejected at a confidence level of $\alpha = 0.05$ for nearly 1.9% of the posts whereas for the case of two log-normals this quantity is reduced to 0.02% approximately. Globally, the double log-normal model provides a better description of the activity.

A more elaborated justification for this difference can be obtained if we look at the dependence of the p-values on the publishing hour the post. We compute the mean and standard deviation of the p-values after grouping their corresponding posts according to the hour of the day of the publication. As figure 7.19b shows, all the mean p-values obtained using a single log-normal (blue curve) are smaller than the corresponding double log-normal approximations (red curve). Moreover, there is a dependence on the circadian cycle in the single log-normal approximation. P-values corresponding to posts written during the day are higher than p-values obtained for posts written during the night. This dependence is almost removed when we use a double log-normal model, suggesting that posts written during hours of less activity have a temporal profile different from posts written during working hours. These two types of profiles seem to be better captured by the double log-normal model.

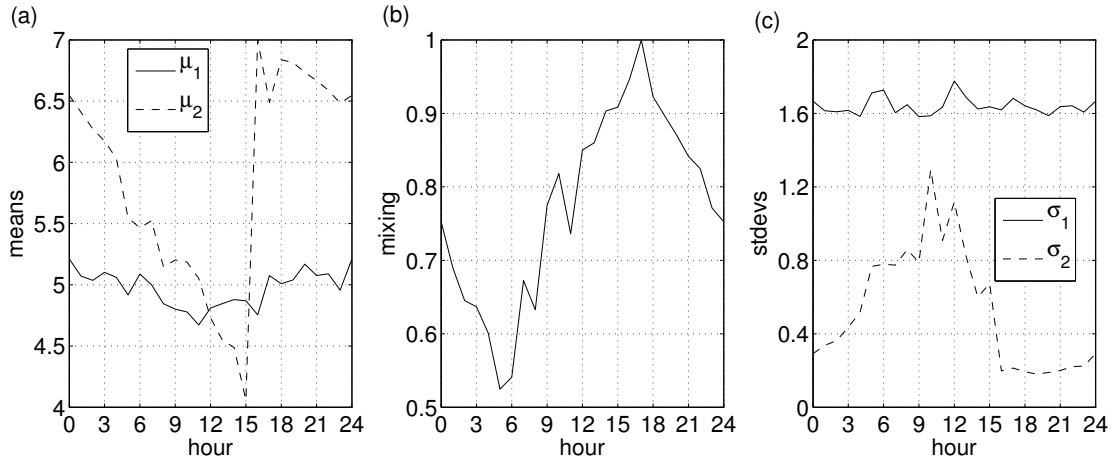

Figure 7.20: Evolution of the parameters which best approximate a mixture of two log-normal distributions to an aggregate of all PCIs corresponding to a post published in a particular hour of the day (see the text). **(a)** Means of the first and the second log-normals, **(b)** mixing coefficient and **(c)** their standard deviations.

.

An explanation: Two waves of activity

Is there an intuitive explanation for this discrepancy? To find what kind of information is captured by the double log-normal model we proceed in the following way: we group all PCIs of posts according to the publishing hour of their corresponding posts, resulting in 24 aggregates of PCIs which summarize the temporal profiles of posts concentrated in a particular hour of the day. For each of these 24 aggregates we perform a double log-normal fit. This methodology allows to analyze the temporal evolution of the double log-normal parameters of these approximations. The results are shown in Figure 7.20.

We first analyze the means μ_1 and μ_2, which indicate the center of the two log-normal kernels. From the first plot we can see that μ_1 only shows small fluctuations around 5. Conversely, the global trend of μ_2 differs significantly. At midnight μ_2 is distant from μ_1 and as time evolves, μ_2 approaches progressively until both coincide at 12pm. At 4pm there is a sudden increase to 7 and the decay starts again until 3pm of the next day, so a cyclic behavior is observed in μ_2.

The mixing parameter w (middle-plot) tells us the proportion of the first log-normal used in the mixture. On one hand, for values $w \approx 0.5$ the two log-normals are necessary to describe the temporal profile. On the other hand, for values $w \approx 1$, only the first log-normal suffices. Concerning the evolution of w we observe also a cyclic behaviour, with minimum and maximum at 4am and 4pm respectively. Interestingly, this oscillation correlates with the circadian cycle reported before (see Figure 7.17b).

Finally, the standard deviations also show interesting profiles. As in the case of μ_1, we see that σ_1

only show minor fluctuations whereas σ_2 shows a weak cyclic behavior, with a minimum at 3pm and a small and irregular increase until 10am.

What conclusions can we draw from all these results? First, the steady profile governing both μ_1 and σ_1 parameters of the first log-normal kernel would suggest that a first wave of activity starts at the moment of the publication of the post. This wave of activity is, despite small fluctuations, independent of the publishing hour of the post. Second, the cyclic behavior found in the parameters μ_2 and σ_2 shows evidence of the existence of a second wave, in this case, relative to the hour where the post has been published. This second wave differs mostly from the first wave at 4pm, although this difference coincides with the time where w reaches its maximum, and therefore only the first log-normal suffices to describe the data. Combining the evolution of w and the discrepancy between both log-normal parameters, we can identify a period which spans approximately between 11pm and 6am where both log-normal distributions differ considerably and still the mixing coefficient w is significant ($w << 1$). This interpretation can also provide an explanation of why the minimum of the performance using only one single log-normal occurs between 11pm and 4am (see Figure 7.19b).

7.6 Conclusions and future work

Our analysis represents a step toward the understanding of the structure of networks in which relations are hidden and more generic than explicit, well-defined links such as friendships or affiliations. The Slashdot network exhibits some special features that deviate from traditional social networks: neutral mixing by degree, almost identical in and out degree distributions, only moderated reciprocity, and absence of a complex community structure.

We conjecture that most of the reactions in Slashdot arise when high diversity in opinions occur. Users are therefore more inclined to be linked to people who express different points of view (Munroe, 2008). The nature of this interaction seems to be a key aspect to understand the obtained results.

Unlike the BBS network (Zhongbao and Changshui, 2003; Goh et al., 2006) where discussions are unrestricted, the scoring system of Slashdot guarantees a high quality and representativity of the social interaction. This particular feature allowed us to find a correlation between scores and number of received replies and to distinguish clearly between two classes of users: good writers who, on average achieve high scores for their comments, and regular writers. The number of replies of a comment depends mostly on its quality (the score it achieved) but we find some weak evidence for user reputation influencing the connectivity in the network. Good writers are more likely than regular ones to receive replies to occasional comments with low scores. However, this effect is not strong enough to cause assortative mixing by score since the opposite is not true. Regular writers can expect a similar number of replies as good writers to their comments with high scores, so there is no negative

effect of a user's reputation.

When analyzing the tree structure generated by the nesting of comments, we find interesting properties which characterize a discussion thread: the probability distributions of the branching factors and the mean score values remain constant as one goes deeper and deeper into the radial tree, suggesting that, despite the strong heterogeneity in the shapes of the discussions visible in their radial tree representation, a simple depth-invariant mechanism exists which is responsible for their evolution. A detailed study of the dynamics governing the growth of nested discussions is a topic of ongoing research.

To measure the degree of controversy of a discussion, a recent approach (Mishne and Glance, 2006) trains a classifier using features that combine semantic and structural information. Our proposed measure, based on the h-index, appears to be a more convenient indicator because of its simplicity, objectivity and robustness. It can be calculated efficiently and is monotonic (it never decreases), which makes it also a stable quantity to monitor and rank a discussion thread while it is still alive and receiving contributions. Its robustness and stability make this measure an appropriate candidate to study the evolution of the discussions. However, human based visual validation is necessary to check how it correlates with subjective sensation of controversy.

The h-index can not only be useful to determine the degree of controversy of a news post, but also to evaluate other magnitudes. For instance, it can be directly calculated for each user, which would give then a similar interpretation as the scientific citations.

We would expect that the time-spans between publishing and reading of a post also follow log-normal patterns. This could be easily verified checking the server logs of Slashdot or access-times of an external homepage linked by a Slashdot post. Such a study has been performed to show the Slashdot effect (Adler, 1999), but the scale of the data presented does not allow to draw significant conclusions. Further investigation is needed to verify this claim.

Log-normal temporal patterns similar to those described above were found in person-to-person communication by Stouffer et al. (2006), who investigated the waiting and inter-event times of an e-mail activity dataset. A second coincidence between their study and our findings is that the number of comments (or e-mails in their case) can be well approximated by the same distribution (a truncated log-normal in this case). The temporal patterns of the e-mail data were previously claimed to show power-law behavior, which would be explained by a queuing model (Barabási, 2005). Although this model might allow insight into other types of human activity (Vázquez et al., 2006) it is not able to account for the observed log-normal temporal patterns.

The remarkable regular temporal patterns present in Slashdot allow to devise a procedure to predict the number of comments that a post will generate. The transient profile of in the PCIs (e.g. the sharp initial raise) makes accurate prediction nearly impossible using standard time-series methods.

A first step in that direction, which predicts with moderate accuracy the expected reaction to a post when only a small fraction of its total number of comments has been received, has been already developed (Kaltenbrunner et al., 2007a). This approach could allow, for instance, dynamic pricing or placing of online advertisements according to the expected reaction to a post, or early adaptation of online marketing campaigns.

Finally, there is the question whether the temporal behavior observed in Slashdot can be explained by means of a simple stochastic model, similar to the one studied in the first part of this thesis. We have shown that global activity is generated in large correlation with the circadian rhythm but, apart from this, does not fluctuate significantly. Thus there exists a "stable" number of comments which is continuously originated and has to be delivered to the posts which are most controversial and not yet saturated. These two factors which characterize a discussion thread, namely, controversy and novelty, seem to play a crucial role in a yet undiscovered model which would explain the observed dynamics. Finding such a model will provide theoretical understanding of the underlying phenomena responsible for this apparently quite general behavior pattern.

Part IV

Bibliography

Journal publications

Gómez, V., Mooij, J. M., and Kappen, H. J. (2007). Truncating the loop series expansion for belief propagation. *Journal of Machine Learning Research*, 8(Sept):1987–2016.

Kaltenbrunner, A., Gómez, V., and López, V. (2007a). Phase transition and hysteresis in an ensemble of stochastic spiking neurons. *Neural Computation*, 19(11):3011–3050.

Kaltenbrunner, A., Gómez, V., Moghnieh, A., Meza, R., Blat, J., and López, V. (2007b). Homogeneous temporal activity patterns in a large online communication space. *IADIS International Journal on WWW/INTERNET*. (to appear).

Conference proceedings

Gómez, V., Kaltenbrunner, A., and López, V. (2006). Event modeling of message interchange in stochastic neural ensembles. In *IJCNN'06 Proceedings*, pages 81–88. IEEE World Congress of Computational Intelligence, Vancouver, BC, Canada.

Gómez, V., Kaltenbrunner, A., and López, V. (2008). Statistical analysis of the social network and discussion threads in slashdot. In *WWW '08: Proceedings of the 17th international conference on World Wide Web*, Beijing, China. ACM.

Kaltenbrunner, A., Gómez, V., and López, V. (2007a). Description and prediction of Slashdot activity. In *Proceedings of the 5th Latin American Web Congress (LA-WEB 2007)*, Santiago de Chile. IEEE Computer Society.

Kaltenbrunner, A., Gómez, V., Moghnieh, A., Meza, R., Blat, J., and López, V. (2007b). Homogeneous temporal activity patterns in a large online communication space. In *Proceedings of the BIS 2007 Workshop on Social Aspects of the Web (SAW 2007)*. Poznan, Poland.

Bibliography

Abeles, M. (1991). *Corticonics: Neural circuits of the cerebral cortex*. Cambridge University Press, Cambridge. 38

Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. (1988). A learning algorithm for Boltzmann machines. *Neurocomputing: foundations of research*, pages 635–649. 9

Adamic, L. A. (1999). The small world web. In *Proc. 3rd ECDL '99*, pages 443–452, London, UK. Springer-Verlag. 104

Adler, S. (1999). The Slashdot effect, an analysis of three Internet publications. http://ssadler.phy.bnl.gov/adler/SDE/SlashDotEffect.html. 122, 152

Albert, R. and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97. 108, 115

Albert, R., Jeong, H., and Barabási, A.-L. (1999). Diameter of the world-wide web. *Nature*, 401:130. 104

Bachnik, W., Szymczyk, S., Leszczynski, P., Podsiadlo, R., Rymszewicz, E., Kurylo, L., Makowiec, D., and Bykowska, B. (2005). Quantitative and sociological analysis of blog networks. *Acta Physica Polonica B*, 36:2435–2446. 107

Baeza-Yates, R., Castillo, C., and López, V. (2005). Characteristics of the Web of Spain. *Cybermetrics*, 9(1). 104

Baggio, R. (2007). The web graph of a tourism system. *Physica A: Statistical Mechanics and its Applications*, 379:727–734. 104

Bair, W., Koch, C., Newsome, W., and Britten, K. (1994). Power spectrum analysis of bursting cells in area MT in the behaving monkey. *Journal Of Neuroscience*, 14:2870–2892. 33

Bak, P. (1996). *How nature works: The Science of Self-Organized Criticality*. Springer. 61

Bak, P., Christensen, K., Danon, L., and Scanlon, T. (2002). Unified scaling law for earthquakes. *Physical Review Letters*, 88(17):178501. 37

Bak, P., Tang, C., and Wiesenfeld, K. (1987). Self-organized criticality - An explanation of 1/f noise. *Physical Review Letters*, 59:381–384. 37

Baoill, A. Ó. (2000). Slashdot and the public sphere. *First Monday*, 5(9). 123

Barabási, A.-L. (2005). The origin of bursts and heavy tails in human dynamics. *Nature*, 435:207–211. 106, 152

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–512. 114

Barabási, A.-L., Goh, K. I., and Vázquez, A. (2005). Reply to comment on "The origin of bursts and heavy tails in human dynamics". 106

Barrat, A., Barthélemy, M., Pastor-Satorras, R., and Vespignani, A. (2004). The architecture of complex weighted networks. *Proceedings Of The National Academy Of Sciences*, 101(11):3747–3752. 105, 128

Barrat, A. and Weigt, M. (2000). On the properties of small-world network models. *European Physical Journal B*, 13:547–560. 109

Bertschinger, N. and Natschlager, T. (2004). Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation*, 16(7):1413–1436. 36

Bethe, H. A. (1935). Statistical theory of superlattices. *Proc. Royal Society A*, 150:552–575. 2

Bi, G. Q. and Poo, M. M. (1998). Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal Of Neuroscience*, 18:10464–10472. 22, 55

Bienenstock, E. (1995). A model of neocortex. *Network-Computation In Neural Systems*, 6:179–224. 38

Bishop, P. O., Levick, W. R., and Williams, W. O. (1964). Statistical analysis of the dark discharge of lateral geniculate neurones. *Journal Of Physiology*, 170:598–612. 33

Bornmann, L. and Daniel, H. D. (2007). What do we know about the h index? *Journal Of The American Society For Information Science And Technology*, 58:1381–1385. 143

Braun, T., Glanzel, W., and Schubert, A. (2005). A Hirsch-type index for journals. *Scientist*, 19:8–8. 143

Braunstein, A., Mézard, M., and Zecchina, R. (2005). Survey propagation: An algorithm for satisfiability. *Random Struct. Algorithms*, 27(2):201–226. 36, 74

Brette, R. (2006). Exact simulation of integrate-and-fire models with synaptic conductances. *Neural Computation*, 18(8):2004–2027. 21

Brette, R. (2007). Exact simulation of integrate-and-fire models with exponential currents. *Neural Computation*, 19(10):2604–2609. 21

Brette, R., Rudolph, M., Carnevale, T., Hines, M., Beeman, D., Bower, J. M., Diesmann, M., Morrison, A., Goodman, P. H., Harris, Jr., F. C., Zirpe, M., Natschlager, T., Pecevski, D., Ermentrout, B., Djurfeldt, M., Lansner, A., Rochel, O., Vieville, T., Muller, E., Davison, A. P., El Boustani, S., and Destexhe, A. (2007). Simulation of networks of spiking neurons: A review of tools and strategies. *Journal of Computational Neuroscience*. 21

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J. (2000). Graph structure in the web. In *Proc. 9th international WWW conference on Computer networks*, pages 309–320, The Netherlands. 104

Brunel, N. and Hakim, V. (1999). Fast global oscillations in networks of integrate-and-fire neurons with low firing rates. *Neural Computation*, 11(7):1621–1671. 15

Brush, A. B., Wang, X., Turner, T. C., and Smith, M. A. (2005). Assessing differential usage of usenet social accounting meta-data. In *Proc. SIGCHI '05*, pages 889–898, New York, USA. ACM. 107

Bullen, P. S. (2003). *Handbook of Means and Their Inequalities*. Kluwer, Dordrecht, The Netherlands. 47

Burkitt, N. (2006). A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological Cybernetics*, 95(1):1–19. 11, 12

Chandrasekhar, S. (1943). Stochastic problems in physics and astronomy. *Review Modern Physics*, 15:1–89. Reprinted in Noise and Stochastic Processes (Ed. N. Wax). New York: Dover, pp. 3-91, 1954. 15

Chen, C. C. (1994). Threshold effects on synchronization of pulse-coupled oscillators. *Physical Review E*, 49:2668–2672. 37

Chen, Q., Chang, H., Govindan, R., Jamin, S., Shenker, S., and Willinger, W. (2002). The origin of power laws in Internet topologies revisited. In *Proceedings of 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02),*, volume 2, pages 608–617. 105

Chertkov, M. and Chernyak, V. Y. (2006a). Loop calculus helps to improve belief propagation and linear programming decodings of LDPC codes. In *invited talk at 44th Allerton Conference*. 98

Chertkov, M. and Chernyak, V. Y. (2006b). Loop series for discrete statistical models on graphs. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(06):P06009. 3, 65, 66, 67, 68, 70, 97, 99

Clauset, A., Shalizi, C. R., and Newman, M. E. J. (2007). Power-law distributions in empirical data. 117, 130

Cox, D. (1984). *Long-range dependence: a review*, pages 55–74. Iowa State University Press. 106

Crovella, M. E. and Bestavros, A. (1997). Self-similarity in world wide web traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846. 104, 106

Dechter, R., Kask, K., and Mateescu, R. (2002). Iterative join-graph propagation. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI-02)*, pages 128–13, San Francisco, CA. Morgan Kaufmann. 67

DeGroot, M. H. and Schervish, M. J. (2002). *Probability and Statistics*, chapter 9.6, pages 566–572. Addison-Wesley, New York, 3rd edition. 120

Delorme, A. and Thorpe, S. J. (2003). SpikeNET: an event-driven simulation package for modelling large networks of spiking neurons. *Network: Computation in Neural Systems*, 14:613–627. 21

Dewes, C., Wichmann, A., and Feldmann, A. (2003). An analysis of Internet chat systems. In *IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 51–64, New York, USA. ACM Press. 106

Dezsö, Z., Almaas, E., Lukács, A., Rácz, B., Szakadát, I., and Barabási, A.-L. (2006). Dynamics of information access on the web. *Physical Review E*, 73(6):066132. 106

Diesmann, M., Gewaltig, M. O., and Aertsen, A. (1999). Stable propagation of synchronous spiking in cortical neural networks. *Nature*, 402:529–533. 38

Dorogovtsev, S. N. and Mendes, J. F. F. (2002). Evolution of networks. *Advances In Physics*, 51(4):1079–1187. 2, 108

Dorogovtsev, S. N., Mendes, J. F. F., and Samukhin, A. N. (2000). Structure of growing networks with preferential linking. *Physical Review Letters*, 85(21):4633–4636. 105, 115

Drineas, P., Krishnamoorthy, M., Sofka, M., and Yeneri, B. (2004). Studying e-mail graphs for intelligence monitoring and analysis in the absence of semantic information. *Lecture Notes in Computer Science*, pages 297–306. 104

Duarte, F., Mattos, B., Bestavros, A., Almeida, V., and Almeida, J. (2007). Traffic characteristics and communication patterns in blogosphere. In *International Conference on Weblogs and Social Media (ICWSM-07)*, Boulder, Colorado, USA. 107, 123

Durstewitz, D., Seamans, J. K., and Sejnowski, T. J. (2000). Neurocomputational models of working memory. *Nature Neuroscience Supplement*, 3:1184–1191. 44

Ebel, H., Mielsch, L.-I., and Bornholdt, S. (2002). Scale-free topology of e-mail networks. *Physical Review E*, 66(3):035103. 104

Elidan, G., McGraw, I., and Koller, D. (2006). Residual belief propagation: Informed scheduling for asynchronous message passing. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, Boston, Massachussetts. AUAI Press. 83

Erdös, P. and Rény, A. (1959). On random graphs. *Publications Matematicae*, 290(6). 108

Erdös, P. and Rény, A. (1960). On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, 17(5). 108

Ernst, U., Pawelzik, K., and Geisel, T. (1998). Delay-induced multistable synchronization of biological oscillators. *Physical Review E*, 57:2150–2162. 33, 38

Everitt, B. S. (1993). *Cluster Analysis*. John Wiley and Sons Inc, 3rd. edition. 112

Faloutsos, M., Faloutsos, P., and Faloutsos, C. (1999). On power-law relationships of the Internet topology. In *SIGCOMM '99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, pages 251–262, New York, USA. ACM Press. 104

Feder, H. J. S. and Feder, J. (1991). Self-organized criticality in a stick-slip process. *Physical Review Letters*, 66(20):2669–2672. 37

Feller, W. (1968). *An Introduction to probability theory and its applications*. Wiley. 23, 25

Fenner, T., Levene, M., and Loizou, G. (2006). A stochastic model for the evolution of the web allowing link deletion. *ACM Transactions on Internet Technology (TOIT)*, 6(2):117–130. 105

Fisher, D., Smith, M., and Welser, H. T. (2006). You are who you talk to: Detecting roles in usenet newsgroups. In *Proc. HICSS '06*, Washington, USA. IEEE CS. 107

Freeman, L. C. (1977). A set of measures of centrality based upon betweenness. *Sociometry*, 40:35–41. 112

Freeman, W. T., Pasztor, E. C., and Carmichael, O. T. (2000). Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47. 66

Frette, V., Christensen, K., Malthe-Sorensen, A., Feder, J., Jossang, T., and Meakin, P. (1996). Avalanche dynamics in a pile of rice. *Nature*, 379:49–52. 37

Fu, F., Liu, L., Yang, K., and Wang, L. (2006). The structure of self-organized blogosphere. *Physica A (submitted)*. 107

Gallagher, R. G. (1963). *Low-density parity check codes*. MIT Press. 2, 66

Garlaschelli, D. and Loffredo, M. I. (2004). Patterns of link reciprocity in directed networks. *Physical Review Letters*, 93(26):268701–+. 111, 112, 128

Gerstein, G. L. and Mandelbrot, B. (1964). Random walk models for the spike activity of a single neuron. *Biophys J.*, 4:41–68. 12, 17

Girvan, M. and Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826. 112

Goh, K.-I., Eom, Y.-H., Jeong, H., Kahng, B., and Kim, D. (2006). Structure and evolution of online social relationships: Heterogeneity in unrestricted discussions. *Physical Review E*, 73(6):066123. 104, 107, 128, 129, 151

Goldstein, M. L., Morris, S. A., and Yen, G. G. (2004). Problems with fitting to the power-law distribution. *The European Physical Journal B*, 41(2):255–258. 105, 116, 120

Gotlieb, C. G. and Corneil, D. G. (1967). Algorithms for finding a fundamental set of cycles for an undirected linear graph. *Commun. ACM*, 10(12):780–783. 78

Habermas, J. (1991). *The Structural Transformation of the Public Sphere: Inquiry into a Category of Bourgeois Society*. Cambridge, MA: MIT Press. 123

Harder, U. and Paczuski, M. (2006). Correlated dynamics in human printing behavior. *Physica A: Statistical Mechanics and its Applications*, 361(1):329–336. 106

Hastings, M. B. (2006). Community detection as an inference problem. *Physical Review E*, 74(3):035102. 113

Haykin, S., Principe, J. C., Sejnowski, T. J., and McWhirter, J. G. (2007). *New Directions in Statistical Signal Processing: From Systems to Brain*, pages 127–154. MIT Press. 36

Hentall, I. D. (2000). Interactions between brainstem and trigeminal neurons detected by cross-spectral analysis. *Neuroscience*, 96:601–610. 14

Herring, S. C., Kouper, I., Paolillo, J. C., Scheidt, L. A., Tyworth, M., Welsch, P., Wright, E., and Yu, N. (2005). Conversations in the blogosphere: An analysis "from the bottom up". In *Proceedings of HICSS '05*. 107

Herring, S. C., Scheidt, L. A., Bonus, S., and Wright, E. (2004). Bridging the gap: A genre analysis of weblogs. In *Proc. 37th HICSS'04 - Track 4*, page 40101.2, Washington, DC, USA. IEEE Computer Society. 106

Heskes, T., Albers, K., and Kappen, H. J. (2003). Approximate inference and constrained optimization. In *Proceedings of the 19th Annual conference on Uncertainty in Artificial Intelligence (UAI-03)*, pages 313–320, San Francisco, CA. Morgan Kaufmann Publishers. 67, 82

Hines, M. and Carnevale, N. (2000). Discrete event simulation in the neuron environment. *Neurocomputing*, 58:1117–1122. 21

Hinton, G. E. and Sejnowski, T. J. (1983). Optimal perceptual inference. pages 448–453. 9

Hirsch, J. E. (2005). An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences*, 102(46):16569–16572. 142

Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal Of Physiology-London*, 117:500–544. 8

Holme, P., Edling, C. R., and Liljeros, F. (2004). Structure and time evolution of an internet dating community. *Social Networks*, 26(2):155–174. 104, 128

Honkanen, P. A. (1978). Circuit enumeration in an undirected graph. In *ACM-SE 16: Proceedings of the 16th annual Southeast regional conference*, pages 49–53, New York, USA. ACM. 78

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–8. 10

Huberman, B. A. and Adamic, L. A. (1999a). Evolutionary dynamics of the World Wide Web. *Technical report*. Xerox Palo Alto Research Center. 104

Huberman, B. A. and Adamic, L. A. (1999b). Internet: Growth dynamics of the World-Wide Web. *Nature*, 401(6749):131–131. Brief Communication. 104

Ikegaya, Y., Aaron, G., Cossart, R., Aronov, D., Lampl, I., Ferster, D., and Yuste, R. (2004). Synfire chains and cortical songs: Temporal modules of cortical activity. *Science*, 304:559–564. 38

Izhikevich, E. M. (2006). Polychronization: Computation with spikes. *Neural Computation*, 18:245–282. 38

Izhikevich, E. M., Gally, J. A., and Edelman, G. M. (2004). Spike-timing dynamics of neuronal groups. *Cerebral Cortex*, 14:933–944. 38

Jaakkola, T. and Jordan, M. I. (1999). Variational probabilistic inference and the QMR-DT network. *Journal of Artificial Intelligence Research*, 10:291–322. 93

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks. Technical Report gMD Report 148, German National Research Center for Information Technology. 36

Johansen, A. (2004). Probing human response times. *Physica A: Statistical Mechanics and its Applications*, 338(1–2):286–291. 106

Johansen, A. (2005). Comment on Barábasi, nature 435, 207 (2005). 106

Johnson, N. L. and Kotz, S. (1969). *Discrete Distributions*. Wiley. 25

Jordan, M., Ghahramani, Z., Jaakkola, T. S., and Saul, L. (1999). An introduction to variational methods for graphical models. In *Learning in Graphical Models*, pages 105–161. Cambridge, MA: MIT Press. 66

Kaltenbrunner, A., Gómez, V., and López, V. (2007a). Description and prediction of Slashdot activity. In *Proceedings of the 5th Latin American Web Congress (LA-WEB 2007)*, Santiago de Chile. IEEE Computer Society. 137, 153

Kaltenbrunner, A., Gómez, V., and López, V. (2007b). Phase transition and hysteresis in an ensemble of stochastic spiking neurons. *Neural Computation*, 19(11):3011–3050. 32, 38, 60, 61

Kaneko, K. (1990). Clustering, coding, switching, hierarchical ordering, and control in a network of chaotic elements. *Physica D*, 41:137–172. 37

Kapteyn, J. (1903). Skew frequency curves in biology and statistics. *Astronomical Laboratory. Groningen, Noordhoff*. 119

Karinthy, F. (1929). *Minden máskuéppen van (Everything Is the Other Way)*, page 85. Atheneum Press. Láncszemek (Chain-Links). Translated from Hungarian and annotated by Adam Makkai and Enikö Jankó. 104

Kentsis, A. (2006). Correspondence patterns: Mechanisms and models of human dynamics. *Nature*, 441:E5. 106

Kinouchi, O. and Copelli, M. (2006). Optimal dynamical range of excitable networks at criticality. *Nature Physics*, 2:348. 36, 61

Kirk, V. and Stone, E. (1997). Effect of a refractory period on the entrainment of pulse-coupled integrate-and-fire oscillators. *Physics Letters A*, 232:70–76. 37

Kirkpatrick, S., C. D. Gelatt, J., and Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4598):671–680. 2

Kirkpatrick, S. and Selman, B. (1994). Critical Behavior in the Satisfiability of Random Boolean Expressions. *Science*, 264:1297–1301. 2, 36

Kleban, S. D. and Clearwater, S. H. (2003). Hierarchical dynamics, interarrival times, and performance. In *SC '03: Proceedings of the 2003 ACM/IEEE conference on Supercomputing*, page 28, Washington, DC, USA. IEEE Computer Society. 106

Kleinberg, J. (2003). Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397. 106

Kleinberg, J. M., Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. S. (1999). The Web as a graph: Measurements, models and methods. In *Computing and Combinatorics: 5th Annual International Conference, (COCOON '99)*, volume 1627, pages 1–17, Tokyo, Japan. 104

Krapivski, P., Rodgers, G., and Redner, S. (2001). Degree distributions of growing networks. *Physical Review Letters*, 86:5401–5404. 114

Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519. 65, 68

Kumar, R., Novak, J., Raghavan, P., and Tomkins, A. (2003). On the bursty evolution of blogspace. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 568–576, New York, USA. ACM Press. 107

Kumar, R., Novak, J., Raghavan, P., and Tomkins, A. (2004). Structure and evolution of blogspace. *Communications of the ACM*, 47(12):35–39. 106

Kumar, R., Novak, J., and Tomkins, A. (2006). Structure and evolution of online social networks. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 611–617, New York, USA. ACM Press. 104

Kuramoto, Y. (2003). *Chemical Oscillations, Waves, and Turbulence*. Dover Publications. Dover Ed edition. 37

Lakhina, A., Byers, J., Crovella, M. E., and Xie, P. (2003). Sampling biases in IP topology measurements. In *IEEE INFOCOM*, volume 1, pages 332–341. 105

Lampe, C. and Resnick, P. (2004). Slash(dot) and burn: Distributed moderation in a large online conversation space. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 543–550, New York, USA. ACM Press. 123

Langton, C. G. (1990). Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D Nonlinear Phenomena*, 42:12–37. 36

Lapicque, L. (1907). Recherches quantitatives sur l'excitation electrique des nerfs traitee comme une polarisation. *J. Physiol.*, 9:620–635. 8

Lauritzen, S. L. and Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical society. Series B-Methodological*, 50(2):154–227. 67, 82

Lawrence, R. (1988). *Lognormal Distributions: Theory and Applications*. Marcel Dekker, INC., New York. Edwin L. Crow and Kunio Shimizu, editors. 105, 119

Legenstein, R. and Maass, W. (2007). 2007 special issue: Edge of chaos and prediction of computational performance for neural circuit models. *Neural Networks*, 20(3):323–334. 36

Leisink, M. A. R. and Kappen, H. J. (2001). A tighter bound for graphical models. *Neural Computation*, 13(9):2149–2171. 66

Leland, W. E., Taqqu, M. S., Willinger, W., and Wilson, D. V. (1994). On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–15. 106

Levine, M. W. (1991). The distribution of the intervals between neural impulses in the maintained discharges of retinal ganglion cells. *Biological Cybernetics*, 65(6):459–467. 14

Lux, T. and Marchesi, M. (1999). Scaling and criticality in a stochastic multi-agent model of a financial market. *Nature*, 397:498–500. 37

Maass, W., Natschlager, T., and Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560. 36

Malamud, B., Morein, G., and Turcotte, D. (1998). Forest Fires: An Example of Self-Organized Critical Behavior. *Science*, 281(5384):1840–1842. 37

Malioutov, D., Johnson, J., and Willsky, A. (2006). Walk-sums and belief propagation in gaussian graphical models. *Journal of Machine Learning Research*, 7(Oct):2031–2064. 99

Marian, I., Reilly, R. G., and Mackey, D. (2002). Efficient event-driven simulation for spiking neural networks. In *Proceedings of 3rd WSES International Conference on Neural Networks and Applications*, Interlaken, Switzerland. 21

Matsumura, N., Goldberg, D., and Llora, X. (2005a). Mining directed social network from message board. In *Proceedings of the 14th international conference on World Wide Web (WWW '05)*, pages 1092–1093, New York, USA. ACM Press. 107

Matsumura, N., Miura, A., Shibanai, Y., Ohsawa, Y., and Nishida, T. (2005b). The dynamism of 2channel. *AI and Society*, 19(1):84–92. (in Japanese). 107

Matsumura, N., Yukio, O., and Mitsuru, I. (2003). Profiling participants in online-community based on influence diffusion model. *Transactions of the Japanese Society for Artificial Intelligence*, 18:165–172. (in Japanese). 107

Mattia, M. and Giudice, P. D. (2000). Efficient event-driven simulation of large networks of spiking neurons and dynamical synapses. *Neural Comput*, 12(10):2305–2329. 20, 21

McCulloch, W. S. and Pitts, W. H. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133. 7, 8

McEliece, R., MacKay, D. J. C., and Cheng, J. (1998). Turbo decoding as an instance of Pearl's belief propagation algorithm. *Journal on Selected Areas of Communication*, 16(2):140–152. 66

McKeegan, D. E. F. (2002). Spontaneous and odour evoked activity in single avian olfactory bulb neurones. *Brain Research*, 929:48–58(11). 14

Mézard, M., Parisi, G., and Zecchina, R. (2002). Analytic and algorithmic solution of random satisfiability problems. *Science*, 297(5582):812–815. 36, 66

Milgram, S. (1967). The small world problem. *Psychology Today*, pages 62–67. 104

Mirollo, R. and Strogatz, S. (1990). Synchronization of pulse-coupled biological oscillators. *Siam Journal On Applied Mathematics*, 50(6):1645–1662. 37

Mishne, G. and Glance, N. (2006). Leave a reply: An analysis of weblog comments. In *Third Annual Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics (WWW-2006)*, Edumburg, UK. 107, 123, 152

Mitzenmacher, M. (2003). A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 1(2):226–251. 129, 130

Mitzenmacher, M. (2006). Editorial: The future of power law research. *Internet Mathematics*, 2(4):525–534. 106

Montanari, A. and Rizzo, T. (2005). How to compute loop corrections to the Bethe approximation. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(10):P10011. 98

Mooij, J. M. and Kappen, H. J. (2005). On the properties of the Bethe approximation and loopy belief propagation on binary networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(11):P11012. 84

Mooij, J. M. and Kappen, H. J. (2007). Loop corrections for approximate inference on factor graphs. *Journal of Machine Learning Research*, 8(May):1113–1143. 94, 95, 98

Mooij, J. M., Wemmenhove, B., Kappen, H. J., and Rizzo, T. (2007). Loop corrected belief propagation. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS-07)*. 98

Morrison, A., Mehring, C., Geisel, T., Aertsen, A., and Diesmann, M. (2005). Advancing the boundaries of high-connectivity network simulation with distributed computing. *Neural Comput.*, 17:1776–1801. 21, 33

Muniruzzaman, A. N. M. (1957). The pricing of options and corporate liabilities. *Butlletin of the Calcutta Statistical Association*, 115(7). 116, 117

Munroe, R. (2008). http://xkcd.com/386/. 151

Murphy, K. P., Weiss, Y., and Jordan, M. I. (1999). Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 467–475, San Francisco, CA. Morgan Kaufmann Publishers. 66

Naruse, K. and Kubo, M. (2006). Lognormal distribution of BBS articles and its social and generative mechanism. In *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 103–112, Washington, DC, USA. IEEE Computer Society. 107

Newman, M. E. (2001). Scientific collaboration networks. I. Network construction and fundamental results. *Physical Review E*, 64(1):016131. 104

Newman, M. E. J. (2002). Assortative mixing in networks. *Physical Review Letters*, 89(20):208701. 111

Newman, M. E. J. (2003a). Mixing patterns in networks. *Physical Review E*, 67(2):026126. 110

Newman, M. E. J. (2003b). The structure and function of complex networks. *SIAM Review*, 45(2):167–256. 2, 108, 127, 135

Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6):066133. 112

Newman, M. E. J. (2005). Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, 46:323–351. 116

Newman, M. E. J., Forrest, S., and Balthrop, J. (2002). Email networks and the spread of computer viruses. *Physical Review E*, 66(3):035101. 104

Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113. 112

Newman, M. E. J. and Park, J. (2003). Why social networks are different from other types of networks. *Physical Review E*, 68(3):036122. 128

Östborn, P. (2002). Phase transition to frequency entrainment in a long chain of pulse-coupled oscillators. *Physical Review E*, 66:016105. 37

Östborn, P., Åberg, S., and Ohlén, G. (2003). Phase transitions towards frequency entrainment in large oscillator lattices. *Physical Review E*, 68:015004. 37

Packard, N. H. (1988). *Adaptation toward the edge of chaos. In: Dynamics Patterns in Complex Systems*, pages 293–301. World Scientific: Singapore. A. J. Mandell, J. A. S. Kelso, and M. F. Shlesinger, editors. 36

Parisi, J. and Slanina, F. (2006). Loop expansion around the Bethe-Peierls approximation for lattice models. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(02):L02003. 98

Paxson, V. and Floyd, S. (1994). Wide-area traffic: the failure of poisson modeling. In *SIGCOMM '94*, pages 257–268, New York, USA. ACM Press. 106

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, San Francisco, CA. 2, 65, 66, 68

Peierls, R. E. (1936). On Isings model of ferromagnetism. *Proc. Cambridge Philos. Soci.*, 32(477). 2

Pelizzola, A. (2005). Cluster variation method in statistical physics and probabilistic graphical models. *Journal of Physics A: Mathematical and General*, 38(33):R309–R339(1). 67

Pennock, D. M., Flake, G. W., Lawrence, S., Glover, E. J., and Giles, C. L. (2002). Winners don't take all: Characterizing the competition for links on the web. *Proceedings Of The National Academy Of Sciences*, 99(8):5207–5211. 105, 115

Poor, N. (2005). Mechanisms of an online public sphere: The web site Slashdot. *Journal of Computer-Mediated Communication*, 10(2). 123

Potamianos, G. and Goutsias, J. (1997). Stochastic approximation algorithms for partition function estimation of Gibbs random fields. *IEEE Trans. on Inf. Theory*, 43(6):1948–1965. 66

Pratt, G. A. (1989). *Pulse Computation*. PhD thesis, MIT Massachusetts Institute of Technology. 20

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1992). *Numerical Recipes: The Art of Scientific Computing*, chapter 14.3, page 624. Cambridge University Press, Cambridge (UK) and New York, 2nd edition. 120

Price, D. D. S. (1976). A general theory of bibliometric and other commutative advantage processes. *Journal of the American Society for Information Science*, 27(292):292–306. 114, 115

Rapoport, A. (1957). Contribution to the theory of random and biased nets. *Bull. Math. Biophys.*, 19:257–277. 108

Reutimann, J., Giugliano, M., and Fusi, S. (2003). Event-driven simulation of spiking neurons with stochastic dynamics. *Neural Comput*, 15:811–830. 20, 21, 22

Rieke, F., Warland, D., van Steveninck, R. d. R., and Bialek, W. (1997). *Spikes: Exploring The Neural Code*. The MIT Press, London, England. 22

Rochel, O. and Martinez, D. (2003). An event-driven framework for the simulation of networks of spiking neurons. *ESANN'2003 proceedings*, pages 295–300. 19

Rodríguez, F. B., Suárez, A., and López, V. (2002). A discrete model of neural ensembles. *Philosophical Transactions: Mathematical, Physical & Engineering Sciences*, 360(1792):559–573. 17, 32, 39

Rodríguez, F. B., Suárez, A. B., and López, V. B. (2001). Period focusing induced by network feedback in populations of noisy integrate-and-fire neurons. *Neural Computation*, 13(11):2495–2516. 17, 18, 38, 40, 60

Ros, E., Carrillo, R., Ortigosa, E. M., Barbour, B., and Agís, R. (2006). Event-driven simulation scheme for spiking neural networks using lookup tables to characterize neuronal dynamics. *Neural Computation*, 18(12):2959–2993. 21

Rosenblatt, F. (1962). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books. Washington DC. 9

Rudolph, M. and Destexhe, A. (2006). Analytical integrate-and-fire neuron models with conductance-based dynamics for event-driven simulation strategies. *Neural Computation*, 18(9):2146–2210. 21

Sack, W. (2000). Discourse diagrams: Interface design for very large-scale conversations. In *Proc. HICSS '00. Volume 3*, page 3034, Washington, DC, USA. IEEE CS. 107

Schnegg, M. (2006). Reciprocity and the emergence of power laws in social networks. *International Journal of Modern Physics C*, 17:1067–1076. 105

Selman, B., Levesque, H., and Mitchell, D. (1992). Hard and easy distributions of sat problems. In *Proc. of AAAI-92*, pages 459–465, San Jose, CA. 2, 36

Senn, W. and Urbanczik, R. (2000). Similar nonleaky integrate-and-fire neurons with instantaneous couplings always synchronize. *SIAM Journal On Applied Mathematics*, 61:1143–1155. 37, 44

Serrano, M. Á. and Boguñá, M. (2003). Topology of the world trade web. *Phisical Review E*, 68(1):015101. 104

Shwe, M. A., Middleton, B., Heckerman, D. E., Henrion, M., Horvitz, E. J., Lehman, H. P., and Cooper, G. F. (1991). Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base. I. The probabilistic model and inference algorithms. *Methods of Information in Medicine*, 30(4):241–55. 93

Sidiropoulos, A. and Manolopoulos, Y. (2006). Generalized comparison of graph-based ranking algorithms for publications and authors. *Journal of Systems and Software*, 79(12):1679–1700. 143

Siganos, G., Faloutsos, M., Faloutsos, P., and Faloutsos, C. (2003). Power laws and the AS-level internet topology. *IEEE/ACM Transactions on Networking*, 11(4):514–524. 105

Simon, H. A. (1955). On a class of skew distribution functions. *Biometrika*, 42(3/4):425–440. Reprinted in: Simon, H., 1957. Models of Man. 104, 114

Smith, M. (2002). Tools for navigating large social cyberspaces. *Commun. ACM*, 45(4):51–55. 107

Solomonoff, R. and Rapoport, A. (1951). Connectivity on random nets. *Bull. Math. Biophys.*, 13:107–117. 108

Song, S., Miller, K. D., and Abbott, L. F. (2000). Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, 3(9):919–926. 55

Stein, R. B. (1967). Some models of neuronal variability. *Biophysical Journal*, 7:37–68. 13, 33

Stirzaker, D. (2005). *Stochastic Processes and Models*, chapter 5.2, page 220. Oxford University Press. 12

Stouffer, D. B., Malmgren, R. D., and Amaral, L. A. N. (2005). Comment on barábasi, nature 435, 207 (2005). 106

Stouffer, D. B., Malmgren, R. D., and Amaral, L. A. N. (2006). Log-normal statistics in e-mail communication patterns. http://arxiv.org/abs/physics/0605027. 106, 148, 152

Strang, J. E. and Ostborn, P. (2005). Wave patterns in frequency-entrained oscillator lattices - art. no. 056137. *Physical Review E*, 7205:6137–6137. 33

Sun, J., Li, Y., Kang, S. B., and Shum, H. Y. (2005). Symmetric stereo matching for occlusion handling. In *Proceedings of the 2005 conf. Computer Vision and Pattern Recognition (CVPR-05)*, volume 2, pages 399–406, Washington, DC, USA. IEEE Computer Society. 66

Takinawa, M. and D'Ambrosio, B. (1999). Multiplicative factorization of noisy-MAX. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 622–30, San Francisco, CA. Morgan Kaufmann. 93

Thelwall, M. and Wilkinson, D. (2003). Graph structure in three national academic webs: power laws with anomalies. *Journal of the Americal Society for Information Science and Technology*, 54(8):706–712. 104

Timme, M., Wolf, F., and Geisel, T. (2002). Prevalence of unstable attractors in networks of pulse-coupled oscillators. *Physical Review Letters*, 89(15):154105. 38

Tuckwell, H. C. (1988). *Introduction to Theoretical Neurobiology. Volume 2: Nonlinear and Stochastic Theories*. Cambridge University Press, Cambridge. 11, 13, 14, 26, 33

Uhlenbeck, G. E. and Ornstein, L. S. (1930). On the theory of the brownian motion. *Phys. Rev.*, 36:823–841. Reprinted in Noise and Stochastic Processes (Ed. N. Wax). New York: Dover, pp. 93-111, 1954. 12

Vázquez, A., Oliveira, J. G., Dezsö, Z., Goh, K. I., Kondor, I., and Barabási, A.-L. (2006). Modeling bursts and heavy tails in human dynamics. *Physical Review E*, 73:036127. 106, 152

Vreeswijk, C. V. (1996). Partial synchronization in populations of pulse-coupled oscillators. *Physical Review E*, 54:5522–5537. 38

Vreeswijk, C. V. and Abbott, L. F. (1993). Self-sustained firing in populations of integrate-and-fire neurons. *SIAM J. Appl. Math.*, 53(1):253–264. 17

Wainwright, M., Jaakkola, T., and Willsky, A. (2005). A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51(7):2313–2335. 66

Wang, X. J. (2001). Synaptic reverberation underlying mnemonic persistent activity. *Trends In Neurosciences*, 24(1):455–463. 44

Wasserman, S. and Faust, K. (1994). *Social network analysis*. Cambridge University Press, Cambridge. 112

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of small-world networks. *Nature*, 393(6684):440–442. 104, 109, 113, 128

Watts, L. (1994). Event-driven simulation of networks of spiking neurons. *Advances in neural information processing systems*, 6:927–934. San Mateo, CA. Morgan Kauffman Publishers. 20

Welling, M., Minka, T., and Teh, Y. W. (2005). Structured region graphs: Morphing EP into GBP. In *Proceedings of the 21th Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, page 609, Arlington, Virginia. AUAI Press. 67

Wiegerinck, W., Kappen, H. J., ter Braak, E. W. M. T., Burg, W. J. P. P., Nijman, M. J., O, Y. L., and Neijt, J. P. (1999). Approximate inference for medical diagnosis. *Pattern Recognition Letters*, 20(11):1231–1239(9). 68

Willinger, W., Taqqu, M. S., Sherman, R., and Wilson, D. V. (1997). Self-similarity through high-variability: statistical analysis of ethernet lan traffic at the source level. *IEEE/ACM Trans. Netw.*, 5(1):71–86. 106

Winfree, A. (1967). Biological rhythms and the behavior of populations of coupled oscillators. *J. Theoret. Biol.*, 16:15–42. 37

Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2000). Generalized belief propagation. In Leen, T., Dietterich, T., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13 (NIPS-00)*, pages 689–695. 69

Yedidia, J. S., Freeman, W. T., Weiss, Y., and Yuille, A. L. (2005). Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312. 2, 66, 67, 69

Yook, S. H., Jeong, H., Barabási, A.-L., and Tu, Y. (2001). Weighted evolving networks. *Physical Review Letters*, 86(25):5835–5838. 105

Yuille, A. L. (2002). CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14(7):1691–1722. 67

Yule, G. U. (1925). A Mathematical Theory of Evolution, Based on the Conclusions of Dr. J. C. Willis, F.R.S. *Royal Society of London Philosophical Transactions Series B*, 213:21–87. 104, 114

Zhongbao, K. and Changshui, Z. (2003). Reply networks on a bulletin board system. *Physical Review E*, 67(3):036117. 151

Zilberstein, S. (1996). Using anytime algorithms in intelligent systems. *AI magazine*, 17(3):73–83 (1p.1/4). 98