

Lecture Notes in Mathematics 2141

CIME Foundation Subseries

P.R. Kumar · Martin J. Wainwright
Riccardo Zecchina

Mathematical Foundations of Complex Networked Information Systems

Politecnico di Torino, Verrès, Italy 2009

Fabio Fagnani · Sophie M. Fosson
Chiara Ravazzi *Editors*



FONDAZIONE
CIME
ROBERTO CONTI



Springer

Lecture Notes in Mathematics

2141

Editors-in-Chief:

J.-M. Morel, Cachan

B. Teissier, Paris

Advisory Board:

Camillo De Lellis, Zurich

Mario di Bernardo, Bristol

Alessio Figalli, Austin

Davar Khoshnevisan, Salt Lake City

Ioannis Kontoyiannis, Athens

Gabor Lugosi, Barcelona

Mark Podolskij, Aarhus

Sylvia Serfaty, Paris and NY

Catharina Stroppel, Bonn

Anna Wienhard, Heidelberg

More information about this series at

<http://www.springer.com/series/304>



**FONDAZIONE
CIME**
ROBERTO CONTI

CENTRO INTERNAZIONALE MATEMATICO ESTIVO
INTERNATIONAL MATHEMATICAL SUMMER CENTER

Fondazione C.I.M.E., Firenze

C.I.M.E. stands for *Centro Internazionale Matematico Estivo*, that is, International Mathematical Summer Centre. Conceived in the early fifties, it was born in 1954 in Florence, Italy, and welcomed by the world mathematical community: it continues successfully, year for year, to this day.

Many mathematicians from all over the world have been involved in a way or another in C.I.M.E.'s activities over the years. The main purpose and mode of functioning of the Centre may be summarised as follows: every year, during the summer, sessions on different themes from pure and applied mathematics are offered by application to mathematicians from all countries. A Session is generally based on three or four main courses given by specialists of international renown, plus a certain number of seminars, and is held in an attractive rural location in Italy.

The aim of a C.I.M.E. session is to bring to the attention of younger researchers the origins, development, and perspectives of some very active branch of mathematical research. The topics of the courses are generally of international resonance. The full immersion atmosphere of the courses and the daily exchange among participants are thus an initiation to international collaboration in mathematical research.

C.I.M.E. Director

Elvira MASCOLO

Dipartimento di Matematica "U. Dini"

Università di Firenze

viale G.B. Morgagni 67/A

50134 Florence

Italy

e-mail: mascolo@math.unifi.it

C.I.M.E. Secretary

Paolo SALANI

Dipartimento di Matematica "U. Dini"

Università di Firenze

viale G.B. Morgagni 67/A

50134 Florence

Italy

e-mail: salani@math.unifi.it

For more information see CIME's homepage: <http://www.cime.unifi.it>

CIME activity is carried out with the collaboration and financial support of:

- INdAM (Istituto Nazionale di Alta Matematica)
- MIUR (Ministero dell'Università e della Ricerca)

P.R. Kumar • Martin J. Wainwright •
Riccardo Zecchina

Mathematical Foundations of Complex Networked Information Systems

Politecnico di Torino, Verrès, Italy 2009

Fabio Fagnani, Sophie M. Fosson, Chiara Ravazzi
Editors

 Springer



FONDAZIONE
CIME
ROBERTO CONTI

CENTRO INTERNAZIONALE MATEMATICO ESTIVO
INTERNATIONAL MATHEMATICAL SUMMER CENTER

Authors

P.R. Kumar
CSL and Department of ECE
University of Illinois
Urbana
Illinois, USA

Martin J. Wainwright
Department of Statistics
UC Berkeley
Berkeley
California, USA

Riccardo Zecchina
DISAT
Politecnico di Torino
Torino, Italy

Editors

Fabio Fagnani
DISMA
Politecnico di Torino
Torino, Italy

Sophie M. Fosson
DET
Politecnico di Torino
Torino, Italy

Chiara Ravazzi
DET
Politecnico di Torino
Torino, Italy

ISSN 0075-8434

ISSN 1617-9692 (electronic)

Lecture Notes in Mathematics

ISBN 978-3-319-16966-8

ISBN 978-3-319-16967-5 (eBook)

DOI 10.1007/978-3-319-16967-5

Library of Congress Control Number: 2015940535

Mathematics Subject Classification (2010): 05C80 Graph Theory - Random graph, 05C85 Graph Theory - Graph Algorithms, 05C90 Graph Theory - Applications

Springer Cham Heidelberg New York Dordrecht London

© Springer International Publishing Switzerland 2015

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

In the past 20 years, we have been witnessing a revolution in information and communication technologies, consisting in an astonishing progress in different research fields. Innovations have been developed not only in the classical fields of wireless communication (design of high performance codes very close to the Shannon limit), computer engineering (parallel computing, computer vision, data mining), bioinformatics (genome interpretation), and VLSI technologies, but also in new areas such as wireless sensor networks and smart dust, air traffic control, large scale surveillance, unmanned mobile multi-vehicle systems, biological networks interpretation, and so on.

All of these problems deal with complex systems and large scale optimization, and require a prohibitive amount of computational effort even in low dimensions. The complexity calls for robust and adaptable solutions. In spite of the inherent difficulties, fundamental contributions have already appeared, based on the principle that most of complex systems are the result of the interactions of numerous, but rather simple entities. The complexity is then a consequence of the interactions architecture, that can be described by a network, and solutions can be developed from mathematical models that describe how the local rules originate the global behavior.

The mathematical tools useful to address these new scientific issues come mainly from combinatorics, probability theory, and statistical mechanics. The concept of graph plays a prominent role: not only it is the natural model for any communication network, but also it describes interactions among variables, as exploited, e.g., in graphical models in modern coding theory.

In particular, random graphs are a suitable model to describe complex networks (e.g., internet, wireless communication networks, sensor networks). Randomness enters at different levels, to mimic the development of complex networks, to model faults or noise in the communication links, or also to find typical codes or algorithms with good properties. For a deep analysis of such random models, probabilistic techniques are essential, among which we mention concentration inequalities, large deviation theory, and percolation. In case of aggregation of simple entities, where

complexity rises from the network of interactions, statistical mechanics is a natural approach.

The summer school “Mathematical Foundations of Complex Networked Information Systems” was organized by CIME, with the contribution of Newcom++, in Verrès (Aosta Valley, Italy) in 2009. The aim of the school was to give an introduction to some of the fundamental scientific issues emerging in these disciplines. It consisted of four courses, each of 6 h.

The course of Prof. Béla Bollobás focused on random graphs: it provided all the necessary probabilistic tools and proposed a number of different ways to model randomness.

The course of Prof. P.R. Kumar focused on communication networks, in particular on the issues of the transmission of information along a network where there are simultaneously many potential receivers and many potential transmitters. Prof. Kumar presented the basic information theoretic aspects of wireless communication networks.

The other two courses addressed the analysis of distributed algorithms over networks and, more generally, over graphical models. Particular emphasis was given to the famous message passing algorithms which have applications in a broad variety of contexts including artificial intelligence, distributed inferential statistics, coding theory, and combinatorial optimization. The course of Prof. Riccardo Zecchina described the statistical physics interpretation of these algorithms with special attention to applications in classical combinatorial optimization problems. The course by Prof. Martin J. Wainwright instead focused on the probabilistic aspects of message passing algorithms and their relation with the relaxation techniques in optimization theory.

About 70 people from all over the world attended the school. Among them, there were senior graduate students and post-doc researchers in pure and applied mathematics, information engineering, and physics.

The overall aim of this book is to record the results presented during the summer school.

Torino, Italy

Torino, Italy

Torino, Italy

Fabio Fagnani

Sophie M. Fosson

Chiara Ravazzi

Contents

Some Introductory Notes on Random Graphs	1
Fabio Fagnani, Sophie M. Fosson, and Chiara Ravazzi	
Statistical Physics and Network Optimization Problems	27
Carlo Baldassi, Alfredo Braunstein, Abolfazl Ramezanzpour, and Riccardo Zecchina	
Graphical Models and Message-Passing Algorithms: Some Introductory Lectures	51
Martin J. Wainwright	
Bridging the Gap Between Information Theory and Wireless Networking	109
P.R. Kumar	

Some Introductory Notes on Random Graphs

Fabio Fagnani, Sophie M. Fosson, and Chiara Ravazzi

1 Introduction

In the last decades, the words *complexity* and *networks* have constantly increased their presence in the scientific literature and have become popular even in our everyday life. Born in different contexts, nowadays they are often associated to characterize the so-called complex networked systems, that are of outstanding importance in our society and range from natural phenomena to technological models. A common example is the internet network, today accessed by one third of the world population and in continuous expansion. While most of the internet users have no deep insight of its architecture, everyone has an idea of its networked nature and has made experience of its complexity, which affects its use in terms of bottlenecks and communication delays.

While giving a definition of complex networked system is tricky, it is easy to provide examples, as they are involved in a number of different fields such as physics, computer science, economy, finance, biology, epidemiology, sociology. The common point of such systems is the presence of a large number of objects (individuals, sensors, computers, cells, molecules) that interact in a way that presents strong irregularities, e.g., following patterns that vary in time, sometimes randomly, with unclear modifications of roles and clustering behaviors. It is worth

F. Fagnani (✉)

Department of Mathematical Sciences (DISMA), Politecnico di Torino, Corso Duca degli
Abruzzi 24, 10129 Torino, Italy

e-mail: fabio.fagnani@polito.it

S.M. Fosson • C. Ravazzi

Department of Electronics and Telecommunications (DET), Politecnico di Torino, Corso Duca
degli Abruzzi 24, 10129 Torino, Italy

e-mail: sophie.fosson@polito.it; chiara.ravazzi@polito.it

© Springer International Publishing Switzerland 2015

F. Fagnani et al. (eds.), *Mathematical Foundations of Complex Networked
Information Systems*, Lecture Notes in Mathematics 2141,

DOI 10.1007/978-3-319-16967-5_1

noting that the dimension of the network (that is, the number of interacting components) does not automatically determine the complexity: if they present good scalability properties, large networks may be simple. Complexity arises because of the topology of the interconnections which enhance rapid diffusion of information and because of the non-linear and probabilistic nature of the interaction laws guiding the dynamics over the network. In these cases, the global behavior of the network shows complexity features which by no means can be seen as the addition of the many individual behaviors. Often, such systems can exhibit transition phase phenomena with drastic change in their functional modes when even small and localized perturbations take place. The first seminal work on complexity has been done by physicists: from gas to chaos theory, they have studied the large scale consequences of the interactions among a number of small components, which assumes a networked structure; statistical mechanics is clearly an appropriate language to describe such large scale networks as well the connection between the micro and macro observation scales. Analogously, biologists, economists, anthropologists have analyzed existing systems of their interest considering the complexity generated by connectivity among organisms or individuals. Complex infrastructural networks have also attracted the attention of engineers and computer scientists since a while. The digital revolution has created many new applications: besides the world wide web, there are the wireless cellular networks, and the more futurist sensor networks. Other remarkable examples are the electrical networks including the recent smart grids and transportation networks. In most of these uses, artificial networks have grown in time so much that a fully faithful representation of them is not available. Often these infrastructure networks have been optimally designed for the business-as-usual behavior and they present high fragility to small perturbations which may lead them to unpredictable and unstable dynamics (e.g., cascading failures, blackouts, crashes).

Whatever perspective is considered, it has become fundamental to understand and efficiently describe complex networked systems, as well to establish new paradigms for their design and control.

This series of lectures is aimed at introducing the reader to the mathematical theory of complex networked systems. A few fundamental issues are proposed, that show how research in this field necessarily has to cope with a double difficulty: the problem of constructing coherent mathematical models sufficiently rich to be able to describe the existent complex networks and, on the other side, simple enough to be amenable for a general mathematical theory. While elegant and complete mathematical results already exist for certain specific problems, it is true that the mathematical theory of complex networks is far from being complete and many outstanding problems still wait for an answer.

The required background is limited to undergraduate-level calculus, linear algebra, probability theory, information theory and combinatorics. As the concept of graph plays a fundamental role in every aspect of complex networks, in this first section, we will provide a brief introduction to graph theory with particular focus on models of random graphs for their role of furnishing the simplest and more effective way to exhibit large scale graphs having properties similar to the real

life networks (like for instance the world wide web). After recalling basic notation and concepts on graphs, we will consider three different models of random graphs for their crucial role in the analysis and simulation of large scale networks. We will make an attempt to present both rigorous mathematical results and applications examples, without much details, but with precise references to retrieve them. We will restrict our analysis to specific aspects that are of much interest for us and for the purpose of this book, in particular we will analyze the asymptotic behavior of random graphs when the size tends to infinity, in terms of connectivity, existence of giant components, distances, and local topological structure.

The lecture notes “Statistical physics and network optimization problems” by Carlo Baldassi, Alfredo Braunstein, Abolfazl Ramezanzpour, and Riccardo Zecchina contain a statistical mechanics approach to analyze phase transitions and optimization problems over complex random structures. First, they show the classical Potts model in theoretical physics can be of use to analyze percolation (e.g. the structure of connected components) in random graphs. Afterwards, they show how more complex combinatorial optimization problems can be studied and solved by using more advanced statistical mechanics techniques like the cavity method and the message passing algorithms. The presence of phase transitions in such problems is a key feature which is deeply analyzed in these notes.

Message passing algorithms also constitute the core of “Graphical models and message-passing algorithms: some introductory lectures” by Martin J. Wainwright, which discusses how graphical models provide an almost universal framework for describing statistical dependence in collections of random variables. At their core lie various correspondences between the conditional independence properties of a random vector, and the structure of an underlying graph used to represent its distribution. Message passing algorithms are introduced and analyzed in detail in their standard form (sum-product and max-product) to solve global computation tasks over graphical models and, in particular, in their application to solve statistical inferential problems (e.g., maximum likelihood).

Finally, the lecture notes “Bridging the Gap Between Information Theory and Wireless Networking” by P.R. Kumar focus on a very important applicative example: wireless networks. They provide an account of an approach to studying wireless networks with arbitrary numbers of nodes focusing on the problem of characterizing their capacity, namely the amount of information that can be globally simultaneously transmitted among the various nodes. The chapter begins with a primer on Shannon’s information theory for point-to-point communication, as well as an account of some networks for which the capacity has been characterized. Wireless networks are first modeled on the basis of the current radio technology, providing a framework where capacity can be defined and bound. A key feature of this approach is that it models the physical context of space and distance which are very relevant to wireless networks. Next, challenging the technological model, one attempts to obtain a more definitive characterization of capacity that is not beholden to a particular technology. In this new context nodes are imagined to be embedded in space and attenuation is a function of distance. Performance are analyzed taking into account data rates and distances over which the data are carried,

obtaining a fundamental connection between the capacity of wireless networks and the attenuation properties of the medium.

2 Generalities on Graphs

The mathematical model of a network is the *graph*, which has been studied by many remarkable pure mathematicians in the last decades. Its definition is incredibly simple: a graph is a pair of sets $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where \mathcal{E} is a subset of \mathcal{V}^2 [3]. Typically, \mathcal{V} is known as the set of *nodes* and \mathcal{E} as the set of *edges* that join some nodes. This is the basis for graph theory, which today envisages different branches and has reached deep results which, besides being important in complex networks, have played a pivotal role in other areas of mathematics such as combinatorics, algebra, and topology.

The purpose of these introductory notes is to help the reader to understand what are the objects of study, the phenomena, and the mathematical tools addressed by *random graph theory*. Roughly speaking, a graph is said to be random when its properties are determined by probabilistic rules. In the following, we will present three outstanding models (Erdős and Rényi model, configuration model, and random geometric graphs) that will clarify this concept.

Before presenting the models, we point out that in the theoretical analysis of the statistical aspects of graphs two main methodological lines can be distinguished. The first one concerns the estimation of the proportion of graphs having a certain property. This approach is deterministic and uses the theory of combinatorial enumeration (Polya's enumeration theorems, generating functions, differential operators) in order to obtain exact formulae and to investigate asymptotic behavior.

The second methodological line, which is the one studied in these notes, is nowadays more relevant. It was introduced at the beginning of the 1960s by Erdős and Rényi in their pioneering works [6–9]. Its novelty consists in the use of the *probabilistic method* to prove structural properties, or the existence of a graph with certain properties; in other terms, random graphs are used to prove deterministic properties of the graphs. A probability space, called graph ensemble, is constructed and then it is shown that a randomly chosen graph from such space (named typical graph) satisfies the desired properties with high probability. As explained in the preface of the prominent book of Bollobás [3], the idea of using probability to tackle deterministic problems has a long history, and examples can be found throughout the entire past century in mathematical analysis, combinatorics, number theory, and geometry. Extending the above mentioned procedure, in general the probabilistic method aims at proving that “a structure with certain desired properties exists through the definition of a suitable probability space of structures and then by showing that the desired properties hold in this space with positive probability”, as explained in the preface of the book by Alon and Spencer [1] dedicated to the argument.

The surprising fact is that “the probabilistic ideas proved to be so important in the study of such simple finite structure as graph” [3]. But as Bollobás noticed in his preface, this is likely only the beginning of the diffusion of probabilistic methods to solve more and more complicated problems far beyond graph theory.

Since the seminal works of Erdős and Rényi, the literature on random graph theory has been grown exponentially, due also to the spread of networked social and technological systems in everyday life and to the novel interest for networked systems present in nature. Different communities are involved in this expansion: if mathematicians have been attracted by random graph theory from the beginning, the interest from physicists and biologists has grown due the implications in their research fields. This has originated a heterogeneous amount of works, where the more applications’ oriented are from the physics community, and the more rigorous from mathematics. Providing an exhaustive bibliography is pretty much an impossible mission: even just making a list of all the survey books on the argument is a huge work. For the purpose of these notes, we suggest the reader to refer to the already mentioned book of Bollobás [3], which contains much information and references. A second, more recent reference is the book by Durrett [5], while for more tutorial reading we recommend the lecture notes by van der Hofstad [20]. This is enough to have a good overview on the problem from a rigorous mathematical viewpoint, while specific references for the single models will be given in the next.

The relationship between random graph models and real complex networks is analyzed in a very readable survey by Newman [15], which explains how the research on graphs has changed in the last years due to explosion of large networked systems in real applications. Networks with millions of nodes and links are now the object of interest, which drives the study on graphs to new questions, in particular regarding the behavior of graphs when their size tends to infinity. Moreover, a number of different properties are now taken in consideration like the presence of giant components, clustering, resilience, small-world phenomena. In [15] a wide overview on the most important types of networks is proposed, with distinction between social, information, technological and biological networks. Afterwards, some models are analyzed and their mathematical properties are reviewed, in order to understand how well such models can predict the real complex networks behaviors and which gaps are still present that do not allow to describe some observed phenomena. A bibliography of some hundreds of references is finally provided.

The models presented in the introductory notes will be described through rigorous mathematical results. We will skip most of the details, but we will supply precise bibliographic references for each topic. The aspects we will analyze are those of much interest for the purpose of this book, specifically, the asymptotic behavior of graphs with size tending to infinity, in terms of connectivity and existence of giant components. Generalities on these concepts are given in the next section, which also retrieves the necessary definitions and notations about graphs. Afterwards, we will introduce large scale networks and finally describe the three above mentioned random graph models.

2.1 Basic Definitions and Notation

We now provide some vocabulary and fundamental notations on graphs.

As already sketched, a graph is a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, \dots, n\}$ is the set of nodes (or vertices) of \mathcal{G} , and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges (or arcs), whose elements are oriented pairs (v, w) representing the connecting links between two nodes ($v \in \mathcal{V}$ and $w \in \mathcal{V}$). The cardinality of the vertex set $|\mathcal{V}| = n$ is called the *order* of the graph. A graph can be pictorially described by drawing nodes as dots and edges as arrows: if $(v, w) \in \mathcal{E}$, we connect nodes v and w in such a way v is the tail and w is the head of the arrow. A graph \mathcal{G} is said *undirected* if when $(v, w) \in \mathcal{E}$, then also $(w, v) \in \mathcal{E}$. We say that a graph is *simple* if no self-loop is present, namely if $(v, v) \notin \mathcal{E}$ for all $v \in \mathcal{V}$. We define the in-degree and out-degree as $d_v^{\text{in}} = |\{w \in \mathcal{V} : (w, v) \in \mathcal{E}\}|$ and $d_v^{\text{out}} = |\{w \in \mathcal{V} : (v, w) \in \mathcal{E}\}|$, respectively. If \mathcal{G} is undirected then $d_v^{\text{in}} = d_v^{\text{out}} = d_v$. A graph \mathcal{G} is said *regular* if d_v is constant.

Given two graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $\mathcal{H} = (\mathcal{W}, \mathcal{F})$, they are said to be *isomorphic* if there exists a bijection $\varphi : \mathcal{V} \rightarrow \mathcal{W}$ such that $(v, w) \in \mathcal{E} \iff (\varphi(v), \varphi(w)) \in \mathcal{F}$ and we write $\mathcal{G} \simeq \mathcal{H}$. Two isomorphic graphs essentially simply differ by a different labeling of the vertices. Throughout this book we shall consider two isomorphic graphs as identical.

We define the intersection and the union of two graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $\mathcal{H} = (\mathcal{W}, \mathcal{F})$ as the graphs obtained by $\mathcal{G} \cup \mathcal{H} = (\mathcal{V} \cup \mathcal{W}, \mathcal{E} \cup \mathcal{F})$ and $\mathcal{G} \cap \mathcal{H} = (\mathcal{V} \cap \mathcal{W}, \mathcal{E} \cap \mathcal{F})$. On the other hand, we say that a graph $\mathcal{H} = (\mathcal{W}, \mathcal{F})$ is a *subgraph* of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ ($\mathcal{H} \subseteq \mathcal{G}$) if $\mathcal{W} \subseteq \mathcal{V}$ and $\mathcal{F} \subseteq \mathcal{E}$. Furthermore, given the subset $\mathcal{W} \subseteq \mathcal{V}$ the graph $\mathcal{G}(\mathcal{W}) = (\mathcal{W}, \mathcal{E} \cap \mathcal{W} \times \mathcal{W})$ is said to be the *\mathcal{W} -induced subgraph*.

A *path* in the graph \mathcal{G} is a sequence of vertices $\omega = (v_1, \dots, v_\ell)$ such that $(v_1, v_2) \in \mathcal{E}$, $(v_2, v_3) \in \mathcal{E}$, \dots , $(v_{\ell-1}, v_\ell) \in \mathcal{E}$. The path ω is said to connect v_1 to v_ℓ and ℓ is the length of the path. A graph \mathcal{G} is said to be *strongly connected* if $\forall (v, w) \in \mathcal{V} \times \mathcal{V}$, there exists a ω -path in the graph starting in v and ending in w . If the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is not strongly connected, it is always possible to split the graph into its strongly connected components. We can consider the partition of $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2 \cup \dots \cup \mathcal{V}_m$ such that if we define $\mathcal{E}_j = \mathcal{E} \cap \mathcal{V}_j \times \mathcal{V}_j$, the graphs $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ are called the *connected components* of \mathcal{G} .

It is possible to introduce a distance on the vertices of a graph considering minimal length paths. Precisely, given two vertices u and v in a strongly connected graph \mathcal{G} , we put $d(u, v) = 0$ if $u = v$ and equal to the length of the minimal length path connecting u to v in \mathcal{G} . Finally, the *diameter* of \mathcal{G} is defined as

$$\text{diam}(\mathcal{G}) := \max_{u, v} d(u, v).$$

A cycle in a graph \mathcal{G} is a path $\omega = (v_1, \dots, v_\ell)$ where $v_1 = v_\ell$, $\ell \geq 3$ and all nodes are distinct. An undirected graph \mathcal{G} is called *acyclic* (or *tree*) if it does not possess cycles.

A graph property \mathcal{Q} is called *monotone increasing* if when \mathcal{G} satisfies \mathcal{Q} and $\mathcal{G} \subset \mathcal{H}$ then also \mathcal{H} satisfies \mathcal{Q} . In other terms, property \mathcal{Q} is maintained if we add edges in the graph \mathcal{G} . Typical examples of monotone increasing properties are the strong connectivity or the existence of a specified subgraph in the graph. An example of a property which is not monotone increasing is acyclicity.

In the next we will use the symbol \mathbb{N} and \mathbb{N}_0 to name, respectively, the set of positive and non-negative integers. Moreover, we will use \xrightarrow{p} to indicate the convergence in probability.

2.2 Large Scale Networks

In these notes we will focus on large size graphs, motivated by the plethora of applications that envisage large networks. Nowadays, on one hand there is interest in considering large scale infrastructure networks like internet, electrical grids, wireless sensor networks, or web based social networks like Facebook and Twitter. On the other hand, large scale graphs naturally appear in the design of high performance algorithms for coding and compressing signals. Typical networks encountered in such applications have some common properties about connectivity and degree which can be accurately detected in real examples; the key point is to understand how to construct abstract graphs possessing exactly these properties. The reason is quite evident: simulations in real infrastructure networks or in social networks are difficult and expensive, and one hopes to be able to use theoretical simple models to perform numerical simulations.

Two popular properties of large scale networks, like internet and social networks, are the following:

- the diameter exhibits a slow growth in the number of nodes, typically of logarithmic type;
- the distribution of the degrees of nodes follows the so-called *power law*.

The construction of families of graphs exhibiting such properties is thus relevant for the analysis and simulation purposes of complex networks. In other contexts, where the graph represents the architecture of an underlying distributed algorithm, similar requirements occur. For instance, in the design of graphs describing low density parity check codes, the following issues are fundamental:

- the distribution of degrees of nodes follows some prespecified strict requirement;
- all cycles are all of large length (typically logarithmic in the number of nodes) so that the graphs locally exhibit a tree-like structure.

A remarkable achievement of modern graph theory is to prove the existence of families of graphs with specific properties as those discussed above and, even more important, to give explicit simple recipes to construct them concretely. Random graph theory is one of the most powerful tools to address such problems.

3 Erdős-Rényi Model

Roughly speaking, the Erdős-Rényi graph [6, 7] is a random graph $\mathcal{G}_{n,p}$ of order n in which the edges connecting nodes are included independently with probability $p \in (0, 1)$. More precisely, if \mathcal{G} is a specific graph over $\mathcal{V}_n = \{1, \dots, n\}$ with m edges then $\mathbb{P}(\mathcal{G}_{n,p} = \mathcal{G}) = p^m(1-p)^{\binom{n}{2}-m}$. If $p = 1/2$ then all graphs with n vertices are chosen with equal probability.

All the results we now present about Erdős-Rényi graphs can be retrieved, with more details, in a variety of books, e.g., in the comprehensive book [3] and in the more recent notes [20].

It should be noted that as p increases from 0 to 1, the model becomes more and more likely to select graphs with more edges (see Fig. 1). The following result is thus quite intuitive.

Proposition 1 *If \mathcal{D} is a monotonic increasing property, then the probability $\mathbb{P}(\mathcal{G}_{n,p}$ satisfies \mathcal{D}) is an increasing function of $p \in (0, 1)$.*

Remark Although simple, it is instructive to see a proof of the above proposition, for which an interesting relationship between $\mathcal{G}_{n,p}$ and $\mathcal{G}_{n,q}$ with $0 \leq p < q \leq 1$ can be used. It is immediate to notice that the ensemble $\mathcal{H} = \mathcal{G}_{n,p} \cup \mathcal{G}_{n, \frac{q-p}{1-p}}$ has the same probability distribution than $\mathcal{G}_{n,q}$. As $\mathcal{G}_{n,p} \subseteq \mathcal{H}$, we conclude that

$$\mathbb{P}(\mathcal{G}_{n,p} \text{ satisfies } \mathcal{D}) \leq \mathbb{P}(\mathcal{H} \text{ satisfies } \mathcal{D}) = \mathbb{P}(\mathcal{G}_{n,q} \text{ satisfies } \mathcal{D}).$$

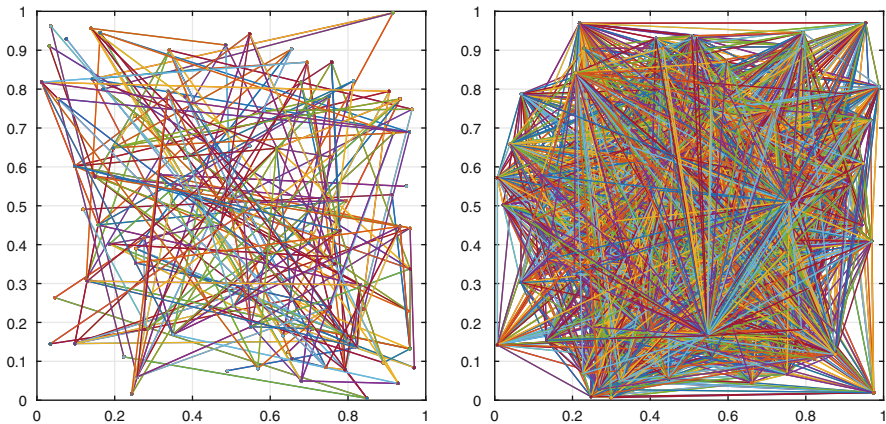


Fig. 1 Examples of Erdős-Rényi graphs with $n = 100$ nodes. *Left:* An edge is drawn with probability $p = 0.05$. *Right:* An edge is drawn with probability $p = 0.5$

Main focus in these notes will be on the behavior of random graphs in the so called large scale limit, namely when the number of vertices n tends to infinity. The parameter p will be a function of n and, typically, the ensemble will exhibit drastic change in graph properties (e.g. connectivity) as p crosses some threshold. Given a monotonic increasing property \mathcal{Q} , we say that a function $\zeta(n)$ is a threshold function for \mathcal{Q} if the following happens, for $n \rightarrow +\infty$:

- if $p \gg \zeta(n)$, $\mathcal{G}_{n,p}$ satisfies property \mathcal{Q} with probability 1;
- if $p \ll \zeta(n)$, $\mathcal{G}_{n,p}$ satisfies property \mathcal{Q} with probability 0.

When such a threshold function exists, we say that a phase transition occurs at that threshold. It is a remarkable fact that, for many key properties a threshold function indeed exists for the Erdős-Rényi ensemble. Below we will study in detail two important instances of this phenomenon which are the connectivity and the presence of a giant connected component in the graph.

3.1 Connectivity and Giant Component

Regarding connectivity, the following result holds, which implies that $\zeta(n) = \ln n/n$ is a threshold function for such property.

Theorem 1 *Let $\omega_n \rightarrow +\infty$ for $n \rightarrow +\infty$. The following facts hold true.*

1. If $p = \frac{\ln n + \omega_n}{n}$, then $\mathbb{P}(\mathcal{G}_{n,p} \text{ is connected}) \xrightarrow{n \rightarrow +\infty} 1$.
2. If $p = \frac{\ln n - \omega_n}{n}$, then $\mathbb{P}(\mathcal{G}_{n,p} \text{ is connected}) \xrightarrow{n \rightarrow +\infty} 0$.

We illustrate this result through numerical simulations in Fig. 2. On the left, the empirical probability of the event $\{\mathcal{G}_{n,c \ln(n)/n} \text{ is connected}\}$ is depicted as a function of the parameter $c \in (0, 2)$ averaged over 50 independent graph samples for five network sizes, i.e., for $n \in \{100, 200, 500, 1000, 5000\}$.

Remark The proof of the above theorem is not particularly difficult. Here we only illustrate some methodological ideas involved in the proof since they are quite ubiquitous in such field and also because some striking facts indeed happen.

Let N_1 be the number of isolated vertices in $\mathcal{G}_{n,p}$. It is clearly a random variable whose mean satisfies

$$\mathbb{E}(N_1) \leq n(1-p)^{n-1}. \quad (1)$$

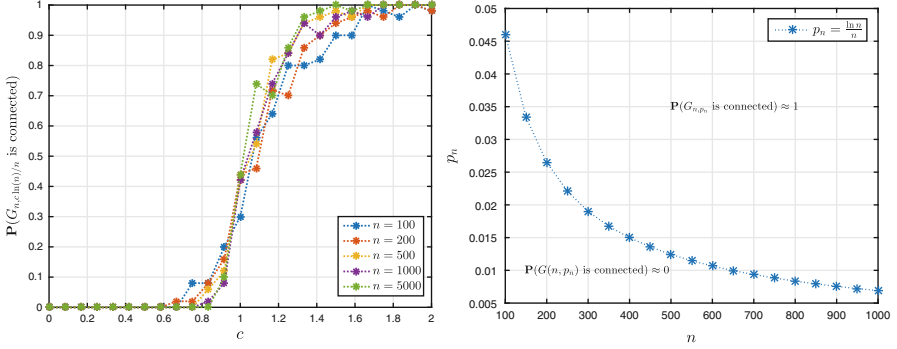


Fig. 2 *Left*: Empirical probability that $\mathcal{G}_{n,c \ln(n)/n}$ is connected as a function of the parameter $c \in (0, 2)$ averaged over 30 independent graph samples. *Right*: Theoretical threshold function for connectivity

Simple calculus derivations show that

$$\lim_{n \rightarrow +\infty} \mathbb{E}(N_1) = \begin{cases} 0 & \text{if } p = \frac{\ln n + \omega_n}{n} \\ +\infty & \text{if } p = \frac{\ln n - \omega_n}{n}. \end{cases}$$

This seems to suggest that above the threshold isolated vertices are absent while there are plenty of them below the threshold. Unfortunately, this computation is by itself not sufficient to conclude anything rigorous regarding the connectivity. On one hand, the absence of isolated vertices does not yield connectivity of the graph and, on the other hand, when $\mathbb{E}(N_1)$ blows up, nothing can be concluded on $\mathbb{P}(N_1 > 1)$ since Markov inequality only yields $\mathbb{P}(N_1 > 1) < \mathbb{E}(N_1)$. Below we will briefly describe the arguments needed to make rigorous conclusions.

Let N_k be the number of connected components of cardinality exactly k inside $\mathcal{G}_{n,p}$ and notice that

$$\begin{aligned} \{\mathcal{G}_{n,p} \text{ is disconnected}\} &\subseteq \left\{ \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor} N_k \geq 1 \right\} \\ \{\mathcal{G}_{n,p} \text{ is connected}\} &\supseteq \{N_1 = 0\} \end{aligned}$$

The proof of point 2. in Theorem 1 is obtained by first estimating $\mathbb{E}(N_k)$ by

$$\mathbb{E}(N_k) \leq \binom{n}{k} (1-p)^{k(n-k)} \quad (2)$$

This inequality simply comes from counting those subsets of k nodes isolated from the rest of the graph, and is indeed an equality for $k = 1$. Markov inequality yields

$$\mathbb{P}(\mathcal{G}_{n,p} \text{ is disconnected}) \leq \mathbb{P}\left(\sum_{k=1}^{\lfloor \frac{n}{2} \rfloor} N_k \geq 1\right) \leq \mathbb{E}\left(\sum_{k=1}^{\lfloor \frac{n}{2} \rfloor} N_k\right) \quad (3)$$

and result then follows by applying careful asymptotic estimations to (3). The proof of point 1. in Theorem 1 instead is obtained by Chebyshev inequality:

$$\begin{aligned} \mathbb{P}(\mathcal{G}_{n,p} \text{ is connected}) &\leq \mathbb{P}(N_1 = 0) \leq \mathbb{P}(|N_1 - \mathbb{E}(N_1)|^2 \geq |\mathbb{E}(N_1)|^2) \\ &\leq \frac{\mathbb{E}|N_1 - \mathbb{E}(N_1)|^2}{|\mathbb{E}(N_1)|^2} \end{aligned}$$

by explicitly computing the variance of N_1 . What is remarkable is the fact that such simple and apparently rough idea might work: the threshold for connectivity apparently is the same as the threshold for the disappearance of isolated nodes in the graph!

Below the threshold of connectivity, isolated vertices appear as well as other small isolated subgraphs. However, most of the graph remains connected, in other words the graphs still exhibits the presence of a so called giant component whose threshold of appearance is sensibly smaller at $\zeta(n) = 1/n$. Given a node $x \in \mathcal{V}_n$, we denote by C_x the connected component containing x . The following result captures the phenomenon (and is actually much more informative than just determining the threshold function).

Theorem 2 *The following facts hold true.*

1. *If $p = \frac{c}{n}$ with $c > 1$ then, there exists $\beta > 0$ such that*

$$\mathbb{P}(|\{x \in \mathcal{V}_n \text{ s.t. } |C_x| \geq \beta \ln n\}| = 1) \rightarrow 1 \quad \text{as } n \rightarrow \infty.$$

Moreover, the only component $C_{\bar{x}}$ such that $|C_{\bar{x}}| \geq \beta \ln n$, indeed satisfies $|C_{\bar{x}}| \sim \theta n$ for some $\theta \in]0, 1[$.

2. *If $p = \frac{c}{n}$ with $c < 1$, then for any $\beta > 0$ such that*

$$\mathbb{P}(\max_{x \in \mathcal{V}_n} |C_x| \geq \beta n) \rightarrow 0 \quad \text{as } n \rightarrow \infty.$$

It is important to remark the type of phase transition occurring when we pass the threshold $\zeta(n) = 1/n$. Below it only components of logarithmic size exist, while, above it, suddenly a giant component containing a non vanishing fraction of nodes exists while all other components are small and remain of logarithmic size.

The fraction size θ of the giant component when we choose $p = \frac{c}{n}$ with $c > 1$ is a function of the parameter c and, quite amazingly, can be analytically characterized and numerically computed. In Fig. 3 the linear growth rate of the giant component of the $\mathcal{G}_{n,c/n}$ is shown as a function of the parameter $c \in (0, 4)$ averaged over 50 independent samples of the graph.

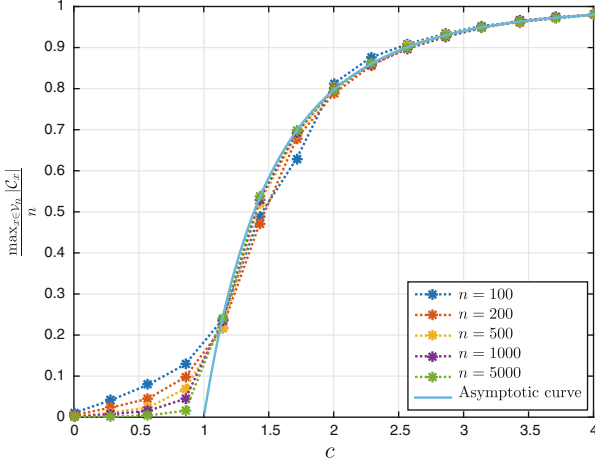


Fig. 3 The linear growth rate of the giant component of the $\mathcal{G}_{n,c}/n$ is shown as a function of the parameter $c \in (0, 4)$ averaged over 50 independent samples of the graph

3.2 Branching Processes

The theory of branching processes plays a key role in the proof of Theorem 2 and also in the next models we will describe. Here, we review the basics of such theory.

Branching processes model the growth of a population of individuals where each individual reproduces with the same offspring distribution. Formally, let us consider a family of i.i.d random variables ξ_i^t (with indices $t \in \mathbb{N}$ and $i \in \mathbb{N}$) taking values over \mathbb{N} to be interpreted as the number of descendants of individual i of generation t . The branching process Z_t is a Markov process defined by the recursive formula

$$Z_{t+1} = \begin{cases} \sum_{i=1}^{Z_t} \xi_i^t & \text{if } Z_t > 0 \\ 0 & \text{if } Z_t = 0 \end{cases}$$

with the initial condition $Z_0 = 1$. The main issue regarding the branching process is to establish the so called extinction probability, formally defined as

$$q := \mathbb{P}\{\exists t \in \mathbb{N} \mid Z_t = 0\}.$$

Assuming that $\mathbb{E}(\xi_i^t) = \mu < +\infty$, a simple conditioning argument, shows that $\mathbb{E}[Z_t] = \mu^t$ so that quite a different behavior is showing depending if $\mu < 1$ or $\mu > 1$.

In the first case $\mu < 1$, indeed, we have that, by Markov inequality $\mathbb{P}(Z_t \geq 1) \leq \mu^t$ and a straightforward application of Borel-Cantelli lemma yields that $\mathbb{P}(Z_t \geq 1, \forall t) = 0$. Thus, in the case $\mu < 1$ we have that $q = 1$, namely the process will extinguish in finite time with probability 1.

To analyze the case when $\mu > 1$, it is useful to study the evolution of $q_t := \mathbb{P}(Z_t = 0)$ the probability that, by time t , the process has extinguished. Notice that q_t is an increasing sequence converging to q . This sequence allows for a recursive characterization based on conditioning on the first generation offspring. Put $p_k = \mathbb{P}(\xi_1^t = k)$. We have that

$$q_{t+1} = \sum_k \mathbb{P}(Z_t = 0 \mid Z_1 = k) \mathbb{P}(Z_1 = k) = \sum_k q_t^k p_k.$$

If we consider the generating function $f(z) = \sum_k z^k p_k$, we can rewrite this as $q_{t+1} = f(q_t)$. Notice that $f'(z) \geq 0$ and $f''(z) \geq 0$ so that $f : [0, 1] \rightarrow [0, 1]$ is an increasing convex function such that $f(0) = p_0$ and $f(1) = 1$. The sequence q_t will converge to the smallest solution of the equation $f(z) = z$ which thus coincides with the extinction probability q . Notice that in the case when $\mu > 1$, there are always exactly two solutions to such equation, the smallest of which in $[0, 1]$. This implies that the extinction probability q is always strictly less than one; in other words there is a non vanishing probability that the process will not die out.

What is the relation between branching processes and Erdős-Rényi random graphs? Imagine you have fixed a node v and you reveal the graph $\mathcal{G}_{n,p}$ step by step, starting by v , then considering its first neighbors, then the second neighbors and so on. Interpreting the neighbors of a node as offspring, we get the connection with branching processes. More precisely, the number of nodes at distance exactly t from v can be upper bounded by a branching process Z_t with an offspring distribution given by a binomial $B(n - 1, p)$. It is not an equality because of two facts: (1) there may be cycles in the graph corresponding to intersections in the offspring of different nodes; (2) as we proceed revealing the graph, possible neighbors can only be drawn from smaller subsets, then the actual offspring is a binomial $B(k, p)$ with $k < n - 1$ (see Fig. 4).

We can thus assert that

$$|C_v| \leq \sum_t Z_t.$$

In the case when $c < 1$, the mean $\mu = (n - 1)p < 1$ and we know that the branching process will die out with probability 1. This will also imply that the cardinality of C_v will be bounded, with probability 1, with respect to n . In more informal terms, in the case when $c < 1$, if we pick a node and consider the connected component in $\mathcal{G}_{n,p}$ containing this node, with high probability it will be “small” not growing when $n \rightarrow +\infty$. While this fact is not sufficient to prove item 2. in Theorem 2,

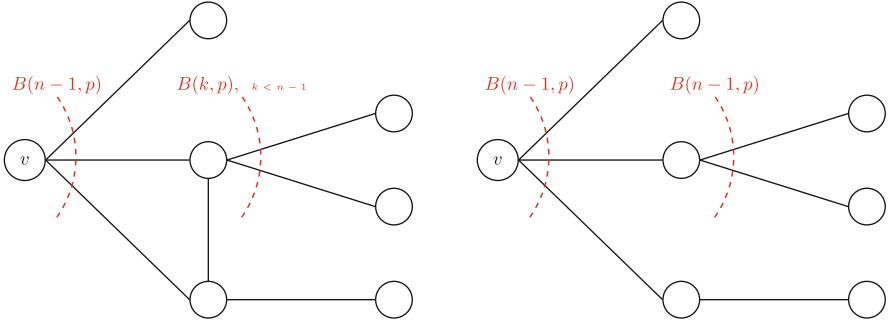


Fig. 4 Approximation of an Erdős-Rényi graph (*left*) with a branching process (*right*). The number of nodes of an Erdős-Rényi graph can be upper bounded by a branching process with offspring distribution $B(n-1, p)$. We have not an equality because in an Erdős-Rényi graph there can be cycles and also the possible number of nodes decreases while exploring the graph

it is the first step in that direction and it gives an intuitive understanding of the reason for the absence of a large connected component. In the case when $c > 1$ the corresponding branching process has an extinction probability $q < 1$. Since the one above is only an upper bound, in principle it does not infer anything regarding the existence of a giant component. More refined arguments who estimate the gap between the branching process and the real random graph, however allow to prove also item 1. in Theorem 2. The interesting fact is that the fraction size of the giant component θ is equal to $1 - q$ the probability that the corresponding branching process will not extinguish in finite time. Intuitively we can imagine the following scenario: for every node v you consider the branching process as described above, with probability $1 - q$ it will not die out. The average number of nodes having a non extinguishing branching process will be $(1 - q)n$: they will form the giant component of the graph.

3.3 Behavior at the Giant Component Threshold

The regime where $p = \frac{c}{n}$ where c is a constant > 1 is of particular applicative importance. Indeed, as seen before, in this regime a giant component appears with probability 1 as $n \rightarrow +\infty$. A number of other interesting facts indeed happen and we are going to review them below. First notice the way degrees behave: given a node v , its degree d_v is a binomial random variable $B(n, p)$, which, in the regime considered, converges, when $n \rightarrow +\infty$, to a Poisson random variable $\mathcal{P}(c)$ having mean equal to c . Differently from the connectivity threshold, where degrees have unbounded mean, in this regime degrees are bounded: in many applicative contexts this property is indeed quite natural to have.

Another interesting feature of such graphs is the tree-like structure they exhibit. Indeed, for a fixed node $v \in \mathcal{V}_n$, let $N_k(v)$ be the number of cycles of length k passing through node v in $\mathcal{G}_{n,p}$. A simple enumerative argument yields

$$\mathbb{E}[N_k(v)] = \frac{1}{2}(n-1)(n-2)\dots(n-k+1)p^k.$$

Using Markov's inequality, it thus follows that

$$\mathbb{P}(\exists \text{ cycle of length } \leq k \text{ containing } v) \leq \frac{1}{n} \frac{c^3 c^{k-2} - 1}{2(c-1)}.$$

This shows that cycle of logarithmic length through a node are unlikely. Indeed, can also be rewritten as

$$\mathbb{P}(\exists \text{ cycle of length } \leq a \log n \text{ containing } v) \leq \frac{cn^{a \log c - 1}}{2(c-1)} \xrightarrow{n \rightarrow +\infty} 0,$$

for all $a < 1/\log c$.

4 Configuration Model

The possibility to construct graphs possessing a specified degree distribution is very useful for its application to real world networks [4]. Real world networks, in fact, often exhibit power law behavior, which does not show up in the classical Erdős-Rényi model. This section is devoted to introduce a new random graph model where the degree distribution can be assigned in an arbitrary way.

Given $n \in \mathbb{N}$ and the sequence $\mathbf{d} = (d_1, \dots, d_n)$ such that $\sum_{i=1}^n d_i$ is even, the configuration model is a random graph with n vertices whose degrees are specified by \mathbf{d} . The construction is quite simple: endow each vertex $i \in \{1, \dots, n\}$ with d_i half edges (also called stubs) and then match uniformly at random all the half edges [3]. More precisely, given \mathbf{d} , we consider a sequence where node 1 is repeated d_1 times, node 2 is listed d_2 times, and so on:

$$\underbrace{1, \dots, 1}_{d_1}, \underbrace{2, \dots, 2}_{d_2}, \dots, \underbrace{n, \dots, n}_{d_n},$$

If we then consider a random permutation of the entries of this sequence, and then take the entries sequentially in pairs, we obtain a random matching of the stubs of the nodes. Figure 5 shows an instance of the configuration model obtained via stub matching technique.

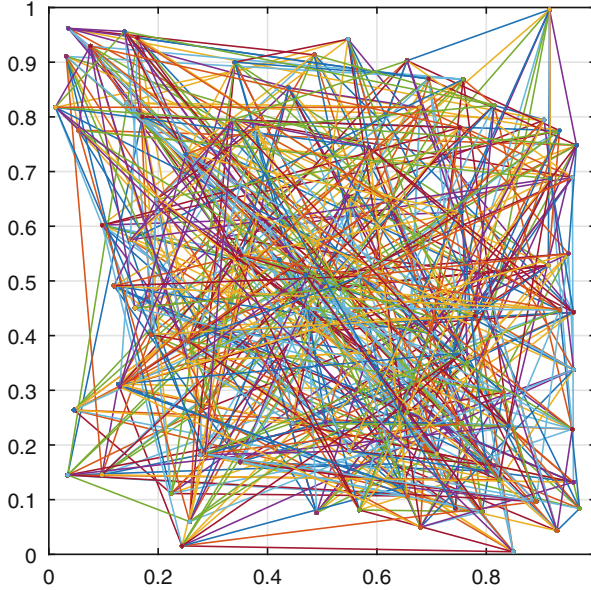


Fig. 5 Example of a graph with $n = 100$ nodes obtained via stub matching technique with degree sequence $\mathbf{d} = 6\mathbf{1}$

In this way, a *multigraph* is built, that is, a graph in which self-loops and multi-edges are allowed. However, the number of self-loops and multi-edges is typically a vanishing fraction when $n \rightarrow \infty$ and can generally be safely ignored. In the next, we denote such random multigraph by $\mathcal{G}_n(\mathbf{d})$

In the spirit of large scale networks, even in this case we are interested in considering ensembles when $n \rightarrow +\infty$. To this purpose we fix a discrete probability distribution over the positive integers $(p_k)_{k \in \mathbb{N}}$ and consider a sequence $\mathbf{d}^{(n)} = (d_1, \dots, d_n)$ such that

$$\frac{|\{i = 1, \dots, n : d_i = k\}|}{n} \xrightarrow{n \rightarrow \infty} p_k.$$

We will refer to $(p_k)_{k \in \mathbb{N}}$ as to the degree distribution of the ensemble. We put

$$\mu := \sum_{k=0}^{\infty} k p_k, \quad \mu_2 = \sum_{k=0}^{\infty} k^2 p_k$$

and we assume to be both finite.

In the following sections, we pursue an analysis analogous to what was done for the Erdős-Renyi model, presenting a number of results studying the threshold for connectivity and appearance of a giant component.

4.1 Connectivity and Giant Component

In this section, we study the properties of the configuration model in terms of connectivity and existence of a giant component. We start with the following theorem which reformulates Theorem 1 in [14] and characterizes a phase transition for the existence of a giant component.

We assume we are considering an ensemble of configuration models having degree distribution $(p_k)_{k \in \mathbb{N}}$ with finite first and second moments μ and μ_2 . Put

$$\nu := \frac{\mu_2 - \mu}{\mu}$$

and consider the generating function $g(z) = \sum_{k=0}^{\infty} p_k z^k$.

Theorem 3 *The following facts hold true.*

1. *If $\nu > 1$ then there exists $\gamma > 0$ such that*

$$\mathbb{P}(|\{x \in \mathcal{V}_n \text{ s.t. } |C_x| \geq \gamma \ln n\}| = 1) \rightarrow 1 \quad \text{as } n \rightarrow \infty.$$

Moreover the largest connected component $C_{\bar{x}}$ satisfies $C_{\bar{x}} \sim (1 - g(\rho))n$ where $\rho \in [0, 1[$ is the unique solution to $g'(\xi) = \mu\xi$.

2. *If $\nu < 1$, then for any $\beta > 0$*

$$\mathbb{P}(\max_{x \in \mathcal{V}_n} |C_x| \geq \beta n) \rightarrow 0 \quad \text{as } n \rightarrow \infty.$$

Remark As for Erdős-Renyi graphs, the growth of clusters can be approximated in the early stages by a suitable branching process. Below we sketch the basic idea of this approximation emphasizing the difference with respect to the Erdős-Renyi model and we give explanation for the role of the parameter ν and of the generating function in the determination of the size of the giant component.

Similarly to the cluster analysis for the Erdős-Renyi graph, let us fix a root $v \in \mathcal{V}$ randomly uniformly chosen, and explore the graph $\mathcal{G}_n(\mathbf{d})$ step by step, starting from v , then considering the first neighbors, and so on. The number of nodes at distance t from v can be upper estimated, as done in Sect. 3.2, by a Branching process which however, in this case, does not have a homogeneous offspring distribution in time. Indeed, the first generation neighbors of node v will follow the degree distribution (p_k) (at least asymptotically in n). However, one moment of thought makes us realize that the neighbors of v will not possess the same degree statistics; indeed, the uniform matching of the stubs used in the construction of the configuration model leads to a non uniform choice of nodes actually privileging high degree ones in a

way that is proportional to the degree. The correct probability that a neighbor of v will have k neighbors is thus given by $kp_k / \sum_j jp_j$ (again asymptotically in n). The right branching process to consider to upper bound the cluster growth is thus a so-called two-phase branching process Z_t defined by the recursive formula

$$Z_{t+1} = \begin{cases} \sum_{i=1}^{Z_t} \xi_i^t & \text{if } Z_t > 0 \\ 0 & \text{if } Z_t = 0 \end{cases} \quad \text{with } Z_0 = 1$$

in which the first generation has an offspring distribution given by $\mathbb{P}(\xi_i^0 = k) = p_k$ and the second and later generations have offspring distribution

$$\mathbb{P}(\xi_i^t = k) = q_k = \frac{(k+1)p_{k+1}}{\mu}$$

with $t \geq 1$. The reason for the index $k+1$ is simply due to the fact that the edge coming from previous generation neighbors is not considered in the offspring. Notice that, while $\mathbb{E}[\xi_i^0] = \mu$, for $t \geq 1$ we have that

$$\mathbb{E}[\xi_i^t] = \sum_k k \frac{(k+1)p_{k+1}}{\mu} = \sum_k \frac{(k-1)kp_k}{\mu} = \frac{\mu_2 - \mu}{\mu} = \nu.$$

Therefore, for $t \geq 1$, it holds $\mathbb{E}[Z_t] = \nu^{t-1}\mu$. If $\nu < 1$, then $\mathbb{E}[\sum_{t=0}^{\infty} Z_t] = 1 + \mu/(1-\nu)$ and, combining Markov inequality with Borel-Cantelli lemma, we obtain that $\mathbb{P}(Z_t \geq 1, \forall t) = 0$. In this case the process extinguish with probability one. The case with $\nu > 1$ is analyzed by studying the evolution of the sequence $(\theta_t)_{t \in \mathbb{N}_0}$ defined by $\theta_t = \mathbb{P}(Z_t = 0)$. As in the case of a standard branching process, conditioning with respect to the first generation offspring allows to express

$$\theta_{t+1} = \sum_k \mathbb{P}(Z_{t+1} = 0 | Z_1 = k) p_k.$$

Considering that after the first step, our branching process becomes a standard one with offspring distribution (q_k) we have that $\mathbb{P}(Z_{t+1} = 0 | Z_1 = k) = \tilde{\theta}_t^k$ where $\tilde{\theta}_t$ is the extinction probability at time t of a standard branching process having offspring distribution (q_k) . Therefore, using the considerations in previous sections, we can conclude that the extinction probability θ_t is governed by the recursive relations

$$\begin{cases} \tilde{\theta}_{t+1} = g_1(\tilde{\theta}_t) \\ \theta_{t+1} = g(\tilde{\theta}_t) \end{cases}$$

where $g(z) = \sum_k z^k p_k$ and $g_1(z) = \sum_{k=0}^{\infty} q_k z^k = g'(z)/\mu$. The sequence $\tilde{\theta}_t$ will thus converge to the unique solution ρ in $[0, 1[$ of the equation $g_1(z) = z$. Since g is continuous the sequence, θ_t will converge to $g(\rho)$. This allows us to conclude that

there is a non vanishing probability that the process will not die out. The fraction of nodes sitting in the giant component is exactly $1 - g(\rho)$ as in the case of the Erdős-Renyi model.

It can be proved [14] that when no giant component occurs, then there exist a constant R and a non-negative function $\omega_n \leq n^{1/8-\epsilon}$ such that the configuration model has fewer than $2R\omega_n^2 \log n$ cycles, and no component has more than one cycle. This states that small components have a tree like structure.

We conclude this discussion by noticing that some previously results about Erdős-Renyi random graphs $\mathcal{G}_{n,p}$ with link formation probability $p = \lambda/n + o(1/n)$ can be obtained from Theorem 3 as a special case. It is well known that for sufficiently large n , the binomial distribution with parameters n and $p = \lambda/n$ is close to the Poisson distribution with parameter λ . More precisely, we have that

$$\mathbb{P}(d_i = k) = e^{-\lambda} \frac{\lambda^k}{k!} + o(1).$$

Therefore, the giant component emerges when $\nu = \frac{\lambda + \lambda^2 - \lambda}{\lambda} > 1$ or, equivalently, when $\lambda > 1$. This fact yields the threshold function $\zeta(n) = 1/n$ for the emergence of the giant component.

We now investigate sufficient conditions ensuring the configuration model to be a connected graph.

Proposition 2 ([20, Theorem 10.14]) *Let us assume that $d_i \geq 3$ for every $i \in [n] := \{1, \dots, n\}$. Then, there exists a constant $c > 0$ such that*

$$\mathbb{P}(\mathcal{G}_n(\mathbf{d}) \text{ disconnected}) \leq \frac{c}{n}.$$

If $d_i \geq 3$ then $p_1 = p_2 = 0$, which implies $\nu > 1$. Therefore, we are considering the supercritical regime and Theorem 3 implies that the largest connected component has size $n(1 + o(1))$ for $n \rightarrow \infty$. The proof is based on a straightforward probabilistic and combinatorial procedure. Let N be the number of connected components inside $\mathcal{G}_n(\mathbf{d})$. The proof of Proposition 2 is obtained as for the Erdős-Renyi graph, by first estimating $\mathbb{E}(N)$ by

$$\mathbb{E}(N) \leq \sum_{\Omega \subset [n]} \frac{(\sum_{i \in \Omega} d_i)! (\sum_{i \in [n] \setminus \Omega} d_i)!}{(\sum_{i \in [n]} d_i)!}$$

and then by using Markov inequality.

The relevance of the assertion in Proposition 2 is in the fact that the graph tends to be connected even if the average degree is bounded, which is a difference with respect to the Erdős-Renyi graph and is more realistic for many applications. Moreover, the proposition requires only the degree to be larger than 3 ($p_1 = p_2 = 0$), which is a mild condition.

5 Random Geometric Graph

In a random geometric graph, nodes are associated with a position in a d -dimensional space and are connected when the distance is smaller than a prescribed radius. Intuitively, this may represent many practical situations. For example, let us consider the so-called ad hoc networks [12], which are formed by mobile nodes communicating over wireless channels and without centralized control; a number of technological applications are envisaged in the ad hoc class, ranging from vehicular networks to wireless sensor networks. In this field, a fundamental issue is how much transmission power is necessary to keep the network connected, assuming that the required power is proportional to the distance between nodes. In particular, the determination of the critical power, say the minimal power that guarantees asymptotic connectivity, is much studied [12]. The asymptotic case (say, the number of nodes tends to infinity) is important because in many applications (for example, in environmental and habitat monitoring) the number of nodes typically is very large [19]. In this regime, one can also observe that significant energy savings can be obtained by disconnecting a small percentage of nodes [17], which attracts the attention to the study of the largest connected component containing a nonvanishing fraction of nodes, that is, the giant component.

Motivated by the aforementioned applications, let us now introduce the theory of random geometric graphs and describe some results about connectivity and presence of giant components, providing some elements of the mathematical tools used for the proofs.

Let f be a probability density function on $[0, 1]^d$, and let x_1, \dots, x_n be independent and identically distributed points on $[0, 1]^d$ with common density f . Given a fixed $r > 0$, let us consider the graph $\mathcal{G}(n, r) = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$ such that to each vertex $v \in \mathcal{V}$ is associated a position x_v in $[0, 1]^d$ and the edge set is given by $\mathcal{E} = \{(u, v) \in \mathcal{V} \times \mathcal{V} : \|x_u - x_v\|_2 \leq r\}$ where $\|\cdot\|_2$ is the Euclidean norm. The graph we obtain is symmetric: if $(u, v) \in \mathcal{E}$ then also $(v, u) \in \mathcal{E}$. This model is called a random geometric graph; other terms which have been used for these graphs include *r-interval graphs* (when $d = 1$), *r-disk graphs* (when $d = 2$), and *proximity graphs*. To keep the presentation as simple as possible we will review some results on connectivity, distribution of degrees, and component sizes only

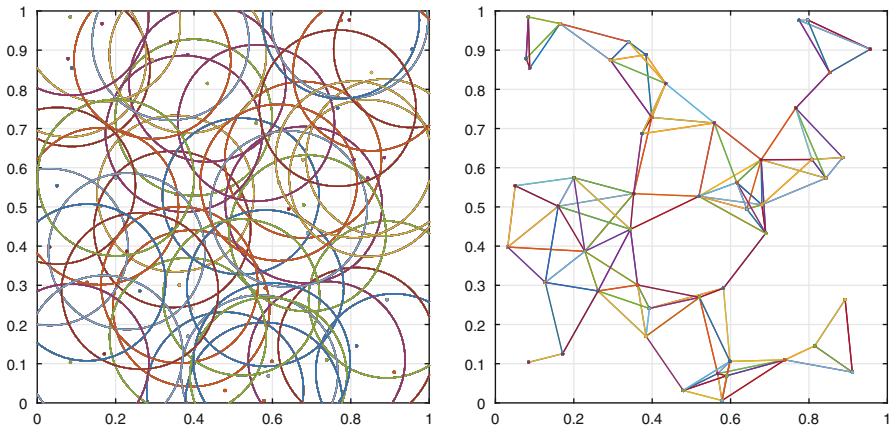


Fig. 6 Example of a random geometric graph with $n = 50$ and $r = 0.2$. *Left:* x_1, \dots, x_{50} are uniformly distributed on $[0, 1]^2$. An edge is drawn if two points are at distance below $r = 0.2$. *Right:* the resulting random geometric graph

when the dimension is $d = 2$ and the points are i.i.d. uniform on the unit square on $[0, 1]^2$. An example of a random geometric graph with $n = 50$ nodes and $r = 0.2$ is provided in Fig. 6.

As in the cases treated in the previous sections, it is difficult to perform precise computation of probabilities for properties of $\mathcal{G}(n, r)$ except for graphs with a small size n . Therefore, we consider sequences $(r_n)_{n \in \mathbb{N}}$ and we focus on the asymptotic properties of graphs $\mathcal{G}(n, r_n)$ when $n \rightarrow \infty$. In particular, we will compare results for random geometric graphs with their counterparts in the Erdős-Rényi models.

Before presenting some results, we make some preliminary observations on the model. If a node $v \in \mathcal{V}$ is placed near the boundary of $[0, 1]^2$ (at a distance smaller than r), then the ball centered at x_v of radius r will cover less area than the one centered in a location point at distance greater than r from the boundary. To avoid border effects, the region $[0, 1]^2$ is meant to form a torus, *i.e.* points at the border are assumed to be immediate neighbors of points at the opposite edge of the area.

With this simplification in mind, there exists a link between nodes (v, w) if and only if node w lies within a ball centered at x_v of radius r . Then the probability for the existence of the link (v, w) is given by πr^2 (the volume of the ball), which is independent of location x_v ; the degree d_v is a binomial random variable $B(n - 1, \pi r^2)$ with expected degree is $(n - 1)\pi r^2$, and the expected number of edges is $\binom{n}{2} \pi r^2$.

It is interesting to notice that both random geometric graphs and Erdős-Rényi graphs \mathcal{G}_{n, p, r^2} with edge probability $p = \pi r^2$ have the same average degree and the same expected number of edges.

5.1 Connectivity

The following proposition provides a necessary condition for connectivity [12].

Proposition 3 *Let $\omega_n \rightarrow \omega$ for $n \rightarrow +\infty$ and $\pi r_n^2 = \frac{\ln n + \omega_n}{n}$. Then*

$$\liminf_{n \rightarrow \infty} \mathbb{P}(\mathcal{G}(n, r_n) \text{ is disconnected}) \geq e^{-\omega} (1 - e^{-\omega}).$$

Remark Let N_k be the number of connected components of order k . Given a node $v \in \mathcal{V}_n$, let us denote by C_v the connected component containing v .

$$\{\mathcal{G}(n, r_n) \text{ is disconnected}\} \supseteq \{N_1 = 1\} = \bigcup_{v \in \mathcal{V}} \{v \text{ is the only isolated node in } \mathcal{G}(n, r_n)\}$$

By the inclusion-exclusion principle we get that

$$\begin{aligned} \mathbb{P}(\mathcal{G}(n, r_n) \text{ is disconnected}) &\geq \sum_{v \in \mathcal{V}} \mathbb{P}(v \text{ is an isolated node in } \mathcal{G}(n, r_n)) & (4) \\ &\quad - \sum_{v \in \mathcal{V}} \sum_{w \in \mathcal{V} \setminus \{v\}} \mathbb{P}(v, w \text{ are isolated nodes in } \mathcal{G}(n, r_n)). & (5) \end{aligned}$$

As already observed the probability of isolation is independent of the node and

$$\mathbb{P}(v \text{ is an isolated node in } \mathcal{G}(n, r_n)) = (1 - \pi r_n^2)^{(n-1)}$$

and

$$\mathbb{P}(v, w \text{ are isolated nodes in } \mathcal{G}(n, r_n)) \leq (1 - 2\pi r_n^2)^{n-2} + 3\pi r_n^2 \left(1 - \frac{5}{4}\pi r_n^2\right)^{n-2} \quad (6)$$

where the first term takes into account the event conditioned to the fact $\|x_v - x_w\| \geq 2r_n$ and the second one considers the case with $r_n \leq \|x_w - x_v\| \leq 2r_n$ (see Fig. 7).

By substituting these expressions in (4) and estimating them asymptotically we obtain

$$\mathbb{P}(\mathcal{G}(n, r_n) \text{ is disconnected}) \geq e^{-\omega} (1 - e^{-\omega}) + o(1), \quad \text{as } n \rightarrow \infty$$

which completes the proof of Proposition 3.

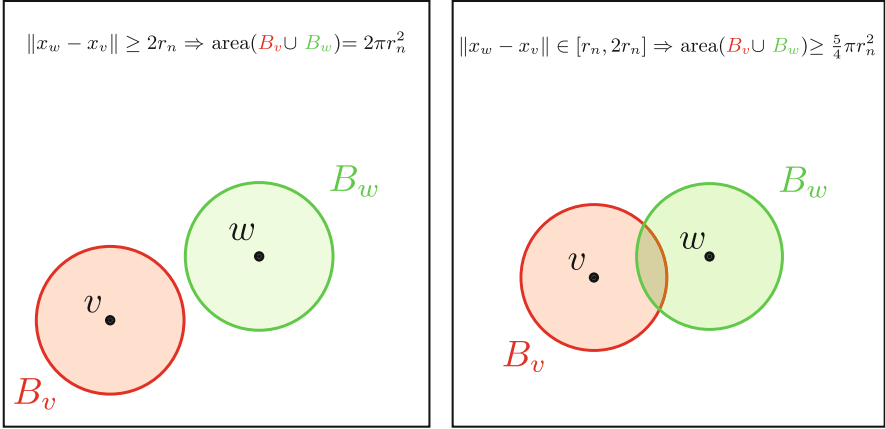


Fig. 7 The upper bound of the probability that v and w are isolated in (6) takes into account both the cases $\|x_v - x_w\| \geq 2r_n$ (left) and $r_n \leq \|x_w - x_v\| \leq 2r_n$ (right)

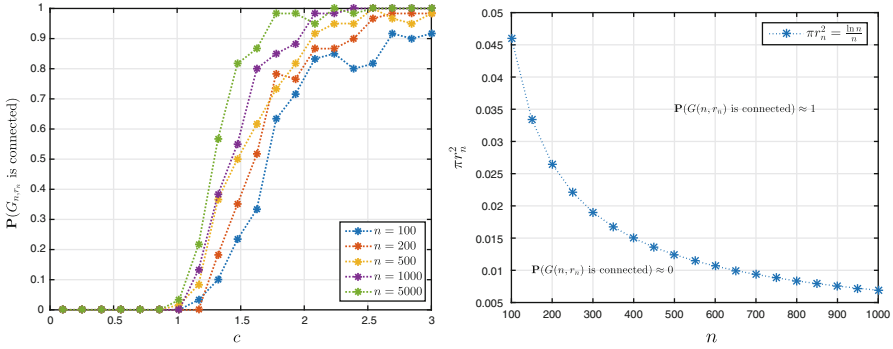


Fig. 8 Left: Empirical probability that \mathcal{G}_{n,r_n} with $\pi r_n^2 = c \log(n)/n$ is connected as a function of the parameter $c \in (0, 3)$ averaged over 60 independent samples of the graph. Right: Threshold function $\pi r_n^2 = \frac{\ln n}{n}$ for connectivity

Proposition 3 implies that a necessary condition for asymptotic connectivity is $\pi r_n^2 = (\ln n + \omega_n)/n$ with $\omega_n \rightarrow \infty$. We now present a sufficient condition for connectivity. More precisely, the following theorem shows that $\xi(n) = \ln n/n$ is a threshold function for connectivity (see Fig. 8).

Theorem 4 Let $\omega_n \rightarrow +\infty$ for $n \rightarrow +\infty$. The following facts hold true.

1. If $\pi r_n^2 = \frac{\ln n + \omega_n}{n}$, then $\mathbb{P}(\mathcal{G}(n, r_n) \text{ is connected}) \rightarrow 1$ as $n \rightarrow \infty$.
2. If $\pi r_n^2 = \frac{\ln n - \omega_n}{n}$, then $\mathbb{P}(\mathcal{G}(n, r_n) \text{ is connected}) \rightarrow 0$ as $n \rightarrow \infty$.

Remark It is surprising that both random geometric graphs $\mathcal{G}(n, r_n)$ and Erdős-Renyi graphs $\mathcal{G}_{n, \pi r_n^2}$ have closely related critical thresholds for connectivity. Although the threshold expression is the same, the proof of asymptotic connectivity for $\mathcal{G}(n, r_n)$ is quite different from that for Erdős-Renyi graphs $\mathcal{G}_{n, \pi r_n^2}$ and involves different tools. This is due to the fact that the event that there are links $(v, w) \in \mathcal{E}$ and $(w, u) \in \mathcal{E}$ is not independent of the event $(v, u) \in \mathcal{E}$ in the random geometric graph and, in particular, $\mathbb{P}((v, u) \in \mathcal{E}) < \mathbb{P}((v, u) \in \mathcal{E} | (v, w) \in \mathcal{E}, (w, u) \in \mathcal{E})$.

We start with a simple computation. As already happens in the Erdős-Renyi model, in the random geometric graphs the main obstruction to being connected is having a vertex with no neighbours. The number N_1 of isolated vertices is estimated as follows by

$$\mathbb{P}(N_1 \geq 1) \leq \sum_{v \in \mathcal{V}} \mathbb{E}[\mathbb{1}_{(d_v=0)}] = n\mathbb{P}\{d_1 = 0\} = n(1 - \pi r_n^2)^{n-1}. \quad (7)$$

Then (7) implies that if $\pi r_n^2 = \frac{\ln n + \omega_n}{n}$, then

$$\mathbb{P}(N_1 \geq 1) \leq e^{-\omega_n + o(1)} \rightarrow 0 \quad \text{as } n \rightarrow \infty$$

and suggests $\mathbb{P}(N_1 \geq 1) \rightarrow 1$ as $n \rightarrow \infty$ if $\pi r_n^2 = \frac{\ln n - \omega_n}{n}$. While it is not hard to calculate the expected number of isolated vertices, it is difficult to show there are no other obstructions to connectivity. In fact, the event $\{N_1 = 0\}$, which is clearly necessary for connectivity, turns out to be asymptotically sufficient. To prove this fact, Gupta and Kumar [12] make use of some results of *continuum percolation* [10, 13, 16], which plays a crucial role in the graph theory. Introduced in [13], the continuum percolation theory assumes that the nodes are distributed according to a Poisson process in \mathbb{R}^2 with rate ρ and an edge exists between two nodes if they are at a distance below r . The goal is to determine a critical threshold for the radius r such that the origin, which is assumed to be a node of the graph, is connected to an infinite-order connected component.

5.2 Giant Component

Recently, the emergence of giant components in random geometric graphs has been studied in different works [11, 16, 18]. Here, we report a result that suits the assumptions made for our exposition (for $d = 2$ and points independent and uniformly distributed over $[0, 1]^2$).

Theorem 5 ([16]) Let $r_n = \frac{\lambda}{\sqrt{n}}$ for any $\lambda > 0$. Then, there exists a critical value λ_c such that the following facts hold true.

1. If $\lambda > \lambda_c$, there exists $\theta > 0$ such that

$$\mathbb{P}(\{x \in \mathcal{V}_n \text{ s.t. } |C_x| \geq \theta n\} = 1) \rightarrow 1 \quad \text{as } n \rightarrow \infty.$$

2. If $\lambda < \lambda_c$, for any $\beta > 0$

$$\mathbb{P}(\max_{x \in \mathcal{V}_n} |C_x| \geq \beta n) \rightarrow 0 \quad \text{as } n \rightarrow \infty.$$

In Fig. 9 the average size of the largest cluster of the $\mathcal{G}_{n, \frac{\lambda}{\sqrt{n}}}$ without toroidal boundary conditions is depicted as a function of the parameter $\lambda \in (0, 10)$ averaged over 50 independent samples of the graph. Let λ_c be the lowest connectivity at which the fraction of vertices in the largest cluster is greater than 0 in the macroscopic limit. It should be noted that the critical value is sharply defined and is around $\lambda_c = 4.52$.

The proof of Theorem 5 requires considerable effort (see [16, Chapters 10–11]). Some simpler proofs have further been provided for subcases of these results. As an example, in [18] the problem is considered with $\pi r_n^2 = \frac{g(n)+c}{n}$, where c is any fixed real number and $g(n)$ is a positive function such that $\lim_{n \rightarrow \infty} g(n) = \infty$ and $\lim_{n \rightarrow \infty} \frac{g(n)}{\log n} = 0$. In particular, in [18, Theorem 5], the case of vertices uniformly distributed over $[0, 1]^2$. Its proof is based on percolation arguments.

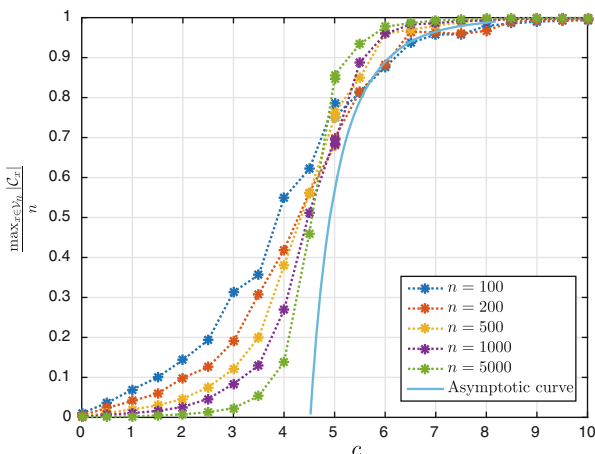


Fig. 9 The linear growth rate of the giant component of the $\mathcal{G}_{n, \frac{\lambda}{\sqrt{n}}}$ is shown as a function of the parameter $\lambda \in (0, 10)$ averaged over 30 independent samples of the graph

A further recent work on giant components of random geometric graphs is [11]. In [11, Theorem 1] the more general condition $\frac{c_1}{n} \leq r_n^2 \leq \frac{c_2 \log n}{n}$, with c_1, c_2 fixed constants, is assumed to prove the existence of a giant component with high probability for $d = 2$. Further suggested readings are [19], where a shadowing model is considered to investigate more practical transmission power issues, and in [2], where hyperbolic graphs are studied, in which distances are not Euclidean.

References

1. N. Alon, J. Spencer, *The Probabilistic Method*. Wiley-Interscience Series in Discrete Mathematics and Optimization, 2nd edn. (Wiley, New York, 2000)
2. M. Bode, N. Fountoulakis, T. Müller, On the giant component of random hyperbolic graphs, in *Proceedings of the 7th European Conference on Combinatorics, Graph Theory and Applications, EUROCOMB'13* (2013), pp. 425–429
3. B. Bollobás, *Random Graphs*. Cambridge Studies in Advanced Mathematics, vol. 73, 2nd edn. (Cambridge University Press, Cambridge, 2001)
4. S. Chatterjee, P. Diaconis, A. Sly, Random graphs with a given degree sequence. *Ann. Appl. Probab.* **21**(4), 1215–1625 (2011)
5. R. Durrett, *Random Graph Dynamics*. Cambridge Series in Statistical and Probabilistic Mathematics (Cambridge University Press, New York, 2007)
6. P. Erdős, A. Rényi, On random graphs I. *Publ. Math. Debr.* **6**, 290–297 (1959)
7. P. Erdős, A. Rényi, On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kutat ó Int. Közl.* **5**, 17–61 (1960)
8. P. Erdős, A. Rényi, On the evolution of random graphs. *Bull. Inst. Int. Stat.* **38**, 343–347 (1961)
9. P. Erdős, A. Rényi, On the strength of connectedness of a random graph. *Acta Math. Acad. Sci. Hung.* **12**, 261–267 (1961)
10. M. Franceschetti, R. Meester, *Random Networks for Communication: From Statistical Physics to Information Systems* (Cambridge University Press, Cambridge, 2007)
11. G. Ganesan, Size of the giant component in a random geometric graph. *Ann. Inst. Henri. Poincaré Probab. Stat.* **49**(4), 1130–1140 (2013)
12. P. Gupta, P.R. Kumar, Critical power for asymptotic connectivity in wireless networks, in *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming*, ed. by W.M. McEneaney, G. Yin, Q. Zhang (Birkhauser, Boston, 1998), pp. 547–566
13. R. Meester, R. Roy, *Continuum Percolation* (Cambridge University Press, Cambridge, 1996)
14. M. Molloy, B. Reed, A critical point for random graphs with a given degree sequence. *Random Struct. Algorithm* **6**(2–3), 161–180 (1995)
15. M.E.J. Newman, The structure and function of complex networks. *SIAM Rev.* **45**(2), 167–256 (2003)
16. M. Penrose, *Random Geometric Graphs*. Oxford Studies in Probability (Oxford University Press, Oxford, 2003)
17. P. Santi, D.M. Blough, The critical transmitting range for connectivity in sparse wireless ad hoc networks. *IEEE Trans. Mob. Comput.* **2**(1), 25–39 (2003)
18. X. Ta, G. Mao, B.D.O. Anderson, On the properties of giant component in wireless multi-hop networks, in *IEEE INFOCOM 2009* (2009), pp. 2556–2560
19. X. Ta, G. Mao, B.D.O. Anderson, On the giant component of wireless multihop networks in the presence of shadowing. *IEEE Trans. Veh. Technol.* **58**(2), 5152–5163 (2009)
20. R. van der Hofstad, *Random Graphs and Complex Networks*. Lecture Notes (2013). <http://www.win.tue.nl/~rhofstad/NotesRGCN.pdf>

Statistical Physics and Network Optimization Problems

**Carlo Baldassi, Alfredo Braunstein, Abolfazl Ramezanpour,
and Riccardo Zecchina**

The scope of these lecture notes is to provide an introduction to modern statistical physics mean-field methods for the study of phase transitions and optimization problems over random structures. We first give a brief introduction to the field using as tutorial example the percolation problem in random graphs. Next we describe the so called cavity method and the related message-passing algorithms (Belief Propagation and variants) which can be used to analyze and solve optimization problems over random structures.

1 Statistical Physics and Optimization

Equilibrium statistical mechanics and combinatorial optimization have common roots. Phase transitions are mathematical phenomena which are not limited to physical systems but are typical of many combinatorial problems, one famous example being the percolation transition in random graphs. Similarly, the understanding of relevant physical problems, such as three dimensional lattice statistics

C. Baldassi (✉)

DISAT and Center for Computational Sciences, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy
e-mail: carlo.baldassi@polito.it

A. Braunstein • R. Zecchina

Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

Human Genetics Foundation, Via Nizza 52, 10126 Torino, Italy

Collegio Carlo Alberto, Via Real Collegio 30, 10024 Moncalieri, Italy

A. Ramezanpour

Department of Physics, University of Neyshabur, P.O. Box 91136-899, Neyshabur, Iran

© Springer International Publishing Switzerland 2015

F. Fagnani et al. (eds.), *Mathematical Foundations of Complex Networked Information Systems*, Lecture Notes in Mathematics 2141,

DOI 10.1007/978-3-319-16967-5_2

or two dimensional quantum statistical mechanics problems, is strictly related to the question of purely combinatorial origin of solving counting problems over non planar lattices. Most of the tools and concepts which have allowed to solve problems in one field have a natural counterpart in the other. While the possibility of solving exactly physical models is related to the presence of algebraic properties which guarantee integrability, in the combinatorial approach the emphasis is more on algorithms that can be applied to specific problem instances.

Many interesting combinatorial optimization problems are NP-hard, implying that there is no known polynomial algorithm that is able to solve any instance of the problem. However, there is no a priori reason to assume that instances with real-world relevance will be specially hard, so the recent years have seen an upsurge of interest in the theory of typical-case complexity. In practice, hardness can still be present in random ensembles and one may try to identify those ensembles of optimization problems which are hard to solve, and the reason for this difficulty.

Combinatorial problems are usually written as Constraint Satisfaction Problems (CSP): N discrete variables are given which have to satisfy M constraints, all at the same time. Each constraint can take different forms depending on the problem under study. Well known examples are the K-Satisfiability (K-SAT) problem in which constraints are an ‘OR’ function of K variables in the ensemble (or their negations) and the Graph Q -coloring problem in which constraints simply enforce the condition that the end points of the edges in the graph must not have the same color (among the Q possible ones). A generic CSP can be written as the problem of finding a zero energy ground state of an appropriate energy function and its analysis amounts at performing a zero temperature statistical physics study. Hard combinatorial problems correspond to physical model systems with competing interaction, also called frustrated models.

In these lectures notes we will focus on two tutorial topics. The first one is to provide a brief introduction to statistical mechanics by showing how it can be used to study percolation in random graphs. The second is to give an introduction to the main statistical physics techniques which are used to study optimization problems over random structures.

2 Elements of Statistical Physics

The objective of statistical physics is to provide a probabilistic description of the macroscopic behaviour of a system at equilibrium from the knowledge of its microscopic components.

The implementation of this idea has required the introduction of revolutionary concepts and the interested reader can consult textbooks e.g. [1–3]. We shall adopt an operative approach and start from the following postulate.

A configuration C of the system, e.g. the specification of the N particle positions $\{\mathbf{r}_j\}$, has a probability $p(C)$ to be realized at any time when the system is in equilibrium. In other words, the probability of observing the system in configuration

C is given by¹

$$p(C) = \frac{1}{Z} \exp\left(-\frac{1}{T} E(C)\right). \quad (1)$$

In the above expression, T is the temperature and E is the *energy* of the system (a real-valued function over the set of configurations). The *partition function* Z ensures the correct normalization of the probability distribution p ,

$$Z = \sum_C \exp\left(-\frac{1}{T} E(C)\right). \quad (2)$$

The role of the temperature can be understood by considering two limiting cases:

1. *infinite temperature* $T = \infty$: the probability $p(C)$ becomes independent of C and all configurations are equiprobable. The system is said to be in a “disordered” phase (physically one may think to a gas or to a paramagnet).
2. *zero temperature* $T = 0$: the probability $p(C)$ is concentrated on the minimum of the energy function E , called the *ground state*.

The connection with combinatorics is easily understood by observing that the normalization factor Z is nothing but the *generating function* of the configuration energies.

We may rewrite the partition function (2) as

$$Z = \sum_E N(E) \exp(-\beta E), \quad (3)$$

where $N(E)$ is the number of configurations C having energies $E(C)$ precisely equal to E and $\beta = \frac{1}{T}$. If we set $x = \exp(-\beta)$ then $Z(x)$ becomes simply the generating function of the coefficients $N(E)$ as usually defined in combinatorics $Z = \sum_E N(E) x^E$.

For any configuration C , the *entropy* is defined as $\hat{S}(E(C)) = \log N(E(C))$. This quantity is in general very hard to compute, but in the large size limit $N \gg 1$, and in most cases of interest in statistical physics E is sharply peaked around its thermal average $\langle E \rangle_T$, and therefore one can compute the average entropy $S \simeq \sum_C \hat{S}(E(C)) p(C) \simeq \hat{S}(\langle E \rangle_T)$, which turns out to be equivalent to the *Shannon entropy* of the probability distribution p , as usually defined in information theory,

¹Throughout this chapter, we will omit for simplicity the Boltzmann’s constant k ; this is always possible by choosing appropriate measurement units, such that $k = 1$.

and which is known as the *Gibbs entropy* in statistical physics:

$$S = - \sum_C p(C) \log p(C) = -\frac{1}{T} \left(F(T) - \langle E \rangle_T \right), \quad (4)$$

where

$$F(T) = -T \log Z(T) \quad (5)$$

is called the *free-energy* of the system. Therefore, it is usual to refer to S simply as the entropy of the system.

The entropy is an increasing function of temperature, which can be readily verified by computing the derivative of (4). At zero temperature, it corresponds to the logarithm of the number of absolute minima of the energy function $E(C)$.

The fact that $S \simeq \hat{S}(\langle E \rangle_T)$ is an instance of a more general phenomenon: for common statistical physics models, intensive statistical quantities such as the energy density E/N and the entropy density \hat{S}/N become essentially fixed, when N is large, at any given value of T (except at most at phase transition boundaries, where discontinuities may appear): their probability distribution over the measure $p(C)$ tends exponentially to a Dirac delta distribution as $N \rightarrow \infty$. This property (known as *concentration of measure* in probability theory) allows to study e.g. the entropy as a function of the energy (by varying T), the phase transitions of the system, the structure of the ground states etc. A rigorous treatment of this subject can be found e.g. in [4].

3 Statistical Physics Approach to Percolation in Random Graphs

We shall attempt to familiarize with the statistical mechanics approach by describing its application to the analysis of the famous percolation problem in random graphs [5, 6]. Consider the complete graph K_N over N vertices. G_{N,N_L} is the set of graphs obtained by taking only $N_L = \gamma N/2$ among the $\binom{N}{2}$ edges of K_N in all possible different ways. A randomly chosen element of G_{N,N_L} with the flat measure is called a *random graph*.

The alternative procedure of deleting edges from K_N with probability $1 - \gamma/N$ lead to similar family of random graphs. In the large N limit, both families share common properties and we shall highlight the differences only when necessary.

The connected components of a given graph G are called *clusters*. Their size is the number of vertices they contain (e.g. isolated vertices are clusters of size one). We denote by $C(G)$ the number of connected components of G and by $c(G)$ their fractional number, $c(G) = \frac{C(G)}{N}$. When c is small, the random graph G is

characterized by few large clusters whereas for c approaching unity there are many clusters of small size.

Percolation theory is concerned with the study of the relationship between the probability p of two vertices being connected with the typical value of c in the $N \rightarrow \infty$ limit.

In what follows we show how such a relationship can be analysed by the study of a statistical mechanics model, the so called Potts model, through a saddle-point technique.

We denote with $\mathcal{P}(G)$ the probability of generating a random graph G through the deletion process from the complete graph K_N . Given that the edge deletions are statistically independent, this probability depends on the number of edges N_L only, and factorizes as

$$\mathcal{P}(G) = p^{N_L(G)} (1 - p)^{\frac{N(N-1)}{2} - N_L(G)}, \quad (6)$$

where $1 - p = 1 - \frac{\gamma}{N}$ is the probability of edge deletion. We want to study the probability density $\rho(c)$ of generating a random graph with c clusters,

$$\rho(c) = \sum_G \mathcal{P}(G) \delta(c - c(G)), \quad (7)$$

where δ indicates the Dirac distribution.

We can introduce a generating function of the cluster probability by

$$\begin{aligned} F(q) &= \int_0^1 dc \rho(c) q^{Nc} \\ &= \int_0^1 dc q^{Nc} \sum_{G \subseteq K_N} \mathcal{P}(G) \delta(c - c(G)) \\ &= \sum_{G \subseteq K_N} \mathcal{P}(G) q^{c(G)} = \sum_{G \subseteq K_N} p^{L(G)} (1 - p)^{\frac{N(N-1)}{2} - L(G)} q^{c(G)}, \quad (8) \end{aligned}$$

with q being a formal parameter.

In the large size limit, $\rho(c)$ is expected to be highly concentrated around some value $c(\gamma)$ equal to the typical fraction of clusters per vertex and depending only the average degree of γ . Random graphs whose $c(G)$ differs enough from $c(\gamma)$ will be exponentially rare in N . Therefore, the quantity

$$\omega(c) = \lim_{N \rightarrow \infty} \frac{1}{N} \log \rho(c) \quad (9)$$

should vanish for $c = c(\gamma)$ and be strictly negative otherwise. In the following, we shall compute $\omega(c)$ and thus obtain information not only on the typical number of clusters but also on the large deviations.

Defining the logarithm $\tilde{f}(q)$ of the cluster generating function as

$$\tilde{f}(q) = \lim_{N \rightarrow \infty} \frac{1}{N} \log F(q), \quad (10)$$

we obtain from a saddle-point calculation on c , see (8), (9),

$$\tilde{f}(q) = \max_{0 \leq c \leq 1} \left[c \log q + \omega(c) \right]. \quad (11)$$

\tilde{f} and ω are conjugated Legendre transforms and it will turn out that a direct computation of \tilde{f} is easier.

3.1 The Potts Model Representation

We proceed by computing the properties of random graphs by using a mapping of the generating function of the cluster probability to the so-called Potts model.

The Potts model [7] is defined in terms of an energy function which depends on N discrete variables σ_i (called *spins* in the physics jargon), one for each vertex of the complete graph K_N , which take q distinct values $\sigma_i = 0, 1, \dots, q-1$. The energy function is written as

$$E[\{\sigma_i\}] = - \sum_{i < j} \delta(\sigma_i, \sigma_j), \quad (12)$$

where $\delta(a, b)$ is the Kronecker delta function: $\delta(a, b) = 1$ if $a = b$ and $\delta(a, b) = 0$ if $a \neq b$. The partition function of the Potts model can be written by summing over all q^N configurations

$$Z_{\text{Potts}} = \sum_{\{\sigma_i=0, \dots, q-1\}} \exp \left[\beta \sum_{i < j} \delta(\sigma_i, \sigma_j) \right] \quad (13)$$

where $\beta = 1/T$ is the inverse temperature.

In order to identify the mapping we need to compare the expansion of Z_{Potts} to the definition of the cluster generating function $F(q)$ of the random graphs.

Following Kasteleyn and Fortuin [8], we start by rewriting Z_{Potts} as a dichromatic polynomial. Upon posing

$$v = e^\beta - 1, \quad (14)$$

one can easily check that (13) can be written in the form

$$Z_{Potts} = \sum_{\{\sigma_i\}} \prod_{i < j} [1 + v \delta(\sigma_i, \sigma_j)]. \quad (15)$$

When σ_i and σ_j take the same value there appears a factor $(1 + v)$ in the product (corresponding to a term e^β in (13)); on the contrary, whenever σ_i and σ_j are different the product remains unaltered. The expansion of the above product reads

$$Z_{Potts} = \sum_{\{\sigma_i\}} \left[1 + v \sum_{i < j} \delta(\sigma_i, \sigma_j) + v^2 \sum_{i < j, k < l / (i, j) \neq (k, l)} \delta(\sigma_i, \sigma_j) \delta(\sigma_k, \sigma_l) + \dots \right]. \quad (16)$$

We obtain $2^{\frac{N(N-1)}{2}}$ terms each of which composed by two factors, the first one given by v raised to a power equal to the number of δ s composing the second factor. It follows that each term corresponds to a possible subset of edges on K_N , each edge weighted by a factor v . There is a one-to-one correspondence between each term of the sum and the sub-graphs G of K_N . The edge structure of each sub-graph is encoded in the product of the δ s. This fact allows to reinterpret the partition function as a sum over sub-graphs

$$Z_{Potts} = \sum_{\{\sigma_i\}} \sum_{G \subseteq K_N} \left[v^{L(G)} \prod_{k=0}^{L(G)} \delta(\sigma_{i_k}, \sigma_{j_k}) \right] \quad (17)$$

where $L(G)$ is the number of edges in the sub-graph G and i_k, j_k are the vertices connected by the k th edge of the sub-graph. The order of the summations can be exchanged to perform the sum over the configurations first. Given a sub-graph G with L links and C clusters (isolated vertices included), the sum over spins configurations will give zero unless all the σ s belonging to a cluster of G have the same value (cfr. the δ functions). In such a cluster, one can set the σ s to any of the q different values and hence the final form of the partition function reads

$$Z_{Potts} = \sum_{G \subseteq K_N} v^{L(G)} q^{C(G)}. \quad (18)$$

By making the identification $p = 1 - e^{-\beta} = v / (1 + v)$, we can rewrite the partition function as

$$\begin{aligned} Z_{\text{Potts}} &= \sum_{G \subseteq K_N} \left(\frac{p}{1-p} \right)^{L(G)} q^{C(G)} \\ &= (1-p)^{-\frac{N(N-1)}{2}} \sum_{G \subseteq K_N} p^{L(G)} (1-p)^{\frac{N(N-1)}{2} - L(G)} q^{C(G)}. \end{aligned} \quad (19)$$

By extracting the prefactor on the r.h.s. of (19), we thus find $Z_{\text{Potts}} = e^{\frac{N\gamma}{2}} F(q)$, at the leading exponential order in N . The large N behaviour of the cluster probability $\omega(c)$ is therefore related to the Potts free-energy,

$$f_{\text{Potts}}(q) = - \lim_{N \rightarrow \infty} \frac{1}{\beta N} \log Z_{\text{Potts}}, \quad (20)$$

through

$$-\frac{\gamma}{2} - f_{\text{Potts}}(q) = \max_{0 \leq c \leq 1} (c \log q + \omega(c)). \quad (21)$$

We are interested in finding the value $c^*(q)$ which maximizes the r.h.s. in (21); since

$$\left. \frac{d\omega(c)}{dc} \right|_{c^*(q)} = -\log q \quad (22)$$

it follows that ω takes its maximum value for $q = 1$. Differentiating Eq. (21) with respect to q , we have

$$-\frac{df_{\text{Potts}}}{dq} = \frac{d}{dq} (c \log q + \omega(c)) = \frac{\partial}{\partial c} (c \log q + \omega(c)) \frac{\partial c}{\partial q} + \frac{c}{q}, \quad (23)$$

which, in virtue of Eq. (22) becomes:

$$c^*(q) = -q \frac{df_{\text{Potts}}}{dq}(q). \quad (24)$$

We have thus shown that the typical fraction of clusters per site, $c^*(q = 1)$, can be obtained, at a given connectivity γ , by computing the Potts free-energy in the vicinity of $q = 1$. Since the Potts model is originally defined for integer values of q , an analytic continuation to real values of q will be necessary.

A straightforward examination of the energy function (12) shows that the latter depends on the configurations only through the fractions $x(\sigma; \{\sigma_i\})$ of variables σ_i in the σ th state ($\sigma = 0, 1, \dots, q-1$) [9],

$$x(\sigma; \{\sigma_i\}) = \frac{1}{N} \sum_{i=1}^N \delta(\sigma_i, \sigma), \quad (\sigma = 0, 1, \dots, q-1). \quad (25)$$

Of course, $\sum_{\sigma} x(\sigma; \{\sigma_i\}) = 1$.

Using these fractions, the energy (12) may be rewritten as

$$E[\{\sigma_i\}] = -\frac{N^2}{2} \sum_{\sigma=0}^{q-1} x(\sigma; \{\sigma_i\})^2 + \frac{N}{2}. \quad (26)$$

Note that the last term on the r.h.s. of (26) can be neglected with respect to the first term whose order of magnitude is $O(N^2)$.

The partition function (13) at inverse temperature $\beta = \gamma/N$ now becomes

$$\begin{aligned} Z_{\text{Potts}} &= \sum_{\{\sigma_i=0,1,\dots,q-1\}} \exp\left(-\frac{\gamma}{2} N \sum_{\sigma=0}^{q-1} x(\sigma, \{\sigma_i\})^2\right) \\ &= \sum_{\{x_{\sigma}=0,1/N,\dots,1\}}^{(\text{Norm})} \exp\left(\frac{\gamma}{2} N \sum_{\sigma=0}^{q-1} x(\sigma)^2\right) \frac{N!}{\prod_{\sigma=0}^{q-1} [N x(\sigma)]!} \\ &= \int_0^1 \sum_{\{x(\sigma)\}}^{(\text{Norm})} \prod_{\sigma=1}^{q-1} dx(\sigma) \exp(-Nf[\{x(\sigma)\}]) \end{aligned} \quad (27)$$

to the leading order in N . The superscript (Norm) indicates that the sum or the integral must be restricted to the normalized subspace $\sum_{\sigma=0}^{q-1} x(\sigma) = 1$. The “free-energy” density functional f appearing in (27) is

$$f[\{x(\sigma)\}] = \sum_{\sigma=0}^{q-1} \left\{ -\frac{\gamma}{2} [x(\sigma)]^2 + x(\sigma) \log x(\sigma) \right\}. \quad (28)$$

In the limit of large N , the integral in (27) may be evaluated by the saddle-point method and the Potts free-energy (20) can be evaluated as

$$f_{\text{Potts}}(q) = \min_{\{x(\sigma)\}} f[\{x(\sigma)\}]. \quad (29)$$

From the definition of the problem, each possible value of σ plays the same role and f is invariant under the permutation symmetry of the different q values. However, we should keep in mind that such a symmetry could be broken in a given minimum. Depending on the value of the connectivity γ , the permutation symmetry

may or may not be broken, leading to a phase transition in the problem which coincides with the birth a giant component in the associated random graph.

3.1.1 Symmetric Saddle-Point

Consider first the symmetric candidate for an extremum of f ,

$$x^{\text{sym}}(\sigma) = \frac{1}{q}, \quad \forall \sigma = 0, \dots, q-1. \quad (30)$$

We have

$$f_{\text{Potts}}^{\text{sym}}(q) = -\log q - \frac{\gamma}{2q}. \quad (31)$$

Taking the Legendre transform of this free-energy, see (21) and (24), we get for the logarithm of the cluster distribution density

$$\omega^{\text{sym}}(c) = -\frac{\gamma}{2} - (1-c)(1 + \log \gamma - \log [2(1-c)]). \quad (32)$$

$\omega^{\text{sym}}(c)$ is maximal and null at $c^{\text{sym}}(\gamma) = 1 - \frac{\gamma}{2}$, a result that cannot be true for connectivities larger than two and must break down somewhere below. Comparison with the rigorous derivation in random graph theory indicates that the symmetric result is exact as long as $\gamma \leq \gamma_c = 1$ and is false above the percolation threshold γ_c . The failure of the symmetric extremum in the presence of a giant component coincides with the appearance of symmetry broken saddle points.

To understand the mechanism responsible for the symmetry breaking, one may look at the local stability of the symmetric saddle-point (30) and compute the eigenvalues of the Hessian matrix. One finds a non degenerate eigenvalue $\lambda_0 = q(q - \gamma)$ and another eigenvalue $\lambda_1 = q - \gamma$ with multiplicity $q - 2$. The analytic continuation of the eigenvalues to real $q \rightarrow 1$ lead to the single value $\lambda = 1 - \gamma$ which changes sign at the percolation threshold γ_c . Therefore, the symmetric saddle-point is not a local minimum of f above γ_c , showing that a more complicated saddle-point has to be found.

3.1.2 Symmetry Broken Saddle-Point

The simplest way to break the symmetry of the problem is to look for solutions in which one among the q values appears more frequently than the others. Therefore

one can look for a saddle-point of the form

$$\begin{aligned} x(0) &= \frac{1}{q} [1 + (1 - q)s] \\ x(\sigma) &= \frac{1}{q} [1 - s], \quad (\sigma = 1, \dots, q - 1). \end{aligned} \quad (33)$$

The symmetric case can be recovered by setting $s = 0$. The free-energy of the Potts model is obtained by plugging the fractions (33) into (28). In the limit $q \rightarrow 1$ of interest,

$$f[\{x(\sigma)\}] = -\frac{\gamma}{2} + (q - 1) f_{\text{Potts}}(s, \gamma) + O((q - 1)^2) \quad (34)$$

with

$$f_{\text{Potts}}(s, \gamma) = \frac{\gamma}{2} \left(1 - \frac{1}{2}s^2\right) - 1 + s + (1 - s) \log(1 - s) \quad (35)$$

Minimization of $f_{\text{Potts}}(s, \gamma)$ with respect to the order parameter s shows that for $\gamma \leq 1$ the symmetric solution $s = 0$ is recovered, whereas for $\gamma > 1$ there exists a non vanishing optimal value $s^*(\gamma)$ of s that is solution of the implicit equation

$$1 - s^* = \exp(-\gamma s^*). \quad (36)$$

The stability analysis shows that the solution is stable for any value of γ .

The interpretation of $s^*(\gamma)$ is straightforward: s^* is the fraction of vertices belonging to the giant cluster. The average fraction of connected components $c(\gamma)$ equals $-f_{\text{Potts}}(s^*(\gamma), \gamma)$, see (24), in perfect agreement with exact results by Erdős and Rényi.

Further results on the properties of random graphs can be extracted from the previous type of calculation, such as the scaling behaviour at the percolation point and large deviations. We refer to [10] for details.

4 Statistical Physics Methods for More Complex Problems

More advanced mean-field methods can be used to analyze and solve problems defined not only over K_N as in the previous example but on sparser graphs. Optimization and constraint satisfaction problems over finite connectivity random networks are the chief examples [11]. The functional mean field techniques which are used to study these problems are known as cavity methods with different levels of possible symmetry breaking. The story of these mathematical approaches is quite old [12, 13] and they have been rediscovered many times within different disciplines.

However it is only in the last decade that their algorithmic power has started to be fully appreciated.

In order to provide a concise presentation we introduce a notation which will be used throughout the rest of this chapter for generic constraint satisfaction problems and Statistical physics models.

We will denote by \mathbf{s} the set of N discrete variables over which the problem is defined, and use by convention the letters i, j, k, \dots to denote variable indices. Each variable s_i can in general take values in a different set X_i , thus $\mathbf{s} = \{s_i \in X_i | i = 1, \dots, N\}$, but it is often the case that $X_i = X$ is common to all variables.

We will denote by C the set of M hard constraints which characterize the problem, and use by convention the letters a, b, \dots to denote hard constraint indices; therefore $C = \{C_a | a = 1, \dots, M\}$. We also use the notation ∂a to indicate the set of all variable indices involved in the a th constraint (and by $s_{\partial a}$ the corresponding set of variables), and ∂i to denote the set of all constraint indices which involve the i th variable. So to each given constraint C_a we associate the indicator function $\mathbb{1}_a(s_{\partial a})$ which is 1 if the constraint is satisfied, 0 otherwise.

A constraint satisfaction problem can thus be mapped onto a bipartite graph (factor graph), with variable nodes representing the variables s_i and factor nodes representing the constraints C_a ; the edges of the graph always connect nodes of different types and encode the structure of the problem (i.e. the sets ∂a , or equivalently ∂i). We denote with (ai) an edge from the factor node a to the variable node i . We denote the set of all edges of the graph by \mathcal{E} .

The problem is deemed *satisfiable* (SAT) as long as there is a solution satisfying all the hard constraints, otherwise it is *unsatisfiable* (UNSAT).

We also introduce soft constraints, associated with either node type, denoted as $E_i(s_i)$ for the variable nodes and $E_a(s_{\partial a})$ for the factor nodes. Together, they provide the energy function $E = \sum_i E_i(s_i) + \sum_a E_a(s_{\partial a})$, which allows to differentiate between valid configurations \mathbf{s} by favouring those configurations which have the lowest energy. An external parameter β , which has the role of an inverse temperature, controls the relative weight of the valid configurations as a function of their energy; in the limit of $\beta \rightarrow 0$, all configurations are equally weighted, while in the zero-temperature limit $\beta \rightarrow \infty$ the soft constraints become equivalent to hard constraints and only the configurations which realize the minima of the energy (ground states) are allowed.

This formalism allows us to map any constraint satisfaction problem onto a statistical physics model defined by the following partition function:

$$Z = \sum_{\mathbf{s}} \prod_a \mathbb{1}_a(s_{\partial a}) e^{-\beta[\sum_i E_i(s_i) + \sum_a E_a(s_{\partial a})]} \quad (37)$$

which corresponds to the following probability measure:

$$\mu(\mathbf{s}) = \frac{1}{Z} \prod_a \mathbb{I}_a(s_{\partial a}) e^{-\beta[\sum_i E_i(s_i) + \sum_a E_a(s_{\partial a})]} \quad (38)$$

As we already noted in Sect. 2, in most common cases intensive statistical quantities such as the energy density are essentially fixed for almost any given value of the problem's parameters, when $N \gg 1$. Furthermore, CSPs and statistical physics models are often considered at the level of ensembles (families) of problems, where the parameters which describe each problem (the structure of the factor graph and of the constraints) are extracted from some probability distribution. A realization of the parameters constitutes the so-called *quenched disorder* of the system. When the probability distribution of some quantity, with respect to the quenched disorder (consider e.g. the thermal average of the energy density as a function of the model's parameters) is also peaked around its mean, the quantity is called *self-averaging*. This implies that any random instance of a problem is representative of the whole family, and, conversely, that studying the problem at the ensemble level is informative about almost all individual instances of the problem. The percolation transition discussed above is an example of this phenomenon. The self-averaging property of a given quantity may need to be assessed on a case-by-case basis, but in some important cases general results can be used (e.g. Talagrand's concentration inequality [14], which applies to Lipschitz-continuous functions).

Following are some examples of how some common problems can be mapped on this formalism:

- In a spin glass (a prototypical problem in statistical physics of disordered systems) the variables are binary, $s_i \in \{-1, +1\}$, and all constraints involve only two distinct variables, so if $\partial a = \{i, j\}$ we can make the identification $a \equiv (ij)$; hard constraints are moot, $\forall i, j : \mathbb{I}_{(ij)}(s_i, s_j) = 1$, and soft constraints are described by local fields h_i and interactions J_{ij} , extracted from some random distribution: $E = -\sum_i h_i s_i - \sum_{(ij) \in \mathcal{E}} J_{ij} s_i s_j$.
- In the q -coloring problem the interactions are again pair-wise, $a \equiv (ij)$, but the graph structure is typically non-trivial (e.g. may be random), each variable can take one of q states, $s_i \in \{0, \dots, q-1\}$, and all constraints are hard: $\mathbb{I}_{(ij)}(s_i, s_j) = 1 - \delta_{s_i, s_j}$ (where δ is the Kronecker symbol).
- In the maximum weight independent set problem the variables are binary, $s_i \in \{0, 1\}$, the interaction are pair-wise, $a \equiv (ij)$, and defined on a non-trivial graph, and we have both hard and soft constraints: $\mathbb{I}_{(ij)}(s_i, s_j) = \delta_{s_i s_j, 0}$ and $E_i(s_i) = -s_i$.
- In the p -spin model (an extension of the spin glass model) variables are binary, $s_i \in \{-1, +1\}$, but each interaction involves p variables, $a \equiv (i_1, \dots, i_p)$, and we have soft constraints $E = -\sum_{(i_1 \dots i_p) \in \mathcal{E}} J_{i_1 \dots i_p} \prod_{l=1}^p s_{i_l}$. When $\beta \rightarrow \infty$, it

becomes equivalent to the problem known as p -XORSAT in computer science, with each energy term corresponding to an XOR hard constraint enforcing the condition

$$\sum_{l=1}^p \frac{(1 + s_{il})}{2} = \frac{(1 + \text{sign}(J_{i_1 \dots i_p}))}{2} \pmod{2} \quad (39)$$

- In the K -SAT problem the variables are binary, $s_i \in \{-1, +1\}$, and each hard constraint C_a involves K variables: $\mathbb{I}_a(s_{\partial a}) = 1 - \prod_{l=1}^K (1 - J_{a i_l} s_{i_l})$, with $J_{a i} \in \{-1, +1\}$.

5 Bethe Approximation and Message Passing Algorithms

5.1 Belief Propagation

Solving a model described by (37) is in general a hard problem. However, a general solution is possible when the underlying factor graph structure has no loops, in which case it is called a *tree*.²

5.1.1 Marginals

Let us define the local marginals

$$\mu_i(s_i) = \sum_{\{s_j\}_{j \neq i}} \mu(\mathbf{s}) \quad (40)$$

$$\mu_a(s_{\partial a}) = \sum_{\{s_j\}_{j \neq \partial a}} \mu(\mathbf{s}). \quad (41)$$

We will show that, on a tree, (38) can be written in terms of the above marginals, in one of the two equivalent forms:

$$\mu(\mathbf{s}) = \prod_i \mu_i(s_i) \prod_a \frac{\mu_a(s_{\partial a})}{\prod_{i \in \partial a} \mu_i(s_i)} \quad (42)$$

$$= \prod_i \mu_i(s_i)^{1-|\partial i|} \prod_a \mu_a(s_{\partial a}). \quad (43)$$

²When the graph does not form a single connected component it is often called a *forest*, but this distinction is moot for our purposes.

The marginals (40) and (41) can be obtained by the so-called Belief Propagation (or Bethe-Peierls) equations. The procedure consists of writing marginals for a graph in which either a variable or some factor nodes are removed, as functions of analogous marginals. These marginals are called *cavity* marginals, and Belief Propagation is an example of a cavity method. The resulting system of equations is then solved by an iterative procedure, and the desired, non-cavity marginals are finally computed. We indicate with the subscript $i \rightarrow a$ the cavity marginals for variable i when the factor node a is removed from the graph, and with the subscript $a \rightarrow i$ the cavity marginal for variable i when all factor nodes $b \in \partial i \setminus a$ are removed.

The Belief Propagation (BP) equations for the cavity marginals are written as:

$$\mu_{i \rightarrow a}(s_i) = \frac{1}{z_{i \rightarrow a}} e^{-\beta E_i(s_i)} \prod_{b \in \partial i \setminus a} \mu_{b \rightarrow i}(s_i) \quad (44)$$

$$\mu_{a \rightarrow i}(s_i) = \frac{1}{z_{a \rightarrow i}} \sum_{s_{\partial a \setminus i}} \mathbb{I}_a(s_{\partial a}) e^{-\beta E_a(s_{\partial a})} \prod_{j \in \partial a \setminus i} \mu_{j \rightarrow a}(s_j) \quad (45)$$

where $z_{i \rightarrow a}$ and $z_{a \rightarrow i}$ are normalization constants. These equations are exact on tree graphs, since, when removing a factor node, all the variables involved in that node become independent from each other, and their probability distribution factorizes. They can be solved iteratively by starting from the leaves of the graph (nodes of connectivity 0 or 1), where their expression becomes trivial, and iterating inwards; therefore, the time required is at worst of order MKq^K , where K is the maximum degree of the function nodes, and q is the maximum number of states which a variable can take. The term q^K can in some common cases be improved by exploiting the structure of the functions \mathbb{I}_a and E_a .

The cavity marginals are also often called *messages*, and the iterative procedure *message passing*.

It is also useful to introduce the non-normalized message passing equations, which define the cavity partition functions:

$$Z_{i \rightarrow a}(s_i) = e^{-\beta E_i(s_i)} \prod_{b \in \partial i \setminus a} Z_{b \rightarrow i}(s_i) \quad (46)$$

$$Z_{a \rightarrow i}(s_i) = \sum_{s_{\partial a \setminus i}} \mathbb{I}_a(s_{\partial a}) e^{-\beta E_a(s_{\partial a})} \prod_{j \in \partial a \setminus i} Z_{j \rightarrow a}(s_j). \quad (47)$$

Using these, we can express the marginals (40) and (41) as:

$$\mu_i(s_i) = \frac{1}{Z} e^{-\beta E_i(s_i)} \prod_{a \in \partial i} Z_{a \rightarrow i}(s_i) \quad (48)$$

$$\mu_a(s_{\partial a}) = \frac{1}{Z} \mathbb{I}_a(s_{\partial a}) e^{-\beta E_a(s_{\partial a})} \prod_{i \in \partial a} Z_{i \rightarrow a}(s_i). \quad (49)$$

These allow us to write the constraints as functions of the marginals and the cavity partition functions; introducing them back into Eq. (38) allows us to prove the relation (42).

It is easy to see that the marginals (40) and (41) can also be written in terms of the cavity marginals as:

$$\mu_i(s_i) = \frac{1}{z_i} e^{-\beta E_i(s_i)} \prod_{a \in \partial i} \mu_{a \rightarrow i}(s_i) \quad (50)$$

$$\mu_a(s_{\partial a}) = \frac{1}{z_a} \mathbb{I}_a(s_{\partial a}) e^{-\beta E_a(s_{\partial a})} \prod_{i \in \partial a} \mu_{i \rightarrow a}(s_i) \quad (51)$$

where again z_i and z_a are normalization constants.

Another useful relationship which immediately follows is:

$$\mu_i(s_i) \propto \mu_{i \rightarrow a}(s_i) \mu_{a \rightarrow i}(s_i) \quad \forall a \in \partial i. \quad (52)$$

5.1.2 Free Energy

The cavity partition functions also allow us to express the total free energy and partition function by:

$$e^{-\beta F} = Z = \sum_{s_i} e^{-\beta E_i(s_i)} \prod_{a \in \partial i} Z_{a \rightarrow i}(s_i) \quad \forall i. \quad (53)$$

We can decompose the total free energy as the sum of local contributions expressed in terms of the cavity messages. Let us define the cavity free energies and free energy shifts by:

$$e^{-\beta F_{i \rightarrow a}} = \sum_{s_i} Z_{i \rightarrow a}(s_i) \quad (54)$$

$$e^{-\beta F_{a \rightarrow i}} = \sum_{s_i} Z_{a \rightarrow i}(s_i) \quad (55)$$

$$e^{-\beta \Delta F_{i \rightarrow a}} = e^{-\beta(F_{i \rightarrow a} - \sum_{b \in \partial i \setminus a} F_{b \rightarrow i})} = z_{i \rightarrow a} \quad (56)$$

$$e^{-\beta \Delta F_{a \rightarrow i}} = e^{-\beta(F_{a \rightarrow i} - \sum_{j \in \partial a \setminus i} F_{j \rightarrow a})} = z_{a \rightarrow i} \quad (57)$$

and analogously for the non-cavity free energy shifts:

$$e^{-\beta \Delta F_i} = z_i \quad (58)$$

$$e^{-\beta \Delta F_a} = z_a. \quad (59)$$

Using the BP equations it is easy to see that:

$$\Delta F_{i \rightarrow a} = \Delta F_i - \Delta F_{ia} \quad (60)$$

$$\Delta F_{a \rightarrow i} = \Delta F_a - \Delta F_{ia} \quad (61)$$

with [cfr. Eq. (52)]

$$e^{-\beta \Delta F_{ia}} = \sum_{s_i} \mu_{i \rightarrow a}(s_i) \mu_{a \rightarrow i}(s_i) = \frac{z_i}{z_{i \rightarrow a}}. \quad (62)$$

With these, we can rewrite the free energy as:

$$F = \sum_i \Delta F_i + \sum_a \Delta F_a - \sum_{(ia)} \Delta F_{ia} \quad (63)$$

or as:

$$F = \sum_a \left(\Delta F_a + \sum_{i \in \partial a} \Delta F_{i \rightarrow a} \right) - \sum_i (|\partial i| - 1) \Delta F_i. \quad (64)$$

The free energy can also be written in terms of the local marginals $\mu_i(s_i)$ and $\mu_a(s_{\partial a})$, as:

$$F = \sum_a F_a - \sum_i (|\partial i| - 1) F_i \quad (65)$$

where

$$F_a = \sum_{s_{\partial a}} E_a(s_{\partial a}) \mu_a(s_{\partial a}) - \frac{1}{\beta} \sum_{s_{\partial a}} \mu_a(s_{\partial a}) \ln \mu_a(s_{\partial a}) \quad (66)$$

$$F_i = \sum_{s_i} E_i(s_i) \mu_i(s_i) - \frac{1}{\beta} \sum_{s_i} \mu_i(s_i) \ln \mu_i(s_i). \quad (67)$$

In the above equations, we adopted the convention that $x \log x = 0$ if $x = 0$.

5.1.3 Graphs with Loops

The above equations are only exact if the factor graph is a tree. When that is not the case, expression (42), known as the Bethe approximation, can still provide good results in a wide range of cases, and even become asymptotically exact in the limit of large N .

Preliminarily, we shall note that, on graphs with loops, the procedure for solving the BP equations must be slightly modified: for example, one could initialize the cavity messages at random, or uniformly, and update them by using Eqs. (44) and (45) in random order, until a fixed point for the whole set is eventually found.

With regard to the approximation estimate, the crucial observation is that the property which was used in the derivation of the BP equations is that the cavity marginals factorize, i.e. that the connected correlations between any two variables involved in an interaction a vanish when the interaction a is removed: this is commonly called the *clustering property*.

In a graph with loops, the clustering property does not hold in general, but the correlations which one neglects by assuming that it does (i.e. that the joint probability of the variables $s_{\partial a}$ can be factorized if a is removed) can become asymptotically small as the problem size grows: one common example where this may happen is that of random graphs in which the typical size of the loops tends to diverge with the size of the graph. This kind of graphs are called *locally tree-like*: removing any factor node a means that the distance along the graph of the variables in $s_{\partial a}$ diverges, and we may expect that they become uncorrelated. This is often, but now always, the case, since there is one additional, more subtle, condition: in the above derivation, we assumed that the BP equations only have a single solution: this is always true on trees, but when loops are present there may be no solution at all, or more than one, even on locally tree-like graphs (see Sect. 5.4). In both cases, the observed effect is that the iterative procedure does not converge, or that some normalization constants $z_{i \rightarrow a}$ or $z_{a \rightarrow i}$ become 0 during the iteration.

In many cases, the BP equations provide reasonable estimates even when the clustering property doesn't hold; in particular, they can be exploited to identify "good" configurations (i.e. low-energy, SAT configurations) for a given instance of a problem, even when optimality is not strictly guaranteed, as we shall see in Sect. 5.3.

5.2 The $\beta \rightarrow \infty$ Limit: Minsum Algorithm

The BP equations (44) and (45) can be studied in the limit $\beta \rightarrow \infty$, assuming that the messages scale as $\mu_{i \rightarrow a}(s_i) \propto e^{-\beta M_{i \rightarrow a}(s_i)}$ and $\mu_{a \rightarrow i}(s_i) \propto e^{-\beta M_{a \rightarrow i}(s_i)}$. In this way we obtain:

$$M_{i \rightarrow a}(s_i) = E_i(s_i) + \sum_{b \in \partial i \setminus a} M_{b \rightarrow i}(s_i) - C_{i \rightarrow a} \quad (68)$$

$$M_{a \rightarrow i}(s_i) = \min_{s_{\partial a \setminus i} : \mathbb{I}_a(s_{\partial a})=1} \left\{ E_a(s_{\partial a}) + \sum_{j \in \partial a \setminus i} M_{j \rightarrow a}(s_j) \right\} - C_{a \rightarrow i}. \quad (69)$$

These are the so called minsum equations. The constants $C_{i \rightarrow a}$ and $C_{a \rightarrow i}$ enforce the conditions $\min_{s_i} M_{i \rightarrow a}(s_i) = \min_{s_i} M_{a \rightarrow i}(s_i) = 0$, which are the analogues to the normalization of the BP messages.

When applied to tree graphs, the resulting minsum algorithm can be ascribed to the family of dynamic programming algorithms.

The minsum messages can be used to obtain an estimate of the minimum energy by taking the limit $\beta \rightarrow \infty$ of the Bethe free energy (63):

$$E_0 = \lim_{\beta \rightarrow \infty} F = \sum_i \Delta E_i + \sum_a \Delta E_a - \sum_{(ia)} \Delta E_{ia} \quad (70)$$

where

$$\Delta E_i = \lim_{\beta \rightarrow \infty} \Delta F_i = \min_{s_i} \left\{ E_i(s_i) + \sum_{a \in \partial i} M_{a \rightarrow i}(s_i) \right\} \quad (71)$$

$$\Delta E_a = \lim_{\beta \rightarrow \infty} \Delta F_a = \min_{s_{\partial a}: \prod_a(s_{\partial a})=1} \left\{ E_a(s_{\partial a}) + \sum_{i \in \partial a} M_{i \rightarrow a}(s_i) \right\} \quad (72)$$

$$\Delta E_{ia} = \lim_{\beta \rightarrow \infty} \Delta F_{ia} = \min_{s_i} \{ M_{a \rightarrow i}(s_i) + M_{i \rightarrow a}(s_i) \} \quad (73)$$

If the ground state is unique, it can be found from the expression of the marginal (50):

$$s_i^* = \operatorname{argmin}_{s_i} \left(E_i(s_i) + \sum_{a \in \partial i} M_{a \rightarrow i}(s_i) \right). \quad (74)$$

However, as for the BP equations, this is only guaranteed to be correct on a tree. The condition of uniqueness of the ground state can be ensured by adding a small noise perturbation to single site energetic terms E_i . On graphs with loops, the iteration often fail to converge. A general strategy both to deal with this situation and to find solutions of CSP at non-zero temperature is presented in the next section.

5.3 Finding a Solution: Decimation and Reinforcement Algorithms

Suppose we want to find a solution of a constraint satisfaction problem described by (38), and that we are able to compute (perhaps approximately) the local marginals $\mu_i(s_i)$. We will introduce two heuristic approaches based on message passing: decimation and reinforcement.

5.3.1 Decimation

A decimation algorithm works as follows.

1. Compute the $\mu_i(s_i)$ (via BP, or minsum, or another algorithm).
2. Fix the most biased³ variable i_{\max} according to the local marginal: $i_{\max} = \operatorname{argmax}_i (\max_{s_i} \mu_i(s_i) - \operatorname{sndmax}_{s_i} \mu_i(s_i))$, where sndmax is the second maximum function: $\operatorname{sndmax}_x f(x) = \max_{x \neq \operatorname{argmax}_y f(y)} f(x)$. Fixing a variable reduces the problem to a new problem with $N - 1$ variables.
3. Repeat the above steps until all variables are fixed.

Note that as long as we can compute the local marginals exactly the above decimation algorithm will end up with the optimal solution, if one solution exists. If the marginals are approximate, this is not guaranteed, and the actual results depend on the problem.

An easy modification which can speed up the algorithm is to fix a small fraction of the variables at a time in step 2, rather than only one.

5.3.2 Reinforcement

Another similar approach is to fix the variables smoothly, by introducing a reinforcement field which changes during the iterative message passing propagation. Let us consider the iterative BP algorithm where messages are initialized at random (or uniformly) and updated in succession according to Eqs. (44) and (45), and let us denote with τ the iterative step.⁴ The reinforced BP equations read:

$$\mu_{i \rightarrow a}^{\tau+1}(s_i) = \frac{1}{z_i^{\tau+1}} e^{h_i^\tau(s_i)} e^{-\beta E_i(s_i)} \prod_{b \in \partial i \setminus a} \mu_{b \rightarrow i}^\tau(s_i) \quad (75)$$

$$\mu_{a \rightarrow i}^\tau(s_i) = \frac{1}{z_{a \rightarrow i}^\tau} \sum_{s_{\partial a \setminus i}} \mathbb{I}_a(s_{\partial a}) e^{-\beta E_a(s_{\partial a})} \prod_{j \in \partial a \setminus i} \mu_{j \rightarrow a}^\tau(s_j) \quad (76)$$

where we introduced the reinforcement field h_i^τ :

$$h_i^\tau(s_i) = r(\tau) \ln \mu_i^\tau(s_i) \quad (77)$$

$$\mu_i^{\tau+1}(s_i) = \frac{1}{z_i^{\tau+1}} e^{h_i^\tau(s_i)} e^{-\beta E_i(s_i)} \prod_{a \in \partial i} \mu_{a \rightarrow i}^\tau(s_i). \quad (78)$$

³Using the most biased variable is a simple and reasonable heuristic which works well in practice, but other strategies may be considered.

⁴One step may correspond to one update of all messages (*synchronous* update scheme), or more often to the update of the messages associated to one randomly chosen variable or function node (*asynchronous* update scheme).

Here, $r(\tau)$ is the reinforcement parameter: it is initialized as $r(0) = 0$ and increased slowly, until the algorithm converges to a single solution of the problem, i.e. until $\mu_i^\tau(s_i) \simeq \delta_{s_i, s_i^*}$. The underlying idea is to use the approximate marginals obtained after τ iterations, and use them to bias the problem in the following iterations: rather like decimation, but in an ongoing fashion. Compared to decimation, this reinforced iterative scheme has the advantage of being potentially quicker [this depends on the growth rate of $r(\tau)$], and of being applicable in some cases even when the standard BP equations don't admit a single fixed point.

The reinforcement scheme can be straightforwardly extended to the minsum algorithm of Sect. 5.2 by taking the limit $\beta \rightarrow \infty$.

5.4 Replica Symmetry Breaking and Higher Levels of BP

The multiple-BP-fixed-points situation mentioned at the end of Sect. 5.1 may occur e.g. when the space of solutions is fragmented in clusters, each one corresponding to a different fixed point of the BP equations (the clustering property is valid within each cluster, but not globally). In the context of the statistical physics of disordered systems, this phenomenon is known under the name of *replica symmetry breaking* (RSB), while the situation where the space of solutions forms a single connected cluster is indicated as *replica symmetric* (RS). These names originate from the studies on the typical structure of the phase space in disordered models via the so-called *replica method*; the computations are carried over by using the saddle point method, and the structure of said saddle point is described in terms of its level of symmetry breaking, with the symmetric solution being called RS and the successive levels 1RSB, 2RSB, etc., up to full-RSB. When the correct level of RSB is not taken into account properly, the estimates of the order parameters are only approximate. The underlying theory (see [12] for a full exposition) is not rigorously proven yet, but has been highly successful: as yet, its results on a wide class of models (called mean-field models) have always been in agreement with rigorous theoretical results (whenever available), and with numerical experiments. Note that the last statement does not contain a precise conjecture and is somehow tautological, as it is customary to define mean-field models as the ones that can be resolved by this theory.

The BP equations are apt to describe problems at the RS level, and thus fail to give correct results when RSB occurs. The cavity method approach is nonetheless still applicable: for example, a correct description at the 1RSB level can be obtained by propagating messages which describe probability distributions over BP messages. Higher levels of RSB can in principle be dealt with in the same way, by using distributions over distributions of messages, and so on. The detailed procedure can be in general described as follows: given a constraint satisfaction problem, write the corresponding BP equations; then, consider the BP messages as the variables of a new problem, and the BP equations as its constraints over such variables, and write

higher-level BP equations for this new meta-model.⁵ Iterate this procedure for the desired level of symmetry breaking.

More formally, a 1RSB description is obtained by studying the model described by the following partition function:

$$Z_{1RSB} = \sum_{\mu} e^{-m\beta F(\mu_{\partial i}, \mu_{\partial a})} \prod_i \mathbb{I}_i(\mu_{\partial i}) \prod_a \mathbb{I}_a(\mu_{\partial a}) \quad (79)$$

where

- $\mu_{\partial i} = \{\mu_{i \rightarrow a} | a \in \partial i\} \cup \{\mu_{a \rightarrow i} | a \in \partial i\}$, and analogous for $\mu_{\partial a}$;
- \mathbb{I}_i and \mathbb{I}_a enforce the BP equations (44) and (45);
- $F(\mu_{\partial i}, \mu_{\partial a})$ is the Bethe free energy (63);
- m is the so-called *Parisi parameter*, which can be used to control how the different clusters of configurations are weighted with respect to each other. The RS solution corresponds to $m = 1$, while in the RSB phase $m < 1$. Setting $m = 0$ with β finite amounts at weighting all clusters equally. The limit $m \rightarrow 0$, $\beta \rightarrow \infty$ with $y = m\beta$ finite can be used to weight each cluster α according to its average energy E^α , as e^{-yE^α} .

The procedure as sketched above can of course have prohibitive computational costs, but not necessarily: for example, hard instances of the random K -SAT problem, where 1-step symmetry breaking occurs, can be efficiently solved by using the so-called Survey Propagation algorithm, which is equivalent to a simplified version of a 2-level BP, with a decimation procedure used to identify a solution (see [15, 16]).

References

1. R. Reif, *Fundamentals of Statistical and Thermal Physics* (McGraw-Hill, New York, 1965)
2. S. Ma, *Statistical Mechanics* (World Scientific, Singapore, 1985)
3. K. Huang, *Statistical Mechanics* (Wiley, New York, 1967)
4. R.S. Ellis, *Entropy, Large Deviations, and Statistical Mechanics* (Springer, New York, 1985)
5. P. Erdős, A. Rényi, On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.* **5**, 17 (1960)
6. B. Bollobás, *Random Graphs* (Academic, New York, 1985)
7. R. Potts, *Proc. Camb. Philos. Soc.* **48**, 106 (1952)
8. P. Kasteleyn, C. Fortuin, *J. Phys. Soc. Jpn. Suppl.* **26**, 1114 (1969)
9. F. Wu, The potts model. *Rev. Mod. Phys.* **54**, 235 (1982)
10. A. Engel, R. Monasson, A.K. Hartmann, On large-deviations properties of Erdős-Rényi random graphs. *J. Stat. Phys.* **117**, 387 (2004)

⁵This requires to extend the BP equations to models with continuous variables, which was omitted here for simplicity, but is rather straightforward.

11. O. Martin, R. Monasson, R. Zecchina, Statistical mechanics methods and phase transitions in optimization problems. *Theor. Comput. Sci.* **265**, 3 (2001)
12. M. Mézard, G. Parisi, M.A. Virasoro, *Spin-Glass Theory and Beyond*. Lecture Notes in Physics, vol. 9 (World Scientific, Singapore, 1987)
13. M. Mézard, A. Montanari, *Information, Physics, and Computation* (Oxford University Press, Oxford, 2009)
14. M. Talagrand, Concentration of measure and isoperimetric inequalities in product spaces. *Publ. Math. de L'IHÉS* **81**(1), 73–205 (1995)
15. M. Mézard, R. Zecchina, Random K-satisfiability: from an analytic solution to a new efficient algorithm. *Phys. Rev. E* **66**, 056126 (2002)
16. A. Braunstein, M. Mézard, R. Zecchina, Survey propagation: an algorithm for satisfiability. *Random Struct. Algorithm* **27**, 201–226 (2005)

Graphical Models and Message-Passing Algorithms: Some Introductory Lectures

Martin J. Wainwright

1 Introduction

Graphical models provide a framework for describing statistical dependencies in (possibly large) collections of random variables. At their core lie various correspondences between the conditional independence properties of a random vector, and the structure of an underlying graph used to represent its distribution. They have been used and studied within many sub-disciplines of statistics, applied mathematics, electrical engineering and computer science, including statistical machine learning and artificial intelligence, communication and information theory, statistical physics, network control theory, computational biology, statistical signal processing, natural language processing and computer vision among others.

The purpose of these notes is to provide an introduction to the basic material of graphical models and associated message-passing algorithms. We assume only that the reader has undergraduate-level background in linear algebra, multivariate calculus, probability theory (without needing measure theory), and some basic graph theory. These introductory lectures should be viewed as a pre-cursor to the monograph [64], which focuses primarily on some more advanced aspects of the theory and methodology of graphical models.

M.J. Wainwright (✉)

Department of Statistics, UC Berkeley, Berkeley, CA 94720, USA

e-mail: wainwrig@stat.berkeley.edu

© Springer International Publishing Switzerland 2015

F. Fagnani et al. (eds.), *Mathematical Foundations of Complex Networked Information Systems*, Lecture Notes in Mathematics 2141,

DOI 10.1007/978-3-319-16967-5_3

2 Probability Distributions and Graphical Structure

In this section, we define various types of graphical models, and discuss some of their properties. Before doing so, let us introduce the basic probabilistic notation used throughout these notes. Any graphical model corresponds to a family of probability distributions over a random vector $X = (X_1, \dots, X_N)$. Here for each $s \in [N] := \{1, 2, \dots, N\}$, the random variable X_s take values in some space \mathcal{X}_s , which (depending on the application) may either be continuous (e.g., $\mathcal{X}_s = \mathbb{R}$) or discrete (e.g., $\mathcal{X}_s = \{0, 1, \dots, m-1\}$). Lower case letters are used to refer to particular elements of \mathcal{X}_s , so that the notation $\{X_s = x_s\}$ corresponds to the event that the random variable X_s takes the value $x_s \in \mathcal{X}_s$. The random vector $X = (X_1, X_2, \dots, X_N)$ takes values in the Cartesian product space $\prod_{s=1}^N \mathcal{X}_s := \mathcal{X}_1 \times \mathcal{X}_2 \times \dots \times \mathcal{X}_N$. For any subset $A \subseteq [N]$, we define the subvector $X_A := (X_s, s \in A)$, corresponding to a random vector that takes values in the space $\mathcal{X}_A = \prod_{s \in A} \mathcal{X}_s$. We use the notation $x_A := (x_s, s \in A)$ to refer to a particular element of the space \mathcal{X}_A . With this convention, note that $\mathcal{X}_{[N]}$ is shorthand notation for the full Cartesian product $\prod_{s=1}^N \mathcal{X}_s$. Given three disjoint subsets A, B, C of $[N]$, we use $X_A \perp\!\!\!\perp X_B \mid X_C$ to mean that the random vector X_A is conditionally independent of X_B given X_C . When C is the empty set, then this notion reduces to marginal independence between the random vectors X_A and X_B .

2.1 Directed Graphical Models

We begin our discussion with directed graphical models, which (not surprisingly) are based on the formalism of directed graphs. In particular, a directed graph $\mathcal{D} = (\mathcal{V}, \vec{\mathcal{E}})$ consists of a vertex set $\mathcal{V} = \{1, \dots, N\}$ and a collection $\vec{\mathcal{E}}$ of directed pairs $(s \rightarrow t)$, meaning that s is connected by an edge directed to t . When there exists a directed edge $(t \rightarrow s) \in \mathcal{E}$, we say that node s is a *child* of node t , and conversely that node t is a *parent* of node s . We use $\pi(s)$ to denote the set of all parents of node s (which might be an empty set). A *directed cycle* is a sequence of vertices $(s_1, s_2, \dots, s_\ell)$ such that $(s_\ell \rightarrow s_1) \in \vec{\mathcal{E}}$, and $(s_j \rightarrow s_{j+1}) \in \vec{\mathcal{E}}$ for all $j = 1, \dots, \ell-1$. A *directed acyclic graph*, or DAG for short, is a directed graph that contains no directed cycles. As an illustration, the graphs in panels (a) and (b) are both DAGs, whereas the graph in panel (c) is *not* a DAG, since it contains (among others) a directed cycle on the three vertices $\{1, 2, 5\}$.

Any mapping $\rho : [N] \rightarrow [N]$ defines an ordering of the vertex set $\mathcal{V} = \{1, 2, \dots, N\}$, and of interest to us are particular orderings.

Definition 1 The ordering $\{\rho(1), \dots, \rho(N)\}$ of the vertex set \mathcal{V} of a DAG is *topological* if for each $s \in \mathcal{V}$, we have $\rho(t) < \rho(s)$ for all $t \in \pi(s)$.

Alternatively stated, in a topological ordering, children always come after their parents. It is an elementary fact of graph theory that any DAG has at least one topological ordering, and this fact plays an important role in our analysis of directed graphical models. So as to simplify our presentation, we assume throughout these notes that the *canonical ordering* $\mathcal{V} = \{1, 2, \dots, N\}$ is topological. Note that this assumption entails no loss of generality, since we can always re-index the vertices so that it holds. With this choice of topological ordering, vertex 1 cannot have any parents (i.e., $\pi(1) = \emptyset$), and moreover vertex N cannot have any children.

With this set-up, we are now ready to introduce probabilistic notions into the picture. A directed graphical model is a family of probability distributions defined by a DAG. This family is built by associating each node s of a DAG with a random variable X_s , and requiring the joint probability distribution over (X_1, \dots, X_N) factorize according to the DAG. Consider the subset of vertices $(s, \pi(s))$ corresponding to a given vertex s and its parents $\pi(s)$. We may associate with this subset a real-valued function $f_s : \mathcal{X}_s \times \mathcal{X}_{\pi(s)} \rightarrow \mathbb{R}_+$ that maps any given configuration $(x_s, x_{\pi(s)}) \in \mathcal{X}_s \times \mathcal{X}_{\pi(s)}$ to a real number $f_s(x_s, x_{\pi(s)}) \geq 0$. We assume moreover that f_s satisfies the normalization condition

$$\sum_{x_s} f_s(x_s, x_{\pi(s)}) = 1 \quad \text{for all } x_{\pi(s)} \in \mathcal{X}_{\pi(s)}. \quad (1)$$

Definition 2 (Factorization for Directed Graphical Models) The *directed graphical model* based on a given DAG \mathcal{D} is the collection of probability distributions over the random vector (X_1, \dots, X_N) that have a factorization of the form

$$p(x_1, \dots, x_N) = \frac{1}{Z} \prod_{s=1}^N f_s(x_s, x_{\pi(s)}), \quad (2)$$

for some choice of non-negative parent-to-child functions (f_1, \dots, f_N) that satisfy the normalization condition (1). We use $\mathcal{F}_{\text{Fac}}(\mathcal{D})$ to denote the set of all distributions that factorize in the form (2).

In the factorization (2), the quantity Z denotes a constant chosen to ensure that p sums to one.

Let us illustrate this definition with some examples.

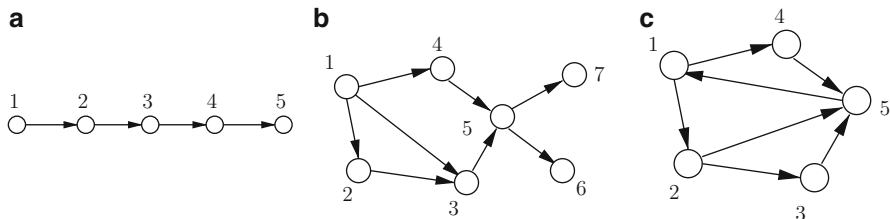


Fig. 1 (a) The simplest example of a DAG is a chain, which underlies the familiar Markov chain. The canonical ordering $\{1, 2, \dots, N\}$ is the only topological one. (b) A more complicated DAG. Here the canonical ordering $\{1, 2, \dots, 7\}$ is again topological, but it is no longer unique: for instance, $\{1, 4, 2, 3, 5, 7, 6\}$ is also topological. (c) A directed graph with cycles (non-DAG). It contains (among others) a directed cycle on vertices $\{1, 2, 5\}$

Example 1 (Markov Chain as a Directed Graphical Model) Perhaps the simplest example of a directed acyclic graph is the chain on N nodes, as shown in panel (a) of Fig. 1. Such a graph underlies the stochastic process (X_1, \dots, X_N) known as a Markov chain, used to model various types of sequential dependencies. By definition, any Markov chain can be factorized in the form

$$p(x_1, \dots, x_N) = p(x_1) p(x_2 | x_1) p(x_3 | x_2) \cdots p(x_N | x_{N-1}). \quad (3)$$

Note that this is a special case of the factorization (2), based on the functions $f_s(x_s, x_{\pi(s)}) = p(x_s | x_{s-1})$ for each $s = 2, \dots, N$, and $f_1(x_1, x_{\pi(1)}) = p(x_1)$.



We now turn to a more complex DAG.

Example 2 (Another DAG) Consider the DAG shown in Fig. 1b. It defines the family of probability distributions that have a factorization of the form

$$p(x_1, \dots, x_7) \propto f_1(x_1) f_2(x_2, x_1) f_3(x_3, x_1, x_2) f_4(x_4, x_1) f_5(x_5, x_3, x_4) f_6(x_6, x_5) f_7(x_7, x_5).$$

for some collection of non-negative and suitably normalized functions $\{f_s, s \in \mathcal{V}\}$.



The factorization (3) of the classical Markov chain has an interesting property, in that the normalization constant $Z = 1$, and all the local functions f_s are equal to conditional probability distributions. It is not immediately apparent whether or not this property holds for the more complex DAG discussed in Example 2, but in fact, as shown by the following result, it is a generic property of directed graphical models.

Proposition 1 *For any directed acyclic graph \mathcal{D} , any factorization of the form factorization (2) with $Z = 1$ defines a valid probability distribution. Moreover, we necessarily have $f_s(x_s, x_{\pi(s)}) = p(x_s | x_{\pi(s)})$ for all $s \in \mathcal{V}$.*

Proof Throughout the proof, we assume without loss of generality (re-indexing as necessary) that $\{1, 2, \dots, N\}$ is a topological ordering. In order to prove this result, it is convenient to first state an auxiliary result.

Lemma 1 *For any distribution p of the form (2), we have*

$$p(x_1, \dots, x_t) = \frac{1}{Z} \prod_{s=1}^t f_s(x_s, x_{\pi(s)}) \quad \text{for each } t = 1, \dots, N. \quad (4)$$

We first use this result to establish the main claims before returning to prove it. If we apply Lemma 1 with $t = 1$, then we obtain that $p(x_1) = f_1(x_1)/Z$, and hence that $Z = 1$ by the normalization condition on f_1 . Otherwise, for any $t \in \{2, \dots, N\}$, applying the representation (4) to both t and $t - 1$ yields

$$\frac{p(x_1, \dots, x_t)}{p(x_1, \dots, x_{t-1})} = f_t(x_t, x_{\pi(t)}) \quad \text{for all } (x_1, \dots, x_t).$$

Since the right-hand side depends only on $x_{\pi(t)}$, so must the left-hand side. When the left-hand side depends only on $x_{\pi(t)}$, then it is equal to the conditional $p(x_t | x_{\pi(t)})$, and so we conclude that $p(x_t | x_{\pi(t)}) = f_t(x_t, x_{\pi(t)})$ as claimed.

It remains to prove Lemma 1. We may assume without loss of generality (re-indexing as necessary) that $\{1, 2, \dots, N\}$ is a topological ordering. Consequently, node N has no children, so that we may write

$$p(x_1, \dots, x_{N-1}, x_N) = \frac{1}{Z} \left[\prod_{s=1}^{N-1} f_s(x_s, x_{\pi(s)}) \right] f_N(x_N, x_{\pi(N)}).$$

Marginalizing over x_N yields that

$$\begin{aligned} p(x_1, \dots, x_{N-1}) &= \frac{1}{Z} \left[\prod_{s=1}^{N-1} f_s(x_s, x_{\pi(s)}) \right] \left[\sum_{x_N} f_N(x_N, x_{\pi(N)}) \right] \\ &= \frac{1}{Z} \prod_{s=1}^{N-1} f_s(x_s, x_{\pi(s)}), \end{aligned}$$

where we have used the facts that x_N appears only in one term (since N is a leaf node), and that $\sum_{x_N} f_N(x_N, x_{\pi(N)}) = 1$. We have thus shown that if the claim holds for $t = N$, then it holds for $t = N - 1$. By recursively applying this same argument, the claim of Lemma 1 follows. \square

Proposition 1 shows that the terms f_i in the factorization (2) have a concrete interpretation as the child-parent conditional probabilities (i.e., $f_s(x_s, x_{\pi(s)})$ is equal to the conditional probability of $X_s = x_s$ given that $X_{\pi(s)} = x_{\pi(s)}$). This local interpretability, which (as we will see) is not shared by the class of undirected graphical models, has some important consequences. For instance, sampling a configuration $(\tilde{X}_1, \dots, \tilde{X}_N)$ from any DAG model is straightforward: assuming the canonical topological ordering, we first sample $\tilde{X}_1 \sim f_1(\cdot)$, and then for $s = 2, \dots, N$, sample $\tilde{X}_s \sim f_s(\cdot, \tilde{X}_{\pi(s)})$. This procedure is well-specified: due to the topological ordering, we are guaranteed that the variable $\tilde{X}_{\pi(s)}$ has been sampled before we move on to sampling \tilde{X}_s . Moreover, by construction, the random vector $(\tilde{X}_1, \dots, \tilde{X}_N)$ is distributed according to the probability distribution (2).

2.1.1 Conditional Independence Properties for Directed Graphs

Thus far, we have specified a joint distribution over the random vector $X = (X_1, \dots, X_N)$ in terms of a particular parent-to-child factorization. We now turn to a different (but ultimately equivalent) characterization in terms of conditional independence. (The reader should recall our standard notation for conditional independence properties from the beginning of Sect. 2.) Throughout the discussion to follow, we continue to assume that the canonical ordering $\{1, 2, \dots, N\}$ is topological.

Given any vertex $s \in \mathcal{V} \setminus \{1\}$, our choice of topological ordering implies that the parent set $\pi(s)$ is contained within the set $\{1, 2, \dots, s - 1\}$. Note that for $s = 1$, the parent set must be empty. We then define the set $v(s) = \{1, 2, \dots, s - 1\} \setminus \pi(s)$. Our basic conditional independence properties are based on the three disjoint subsets $\{s\}$, $\pi(s)$ and $v(s)$.

Definition 3 (Markov Property for Directed Graphical Models) The random vector $X = (X_1, \dots, X_N)$ is Markov with respect to a directed graph if

$$X_s \perp\!\!\!\perp X_{v(s)} \mid X_{\pi(s)} \quad \text{for all } s \in \mathcal{V}. \quad (5)$$

We use $\mathcal{F}_{\text{Mar}}(\mathcal{D})$ to denote the set of all distributions that are Markov with respect to \mathcal{D} .

Let us illustrate this definition with our running examples.

Example 3 (Conditional Independence for Directed Markov Chain) Recall the directed Markov chain first presented in Example 1. Each node $s \in \{2, \dots, N\}$ has a unique parent $\pi(s) = s - 1$, so that the (non-trivial) basic conditional properties are of the form $X_s \perp\!\!\!\perp (X_1, \dots, X_{s-2}) \mid X_{s-1}$ for $s \in \{3, \dots, N\}$. Note that these basic conditional independence statements imply other (non-basic) properties as well. Perhaps the most familiar is the assertion that

$$(X_s, X_{s+1}, \dots, X_N) \perp\!\!\!\perp (X_1, \dots, X_{s-2}) \mid X_{s-1} \quad \text{for all } s \in \{3, \dots, N\},$$

corresponding the fact that the past and future of a Markov chain are conditionally independent given the present (X_{s-1} in this case). \clubsuit

As a second example, let us now return to the DAG shown in Fig. 1b.

Example 4 (Conditional Independence for a More Complex DAG) For this DAG, the basic conditional independence assertions (using the canonical ordering) can again be read off from the graph. In particular, the non-trivial relations are $X_3 \perp\!\!\!\perp X_1 \mid X_2$, $X_5 \perp\!\!\!\perp (X_1, X_2) \mid (X_3, X_4)$, as well as

$$X_6 \perp\!\!\!\perp (X_1, X_2, X_3, X_4) \mid X_5, \quad \text{and} \quad X_7 \perp\!\!\!\perp (X_1, X_2, X_3, X_4, X_6) \mid X_5.$$

\clubsuit

2.1.2 Equivalence of Representations

For any directed graph \mathcal{D} , we have now defined two families of probability distributions: the family $\mathcal{F}_{\text{Fac}}(\mathcal{D})$ of all distributions with a factorization of the form (2), and the family $\mathcal{F}_{\text{Mar}}(\mathcal{D})$ of all distributions that satisfy the basic Markov properties (5). It is natural to ask how these two families are related; pleasingly, they are equivalent, as shown in the following result.

Theorem 1 (Equivalence for Directed Graphical Models) *For any directed acyclic graph \mathcal{D} , we have $\mathcal{F}_{\text{Fac}}(\mathcal{D}) = \mathcal{F}_{\text{Mar}}(\mathcal{D})$.*

Proof The proof of this result is straightforward given our development thus far. We begin with the inclusion $\mathcal{F}_{\text{Mar}}(\mathcal{D}) \subseteq \mathcal{F}_{\text{Fac}}(\mathcal{D})$. From Lemma 1 used in the proof of Proposition 1, for any vertex $t \in \mathcal{V}$, we have

$$\begin{aligned} p(x_1, \dots, x_t) &= \prod_{s=1}^t p(x_s \mid x_{\pi(s)}) = \prod_{s=1}^{t-1} p(x_s \mid x_{\pi(s)}) p(x_t \mid x_{\pi(t)}) \\ &= p(x_1, \dots, x_{t-1}) p(x_t \mid x_{\pi(t)}). \end{aligned}$$

By the definition of $\nu(t)$, we have $\{1, \dots, t-1\} = \pi(t) \cup \nu(t)$, and consequently we can write

$$\frac{p(x_{\pi(t)}, x_{\nu(t)}, x_t)}{p(x_{\pi(t)})} = \frac{p(x_{\pi(t)}, x_{\nu(t)})}{p(x_{\pi(t)})} p(x_t | x_{\pi(t)}) = p(x_{\nu(t)} | x_{\pi(t)}) p(x_t | x_{\pi(t)}),$$

which shows that $X_t \perp\!\!\!\perp X_{\nu(t)} | X_{\pi(t)}$.

In order to establish the reverse inclusion, suppose that the basic Markov properties hold, where we are still using $\{1, 2, \dots, N\}$ as our topological ordering. Using the chain rule for probability, we have

$$\begin{aligned} p(x_1, \dots, x_N) &= p(x_1) \prod_{s=2}^N p(x_s | x_1, \dots, x_{s-1}) \\ &\stackrel{(i)}{=} p(x_1) \prod_{s=2}^N p(x_s | x_{\pi(s)}), \end{aligned}$$

where equality (i) follows by applying the Markov properties. \square

2.2 Undirected Graphical Models

We now turn to discussion of undirected graphical models, which are also known as *Markov random fields* or *Gibbs distributions*. Naturally, these models are built using an undirected graphs, by which we mean a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, N\}$ is the vertex set (as before), and \mathcal{E} is a collection of undirected edges, meaning that there is no distinction between the edge (s, t) and the edge (t, s) . As before, we associate a random variable X_s with each vertex $s \in \mathcal{V}$ of graph, and our interest is in characterizing the joint distribution of the random vector $X = (X_1, \dots, X_N)$.

As with directed graphical models, there are two different ways in which the probabilistic structure of the random vector X can be linked to the graphical structure: factorization and conditional independence properties. Let us begin our exploration with the former property.

2.2.1 Factorization for Undirected Models

For undirected graphical models, the factorization properties are specified in terms of cliques of the graph. A *clique* C of an undirected graph \mathcal{G} is a fully connected subset C of the vertex set \mathcal{V} (i.e., $(s, t) \in \mathcal{E}$ for all $s, t \in C$). A clique is *maximal* if it is not contained within any other clique. Thus, any singleton set $\{s\}$ is always a clique, but it is not maximal unless s has no neighbors in the graph. See Fig. 2a for an illustration of some other types of cliques.

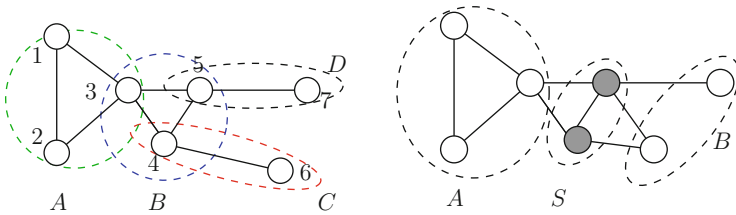


Fig. 2 (a) Illustration of cliques in an undirected graph. Sets $A = \{1, 2, 3\}$ and $B = \{3, 4, 5\}$ are 3-cliques, whereas sets $C = \{4, 6\}$ and $D = \{5, 7\}$ are 2-cliques (that can be identified with edges). (b) Illustration of a vertex cutset: removal of the vertices in S breaks the graph into the separate pieces indexed by subsets A and B

Given an undirected graph, we use \mathcal{C} to denote the set of all its cliques. With each clique $C \in \mathcal{C}$, we associate a *compatibility function* $\psi_C : \mathcal{X}_C \rightarrow \mathbb{R}_+$: it assigns a non-negative number $\psi_C(x_C)$ to each possible configuration $x_C = (x_s, s \in C) \in \mathcal{X}_C$. The factorization property for an undirected graphical model is stated in terms of these compatibility functions.

Definition 4 (Factorization for Undirected Graphical Models) A probability distribution p factorizes over the graph \mathcal{G} if

$$p(x_1, \dots, x_N) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C), \tag{6}$$

for some choice of non-negative compatibility function $\psi_C : \mathcal{X}_C \rightarrow \mathbb{R}_+$ for each clique $C \in \mathcal{C}$. We use $\mathcal{F}_{\text{Fac}}(\mathcal{G})$ to denote the set of all distributions that have a factorization of the form (6).

As before, the quantity Z is a constant chosen to ensure that the distribution is appropriately normalized. (In contrast to the directed case, we cannot take $Z = 1$ in general.) Note that there is a great deal of freedom in the factorization (6); in particular, in the way that we have written it, there is no unique choice of the compatibility functions and the normalization constant Z . For instance, if we multiply any compatibility function by some constant $\alpha > 0$ and do the same to Z , we obtain the same distribution.

Let us illustrate these definitions with some examples.

Example 5 (Markov Chain as an Undirected Graphical Model) Suppose that we remove the arrows from the edges of the chain graph shown in Fig. 1a; we then obtain an undirected chain. Any distribution that factorizes according to this

undirected chain has the form

$$p(x_1, \dots, x_5) = \frac{1}{Z} \prod_{s=1}^5 \psi_s(x_s) \psi_{12}(x_1, x_2) \psi_{12}(x_1, x_2) \psi_{12}(x_1, x_2) \psi_{12}(x_1, x_2) \quad (7)$$

for some choice of non-negative compatibility functions. Unlike the directed case, in general, these compatibility functions are *not* equal to conditional distributions, nor to marginal distributions. Moreover, we have no guarantee that $Z = 1$ in this undirected representation of a Markov chain. The factorization (7) makes use of both maximal cliques (edges) and non-maximal cliques (singletons). Without loss of generality, we can always restrict to only edge-based factors. However, for applications, it is often convenient, for the purposes of interpretability, to include both types of compatibility functions. ♣

Example 6 As a second example, consider the undirected graph shown in Fig. 2a. Any distribution that respects its clique structure has the factorization

$$p(x_1, \dots, x_7) = \frac{1}{Z} \psi_{123}(x_1, x_2, x_3) \psi_{345}(x_3, x_4, x_5) \psi_{4.6}(x_4, x_6) \psi_{57}(x_5, x_7),$$

for some choice of non-negative compatibility functions. In this case, we have restricted our factorization to the maximal cliques of the graph. ♣

2.2.2 Markov Property for Undirected Models

We now turn to the second way in which graph structure can be related to probabilistic structure, namely via conditional independence properties. In the case of undirected graphs, these conditional independence properties are specified in terms of vertex subsets of the graph. A *vertex cutset* is a subset S of the vertex set \mathcal{V} such that, when it is removed from the graph, breaks the graph into two or more disconnected subsets of vertices. For instance, as shown in Fig. 2b, removing the subset S of vertices breaks the graph into two disconnected components, as indexed by the vertices in A and B respectively. For an undirected graph, the Markov property is defined by these vertex cutsets:

Definition 5 (Markov Property for Undirected Graphical Models) A random vector (X_1, \dots, X_N) is *Markov with respect to an undirected graph* \mathcal{G} if, for all vertex cutsets S and associated components A and B , the conditional independence condition $X_A \perp\!\!\!\perp X_B \mid X_C$ holds. We use $\mathcal{F}_{\text{Mar}}(\mathcal{G})$ to denote the set of all distributions that satisfy all such Markov properties defined by \mathcal{G} .

For example, for any vertex s , its *neighborhood set*

$$\mathcal{N}(s) := \{t \in \mathcal{V} \mid (s, t) \in \mathcal{E}\} \quad (8)$$

is always a vertex cutset. Consequently, any distribution that is Markov with respect to an undirected graph satisfies the conditional independence relations

$$X_s \perp\!\!\!\perp X_{\mathcal{V} \setminus \{s\} \cup \mathcal{N}(s)} \mid X_{\mathcal{N}(s)}. \quad (9)$$

The set of variables $X_{\mathcal{N}(s)} = (X_t, t \in \mathcal{N}(s))$ is often referred to as the *Markov blanket* of X_s , since it is the minimal subset required to render X_s conditionally independent of all other variables in the graph. This particular Markov property plays an important role in pseudolikelihood estimation of parameters [11], as well as related approaches for graphical model selection by neighborhood regression (see the papers [48, 52] for further details).

Let us illustrate with some additional concrete examples.

Example 7 (Conditional Independence for a Markov Chain) Recall from Example 5 the view of a Markov chain as an undirected graphical model. For this chain graph, the minimal vertex cutsets are singletons, with the non-trivial ones being the subsets $\{s\}$ for $s \in \{2, \dots, N-1\}$. Each such vertex cutset induces the familiar conditional independence relation

$$\underbrace{(X_1, \dots, X_{s-1})}_{\text{Past}} \perp\!\!\!\perp \underbrace{(X_{s+1}, \dots, X_N)}_{\text{Future}} \mid \underbrace{X_s}_{\text{Present}}, \quad (10)$$

corresponding to the fact that the past and future of a Markov chain are conditionally independent given the present. ♣

Example 8 (More Markov Properties) Consider the undirected graph shown in Fig. 2b. As previously discussed, the set S is a vertex cutset, and hence it induces the conditional independence property $X_A \perp\!\!\!\perp X_B \mid X_S$. ♣

2.2.3 Hammersley-Clifford Equivalence

As in the directed case, we have now specified two ways in which graph structure can be linked to probabilistic structure. This section is devoted to a classical result that establishes a certain equivalence between factorization and Markov properties, meaning the two families of distributions $\mathcal{F}_{\text{Fac}}(\mathcal{G})$ and $\mathcal{F}_{\text{Mar}}(\mathcal{G})$ respectively. For strictly positive distributions p , this equivalence is complete.

Theorem 2 (Hammersley-Clifford) *If the distribution p of random vector $X = (X_1, \dots, X_N)$ factorizes over a graph \mathcal{G} , then X is Markov with respect to \mathcal{G} . Conversely, if a random vector X is Markov with respect to \mathcal{G} and $p(x) > 0$ for all $x \in \mathcal{X}_{[N]}$, then p factorizes over the graph.*

Proof We begin by proving that $\mathcal{F}_{\text{Fac}}(\mathcal{G}) \subseteq \mathcal{F}_{\text{Mar}}(\mathcal{G})$. Suppose that the factorization (6) holds, and let S be an arbitrary vertex cutset of the graph such that subsets A and B are separated by S . We may assume without loss of generality that both A and B are non-empty, and we need to show that $X_A \perp\!\!\!\perp X_B \mid X_S$. Let us define subsets of cliques by $\mathcal{C}_A := \{C \in \mathcal{C} \mid C \cap A \neq \emptyset\}$, $\mathcal{C}_B := \{C \in \mathcal{C} \mid C \cap B \neq \emptyset\}$, and $\mathcal{C}_S := \{C \in \mathcal{C} \mid C \subseteq S\}$. We claim that these three subsets form a disjoint partition of the full clique set—namely, $\mathcal{C} = \mathcal{C}_A \cup \mathcal{C}_S \cup \mathcal{C}_B$. Given any clique C , it is either contained entirely within S , or must have non-trivial intersection with either A or B , which proves the union property. To establish disjointness, it is immediate that \mathcal{C}_S is disjoint from \mathcal{C}_A and \mathcal{C}_B . On the other hand, if there were some clique $C \in \mathcal{C}_A \cap \mathcal{C}_B$, then there would exist nodes $a \in A$ and $b \in B$ with $\{a, b\} \in C$, which contradicts the fact that A and B are separated by the cutset S .

Consequently, we may write

$$p(x_A, x_S, x_B) = \frac{1}{Z} \underbrace{\left[\prod_{C \in \mathcal{C}_A} \psi_C(x_C) \right]}_{\Psi_A(x_A, x_S)} \underbrace{\left[\prod_{C \in \mathcal{C}_S} \psi_C(x_C) \right]}_{\Psi_S(x_S)} \underbrace{\left[\prod_{C \in \mathcal{C}_B} \psi_C(x_C) \right]}_{\Psi_B(x_B, x_S)}.$$

Defining the quantities

$$Z_A(x_S) := \sum_{x_A} \Psi_A(x_A, x_S), \quad \text{and} \quad Z_B(x_S) := \sum_{x_B} \Psi_B(x_B, x_S),$$

we then obtain the following expressions for the marginal distributions of interest

$$p(x_S) = \frac{Z_A(x_S) Z_B(x_S)}{Z} \Psi_S(x_S) \quad \text{and} \quad p(x_A, x_S) = \frac{Z_B(x_S)}{Z} \Psi_A(x_A, x_S) \Psi_S(x_S),$$

with a similar expression for $p(x_B, x_S)$. Consequently, for any x_S for which $p(x_S) > 0$, we may write

$$\begin{aligned} \frac{p(x_A, x_S, x_B)}{p(x_S)} &= \frac{\frac{1}{Z} \Psi_A(x_A, x_S) \Psi_S(x_S) \Psi_B(x_B, x_S)}{\frac{Z_A(x_S) Z_B(x_S)}{Z} \Psi_S(x_S)} \\ &= \frac{\Psi_A(x_A, x_S) \Psi_B(x_B, x_S)}{Z_A(x_S) Z_B(x_S)}. \end{aligned} \tag{11}$$

Similar calculations yield the relations

$$\frac{p(x_A, x_S)}{p(x_S)} = \frac{\frac{Z_B(x_S)}{Z} \Psi_A(x_A, x_S) \Psi_S(x_S)}{\frac{Z_A(x_S)Z_B(x_S)}{Z} \Psi_S(x_S)} = \frac{\Psi_A(x_A, x_S)}{Z_A(x_S)}, \quad \text{and} \quad (12a)$$

$$\frac{p(x_B, x_S)}{p(x_S)} = \frac{\frac{Z_A(x_S)}{Z} \Psi_B(x_B, x_S) \Psi_S(x_S)}{\frac{Z_A(x_S)Z_B(x_S)}{Z} \Psi_S(x_S)} = \frac{\Psi_B(x_B, x_S)}{Z_B(x_S)}. \quad (12b)$$

Combining Eq. (11) with Eqs. (12a) and (12b) yields

$$\begin{aligned} p(x_A, x_B \mid x_S) &= \frac{p(x_A, x_B, x_S)}{p(x_S)} = \frac{p(x_A, x_S)}{p(x_S)} \frac{p(x_B, x_S)}{p(x_S)} \\ &= p(x_A \mid x_S) p(x_B \mid x_S), \end{aligned}$$

thereby showing that $X_A \perp\!\!\!\perp X_B \mid X_S$, as claimed.

In order to prove the opposite inclusion—namely, that the Markov property for a strictly positive distribution implies the factorization property—we require a version of the *inclusion-exclusion formula*. Given a set $[N] = \{1, 2, \dots, N\}$, let $\mathcal{P}([N])$ be its power set, meaning the set of all subsets of $[N]$. With this notation, the following inclusion-exclusion formula is classical:

Lemma 2 (Inclusion-Exclusion) *For any two real-valued functions Ψ and Φ defined on the power set $\mathcal{P}([N])$, the following statements are equivalent:*

$$\Phi(A) = \sum_{B \subseteq A} (-1)^{|A \setminus B|} \Psi(B) \quad \text{for all } A \in \mathcal{P}([N]). \quad (13a)$$

$$\Psi(A) = \sum_{B \subseteq A} \Phi(B) \quad \text{for all } A \in \mathcal{P}([N]). \quad (13b)$$

Proofs of this result can be found in standard texts in combinatorics (e.g., [61]). Returning to the main thread, let $y \in \mathcal{X}_{[N]}$ be some fixed element, and for each subset $A \in \mathcal{P}([N])$, define the function $\phi : \mathcal{X}_{[N]} \rightarrow \mathbb{R}$ via

$$\phi_A(x) := \sum_{B \subseteq A} (-1)^{|A \setminus B|} \log \frac{p(x_B, y_{B^c})}{p(y)}. \quad (14)$$

(Note that taking logarithms is meaningful since we have assumed $p(x) > 0$ for all $x \in \mathcal{X}_{[N]}$.) From the inclusion-exclusion formula, we have $\log \frac{p(x_A, y_{A^c})}{p(y)} =$

$\sum_{B \subseteq A} \phi_B(x)$, and setting $A = [N]$ yields

$$p(x) = p(y) \exp \left\{ \sum_{B \in \mathcal{P}([N])} \phi_B(x) \right\}. \quad (15)$$

From the definition (14), we see that ϕ_A is a function only of x_A . In order to complete the proof, it remains to show that $\phi_A = 0$ for any subset A that is *not* a graph clique. As an intermediate result, we claim that for any $t \in A$, we can write

$$\phi_A(x) = \sum_{B \subseteq A \setminus \{t\}} (-1)^{|A-B|} \log \frac{p(x_t | x_B, y_{B^c \setminus \{t\}})}{p(y_t | x_B, y_{B^c \setminus \{t\}})}. \quad (16)$$

To establish this claim, we write

$$\begin{aligned} \phi_A(x) &= \sum_{\substack{B \subseteq A \\ B \not\ni t}} (-1)^{|A \setminus B|} \log \frac{p(x_B, y_{B^c})}{p(y)} + \sum_{\substack{B \subseteq A \\ B \ni t}} (-1)^{|A \setminus B|} \log \frac{p(x_B, y_{B^c})}{p(y)} \\ &= \sum_{B \subseteq A \setminus \{t\}} (-1)^{|A \setminus B|} \left\{ \log \frac{p(x_B, y_{B^c})}{p(y)} - \log \frac{p(x_{B \cup t}, y_{B^c \setminus \{t\}})}{p(y)} \right\} \\ &= \sum_{B \subseteq A \setminus \{t\}} (-1)^{|A \setminus B|} \log \frac{p(x_B, y_{B^c})}{p(x_{B \cup t}, y_{B^c \setminus \{t\}})} \end{aligned} \quad (17)$$

Note that for any $B \subseteq A \setminus \{t\}$, we are guaranteed that $t \notin B$, whence

$$\frac{p(x_B, y_{B^c})}{p(x_{B \cup t}, y_{B^c \setminus \{t\}})} = \frac{p(y_t | x_B, y_{B^c \setminus \{t\}})}{p(x_t | x_B, y_{B^c \setminus \{t\}})}.$$

Substituting into Eq. (17) yields the claim (16).

We can now conclude the proof. If A is not a clique, then there must some exist some pair (s, t) of vertices *not* joined by an edge. Using the representation (16), we can write ϕ_A as a sum of four terms (i.e., $\phi_A = \sum_{i=1}^4 T_i$), where

$$\begin{aligned} T_1(x) &= \sum_{B \subseteq A \setminus \{s, t\}} (-1)^{|A \setminus B|} \log p(y_t | x_B, y_{B^c \setminus \{t\}}), \\ T_2(x) &= \sum_{B \subseteq A \setminus \{s, t\}} (-1)^{|A \setminus (B \cup \{t\})|} \log p(x_t | x_B, y_{B^c \setminus \{t\}}) \\ T_3(x) &= \sum_{B \subseteq A \setminus \{s, t\}} (-1)^{|A \setminus (B \cup \{s\})|} \log p(y_t | x_{B \cup \{s\}}, y_{B^c \setminus \{s, t\}}), \quad \text{and} \\ T_4(x) &= \sum_{B \subseteq A \setminus \{s, t\}} (-1)^{|A \setminus (B \cup \{s, t\})|} \log p(x_t | x_{B \cup \{s\}}, y_{B^c \setminus \{s, t\}}). \end{aligned}$$

Combining these separate terms, we obtain

$$\phi_A(x) = \sum_{B \subseteq A \setminus \{s,t\}} (-1)^{|A \setminus B|} \log \frac{p(y_t | x_B, y_{B^c \setminus \{t\}}) p(x_t | x_{B \cup \{s\}}, y_{B^c \setminus \{s,t\}})}{p(x_t | x_B, y_{B^c \setminus \{t\}}) p(y_t | x_{B \cup \{s\}}, y_{B^c \setminus \{s,t\}})}$$

But using the Markov properties of the graph, each term in this sum is zero. Indeed, since $s \notin N(t)$, we have

$$p(x_t | x_B, y_{B^c \setminus \{t\}}) = p(x_t | x_{B \cup \{s\}}, y_{B^c \setminus \{s,t\}}), \quad \text{and}$$

$$p(y_t | x_B, y_{B^c \setminus \{t\}}) = p(y_t | x_{B \cup \{s\}}, y_{B^c \setminus \{s,t\}}),$$

which completes the proof. □

2.2.4 Factor Graphs

For large graphs, the factorization properties of a graphical model, whether undirected or directed, may be difficult to visualize from the usual depictions of graphs. The formalism of *factor graphs* provides an alternative graphical representation, one which emphasizes the factorization of the distribution [43, 46].

Let \mathcal{F} represent an index set for the set of factors defining a graphical model distribution. In the undirected case, this set indexes the collection \mathcal{C} of cliques, while in the directed case \mathcal{F} indexes the set of parent–child neighborhoods. We then consider a bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{F}, \mathcal{E})$, in which \mathcal{V} is (as before) an index set for the variables, and \mathcal{F} is an index set for the factors. Given the bipartite nature of the graph, the edge set \mathcal{E} now consists of pairs (s, a) of nodes $s \in \mathcal{V}$ such that the fact $a \in \mathcal{N}(s)$, or equivalently such that $a \in \mathcal{N}(s)$. See Fig. 3b for an illustration.

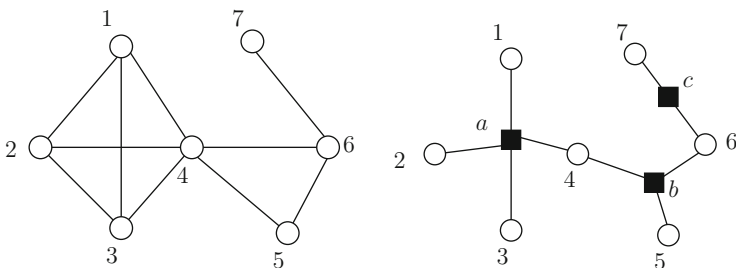


Fig. 3 Illustration of undirected graphical models and factor graphs. (a) An undirected graph on $N = 7$ vertices, with maximal cliques $\{1, 2, 3, 4\}$, $\{4, 5, 6\}$ and $\{6, 7\}$. (b) Equivalent representation of the undirected graph in (a) as a factor graph, assuming that we define compatibility functions only on the maximal cliques in (a). The factor graph is a bipartite graph with vertex set $\mathcal{V} = \{1, \dots, 7\}$ and factor set $\mathcal{F} = \{a, b, c\}$, one for each of the compatibility functions of the original undirected graph

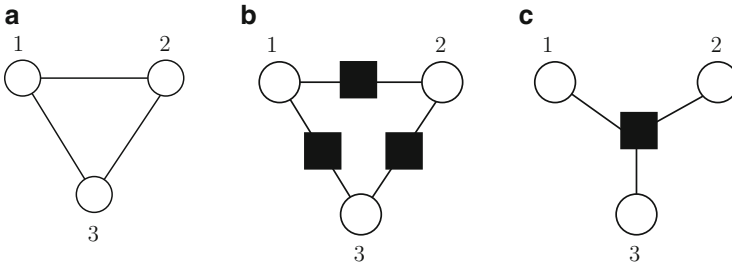


Fig. 4 (a) A three-vertex cycle graph that can be used to represent any distribution over three random variables. (b) Factor graph representation of the pairwise model (19). (c) Factor graph representation of the triplet interaction model

For undirected models, the factor graph representation is of particular value when C consists of more than the maximal cliques. Indeed, the compatibility functions for the nonmaximal cliques do not have an explicit representation in the usual representation of an undirected graph—however, the factor graph makes them explicit. This explicitness can be useful in resolving certain ambiguities that arise with the standard formalism of undirected graphs. As one illustration, consider the three-vertex cycle shown in Fig. 4a. Two possible factorizations that are consistent with this graph structure are the *pairwise-only factorization*

$$p(x_1, x_2, x_3) = \frac{1}{Z} \psi_{12}(x_1, x_2) \psi_{23}(x_2, x_3) \psi_{13}(x_1, x_3), \quad (18)$$

which involves only the non-maximal pairwise cliques, versus the *generic triplet factorization* $p(x_1, x_2, x_3) = \frac{1}{Z'} \psi_{123}(x_1, x_2, x_3)$. The undirected graph in panel (a) does not distinguish between these two possibilities. In contrast, the factor graph representations of these two models are distinct, as shown in panels (b) and (c) respectively. This distinction between the pairwise and triplet factorization is important in many applications, for instance when testing for the presence of ternary interactions between different factors underlying a particular disease.

3 Exact Algorithms for Marginals, Likelihoods and Modes

In applications of graphical models, one is typically interested in solving one of a core set of computational problems. One example is the problem of *likelihood computation*, which arises in parameter estimation and hypothesis testing with graphical models. A closely related problem is that of computing the marginal distribution $p(x_A)$ over a particular subset $A \subset \mathcal{V}$ of nodes, or similarly, computing the conditional distribution $p(x_A | x_B)$, for disjoint subsets A and B of the vertex set. This *marginalization problem* is needed in order to perform filtering or

smoothing of time series (for chain-structured graphs), or the analogous operations for more general graphical models. A third problem is that of *mode computation*, in which the goal is to find a configuration $\hat{x} \in \mathcal{X}_{[N]}$ of highest probability—that is, $\hat{x} \in \max_{x \in \mathcal{X}_{[N]}} p(x)$. In typical applications, some subset of the variables are observed, so these computations are often performed for conditional distributions, with the observed values fixed. It is straightforward to incorporate observed values into the algorithms that we discuss in this section.

At first sight, these problems—likelihood computation, marginalization and mode computation—might seem quite simple and indeed, for small graphical models (e.g., $N \approx 10$ – 20), they can be solved directly with brute force approaches. However, many applications involve graphs with thousands (if not millions) of nodes, and the computational complexity of brute force approaches scales very poorly in the graph size. More concretely, let us consider the case of a discrete random vector $(X_1, \dots, X_N) \in \mathcal{X}_{[N]}$ such that, for each node $s \in \mathcal{V}$, the random variable X_s takes values in the state space $\mathcal{X}_s = \{0, 1, \dots, m-1\}$. A naive approach to computing a marginal at a single node—say $p(x_s)$ —entails summing over all configurations of the form $\{x' \in \mathcal{X}^m \mid x'_s = x_s\}$. Since this set has m^{N-1} elements, it is clear that a brute force approach will rapidly become intractable. Given a graph with $N = 100$ vertices (a relatively small problem) and binary variables ($m = 2$), the number of terms is 2^{99} , which is already larger than estimates of the number of atoms in the universe. This is a vivid manifestation of the “curse-of-dimensionality” in a computational sense.

A similar curse applies to the problem of mode computation for discrete random vectors, since it is equivalent to solving an integer programming problem. Indeed, the mode computation problem for Markov random fields includes many well-known instances of NP-complete problems (e.g., 3-SAT, MAX-CUT etc.) as special cases. Unfortunately, for continuous random vectors, the problems are no easier and typically harder, since the marginalization problem involves computing a high-dimensional integral, whereas mode computation corresponds to a generic (possibly non-convex) optimization problem. An important exception to this statement is the Gaussian case where the problems of both marginalization and mode computation can be solved in polynomial-time for any graph (via matrix inversion), and in linear-time for tree-structured graphs via the Kalman filter.

In the following sections, we develop various approaches to these computational inference problems, beginning with discussion of a relatively naive but pedagogically useful scheme known as elimination, then moving onto more sophisticated message-passing algorithms, and culminating in our derivation of the junction tree algorithm.

3.1 Elimination Algorithm

We begin with an exact but relatively naive method known as the elimination algorithm. It operates on an undirected graphical model equipped with a factorization of the form

$$p(x_1, \dots, x_N) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C).$$

As input to the elimination algorithm, we provide it with a *target vertex* $T \in \mathcal{V}$ (or more generally a target subset of vertices), an *elimination ordering* \mathcal{I} , meaning an ordering of the vertex set \mathcal{V} such that the target T appears first; as well as the collection of compatibility functions $\{\psi_C, C \in \mathcal{C}\}$ defining the factorization (6). As output, the elimination algorithm returns the marginal distribution $p(x_T)$ over the variables x_T at the target vertices.

Elimination Algorithm for Marginalization

1. Initialization:

- (a) Initialize active set of compatibility functions $\mathcal{A} = \{\psi_C, C \in \mathcal{C}\}$.
- (b) Initialize elimination ordering \mathcal{I} (with the target vertex T appearing first).

2. Elimination: while $\text{card}(\mathcal{I}) > \text{card}(T)$:

- (a) Activate last vertex s on the current elimination order:
- (b) Form product

$$\phi(x_s, x_{\mathcal{N}(s)}) = \prod_{\psi_C \text{ involving } x_s} \psi_C(x_C),$$

where $\mathcal{N}(s)$ indexes all variables that share active factors with s .

- (c) Compute partial sum:

$$\tilde{\phi}_{\mathcal{N}(s)}(x_{\mathcal{N}(s)}) = \sum_{x_s} \phi(x_s, x_{\mathcal{N}(s)}),$$

and remove all $\{\psi_C\}$ that involve x_s from active list.

- (d) Add $\tilde{\phi}_{\mathcal{N}(s)}$ to active list, and remove s from index set \mathcal{I} .

(continued)

3. Termination: when $\mathcal{I} = \{T\}$, form the product $\phi_T(x_T)$ of all remaining functions in \mathcal{A} , and then compute

$$Z = \sum_{x_T} \phi_T(x_T) \quad \text{and} \quad p(x_T) = \frac{\phi_T(x_T)}{Z}.$$

We note that in addition to returning the marginal distribution $p(x_T)$, the algorithm also returns the normalization constant Z . The elimination algorithm is best understood in application to a particular undirected graph.

Example 9 (Elimination in Action) Consider the 7-vertex undirected graph shown in Fig. 2a. As discussed in Example 6, it induces factorizations of the form

$$p(x_1, \dots, x_7) \propto \psi_{123}(x_1, x_2, x_3) \psi_{345}(x_3, x_4, x_5) \psi_{46}(x_4, x_6) \psi_{57}(x_5, x_7),$$

Suppose that our goal is to compute the marginal distribution $p(x_5)$. In order to do so, we initialize the elimination algorithm with $T = 5$, the elimination ordering $\mathcal{I} = \{5, 4, 2, 3, 1, 6, 7\}$, and the active set $\mathcal{A} = \{\psi_{123}, \psi_{345}, \psi_{46}, \psi_{57}\}$.

Eliminating x_7 : At the first step, $s = 7$ becomes the active vertex. Since ψ_{57} is the only function involving x_7 , we form $\phi(x_5, x_7) = \psi_{57}(x_5, x_7)$, and then compute the partial sum $\tilde{\phi}_5(x_5) = \sum_{x_7} \psi_{57}(x_5, x_7)$. We then form the new active set $\mathcal{A} = \{\psi_{123}, \psi_{345}, \psi_{46}, \tilde{\phi}_5\}$.

Eliminating x_6 : This phase is very similar. We compute the partial sum $\tilde{\phi}_4(x_4) = \sum_{x_6} \psi_{46}(x_4, x_6)$, and end up with the new active set $\mathcal{A} = \{\psi_{123}, \psi_{345}, \tilde{\phi}_5, \tilde{\phi}_4\}$.

Eliminating x_1 : We compute $\tilde{\phi}_{23}(x_2, x_3) = \sum_{x_1} \psi_{123}(x_1, x_2, x_3)$, and conclude with the new active set $\mathcal{A} = \{\tilde{\phi}_{23}, \psi_{345}, \tilde{\phi}_5, \tilde{\phi}_4\}$.


Eliminating x_3 : In this phase, we first form the product $\phi_{2345}(x_2, x_3, x_4, x_5) = \tilde{\phi}_{23}(x_2, x_3) \psi_{345}(x_3, x_4, x_5)$, and then compute the partial sum

$$\tilde{\phi}_{245}(x_2, x_4, x_5) = \sum_{x_3} \phi_{2345}(x_2, x_3, x_4, x_5).$$

We end up with the active set $\mathcal{A} = \{\tilde{\phi}_{245}, \tilde{\phi}_5, \tilde{\phi}_4\}$.

Eliminating x_2 : The output of this phase is the active set $\mathcal{A} = \{\tilde{\phi}_{45}, \tilde{\phi}_5, \tilde{\phi}_4\}$, where $\tilde{\phi}_{45}$ is obtained by summing out x_2 from $\tilde{\phi}_{245}$.

Eliminating x_4 : We form the product $\phi_{45}(x_4, x_5) = \tilde{\phi}_4(x_4) \tilde{\phi}_{45}(x_4, x_5)$, and then the partial sum $\hat{\phi}(x_5) = \sum_{x_4} \phi_{45}(x_4, x_5)$. We conclude with the active set $\mathcal{A} = \{\tilde{\phi}_5, \hat{\phi}_5\}$.

After these steps, the index set \mathcal{I} has been reduced to the target vertex $T = 5$. In the final step, we form the product $\phi_5(x_5) = \hat{\phi}_5(x_5)\hat{\phi}_5(x_5)$, and then re-normalize it to sum to one. 

A few easy extensions of the elimination algorithm are important to note. Although we have described it in application to an undirected graphical model, a simple pre-processing step allows it to be applied to any directed graphical model as well. More precisely, any directed graphical model can be converted to an undirected graphical model via the following steps:

- remove directions from all the edges
- “moralize” the graph by adding undirected edges between all parents of a given vertex—that is, in detail

for all $s \in \mathcal{V}$, and for all $t, u \in \pi(s)$, add the edge (t, u) .

- the added edges yield cliques $C = (s, \pi(s))$; on each such clique, define the compatibility function $\psi_C(x_C) = p(x_s | x_{\pi(s)})$.

Applying this procedure yields an undirected graphical model to which the elimination algorithm can be applied.

A second extension concerns the computation of conditional distributions. In applications, it is frequently the case that some subset of the variables are observed, and our goal is to compute a conditional distribution of X_T given these observed variables. More precisely, suppose that we wish to compute a conditional probability of the form $p(x_T | \bar{x}_O)$ where $O \subset \mathcal{V}$ are indices of the observed values $X_O = \bar{x}_O$. In order to do so, it suffices to compute a marginal of the form $p(x_T, \bar{x}_O)$. This marginal can be computed by adding binary indicator functions to the active set given as input to the elimination algorithm. In particular, for each $s \in O$, let $\mathbb{I}_{s; \bar{x}_s}(x_s)$ be a zero-one indicator for the event $\{x_s = \bar{x}_s\}$. We then provide the elimination algorithm with the initial active set $\mathcal{A} = \{\psi_C, C \in \mathcal{C}\} \cup \{\mathbb{I}_{s; \bar{x}_s}, s \in O\}$, so that it computes marginals for the modified distribution

$$q(x_1, \dots, x_N) \propto \prod_{C \in \mathcal{C}} \psi_C(x_C) \prod_{s \in O} \mathbb{I}_{s; \bar{x}_s}(x_s),$$

in which $q(x) = 0$ for all configurations $x_O \neq \bar{x}_O$. The output of the elimination algorithm can also be used to compute the likelihood $p(\bar{x}_O)$ of the observed data.

3.1.1 Graph-Theoretic Versus Analytical Elimination

Up to now, we have described the elimination algorithm in an analytical form. It can also be viewed from a graph-theoretic perspective, namely as an algorithm that eliminates vertices, but adds edges as it does so. In particular, during the elimination step, the graph-theoretic version removes the active vertex s from the

current elimination order (as well as all edges involving s), but then adds an edge for each pair of vertices (t, u) that were connected to s in the current elimination graph. These adjacent vertices define the neighborhood set $\mathcal{N}(s)$ used in Step 2(c) of the algorithm. This dual perspective—the analytical and graph-theoretic—is best illustrated by considering another concrete example.

Example 10 (Graph-Theoretic Versus Analytical Elimination) Consider the 3×3 grid-structured graph shown in panel (a) of Fig. 5. Given a probability distribution that is Markov with respect to this grid, suppose that we want to compute the marginal distribution $p(x_5)$, so that the target vertex is $T = 5$. In order to do so, we initialize the elimination algorithm using the ordering $\mathcal{I} = \{5, 2, 8, 6, 4, 1, 3, 7, 9\}$. Panel (b) shows the graph obtained after eliminating vertex 9, now shown in dotted lines to indicate that it has been removed. At the graph-theoretic level, we have removed edges $(8, 9)$ and $(6, 9)$, and added edge $(6, 8)$. At the analytical level, we have performed the summation over x_9 to compute

$$\tilde{\phi}(x_6, x_8) = \sum_{x_9} \psi_{69}(x_6, x_9) \psi_{89}(x_8, x_9),$$

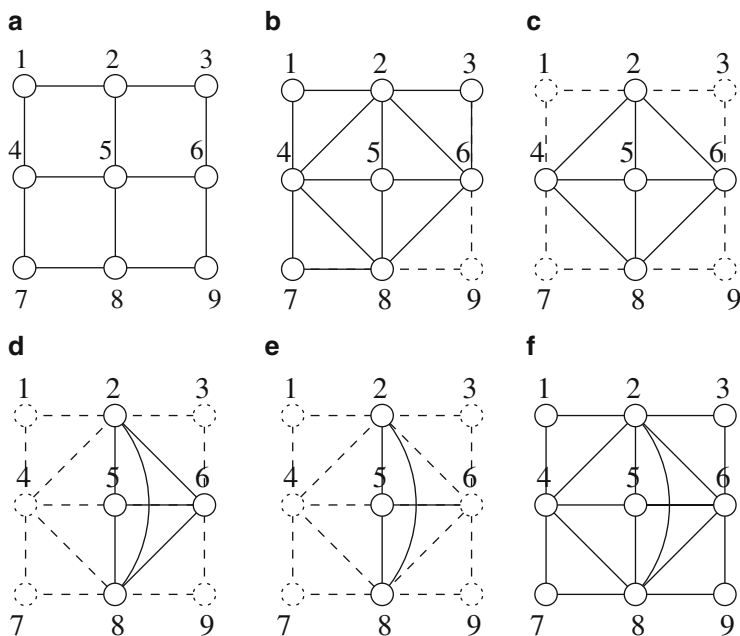


Fig. 5 (a) A 3×3 grid graph in the plane. (b) Result of eliminating vertex 9. Dotted vertices/edges indicate quantities that have been removed during elimination. (c) Remaining graph after having eliminated vertices $\{1, 3, 7, 9\}$. (d) Resulting graph after eliminating vertex 4. (e) Elimination of vertex 6. (f) Final reconstituted graph: original graph augmented with all edges added during the elimination process

and added it to the active list of compatibility functions. Notice how the new function $\tilde{\phi}_{68}$ depends only on the variables (x_6, x_8) , which are associated with the edge (6, 8) added at the graph-theoretic level. Panel (c) shows the result after having eliminated vertices $\{1, 3, 7, 9\}$, in reverse order. Removal of the three vertices (in addition to 9) has a symmetric effect on each corner of the graph. In panel (d), we see the effect of eliminating vertex 4; in this case, we remove edges (4, 5), (4, 2), and (4, 8), and then add edge (2, 8). At the analytical stage, this step will introduce a new compatibility function of the form $\tilde{\phi}_{258}(x_2, x_5, x_8)$. Panel (e) shows the result of removing vertex 6; no additional edges are added at this stage. Finally, panel (f) shows the *reconstituted graph* $\tilde{\mathcal{G}}$, meaning the original graph plus all the edges that were added during the elimination process. Note that the added edges have increased the size of the maximal clique(s) to four; for instance, the set $\{2, 5, 6, 8\}$ is a maximal clique in the reconstituted graph. This reconstituted graph plays an important role in our discussion of junction tree theory to follow. ♣

3.1.2 Complexity of Elimination

What is the computational complexity of the elimination algorithm? Let us do some rough calculations in the discrete case, when $\mathcal{X}_s = \{0, 1, \dots, m-1\}$ for all vertices $s \in \mathcal{V}$. Clearly, it involves at least a linear cost in the number of vertices N . The other component is the cost of computing the partial sums $\tilde{\phi}$ at each step. Supposing that we are eliminating s , let $\mathcal{N}(s)$ be the current set of its neighbors in the elimination graph. For instance, in Fig. 5, when we eliminate vertex $s = 4$, its current neighborhood set is $\mathcal{N}(4) = \{2, 5, 8\}$, and the function $\tilde{\phi}_{258}$ at this stage depends on (x_2, x_5, x_8) . Computing $\tilde{\phi}_{258}$ incurs a complexity of $\mathcal{O}(m^4)$, since we need to perform a summation (over x_4) involving m terms for each of the m^3 possible instantiations of (x_2, x_5, x_8) .

In general, then, the worst-case cost over all elimination steps will scale as $\mathcal{O}(m^{|\mathcal{N}(s)|+1})$, where $|\mathcal{N}(s)|$ is the neighborhood size in the elimination graph when eliminating s . It can be seen that the worst-case neighborhood size (plus one) is equivalent to the size of the largest clique in the reconstituted graph. For instance, in Fig. 5f, the largest cliques in the reconstituted graph have size 4, consistent with the $\mathcal{O}(m^4)$ calculation from above. Putting together the pieces, we obtain a conservative upper bound on the cost of the elimination algorithm—namely, $\mathcal{O}(m^c N)$, where c is the size of the largest clique in the reconstituted graph.

A related point is that for any given target vertex T , there are many possible instantiations of the elimination algorithm that can be used to compute the marginal $p(x_T)$. Remember that our only requirement is that T appear first in the ordering, so that there are actually $(N-1)!$ possible orderings of the remaining vertices. Which ordering should be chosen? Based on our calculations above, in order to minimize the computational cost, one desirable goal would be to minimize the size of the largest clique in the reconstituted graph. In general, finding an optimal elimination ordering of this type is computationally challenging; however, there exist a variety of heuristics for choosing good orderings, as we discuss in Sect. 4.

3.2 Message-Passing Algorithms on Trees

We now turn to discussion of message-passing algorithms for models based on graphs without cycles, also known as trees. So as to highlight the essential ideas in the derivation, we begin by deriving message-passing algorithms for trees with only pairwise interactions, in which case the distribution has form

$$p(x_1, \dots, x_N) = \frac{1}{Z} \prod_{s \in \mathcal{V}} \psi_s(x_s) \prod_{(s,t) \in \mathcal{E}} \psi_{st}(x_s, x_t), \quad (19)$$

where $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ is a given tree, and $\{\psi_s, s \in \mathcal{V}\}$ and $\{\psi_{st}, (s,t) \in \mathcal{E}\}$ are compatibility functions. For a tree, the elimination algorithm takes a very simple form—assuming that the appropriate ordering is followed. Recall that a vertex in a graph is called a *leaf* if it has degree one.

The key fact is that for any tree, it is always possible to choose an elimination ordering \mathcal{I} such that:

- (a) The last vertex in \mathcal{I} (first to be eliminated) is a leaf; and
- (b) For each successive stage of the elimination algorithm, the last vertex in \mathcal{I} remains a leaf of the reduced graph.

This claim follows because any tree always has at least one leaf vertex [17], so that the algorithm can be initialized in Step 1. Moreover, in removing a leaf vertex, the elimination algorithm introduces no additional edges to the graph, so that it remains a tree at each stage. Thus, we may apply the argument recursively, again choosing a leaf vertex to continue the process. We refer to such an ordering as a *leaf-exposing ordering*.

In computational terms, when eliminating leaf vertex s from the graph, the algorithm marginalizes over x_s , and then passes the result to its unique parent $\pi(s)$. The result $\phi_{\pi(s)}$ of this intermediate computation is a function of $x_{\pi(s)}$; following standard notation used in presenting the sum-product algorithm, we can represent it by a “message” of the form $M_{s \rightarrow \pi(s)}$, where the subscript reflects the direction in which the message is passed. Upon termination, the elimination algorithm will return the marginal distribution $p(x_T)$ at the target vertex $T \in \mathcal{V}$; more specifically, this marginal is specified in terms of the local compatibility function ψ_T and the incoming messages as follows:

$$p(x_T) \propto \psi_T(x_T) \prod_{s \in \mathcal{N}(T)} M_{s \rightarrow T}(x_T).$$

Assuming that each variable takes at most m values (i.e., $|\mathcal{X}_s| \leq m$ for all $s \in \mathcal{V}$), the overall complexity of running the elimination algorithm for a given root vertex scales as $O(m^2 N)$, since computing each message amounts to summing m numbers a total of m times, and there are $O(N)$ rounds in the elimination order. Alternatively, since the leaf-exposing ordering adds no new edges to the tree, the reconstituted

graph is *equivalent* to the original tree. The maximal cliques in any tree are the edges, hence of size two, so that the complexity estimate $\mathcal{O}(m^2 N)$ follows from our discussion of the elimination algorithm.

3.2.1 Sum-Product Algorithm

In principle, by running the tree-based elimination algorithm N times—once for each vertex acting as the target in the elimination ordering—we could compute all singleton marginals in time $\mathcal{O}(m^2 N^2)$. However, as the attentive reader might suspect, such an approach is wasteful, since it neglects to consider that many of the intermediate operations in the elimination algorithm would be shared between different orderings. Herein lies the cleverness of the sum-product algorithm: it reuses these intermediate results in the appropriate way, and thereby reduces the overall complexity of computing *all singleton marginals* to $\mathcal{O}(m^2 N)$. In fact, as an added bonus, without any added complexity, it also computes *all pairwise marginal distributions* over variables (x_s, x_t) such that $(s, t) \in \mathcal{E}$.

For each edge $(s, t) \in \mathcal{E}$, the sum-product algorithm maintains two messages—namely, $M_{s \rightarrow t}$ and $M_{t \rightarrow s}$ —corresponding to two possible directions of the edge.¹ With this notation, the sum-product message-passing algorithm for discrete random variables takes the following form:

Sum-Product Message-Passing Algorithm (Pairwise Tree):

1. At iteration $k = 0$:
For all $(s, t) \in \mathcal{E}$, initialize messages

$$M_{t \rightarrow s}^0(x_s) = 1 \quad \text{for all } x_s \in \mathcal{X}_s \quad \text{and} \quad M_{s \rightarrow t}^0(x_t) = 1 \quad \text{for all } x_t \in \mathcal{X}_t.$$

2. For iterations $k = 1, 2, \dots$:
 - (i) For each $(t, s) \in \mathcal{E}$, update messages:

$$M_{t \rightarrow s}^k(x_s) \leftarrow \alpha \sum_{x'_t} \left\{ \psi_{st}(x_s, x'_t) \psi_t(x'_t) \prod_{u \in \mathcal{N}(t)/s} M_{u \rightarrow t}^{k-1}(x'_t) \right\}, \quad (20)$$

where $\alpha > 0$ chosen such that $\sum_{x_s} M_{s \rightarrow s}^k(x_s) = 1$.

(continued)

¹Recall that the message $M_{s \rightarrow t}$ is a vector of $|\mathcal{X}_t|$ numbers, one for each value $x_t \in \mathcal{X}_t$.

(ii) Upon convergence, compute marginal distributions:

$$p(x_s) = \alpha_s \psi_s(x_s) \prod_{t \in \mathcal{N}(s)} M_{t \rightarrow s}(x_s), \quad \text{and} \quad (21a)$$

$$p_{st}(x_s, x_t) = \alpha_{st} \psi_{st}(x_s, x_t) \prod_{u \in \mathcal{N}(s) \setminus \{t\}} M_{u \rightarrow s}(x_s) \prod_{u \in \mathcal{N}(t) \setminus \{s\}} M_{u \rightarrow t}(x_t), \quad (21b)$$

where $\alpha_s > 0$ and $\alpha_{st} > 0$ are normalization constants (to ensure that the marginals sum to one).

The initialization given in Step 1 is the standard “uniform” one; it can be seen that the final output of the algorithm will be same for any initialization of the messages that has strictly positive components. As with our previous discussion of the elimination algorithm, we have stated a form of message-passing updates suitable for discrete random variables, involving finite summations. For continuous random variables, we simply need to replace the summations in the message update (20) with integrals. For instance, in the case of a Gauss-Markov chain, we obtain the Kalman filter as a special case of the sum-product updates.

There are various ways in which the order of message-passing can be scheduled. The simplest to describe, albeit not the most efficient, is the *flooding schedule*, in which every message is updated during every round of the algorithm. Since each edge is associated with two messages—one for each direction—this protocol involves updating $2|\mathcal{E}|$ messages *per iteration*. The flooding schedule is especially well-suited to parallel implementations of the sum-product algorithm, such as on a digital chip or field programmable gate array, in which a single processing unit can be devoted to each vertex of the graph.

In terms of minimizing the total number of messages that are passed, a more efficient scheduling protocol is based on the following rule: any given vertex s updates the message $M_{s \rightarrow t}$ to its neighbor $t \in \mathcal{N}(s)$ only after it has received messages from all its other neighbors $u \in \mathcal{N}(s) \setminus \{t\}$. In order to see that this protocol will both start and terminate, note that any tree has at least one leaf vertex. All leaf vertices will pass a message to their single neighbor at round 1. As in the elimination algorithm, we can then remove these vertices from further consideration, yielding a sub-tree of the original tree, which also has at least one leaf node. Again, we see that a recursive leaf-stripping argument will guarantee termination of the algorithm. Upon termination, a total of $2|\mathcal{E}|$ messages have been passed *over all iterations*, as opposed to per iteration in the flooding schedule.

Regardless of the particular message-update schedule used, the following result states the convergence and correctness guarantees associated with the sum-product algorithm on a tree:

Proposition 2 *For any tree \mathcal{T} with diameter $d(\mathcal{T})$, the sum-product updates have a unique fixed point M^* . The algorithm converges to it after at most $d(\mathcal{T})$ iterations, and the marginals obtained by Eqs. (21a) and (21b) are exact.*

Proof We proceed via induction on the number of vertices. For $N = 1$, the claim is trivial. Now assume that the claim holds for all trees with at most $N - 1$ vertices, and let us show that it also holds for any tree with N vertices. It is an elementary fact of graph theory [17] that any tree has at least one leaf node. By re-indexing as necessary, we may assume that node N is a leaf, and its unique parent is node 1. Moreover, by appropriately choosing the leaf vertex, we may assume that the diameter of \mathcal{T} is achieved by a path ending at node N . By definition of the sum-product updates, for all iterations $k = 1, 2, \dots$, the message sent from N to 1 is given by

$$M_{N \rightarrow 1}^*(x_1) = \alpha \sum_{x_N} \psi_N(x_N) \psi_{N1}(x_N, x_1).$$

(It never changes since vertex N has only one neighbor.) Given this fixed message, we may consider the probability distribution defined on variables (x_1, \dots, x_{N-1}) given by

$$p(x_1, \dots, x_{N-1}) \propto M_{N \rightarrow 1}^*(x_1) \left[\prod_{s=1}^{N-1} \psi_s(x_s) \right] \prod_{(s,t) \in \mathcal{E} \setminus \{(1,N)\}} \psi_{st}(x_s, x_t).$$

Our notation is consistent, in that $p(x_1, \dots, x_{N-1})$ is the marginal distribution obtained by summing out x_N from the original problem. It corresponds to an instance of our original problem on the new tree $\mathcal{T}' = (\mathcal{V}', \mathcal{E}')$ with $\mathcal{V}' = \{1, \dots, N - 1\}$ and $\mathcal{E}' = \mathcal{E} \setminus \{(1, N)\}$. Since the message $M_{N \rightarrow 1}^*$ remains fixed for all iterations $k \geq 1$, the iterates of the sum-product algorithm on \mathcal{T}' are indistinguishable from the iterates on tree (for all vertices $s \in \mathcal{V}'$ and edges $(s, t) \in \mathcal{E}'$).

By the induction hypothesis, the sum-product algorithm applied to \mathcal{T}' will converge after at most $d(\mathcal{T}')$ iterations to a fixed point $M^* = \{M_{s \rightarrow t}^*, M_{t \rightarrow s}^* \mid (s, t) \in \mathcal{E}'\}$, and this fixed point will yield the correct marginals at all vertices $s \in \mathcal{V}'$ and edges $(s, t) \in \mathcal{E}'$. As previously discussed, the message from N to 1

remains fixed for all iterations after the first, and the message from 1 to N will be fixed once the iterations on the sub-tree \mathcal{T}' have converged. Taking into account the extra iteration for node 1 to N , we conclude that sum-product on \mathcal{T} converges in at most $d(\mathcal{T})$ iterations as claimed.

It remains to show that the marginal at node N can be computed by forming the product $p(x_N) \propto \psi_N(x_N) M_{1 \rightarrow N}^*(x_N)$, where

$$M_{1 \rightarrow N}^*(x_N) := \alpha \sum_{x_1} \psi_1(x_1) \psi_{1N}(x_1, x_N) \prod_{u \in N(1) \setminus \{N\}} M_{u \rightarrow 1}^*(x_1).$$

By elementary probability theory, we have $p(x_N) = \sum_{x_1} p(x_N | x_1) p(x_1)$. Since N is a leaf node with unique parent 1, we have the conditional independence relation $X_N \perp\!\!\!\perp X_{\mathcal{T} \setminus \{1\}} | X_1$, and hence, using the form of the factorization (19),

$$p(x_N | x_1) = \frac{\psi_N(x_N) \psi_{1N}(x_1, x_N)}{\sum_{x_N} \psi_N(x_N) \psi_{1N}(x_1, x_N)} \propto \frac{\psi_N(x_N) \psi_{1N}(x_1, x_N)}{M_{N \rightarrow 1}^*(x_1)}. \quad (22)$$

By the induction hypothesis, the sum-product algorithm, when applied to the distribution (22), returns the marginal distribution

$$p(x_1) \propto [M_{N \rightarrow 1}^*(x_1) \psi_1(x_1)] \prod_{t \in N(1) \setminus \{N\}} M_{t \rightarrow 1}^*(x_1).$$

Combining the pieces yields

$$\begin{aligned} p(x_N) &= \sum_{x_1} p(x_N | x_1) p(x_1) \\ &\propto \psi_N(x_N) \sum_{x_1} \frac{\psi_{1N}(x_1, x_N)}{M_{N \rightarrow 1}^*(x_1)} [M_{N \rightarrow 1}^*(x_1) \psi_1(x_1)] \prod_{k \in N(1) \setminus \{N\}} M_{k \rightarrow 1}^*(x_1) \\ &\propto \psi_N(x_N) M_{1 \rightarrow N}^*(x_N), \end{aligned}$$

as required. A similar argument yields that Eq. (21b) yields the correct form of the pairwise marginal for the edge $(1, N) \in \mathcal{E}$; we leave the details as an exercise for the reader. \square

3.2.2 Sum-Product on General Factor Trees

In the preceding section, we derived the sum-product algorithm for a tree with pairwise interactions; in this section, we discuss its extension to arbitrary tree-structured factor graphs. A tree factor graph is simply a factor graph without cycles;

see Fig. 3b for an example. We consider a distribution with a factorization of the form

$$p(x_1, \dots, x_N) = \frac{1}{Z} \prod_{s \in \mathcal{V}} \psi_s(x_s) \prod_{a \in \mathcal{F}} \psi_a(x_a), \quad (23)$$

where \mathcal{V} and \mathcal{F} are sets of vertices and factor nodes, respectively.

Before proceeding, it is worth noting that for the case of discrete random variables considered here, the resulting algorithm is not more general than sum-product for pairwise interactions. Indeed, in any distribution over discrete random variables with a tree-structured factor graph—regardless of the order of interactions that it involves—can be converted to an equivalent tree with pairwise interactions. We refer the reader to Appendix E.3 in the monograph [64] for further details on this procedure. Nonetheless, it is often convenient to apply the non-pairwise form of the sum-product algorithm directly to a factor tree, as opposed to going through this conversion.

We use $M_{s \rightarrow a}$ to denote the message from vertex s to the factor node $a \in \mathcal{N}(s)$, and similarly $M_{a \rightarrow s}$ to denote the message from factor node a to the vertex $s \in \mathcal{N}(a)$. Both messages (in either direction) are functions of x_s , where $M_{a \rightarrow s}(x_s)$ (respectively $M_{s \rightarrow a}(x_s)$) denotes the value taken for a given $x_s \in \mathcal{X}_s$. We let $x_a = \{x_s, s \in \mathcal{N}(a)\}$ denote the sub-vector of random variables associated with factor node $a \in \mathcal{F}$. With this notation, the sum-product updates for a general tree factor graph take the following form:

Sum-Product Updates for Tree Factor Graph:

$$M_{s \rightarrow a}(x_s) \leftarrow \alpha \psi_s(x_s) \prod_{b \in \mathcal{N}(s) \setminus \{a\}} M_{b \rightarrow s}(x_s), \quad \text{and} \quad (24a)$$

$$M_{a \rightarrow s}(x_s) \leftarrow \alpha \sum_{x_t, t \in \mathcal{N}(a) \setminus \{s\}} \left[\psi_a(x_a) \prod_{t \in \mathcal{N}(a) \setminus \{s\}} M_{t \rightarrow a}(x_t) \right]. \quad (24b)$$

Here the quantity α again represents a positive constant chosen to ensure that the messages sum to one. (As before, its value can differ from line to line.) Upon convergence, the marginal distributions over x_s and over the variables x_a are given, respectively, by the quantities

$$p(x_s) = \alpha \psi_s(x_s) \prod_{a \in \mathcal{N}(s)} M_{a \rightarrow s}(x_s), \quad \text{and} \quad (25a)$$

$$p(x_a) = \alpha \psi_a(x_a) \prod_{t \in \mathcal{N}(a)} M_{t \rightarrow a}(x_t). \quad (25b)$$

We leave it as an exercise for the reader to verify that the message-passing updates (24) will converge after a finite number of iterations (related to the diameter of the factor tree), and when equation (25) is applied using the resulting fixed point M^* of the message updates, the correct marginal distributions are obtained.

3.2.3 Max-Product Algorithm

Thus far, we have discussed the sum-product algorithm that can be used to solve the problems of marginalization and likelihood computation. We now turn to the max-product algorithm, which is designed to solve the problem of mode computation. So as to clarify the essential ideas, we again present the algorithm in terms of a factor tree involving only pairwise interactions, with the understanding that the extension to a general factor tree is straightforward.

The problem of mode computation amounts to finding a vector

$$x^* \in \arg \max_{x \in \mathcal{X}_{[N]}} p(x_1, \dots, x_N),$$

where we recall our shorthand notation $\mathcal{X}_{[N]} = \mathcal{X}_1 \times \mathcal{X}_2 \times \dots \times \mathcal{X}_N$. As an intermediate step, let us first consider the problem of computing the so-called *max-marginals* associated with any distribution defined by a tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$. In particular, for each vertex $s \in \mathcal{V}$, we define the *singleton max-marginal*

$$v_s(x_s) := \max_{\{x' \in \mathcal{X}_{[N]} \mid x'_s = x_s\}} p(x'_1, \dots, x'_N), \quad (26)$$

For each edge $(s, t) \in \mathcal{E}$, the *pairwise max-marginal* is defined in an analogous manner:

$$v_{ij}(x_i, x_j) := \max_{\{x' \in \mathcal{X}_{[N]} \mid (x'_i, x'_j) = (x_i, x_j)\}} p(x'_1, \dots, x'_N), \quad (27)$$

Note that the singleton and pairwise max-marginals are the natural analogs of the usual marginal distributions, in which the summation operation has been replaced by the maximization operation.

Before describing how these max-marginals can be computed by the max-product algorithm, first let us consider how max-marginals are relevant for the problem of mode computation. This connection is especially simple if the singleton max-marginals satisfy the *unique maximizer condition*—namely, if for all $s \in \mathcal{V}$, the maximum over $v_s(x_s)$ is achieved at a single element, meaning that

$$\arg \max_{x_s \in \mathcal{X}_s} v_s(x_s) = x_s^* \quad \text{for all } s \in \mathcal{V}. \quad (28)$$

In this case, it is straightforward to compute the mode, as summarized in the following:

Lemma 3 Consider a distribution p whose singleton max-marginals satisfy the unique maximizer condition (28). Then the vector $x^* = (x_1^*, \dots, x_N^*) \in \mathcal{X}_{[N]}$ is the unique maximizer of $p(x_1, \dots, x_N)$.

Proof The proof of this claim is straightforward. Note that for any vertex $s \in \mathcal{V}$, by the unique maximizer condition and the definition of the max-marginals, we have

$$v_s(x_s^*) = \max_{x_s \in \mathcal{X}_s} v_s(x_s) = \max_{x \in \mathcal{X}_{[N]}} p(x_1, \dots, x_N).$$

Now for any other configuration $\tilde{x} \neq x^*$, there must be some index t such that $x_t^* \neq \tilde{x}_t$. For this index, we have

$$\max_{x \in \mathcal{X}_{[N]}} p(x_1, \dots, x_N) = v_t(x_t^*) \stackrel{(i)}{>} v_t(\tilde{x}_t) \stackrel{(ii)}{\geq} p(\tilde{x}),$$

where inequality (i) follows from the unique maximizer condition, and inequality (ii) follows by definition of the max-marginal. We have thus established that $p(\tilde{x}) < \max_{x \in \mathcal{X}_{[N]}} p(x)$ for all $\tilde{x} \neq x^*$, so that x^* uniquely achieves the maximum. \square

If the unique maximizer condition fails to hold, then the distribution p must have more than one mode, and it requires a bit more effort to determine one. Most importantly, it is *no longer* sufficient to extract *some* $x_s^* \in \arg \max_{x_s \in \mathcal{X}_i} v_i(x_s)$, as illustrated by the following toy example.

Example 11 (Failure of Unique Maximizer) Consider the distribution over a pair of binary variables given by

$$p(x_1, x_2) = \begin{cases} 0.45 & \text{if } x_1 \neq x_2, \text{ and} \\ 0.05 & \text{otherwise.} \end{cases} \quad (29)$$

In this case, we have $v_1(x_1) = v_2(x_2) = 0.45$ for all $(x_1, x_2) \in \{0, 1\}^2$, but only configurations with $x_1 \neq x_2$ are globally optimal. As a consequence, a naive procedure that looks only at the singleton max-marginals has no way of determining such a globally optimal configuration. \clubsuit

Consequently, when the unique maximizer condition no longer holds, it is necessary to also take into account the information provided by the pairwise max-marginals (27). In order to do so, we consider a back-tracking procedure that samples some configuration $x^* \in \arg \max_{x \in \mathcal{X}_{[N]}} p(x)$. Let us assume that the tree

is rooted at node 1, and that $\mathcal{V} = \{1, 2, \dots, N\}$ is a topological ordering (i.e., such that $\pi(s) < s$ for all $s \in \{2, \dots, N\}$). Using this topological ordering we may generate an optimal configuration x^* as follows:

Back-Tracking for Choosing a Mode x^*

1. Initialize the procedure at node 1 by choosing any $x_1^* \in \arg \max_{x_1 \in \mathcal{X}_1} v_1(x_1)$.
2. For each $s = 2, \dots, N$, choose a configuration x_s^* at vertex s such that

$$x_s^* \in \arg \max_{x_s \in \mathcal{X}_s} v_{s, \pi(s)}(x_s, x_{\pi(s)}^*).$$

Note that this is a recursive procedure, in that the choice of the configuration x_s^* depends on the previous choice of $x_{\pi(s)}^*$. Our choice of topological ordering ensures that $x_{\pi(s)}^*$ is always fixed before we reach vertex s . We establish that this back-tracking procedure is guaranteed to output a vector $x^* \in \arg \max_{x \in \mathcal{X}_{[N]}} p(x)$ in Proposition 3 below.

Having established the utility of the max-marginals for computing modes, let us now turn to an efficient algorithm for computing them. As alluded to earlier, the max-marginals are the analogs of the ordinary marginals when summation is replaced with maximization. Accordingly, it is natural to consider the analog of the sum-product updates with the same replacement; doing so leads to the following *max-product updates*:

$$M_{t \rightarrow s}(x_s) \leftarrow \alpha \max_{x_t} \left\{ \psi_{st}(x_s, x_t) \psi_t(x_t) \prod_{u \in \mathcal{N}(t)/s} M_{u \rightarrow t}(x_t) \right\}, \quad (30)$$

where $\alpha > 0$ chosen such that $\max_{x_s} M_{t \rightarrow s}(x_s) = 1$. When the algorithm converges to a fixed point M^* , we can compute the max-marginals via Eqs. (21a) and (21b). Let us summarize the properties of the max-product algorithm on any undirected tree:

Proposition 3 *For any tree \mathcal{T} with diameter $d(\mathcal{T})$, the max-product updates converge to their unique fixed point M^* after at most $d(\mathcal{T})$ iterations. Moreover, Eqs. (21a) and (21b) can be used to compute the max-marginals, and the back-tracking procedure yields an element $x^* \in \arg \max_{x \in \mathcal{X}_{[N]}} p(x)$.*

Proof The proof of convergence and correct computation of the max-marginals is formally identical to that of Proposition 2, so we leave the details to the reader

as an exercise. Let us verify the remaining claim—namely, the optimality of any configuration $x^* \in \mathcal{X}_{[N]}$ returned by the back-tracking procedure.

In order to do so, recall that we have assumed without loss of generality (re-indexing as necessary) that $\mathcal{V} = \{1, 2, \dots, N\}$ is a topological ordering, with vertex 1 as the root. Under this ordering, let us use the max-marginals to define the following cost function:

$$J(x_1, \dots, x_N; \nu) = \nu_1(x_1) \prod_{s=2}^N \frac{\nu_{s\pi(s)}(x_s, x_{\pi(s)})}{\nu_{\pi(s)}(x_{\pi(s)})}. \quad (31)$$

In order for division to be defined in all cases, we take $0/0 = 0$.

We first claim that there exists a constant $\alpha > 0$ such that $J(x; \nu) = \alpha p(x)$ for all $x \in \mathcal{X}_{[N]}$, which implies that $\arg \max_{x \in \mathcal{X}_{[N]}} J(x; \nu) = \arg \max_{x \in \mathcal{X}_{[N]}} p(x)$. By virtue of this property, we say that ν defines a *reparameterization* of the original distribution. (See the papers [65, 66] for further details on this property and its utility in analyzing the sum and max-product algorithms.)

To establish the reparameterization property, we first note that the cost function J can also be written in the symmetric form

$$J(x; \nu) = \prod_{s=1}^N \nu_s(x_s) \prod_{(s,t) \in \mathcal{E}} \frac{\nu_{st}(x_s, x_t)}{\nu_s(x_s) \nu_t(x_t)}. \quad (32)$$

We then recall that the max-marginals are specified by the message fixed point M^* via Eqs. (21a) and (21b). Substituting these relations into the symmetric form (32) of J yields

$$\begin{aligned} J(x; \nu) &\propto \left[\prod_{s \in \mathcal{V}} \psi_s(x_s) \prod_{u \in \mathcal{N}(s)} M_{u \rightarrow s}^*(x_s) \right] \left[\prod_{(s,t) \in \mathcal{E}} \frac{\psi_{st}(x_s, x_t)}{M_{s \rightarrow t}^*(x_t) M_{t \rightarrow s}^*(x_s)} \right] \\ &= \prod_{s \in \mathcal{V}} \psi_s(x_s) \prod_{(s,t) \in \mathcal{E}} \psi_{st}(x_s, x_t) \\ &\propto p(x), \end{aligned}$$

as claimed.

Consequently, it suffices to show that any $x^* \in \mathcal{X}_{[N]}$ returned by the back-tracking procedure is an element of $\arg \max_{x \in \mathcal{X}_{[N]}} J(x; \nu)$. We claim that for each vertex $s \in \mathcal{V}$ such that $\pi(s) \neq \emptyset$,

$$\frac{\nu_{s\pi(s)}(x_s^*, x_{\pi(s)}^*)}{\nu_{\pi(s)}(x_{\pi(s)}^*)} \geq \frac{\nu_{s\pi(s)}(x_s, x_{\pi(s)})}{\nu_{\pi(s)}(x_{\pi(s)})} \quad \text{for all } (x_s, x_{\pi(s)}) \in \mathcal{X}_s \times \mathcal{X}_{\pi(s)}. \quad (33)$$

By definition of the max-marginals and our choice of x^* , the left-hand side is equal to one. On the other hand, the definition of the max-marginals implies that $\nu_{\pi(s)}(x_{\pi(s)}) \geq \nu_{s, \pi(s)}(x_s, x_{\pi(s)})$, showing that the right-hand side is less than or equal to one.

By construction, any x^* returned by back-tracking satisfies $\nu_1(x_1^*) \geq \nu_1(x_1)$ for all $x_1 \in \mathcal{X}_1$. This fact, combined with the pairwise optimality (33) and the definition (31), implies that $J(x^*; \nu) \geq J(x; \nu)$ for all $x \in \mathcal{X}_{[N]}$, showing that x^* is an element of the set $\arg \max_{x \in \mathcal{X}_{[N]}} J(x; \nu)$, as claimed. \square

In summary, the sum-product and max-product are closely related algorithms, both based on the basic principle of “divide-and-conquer”. A careful examination shows that both algorithms are based on repeated exploitation of a common algebraic property, namely the *distributive law*. More specifically, for arbitrary real numbers $a, b, c \in \mathbb{R}$, we have

$$a \cdot (b + c) = a \cdot b + a \cdot c, \quad \text{and} \quad a \cdot \max\{b, c\} = \max\{a \cdot b, a \cdot c\}.$$

The sum-product (respectively max-product) algorithm derives its power by exploiting this distributivity to re-arrange the order of summation and multiplication (respectively maximization and multiplication) so as to minimize the number of steps required. Based on this perspective, it can be shown that similar updates apply to any pair of operations (\oplus, \otimes) that satisfy the distributive law $a \otimes (b \oplus c) = a \otimes b \oplus a \otimes c$, and for which \otimes is a commutative operation (meaning $a \otimes b = b \otimes a$). Here the elements a, b, c need no longer be real numbers, but can be more exotic objects; for instance, they could be elements of the ring of polynomials, where (\otimes, \oplus) correspond to multiplication or addition with polynomials. We refer the interested reader to the papers [1, 28, 55, 62] for more details on such generalizations of the sum-product and max-product algorithms.

4 Junction Tree Framework

Thus far, we have derived the sum-product and max-product algorithms, which are exact algorithms for tree-structured graphs. Consequently, given a graph with cycles, it is natural to think about some type of graph transformation—for instance, such as grouping its vertices into clusters—so as to form a tree to which the fast and exact algorithms can be applied. It turns out that this “clustering”, if not done carefully, can lead to incorrect answers. Fortunately, there is a theory that formalizes and guarantees correctness of such a clustering procedure, which is known as the *junction tree framework*.

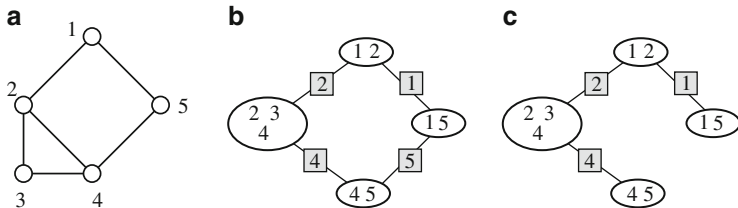


Fig. 6 (a) A graph with cycles on five vertices. (b) Associated clique graph in which circular vertices are maximal cliques of \mathcal{G} ; square gray boxes sitting on the edges represent separator sets, corresponding to the intersections between adjacent cliques. (c) One clique tree extracted from the clique graph. It can be verified that this clique tree fails to satisfy the running intersection property

4.1 Clique Trees and Running Intersection

In order to motivate the development to follow, let us begin with a simple (but cautionary) example. Consider the graph with vertex set $\mathcal{V} = \{1, 2, 3, 4, 5\}$ shown in Fig. 6a, and note that it has four maximal cliques—namely $\{2, 3, 4\}$, $\{1, 2\}$, $\{1, 5\}$ and $\{4, 5\}$. The so-called *clique graph* associated with this graph has four vertices, one for each of these maximal cliques, as illustrated in panel (b). The vertices corresponding to cliques C_1 and C_2 are joined by an edge in the clique graph if and only if $C_1 \cap C_2$ is not empty. In this case, we label the edge joining C_1 and C_2 with the set $S = C_1 \cap C_2$, which is known as the *separator set*. The separator sets are illustrated in gray boxes in panel (b). Finally, one possible tree contained within this clique graph—known as a *clique tree*—is shown in panel (c).

Now suppose that we were given a MRF distribution over the single cycle in panel (a); by the Hammersley-Clifford theorem, it would have the form

$$p(x_1, \dots, x_5) \propto \psi_{234}(x_2, x_3, x_4) \psi_{12}(x_1, x_2) \psi_{15}(x_1, x_5) \psi_{45}(x_4, x_5). \quad (34)$$

We can provide an alternative factorization of (essentially) the same model on the clique graph in panel (b) by making two changes: first, introducing extra copies of each variable from the original graph appears in multiple places on the clique graph, and second, introducing indicator functions on the edges to enforce equality between the different copies. In our particular example, we introduce extra variables x'_s for $s \in \{1, 2, 4, 5\}$, since these variables appear twice in the clique graph. We then form a distribution $q(x, x') = q(x_1, x_2, x_3, x_4, x_5, x'_1, x'_2, x'_4, x'_5)$ that factorizes as

$$q(x, x') \propto \psi_{234}(x_2, x_3, x_4) \psi_{12}(x_1, x'_2) \psi_{15}(x'_1, x_5) \psi_{45}(x'_4, x_5) \times \prod_{s \in \mathcal{V} \setminus \{3\}} \mathbb{I}[x_s = x'_s], \quad (35)$$

where $\mathbb{I}[x_s = x'_s]$ is a $\{0, 1\}$ -valued indicator function for the event $\{x_s = x'_s\}$. By construction, for any $s \in \{1, 2, 3, 4, 5\}$, if we were to compute the distribution

$q(x_s)$ in the expanded model (35), it would be equal to the marginal $p(x_s)$ from the original model (34). Although the expanded model (35) involves additional copies x'_s , the indicator functions that have been added enforce the needed equalities to maintain model consistency.

However, the clique graph in panel (b) and the distribution (35) are not still *not* trees, which leads us to the key question. When is it possible to extract a tree from the clique graph, and have a factorization over the resulting clique tree that remains consistent with the original model (34)? Panel (c) of Fig. 6 shows one clique tree, obtained by dropping the edge labeled with separator set $\{5\}$ that connects the vertices labeled with cliques $\{4, 5\}$ and $\{1, 5\}$. Accordingly, the resulting factorization $r(\cdot)$ over the tree would be obtained by dropping the indicator function $\mathbb{I}[x_5 = x'_5]$, and take the form

$$r(x, x') \propto \psi_{234}(x_2, x_3, x_4) \psi_{12}(x_1, x'_2) \psi_{15}(x'_1, x_5) \psi_{45}(x'_4, x_5) \times \prod_{s \in \mathcal{V} \setminus \{3, 5\}} \mathbb{I}[x_s = x'_s], \quad (36)$$

Of course, the advantage of this clique tree factorization is that we now have a distribution to which the sum-product algorithm could be applied, so as to compute the exact marginal distributions $r(x_i)$ for $i = 1, 2, \dots, 5$. However, the drawback is that these marginals will *not be equal* to the marginals $p(x_i)$ of the original distribution (34). This problem occurs because—in sharp contrast to the clique graph factorization (35)—the distribution (36) might assign non-zero probability to some configuration for which $x_5 \neq x'_5$. Of course, the clique tree shown in Fig. 6c is only one of spanning trees contained within the clique graph in panel (b). However, the reader can verify that none of these clique trees have the desired property.

A bit more formally, the property that we require is the clique tree factorization always have enough structure to enforce the equivalences $x_s = x'_s = x''_s = \dots$, for all copies of a given variable indexed by some $s \in \mathcal{V}$. (Although variables only appeared twice in the example of Fig. 6, the clique graphs obtained from more complicated graphs could have any number of copies.) Note that there are copies x_s and x'_s of a given variable x_s if and only if s appears in at least two distinct cliques, say C_1 and C_2 . In any clique tree, there must exist a unique path joining these two vertices, and what we require is that the equivalence $\{x_s = x'_s\}$ be propagated along this path. In graph-theoretic terms, the required property can be formalized as follows:

Definition 6 A clique tree has the *running intersection property* if for any two clique vertices C_1 and C_2 , all vertices on the unique path joining them contain the intersection $C_1 \cap C_2$. A clique tree with this property is known as a *junction tree*.

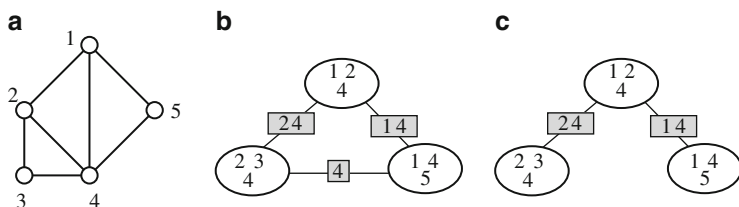


Fig. 7 (a) A modified version of the graph from Fig. 6a, containing the extra edge (1, 4). The modified graph has three maximal cliques, each of size three. (b) The associated clique graph has one node for each of three maximal cliques, with *gray boxes* representing the separator sets. (c) A clique tree extracted from the clique graph; it can be verified that this clique tree satisfies the running intersection property

To illustrate this definition, the clique tree in Fig. 6 fails to satisfy running intersection since 5 belongs to both cliques $\{4, 5\}$ and $\{1, 5\}$, but does not belong to the clique $\{2, 3, 4\}$ along the path joining these two cliques in the clique tree. Let us now consider a modification of this example so as to illustrate a clique tree that does satisfy running intersection, and hence is a junction tree.

Figure 7a shows a modified version of the graph from Fig. 6a, obtained by adding the extra edge (1, 4). Due to this addition, the modified graph now contains three maximal cliques, each of size three. Panel (b) shows the associated clique graph; it contains three vertices, one for each of the three maximal cliques, and the gray boxes on the edges represent the separator sets. Panel (c) shows one clique tree extracted from the clique graph in panel (b). In contrast to the tree from Fig. 6a, this clique tree does satisfy the running intersection property, and hence is a junction tree. (For instance, vertex 4 belongs to both cliques $\{2, 3, 4\}$ and $\{1, 4, 5\}$, and to every clique on the unique path joining these two cliques in the tree.)

4.2 Triangulation and Junction Trees

Thus far, we have studied in detail two particular graphs, one (Fig. 6a) for which it was impossible to obtain a junction tree, and a second (Fig. 7a) for which a junction tree could be found. In this section, we develop a principled basis on which to produce graphs for which the associated clique graph has a junction tree, based on the graph-theoretic notion of *triangulation*.

A cycle in a graph is a sequence of vertices $(s_1, s_2, \dots, s_\ell, s_1)$ such that $(s_\ell, s_1) \in \mathcal{E}$ and $(s_i, s_{i+1}) \in \mathcal{E}$ for all $i = 1, \dots, \ell - 1$. The cycle is *chordless* if there are no edges in the graph joining non-successive vertices in the cycle—that is, the graph does not contain any edges *apart from* those listed above that form the cycle. For example, the cycle (1, 2, 4, 5, 1) in the graph from Fig. 6a is chordless, whereas in contrast, the cycle (1, 2, 3, 4, 5, 1) contains the chord (2, 4).

Definition 7 A graph is *triangulated* if it contains no chordless cycles of length greater than three.

Thus, the graph in Fig. 6a is not triangulated (due to the chordless cycle $(1, 2, 4, 5, 1)$), whereas it can be verified that the graph in Fig. 7a is triangulated. We now state the fundamental connection between triangulation and junction trees:

Theorem 3 *The clique graph associated with a graph \mathcal{G} has a junction tree if and only if \mathcal{G} is triangulated.*

We provide a proof of this result as a part of a more general set of graph-theoretic equivalences in Appendix. The practical significance of Theorem 3 is that it enables us to construct a modified graph $\tilde{\mathcal{G}}$ —by triangulating the original graph \mathcal{G} —such that the clique graph associated with $\tilde{\mathcal{G}}$ is guaranteed to have at least one junction tree. There are a variety of algorithms for obtaining triangulated graphs. For instance, recall the *vertex elimination procedure* discussed in Sect. 3.1; it takes as input a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and then processes the vertices in a pre-specified order so as to produce a new graph.

Lemma 4 *The output $\tilde{\mathcal{G}} = (\mathcal{V}, \tilde{\mathcal{E}})$ of the vertex elimination algorithm is a triangulated graph.*

We leave the proof of Lemma 4 as an exercise for the reader. This fact establishes that there is a connection between the elimination algorithm and the property of triangulation.

Example 12 (Triangulation and Junction Tree) To illustrate the triangulation procedure, consider the 3×3 grid shown in Fig. 8a. If we run the vertex elimination algorithm using the ordering $\mathcal{I} = \{1, 3, 7, 9, 4, 6, 2, 8\}$, then we obtain the graph $\tilde{\mathcal{G}}$ shown in panel (b). (Note that the graph would not be triangulated if the additional edge joining vertices 2 and 8 were not present. Without this edge, the 4-cycle $\{2, 4, 8, 6, 2\}$ would lack a chord.) Note that $\tilde{\mathcal{G}}$ has six maximal cliques: two 4-cliques in the middle of the graph, and four 3-cliques on the boundary. The clique graph associated with $\tilde{\mathcal{G}}$ is illustrated in panel (c), and one clique tree is shown in panel (d). It can be verified that this clique tree satisfies the running intersection property, and so is a junction tree. ♣

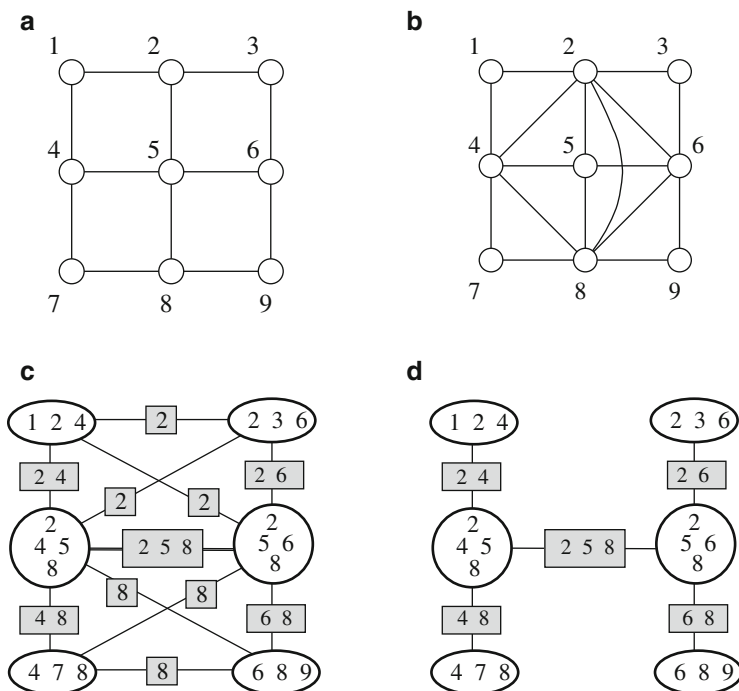


Fig. 8 Illustration of junction tree construction. (a) Original graph is a 3×3 grid. (b) Triangulated version of original graph. Note the two 4-cliques in the middle. (c) Corresponding clique graph for triangulated graph in (b), with maximal cliques depicted within *ellipses*, and separator sets within *rectangles*. (d) A junction tree extracted from the clique graph in (c). It is a clique tree that satisfies the running intersection property

4.3 Constructing the Junction Tree

Given that any triangulated graph has a junction tree (JT), the first step in the JT procedure is to triangulate the graph. At least in principle, this operation is straightforward since it entails adding edges so as to remove chordless cycles from the graph. (However, see the discussion at the end of Sect. 4 for discussion of optimal triangulations.) The focus of this section is the next step of the JT procedure—namely, how to form a junction tree from a triangulated graph. Any graph with cycles has more than one clique tree. Given a triangulated graph, we are guaranteed that at least one of these clique trees has the running intersection property, and so is a junction tree. How to find such a junction tree? It turns out that there is a simple procedure, based on solving a certain maximum weight spanning tree problem, that is always guaranteed to find a junction tree.

Given a triangulated graph, we again consider the clique graph, as previously defined. Recall that vertices C_1 and C_2 (corresponding to cliques from the original graph) are connected by an edge if and only if the associated separator set

$S = C_1 \cap C_2$ is non-empty. Accordingly, for each edge $e = (C_1, C_2)$ of the clique graph, we can associate the positive weight $w(e) = \text{card}(C_1 \cap C_2)$. We refer to the resulting object as the *cardinality-weighted clique graph*. Given this weighted graph, the *maximum weight spanning tree* problem corresponds to finding the tree T (whose vertex set includes every node) such that the weight $w(T) := \sum_{e \in T} w(e)$ is maximized.

Proposition 4 *Any maximal weight spanning tree of the cardinality-weighted clique graph is a junction tree for the original graph.*

Proof Suppose that the triangulated graph $\tilde{\mathcal{G}}$ has M maximal cliques, so that (by definition) the weighted clique graph has M vertices. Let us consider an arbitrary spanning tree T of the weighted clique graph, say with vertices $\{C_1, \dots, C_M\}$ and an associated edge set $\{S_1, \dots, S_{M-1}\}$, where we have identified each edge with its associated separator set. The weight of this spanning tree can be written as $w(T) = \sum_{j=1}^{M-1} \text{card}(S_j)$. Letting $\mathbb{I}[t \in S_j]$ be an $\{0-1\}$ -valued indicator function for the event $\{t \in S_j\}$, we may write $\text{card}(S_j) = \sum_{t \in \mathcal{V}} \mathbb{I}[t \in S_j]$, and hence we have

$$w(T) = \sum_{j=1}^{M-1} \sum_{t \in \mathcal{V}} \mathbb{I}[t \in S_j] = \sum_{t \in \mathcal{V}} \sum_{j=1}^{M-1} \mathbb{I}[t \in S_j]. \quad (37)$$

We now claim that for any $t \in \mathcal{V}$, we have the inequality

$$\sum_{j=1}^{M-1} \mathbb{I}[t \in S_j] \leq \sum_{i=1}^M \mathbb{I}[t \in C_i] - 1, \quad (38)$$

with equality if and only if the subgraph induced by t is connected. To establish this claim, consider the subgraph of T that induced by vertex t —meaning the subgraph formed by the clique vertices C_i that include t , and the associated edges or separator sets S_j that also include t . Since it is a subgraph of the tree T , it must also be acyclic, from which the inequality (38) follows. If the subgraph has a single connected component, then equality holds. When T is actually a junction tree, this equality will hold for any vertex t .

Substituting the bound (38) into the earlier inequality (37), we conclude that

$$w(T) \leq \sum_{t \in \mathcal{V}} \left\{ \sum_{i=1}^M \mathbb{I}[t \in C_i] - 1 \right\},$$

where equality holds if and only if T is a junction tree. Since the given tree T was arbitrary, the claim follows. \square

Putting together the pieces of our development, we are led to the following algorithm:

Junction Tree Algorithm Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$:

1. Run a triangulation procedure to obtain a triangulated graph $\tilde{\mathcal{G}}$.
2. Form the weighted junction graph, and find a maximum weight spanning tree via a standard algorithm (e.g., using Kruskal’s algorithm, or Prim’s algorithm).
3. Given the junction tree, define appropriate compatibility functions on its vertices and edges to reproduce the original distribution, and run the sum-product algorithm.

There are various techniques for finding *some* triangulation of a graph; as noted in Lemma 4, running the vertex elimination algorithm (using any ordering) yields a triangulation. Note, however, that the complexity of tree inference depends on the size of the state spaces at each node in the junction tree, which are determined by the clique sizes in the triangulated graph. It would be desirable, then, to obtain a triangulation with minimal maximal clique size, or equivalently, with minimal treewidth.

Definition 8 The *treewidth* $\kappa(\mathcal{G})$ of an undirected graph \mathcal{G} is the size of largest clique (minus one) in the best triangulation.

As an illustration, any ordinary tree has treewidth one, whereas the graph in Fig. 7a has treewidth two, and the grid in Fig. 8 has treewidth three.

For any fixed treewidth $\kappa = \kappa(\mathcal{G})$, there are efficient algorithms—meaning polynomial-time in the graph size—to test whether the graph has treewidth κ , and if so, to determine an associated junction tree (e.g., [16, 60]). However, the complexity of these algorithms grows exponentially in the treewidth κ , so they are feasible only for graphs of bounded treewidth. For a general graph, the problem of finding an optimal triangulation is NP-hard [6, 67]. Despite this negative result, there are a variety of heuristic algorithms for obtaining “good” triangulations of a graph. Some of most widely studied are the minimum degree and the minimum fill heuristics [4, 36]. The minimum degree heuristic is based on following the elimination ordering obtained by choosing a minimum degree vertex (in the partially reduced graph) at each step. The minimum fill heuristic operates similarly, but instead is based on choosing a vertex s so as to minimize the total number of edges required to make the neighborhood $\mathcal{N}(s)$ a clique, a quantity known as the “fill”.

5 Basics of Graph Estimation

Up to this point, we have considered various types of “forward problems”, by which we mean that the graphical model—both the graph and the functions in its factorization—are known quantities. In this section, we turn our attention to various classes of “inverse problems”, in which either the functions and/or the structure of the graph are unknown. Given a black box that generates samples from the distributions, we consider two possible problems:

- assuming that the clique structure is known, how to estimate the functions in the factorization?
- given no a priori information, how to estimate the clique structure and the functions?

The former problem is one of parameter estimation, since it involves estimating the parameters that specify either the conditional probabilities (for a directed graphical model) or the compatibility functions (for an undirected graphical model). In contrast, the latter problem is a form of model selection, since we need to select the correct graph from competing families of graphs. In the machine learning community, the latter problem is also known as structure learning. We begin our discussion with the former problem, as it is simpler both conceptually and computationally.

5.1 Parameter Estimation for Directed Graphs

The problem of parameter estimation is particularly simple for directed graphical models, in which case, as long as the data is fully observed, the solution can be obtained in closed form.

More concretely, for a known directed graph $\mathcal{D} = (\mathcal{V}, \vec{\mathcal{E}})$, consider the set of all distributions that factorize in the form

$$p(x_1, \dots, x_N) = \prod_{s \in \mathcal{V}} p(x_s \mid x_{\pi(s)}),$$

where $\pi(s) \subseteq \mathcal{V}$ is the parent set of vertex s . Suppose we observe a collection of n samples, each of the form $X_{i\bullet} = (X_{i1}, \dots, X_{iN}) \in \mathcal{X}_{[N]}$. Given the collection of samples $X_{1\bullet}^n := \{X_{1\bullet}, X_{2\bullet}, \dots, X_{n\bullet}\}$, our goal is to estimate the conditional distributions at each vertex $s \in \mathcal{V}$. In these notes, we focus exclusively on the case of discrete variables, in particular with $\mathcal{X}_s = \{0, 1, \dots, m-1\}$ for all vertices $s \in \mathcal{V}$. (The extension to discrete variables with differing numbers of states per vertex is straightforward.)

By definition, each conditional distribution can be written as a ratio of two marginal distributions—namely, as

$$p(x_s | x_{\pi(s)}) = \frac{p(x_s, x_{\pi(s)})}{p(x_{\pi(s)})}.$$

A natural estimate of each marginal distribution are the *empirical marginal distributions*—namely, the quantities

$$\hat{p}(x_s) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}(X_{i_s} = x_s), \quad \text{and}$$

$$\hat{p}(x_s, x_{\pi(s)}) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}(X_{i_s} = x_s, X_{i_{\pi(s)}} = x_{\pi(s)}),$$

where $\mathbb{I}(X_{i_s} = x_s)$ is a zero-one valued indicator function for the event that $\{X_{i_s} = x_s\}$, with the other indicator similarly defined. In words, the empirical marginal $\hat{p}(x_s)$ is simply the relative fraction of times that $X_s = x_s$ in the data.

Given these estimates of the marginal distributions, a natural “plug-in” estimate of the conditional distribution is given by

$$\hat{p}(x_s | x_{\pi(s)}) = \frac{\hat{p}(x_s, x_{\pi(s)})}{\hat{p}(x_{\pi(s)})}. \quad (39)$$

(To be clear, we take $0/0$ to be equal to 0, since these correspond to configurations that never appear in the data set.)

5.2 Parameter Estimation for Undirected Graphs

For undirected graphical models, the problem of parameter estimation is not so simple in general. In particular, given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider the set of all distributions that factorize in the form

$$p(x_1, \dots, x_N; \psi) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C), \quad (40)$$

where \mathcal{C} is the set of all maximal cliques of the graph. In this setting, our goal is to estimate the unknown compatibility functions $\psi_C : \mathcal{X}_C \rightarrow (0, \infty)$ associated with each clique of the graph. A challenge here is that, at least in general, the compatibility functions are not directly related to conditional or marginal distributions of the graph. The exception to this rule is given by undirected trees (and more generally,

junction trees), in which case there is always a choice of compatibility functions that is related to the marginal and/or conditional distributions.

In these notes, we limit our discussion to the method of *maximum likelihood* for estimating the compatibility functions. It is based on choosing the compatibility functions $\psi = \{\psi_C, C \in \mathcal{C}\}$ so as to maximize the (rescaled) log likelihood of the data, namely the quantity

$$\ell(\psi; X_{1\bullet}, \dots, X_{n\bullet}) = \frac{1}{n} \sum_{i=1}^n \log p(X_{i\bullet}; \psi). \quad (41)$$

We use $\hat{\psi} = \{\hat{\psi}_C, C \in \mathcal{C}\}$ to denote the *maximum likelihood estimate*, namely a choice of compatibility functions that maximize the log likelihood. A remark about parameterization before proceeding: since we are considering discrete random variables, the estimate $\hat{\psi}$ can be viewed as a vector of real numbers. In particular, the estimate $\hat{\psi}_C$ can be represented as a sub-vector of $m^{|C|}$ real numbers, one for each possible configuration $J = \{J_s, s \in C\} \in \mathcal{X}_C$ that the variables x_C may assume. By concatenating all of these sub-vectors, we obtain a vector $\hat{\psi} \in \mathbb{R}^D$, where $D = \sum_{C \in \mathcal{C}} m^{|C|}$. Consequently, the rescaled log likelihood is a function from \mathbb{R}^D to \mathbb{R} .

5.2.1 Maximum Likelihood for Undirected Trees

Let us begin by considering the maximum likelihood problem for an undirected tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$, say with a factorization of the form

$$p(x_1, \dots, x_N; \psi) = \frac{1}{Z} \prod_{s \in \mathcal{V}} \psi_s(x_s) \prod_{(s,t) \in \mathcal{E}} \psi_{st}(x_s, x_t). \quad (42)$$

For any tree, we claim that the maximum likelihood estimate $\hat{\psi}$ can be expressed in terms of the empirical marginals at the vertices and edges of the tree—namely, the quantities

$$\hat{p}(x_s = j) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}_{s;j}(X_{i_s}), \quad \text{and} \quad \hat{p}((x_s, x_t) = (j, k)) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}_{st;jk}(X_{i_s}, X_{i_t}), \quad (43)$$

where $\mathbb{I}_{s;j}(x_s)$ is a zero-one indicator for the event that $\{x_s = j\}$, with $\mathbb{I}_{st;jk}(x_s, x_t)$ similarly defined. With these empirical marginals computed from the data, the maximum likelihood estimates of the compatibility functions are given by

$$\hat{\psi}_s(x_s) = \hat{p}(x_s), \quad \text{and} \quad \hat{\psi}_{st}(x_s, x_t) = \frac{\hat{p}(x_s, x_t)}{\hat{p}(x_s)\hat{p}(x_t)}. \quad (44)$$

(As before, we take $0/0 = 0$ for concreteness.) We prove in the following section, as a corollary of a more general result, that the relations (44) do in fact define a global maximizer of the log likelihood (41) for a tree-structured graph.

5.2.2 Maximum Likelihood on General Undirected Graphs

In the case of graphs with cycles, the MLE no longer has a convenient closed-form expression. In order to appreciate the challenges, it is convenient to discuss the maximum likelihood objective using an alternative exponential parameterization. Let us do so for a graph with general clique set \mathcal{C} , understanding that the undirected tree is a special case. For a given clique C and configuration $J = \{J_s, s \in C\} \in \mathcal{X}_C$, we define the function

$$\mathbb{I}_{C;J}(x_C) = \begin{cases} 1 & \text{if } x_C = J \\ 0 & \text{otherwise,} \end{cases} \quad (45)$$

corresponding to a binary-valued indicator for the event that $\{x_C = J\}$. Note that these functions are generalizations of the vertex-based functions $\mathbb{I}_{s;j}$ and edge-based functions $\mathbb{I}_{st;jk}$ discussed previously.

We use $\mathbb{I}_C = \{\mathbb{I}_{C;J}, J \in \mathcal{X}_C\}$ to denote the vector all of $m^{|C|}$ such indicator functions associated with clique C . With this notation, any function $\psi_C : \mathcal{X}_C \rightarrow (0, \infty)$ can be parameterized in the log-linear form

$$\log \psi_C(x_C) = \langle \theta_C, \mathbb{I}_C(x_C) \rangle := \sum_{J \in \mathcal{X}_C} \theta_{C;J} \mathbb{I}_{C;J}(x_C)$$

for some real vector $\theta_C = \{\theta_{C;J}, J \in \mathcal{X}_C\}$ in $m^{|C|}$ dimensions. If we adopt this parameterization for each clique in the graph, then the overall factorization can be re-written in the form

$$p(x_1, \dots, x_N; \theta) = \exp \left\{ \sum_{C \in \mathcal{C}} \langle \theta_C, \mathbb{I}_C(x_C) \rangle - \Phi(\theta) \right\} \quad (46)$$

where $\theta = \{\theta_C, C \in \mathcal{C}\}$ is a real-valued vector of parameters to be estimated, and

$$\Phi(\theta) := \log \sum_{x \in \mathcal{X}_{[N]}} \exp \left\{ \sum_{C \in \mathcal{C}} \langle \theta_C, \mathbb{I}_C(x_C) \rangle \right\} \quad (47)$$

is the log normalization constant. The representation (46) shows that an undirected graphical model can be associated with an *exponential family*, namely one with sufficient statistics $\{\mathbb{I}_C(\cdot), C \in \mathcal{C}\}$ and natural parameter $\theta = \{\theta_C, C \in \mathcal{C}\} \in \mathbb{R}^D$. This connection is useful, since there is a rich and classical statistical literature on the properties of exponential families (e.g., [19, 30, 64]). We make use of some of

these properties here. Moreover, an attractive property of the maximum likelihood estimate is its invariance to reparameterizations: namely, if we can compute the MLE $\hat{\theta}$ in our new, exponential parameterization, then $\hat{\psi} \equiv \exp(\hat{\theta})$ will define the MLE in the original parameterization. We use this fact freely in the discussion to follow.

One convenient property of the exponential parameterization is that the log likelihood $\ell(\theta; X_{1\bullet}, \dots, X_{n\bullet}) := \frac{1}{n} \sum_{i=1}^n \log p(X_{i\bullet}; \theta)$ can be written in terms of the empirical marginals

$$\hat{\mu}_{C;J} = \frac{1}{n} \sum_{i=1}^n \mathbb{I}_{C;J}(X_{iC}), \quad (48)$$

at each clique. Note that $\hat{\mu}_{C;J} = \hat{p}(x_C = J)$ corresponds to the average fraction of times that $\{X_C = J\}$ in the data set. Let us define the vector $\hat{\mu}_C = \{\hat{\mu}_{C;J}, J \in \mathcal{X}_C\} \in m^{|C|}$, as well as the inner product $\langle \theta_C, \hat{\mu}_C \rangle = \sum_{J \in \mathcal{X}_C} \theta_{C;J} \hat{\mu}_{C;J}$. With this notation, the maximum likelihood estimate $\hat{\theta}$ takes the form

$$\hat{\theta} = \arg \max_{\theta \in \mathbb{R}^D} \left\{ \sum_{C \in \mathfrak{c}} \langle \theta_C, \hat{\mu}_C \rangle - \Phi(\theta) \right\}. \quad (49)$$

In order to understand the structure of this optimization problem, we require some useful properties of the function Φ :

Lemma 5 *The function Φ is convex and infinitely differentiable on \mathbb{R}^D , and in particular, we have*

$$\frac{\partial \Phi}{\partial \theta_{C;J}}(\theta) = \underbrace{\sum_{x \in \mathcal{X}} p(x; \theta) \mathbb{I}_{C;J}(x_C)}_{p(x_C=J; \theta)}$$

This result shows that taking the partial derivatives of Φ with respect to the parameters over clique C yields the marginal distribution over that clique. The proof of these properties are relatively straightforward, so we leave them as an exercise for the reader.

Lemma 5 allows us to characterize the structure of the maximum likelihood estimate (49). First, it shows that the rescaled log likelihood ℓ is a concave function of θ , and hence has no local optima that can trap an iterative algorithm. Secondly, it can be used to characterize the stationary conditions that characterize a global

optimum; more precisely, by taking derivatives of the rescaled log likelihood and setting them to zero, we find that the MLE $\hat{\theta}$ satisfies the stationary conditions

$$p(x_C = J; \hat{\theta}) = \hat{\mu}_{C;J} \quad \text{for all } C \in \mathfrak{C}, \text{ and } J \in \mathcal{X}_C. \quad (50)$$

In words, the maximum likelihood estimate $\hat{\theta}$ is such that the marginals computed under the distribution $p(\cdot; \hat{\theta})$ are equal to the empirical marginals determined by the data. This type of moment-matching condition characterizes the MLE in any exponential family; see the monograph [64] for further details.

Let us now return to our previously open question: for a tree-structured graph \mathcal{T} with the factorization (42), why does Eq. (44) specify the maximum likelihood estimate? We simply need to verify that the moment-matching conditions are satisfied. Given the suggested form of $\hat{\psi}$ from Eq. (44), we have

$$p(x_1, \dots, x_N; \hat{\psi}) = \frac{1}{Z} \prod_{s \in \mathcal{V}} \hat{p}(x_s) \prod_{(s,t) \in \mathcal{E}} \frac{\hat{p}(x_s, x_t)}{\hat{p}(x_s) \hat{p}(x_t)}. \quad (51)$$

Our claim is that in this factorization, we must necessarily have $Z = 1$ and moreover

$$p(x_s; \hat{\psi}) = \hat{p}(x_s) \quad \text{for all } s \in \mathcal{V}, \text{ and } p(x_s, x_t; \hat{\psi}) = \hat{p}(x_s, x_t) \quad \text{for all } (s, t) \in \mathcal{E}. \quad (52)$$

In order to verify this claim, we first observe that any undirected tree can be converted to an equivalent directed form as follows: first, designate one vertex, say r , as the root, and then orient all edges away from the root toward the leaves. See Fig. 9 for an illustration of this procedure.

After this conversion, each vertex $s \in \mathcal{V} \setminus \{r\}$ has a unique parent $\pi(s)$, and we can associate the product $\hat{\psi}_s(x_s) \hat{\psi}_{s\pi(s)}(x_s, x_{\pi(s)})$ with the directed edge $(\pi(s) \rightarrow s)$

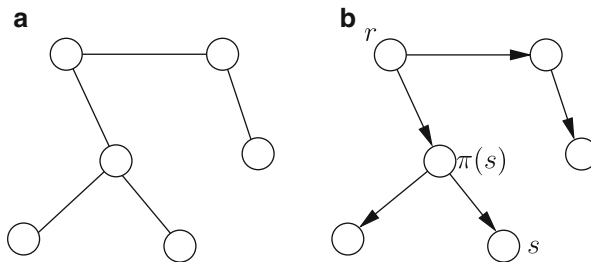


Fig. 9 (a) Undirected tree. (b) Conversion of undirected tree to a directed tree rooted at vertex r . Each vertex $s \in \mathcal{V} \setminus \{r\}$ now has a unique parent $\pi(s)$. Any given parent may have multiple children

and the term $\hat{\psi}_r(x_r)$ with the root vertex. In this way, the factorization (51) can be written the directed form

$$\begin{aligned} p(x_1, \dots, x_N; \hat{\psi}) &= \frac{1}{Z} \hat{\psi}_r(x_r) \prod_{s \in \mathcal{V} \setminus \{r\}} \hat{\psi}_s(x_s) \hat{\psi}_{s\pi(s)}(x_s, x_{\pi(s)}) \\ &= \frac{1}{Z} \underbrace{\hat{p}(x_r)}_{f_r(x_r)} \prod_{s \in \mathcal{V} \setminus \{r\}} \underbrace{\frac{\hat{p}(x_s, x_{\pi(s)})}{\hat{p}(x_{\pi(s)})}}_{f_s(x_s, x_{\pi(s)})}. \end{aligned}$$

Note that this is now a factorization of the general directed form (2), with functions $\{f_s, s \in \mathcal{V}\}$ that satisfy the normalization condition (1). Consequently, an application of Proposition 1 implies that

$$p(x_r; \hat{\psi}) = \hat{p}(x_r), \quad \text{and} \quad \frac{p(x_s, x_{\pi(s)}; \hat{\psi})}{p(x_{\pi(s)}; \hat{\psi})} = \frac{\hat{p}(x_s, x_{\pi(s)})}{\hat{p}(x_{\pi(s)})} \quad \text{for all } \pi(s) \rightarrow s.$$

A recursive argument, moving from the root r out to the leaves, then shows that the marginal-matching condition (52) holds. We leave the details of this final step as an exercise for the reader. (Start with the case of a Markov chain to see the general idea.)

5.2.3 Iterative Proportional Scaling

Thus far, we have seen that in the special case of a tree-structured graph, it is straightforward to find a set of compatibility functions $\hat{\psi}$ (or equivalently, a vector of parameters $\hat{\theta}$) that satisfy the moment-matching conditions (50). The same basic idea can be extended to any triangulated graph, by first building a junction tree, and then following the analogous procedure on the junction tree.

In contrast, for a general undirected graph (with cycles, not triangulated), the MLE problem no longer has a closed-form solution, and needs to be solved by an iterative algorithm. Since the moment-matching conditions (50) characterize the global optimum of the convex program (49) that defines the MLE, a variety of standard algorithms from convex programming [9, 18] could be applied. Here we describe one particular algorithm, attractive due to the simplicity of its updates, known as *iterative proportional scaling*, or IPS for short. As we will see, it amounts to a block co-ordinate ascent of the objective function. The IPS algorithm takes as input the empirical marginals $\{\hat{\mu}_C, C \in \mathcal{C}\}$, computed from the data set as in Eq. (48), as well as a particular ordering of clique set.

Iterative Proportional Scaling (IPS)

1. Initialize estimate $\theta'_C = 0$ for all cliques $C \in \mathfrak{C}$.
2. Cycling over cliques in the given order:
 - (a) Activate clique A , and given current parameter estimate θ' , compute marginal probabilities

$$\mu'_{A;J} = p(x_A = J; \theta') \quad \text{for all } J \in \mathcal{X}_A, \quad (53)$$

- and form the vector $\mu'_A = \{\mu'_{A;J}, J \in \mathcal{X}_A\}$.
- (b) Update the parameter estimate $\theta' \mapsto \theta$ via

$$\theta_C = \begin{cases} \theta'_C & \text{for all } C \in \mathfrak{C} \setminus \{A\} \\ \theta'_A + \log \frac{\hat{\mu}_A}{\mu'_A} & \text{for } C = A. \end{cases}$$

3. Iterate until convergence.

Observe that Step 2(a) of the algorithm requires computation of the marginal distribution $p(x_A; \theta')$ over the currently active clique A . Doing so requires running the elimination algorithm, or building a junction tree on which to run a message-passing algorithm. This step can be computationally expensive if the graph treewidth is high, in which context approximate methods for marginalization become important (see the monograph [64] for more details). The update in Step 2(b) fixes the parameter estimate on all cliques *except* for the active one A ; for this clique, the update can be equivalently written in the form

$$e^{\theta_{A;J}} = e^{\theta'_{A;J}} \frac{\hat{\mu}_{A;J}}{\mu'_{A;J}} \quad \text{for all } J \in \mathcal{X}_A.$$

This form of the update clarifies the origins of the name “proportional scaling”, since the parameters are being adjusted by the relative proportions of the empirical marginal $\hat{\mu}_{A;J}$ and the current model marginal $\mu'_{A;J}$.

It is clear from the IPS updates that any fixed point must satisfy the moment-matching conditions (50), and hence be a global optimum of the log likelihood. Less clear, however, is whether convergence is guaranteed (though it is assumed in the statement of Step 3). Among other properties, the following result shows that the IPS algorithm is indeed guaranteed to converge to the MLE:

Proposition 5 (IPS Properties) *The IPS algorithm has the following properties:*

- (a) We have $\Phi(\theta') = \Phi(\theta)$ over all updates $\theta' \mapsto \theta$ of the algorithm.
 (b) After updating clique A , the estimate θ satisfies

$$p(x_A = J; \theta) = \hat{\mu}_{A;J} \quad \text{for all } J \in \mathcal{X}_A. \quad (54)$$

- (c) The algorithm iterates converge to an MLE solution $\hat{\theta}$.

Proof (a), (b) By definition of the update, for each $J \in \mathcal{X}_A$, we have

$$\begin{aligned} p(x_A = J; \theta) &= \sum_{x \in \mathcal{X}_{[N]}} p(x; \theta) \mathbb{I}_{A;J}(x_A) \\ &= \sum_{x \in \mathcal{X}_{[N]}} p(x; \theta') \frac{e^{\Phi(\theta') - \Phi(\theta)} \hat{\mu}_{A;J}}{\mu'_{A;J}} \mathbb{I}_{A;J}(x_A) \\ &= \left\{ \frac{\sum_{x \in \mathcal{X}_{[N]}} p(x; \theta') \mathbb{I}_{A;J}(x_A)}{\mu'_{A;J}} \right\} \hat{\mu}_{A;J} e^{\Phi(\theta') - \Phi(\theta)} \\ &= \hat{\mu}_{A;J} e^{\Phi(\theta') - \Phi(\theta)}. \end{aligned}$$

Summing both sides over $J \in \mathcal{X}_A$ yields that $\Phi(\theta') = \Phi(\theta)$ as claimed in part (a), and we then see that $p(x_A = J; \theta) = \hat{\mu}_{A;J}$ as claimed in part (b).

- (c) Note that the rescaled log likelihood can be block partitioned in the form $\ell(\theta'_C, C \in \mathcal{C})$. With θ' fixed in all co-ordinates except those indexed by A , let us consider the function

$$\theta_A \mapsto \ell(\theta_A, \theta'_C, C \in \mathcal{C} \setminus A).$$

It is a concave function of θ_A . Moreover, by combining Lemma 5 with the result of part (b), we see that, after updating clique A , the IPS algorithm has maximized this function over θ_A . Thus, the IPS algorithm is equivalent to performing block co-ordinate ascent of the log-likelihood function. By known results on the convergence of such procedures [9], it follows that IPS converges to a global optimum. \square

5.3 Tree Selection and the Chow-Liu Algorithm

In our discussion thus far, we have assumed that the graph structure is known, and that only the compatibility functions need to be estimated. In this section, we consider an instance of the more general problem in which neither is known. As previously mentioned, this problem is known as model selection for graphical models, or graph structure learning. We limit ourselves to the problem of selecting tree-structured graphical models. See the bibliographic section for discussion of other methods suitable for more general forms of graph structure learning.

Given our data set $\{X_{1\bullet}, \dots, X_{n\bullet}\}$, suppose that our goal is to choose the “best-fitting” spanning tree graphical model. More formally, let \mathbb{T} be the set of all spanning trees on N vertices. For a given tree $\mathcal{T} \in \mathbb{T}$ and a given distribution $p(x; \theta(\mathcal{T}))$ that is Markov with respect to \mathcal{T} , we measure its fit in terms of its rescaled log likelihood $\ell(\theta(\mathcal{T})) = \frac{1}{n} \sum_{i=1}^n \log p(X_{i\bullet}; \theta(\mathcal{T}))$.

In principle, the solution to this problem is straightforward: for each tree $\mathcal{T} \in \mathbb{T}$, we compute the maximum likelihood estimate

$$\hat{\theta}(\mathcal{T}) = \arg \max_{\theta(\mathcal{T})} \ell(\theta(\mathcal{T})),$$

thereby obtaining a list $\{\ell(\hat{\theta}(\mathcal{T})), \mathcal{T} \in \mathbb{T}\}$ of maximized likelihoods for every tree \mathcal{T} in the set \mathbb{T} of all spanning trees. We then choose the tree $\hat{\mathcal{T}}$ with the highest maximized likelihood, that is $\hat{\mathcal{T}} = \arg \max_{\mathcal{T} \in \mathbb{T}} \ell(\hat{\theta}(\mathcal{T}))$. However, a graph with N vertices has an enormous number of possible spanning trees— N^{N-2} to be precise [20]—so that a brute force approach is infeasible. Fortunately, as previously shown (44), the MLE for a tree-structured graph has a very special structure, which can be exploited to obtain an efficient algorithm for finding the best-fitting tree.

Chow-Liu Algorithm for Maximum Likelihood Tree

1. Compute the empirical marginals (43) associated with all vertices $s \in \mathcal{V}$ and all distinct vertex pairs (s, t) .
2. For each distinct pair of vertices (s, t) , compute the empirical mutual information

$$\hat{I}_{st} = \sum_{(x_s, x_t) \in \mathcal{X}_s \times \mathcal{X}_t} \hat{p}(x_s, x_t) \log \frac{\hat{p}(x_s, x_t)}{\hat{p}(x_s) \hat{p}(x_t)}. \quad (55)$$

(continued)

3. Using $\{\hat{I}_{st}\}$ as edge weights, compute a maximum weight spanning tree

$$\hat{\mathcal{T}} = \arg \max_{\mathcal{T} \in \mathbb{T}} \sum_{(s,t) \in \mathcal{E}(\mathcal{T})} \hat{I}_{st}. \quad (56)$$

4. Return the maximum likelihood estimate $\hat{\theta}(\hat{\mathcal{T}})$ on the tree from the previous step:

$$\begin{aligned} \hat{\theta}_s(x_s; \hat{\mathcal{T}}) &= \log \hat{p}_s(x_s) \quad \text{for all } s \in \mathcal{V}, \text{ and} \\ \hat{\theta}_{st}(x_s, x_t; \hat{\mathcal{T}}) &= \log \frac{\hat{p}(x_s, x_t)}{\hat{p}(x_s)\hat{p}(x_t)} \quad \text{for all } (s, t) \in \mathcal{E}(\hat{\mathcal{T}}). \end{aligned}$$

Note that Step 3 requires solving a maximum weight spanning tree problem. As mentioned previously during our treatment of the junction tree algorithm, there are various efficient methods for this problem, including Kruskal's algorithm and Prim's algorithm. See the bibliographic section for further discussion.

We claim that the Chow-Liu algorithm is guaranteed to return a maximum likelihood tree—namely, one satisfying

$$\hat{\mathcal{T}} \in \arg \max_{\mathcal{T} \in \mathbb{T}} \ell(\hat{\theta}(\mathcal{T})) = \arg \max_{\mathcal{T} \in \mathbb{T}} \left\{ \max_{\theta(\mathcal{T})} \ell(\theta(\mathcal{T})) \right\}.$$

The proof of this claim is straightforward given our previous discussion. For any fixed tree \mathcal{T} , the maximum likelihood solution has the form (44), whence

$$\begin{aligned} \ell(\hat{\theta}(\mathcal{T})) &= \frac{1}{n} \sum_{i=1}^n \log p(X_{i \bullet}; \hat{\theta}(\mathcal{T})) \\ &= \frac{1}{n} \sum_{i=1}^n \left\{ \sum_{s \in \mathcal{V}} \log \hat{p}(X_{is}) + \sum_{(s,t) \in \mathcal{E}(\mathcal{T})} \log \frac{\hat{p}(X_{is}, X_{it})}{\hat{p}(X_{is})\hat{p}(X_{it})} \right\} \\ &= \sum_{s \in \mathcal{V}} \hat{p}(x_s) \log \hat{p}(x_s) + \sum_{(s,t) \in \mathcal{E}(\mathcal{T})} \hat{p}(x_s, x_t) \log \frac{\hat{p}(x_s, x_t)}{\hat{p}(x_s)\hat{p}(x_t)} \\ &= C + \sum_{(s,t) \in \mathcal{E}(\mathcal{T})} \hat{I}_{st}, \end{aligned}$$

where C denotes a constant that is independent of the tree \mathcal{T} . Consequently, the maximum weight spanning tree obtained in Step 3 corresponds to the maximum likelihood tree.

Intuitively, one might think that an analogous procedure could be extended to junction trees, requiring solving a maximum weight spanning hypertree problem as opposed to a maximum weight spanning tree problem. Unfortunately, unlike the case of spanning trees, computing a maximum weight hypertree is computationally expensive, so that in practice, exact methods are limited to ordinary spanning trees, as in the Chow-Liu algorithm.

6 Bibliographic Details and Remarks

Graphical models were developed independently in a number of different communities. The notion of a Gibbs distribution appeared initially in the statistical physics literature, a notable example being the Ising model [39] that was introduced to model ferromagnetism. Other communities in which graphical models were developed include information theory and coding theory [34, 59], contingency tables and discrete multivariate analysis [13, 38], image processing [12, 35], and artificial intelligence [50].

The Hammersley-Clifford theorem derives its name from a manuscript that was never published (J.M. Hammersley, P. Clifford, 1971, Markov fields on finite graphs and lattices, unpublished). Besag [10] and Grimmett [37] were the first to publish proofs of the result; in these notes, we have adapted the latter proof. See Clifford [22] for some discussion of the history of the result. Lauritzen [44] provides discussion of how the Markov-factorization equivalence can break down when the strict positivity condition fails to hold.

Forms of the sum-product and max-product algorithms were independently developed in many different communities, including the work of Kalman in control theory [41], work on LDPC codes [34] and Viterbi decoding [32, 63] in error-control coding, the alpha-beta or forward-backward algorithms in signal processing [51], Pearl's work [50] in artificial intelligence, and work on non-serial dynamic programming [8]. Dating back to the 1960s, a number of authors (e.g., [49, 54]) studied the relation between graph triangulation and the vertex elimination algorithm, and pointed out the connection to the complexity of Gaussian elimination algorithms for solving linear systems of equations. (In the statistical setting, this problem can be cast as an inference problem for a multivariate Gaussian.) At the end of the 1980s, several groups of researchers independently recognized that the significance of tree decompositions and triangulation extended beyond solving linear equations to more general classes of graph-theoretic problems [14, 45]. Within the statistics and artificial intelligence communities, the junction tree framework was developed by Lauritzen and Spiegelhalter [45]. The paper [15] contains an extensive description of treewidth and related notions of graph complexity. There are various algorithms for solving the maximum weight spanning tree problem, including Kruskal's algorithm and Prim's algorithm [23].

Early work on iterative proportional fitting and scaling algorithms include the papers [27, 31, 58]. These algorithms are actually special cases of more general

Bregman-based projection algorithms [24–26, 29]; see also the work of Amari [2, 3] on the information geometry of exponential families. The Chow-Liu algorithm for finding the maximum likelihood tree appeared in the paper [21]. Karger and Srebro [42, 57] studied generalizations of this problem to junction trees of higher treewidth, establishing the hardness of the problem (in a complexity-theoretic sense) but also proposing an approximation algorithm. There is a very large literature on methods for graphical model selection, local forms of ℓ_1 -regularized regression (e.g., [48, 52]), ℓ_1 -regularization and global likelihoods (e.g., [7, 33, 47, 53]), as well as multiple testing and related approaches (e.g., [5, 40, 56]).

Acknowledgements This work was partially supported by NSF grants CCF-0545862, DMS-0605165 and CCF-0635372, and AFOSR grant 09NL184.

Appendix: Triangulation and Equivalent Graph-Theoretic Properties

In this Appendix, we prove Theorem 3 as part of a more general discussion of triangulation and related graph-theoretic properties. Having already defined the notions of triangulations and junction tree, let us now define the closely related notions of *decomposable* and *recursively simplicial*. The following notion serves to formalize the “divide-and-conquer” nature of efficient algorithms:

Definition 9 A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *decomposable* if either it is complete, or its vertex set \mathcal{V} can be split into the disjoint union of three sets $A \cup B \cup S$ such that (a) A and B are non-empty; (b) the set S separates A and B in \mathcal{G} , and is complete (i.e., $(s, t) \in \mathcal{E}$ for all $s, t \in S$); and (c) $A \cup S$ and $B \cup S$ are also decomposable.

Recall from our discussion of the elimination algorithm in Sect. 3.1 that when a vertex is removed from the graph, the algorithm always connects together all of its neighbors, thereby creating additional edges in the reduced graph. The following property characterizes when there is an elimination ordering such that no edges are added by the elimination algorithm.

Definition 10 A vertex is *simplicial* if its neighbors form a complete sub-graph. A non-empty graph is *recursively simplicial* if it contains a simplicial vertex, and when s is removed, any graph that remains is recursively simplicial.

It should be intuitively clear that these four properties—namely, triangulated, decomposable, recursively simplicial, and having a junction tree—are related. We now show that all four properties are actually equivalent:

Theorem 4 *The following properties of an undirected graph \mathcal{G} are all equivalent:*

- Property (T): \mathcal{G} is triangulated.*
- Property (D): \mathcal{G} is decomposable.*
- Property (R): \mathcal{G} is recursively simplicial.*
- Property (J): \mathcal{G} has a junction tree.*

We prove the sequence of implications $(T) \Rightarrow (D) \Rightarrow (R) \Rightarrow (J) \Rightarrow (T)$.

$(T) \Rightarrow (D)$: We proceed via induction on the graph size N . The claim is trivial for $N = 1$, so let us assume it for all graphs with N vertices, and prove that it also holds for any graph \mathcal{G} with $N + 1$ vertices. If \mathcal{G} is complete, then it is certainly decomposable. Moreover, if \mathcal{G} has more than one connected component, each of which is complete, then it is also decomposable. Otherwise, we may assume that at least one connected component of \mathcal{G} is not complete. (Without loss of generality in the argument to follow, we assume that \mathcal{G} has a single connected component which is not complete.) Since \mathcal{G} is not complete, it contains two non-adjacent vertices a, b . Let S be a minimal set that separates a and b ; the set S must be non-empty since \mathcal{G} has a single connected component. Define A as the set of all vertices connected to a in $\mathcal{V} \setminus S$, and set $B := \mathcal{V} \setminus (A \cup S)$. Clearly, S separates A from B in \mathcal{G} .

Now we need to show that S is complete. If $|S| = 1$, the claim is trivial. Otherwise, for any two distinct vertices $s, t \in S$, there exist paths (s, a_1, \dots, a_i, t) and (s, b_1, \dots, b_j, t) where $a_k \in A, b_k \in B$ and $i, j \geq 1$. (This claim relies on the minimality of S : if there did not exist a path from a to s , then vertex s could be removed from S . Similar reasoning applies to establish a path from a to t , and also the paths involving B .)

We claim that s and t are joined by an edge. If not, take the path from s to t through A with minimal length, and similarly for B . This pair of paths forms a cycle of length at least four, which must have a chord. The chord cannot be in A or B , since this would contradict minimality. It cannot be between vertices in A and B since S separates these two sets. Therefore, s and t are joined, and S is complete.

Finally, we need to show that $A \cup S$ and $B \cup S$ are also decomposable. But they must be triangulated, since otherwise \mathcal{G} would not be triangulated, and they have cardinality strictly smaller than $N + 1$, so the result follows by induction.

(D) \Rightarrow (R): Proof by induction on graph size N . Trivial for $N = 1$. To complete the induction step, we require the following lemma:

Lemma 6 *Every decomposable graph with at least two vertices has at least two simplicial vertices. If the graph is not complete, these vertices can be chosen to be non-adjacent.*

Proof Proof by induction on graph size N . Trivial for $N = 2$. Consider a decomposable graph with $N + 1$ vertices. If the graph is complete, all vertices are simplicial. Otherwise, decompose the graph into disjoint sets A , B and S . The subgraphs $A \cup S$ and $B \cup S$ are also chordless, and hence we have two simplicial vertices in $A \cup S$. If $A \cup S$ is not complete, these can be chosen to be non-adjacent. Given that S is complete, one of the vertices can be taken in A . Otherwise, if $A \cup S$ is complete, choose any node in A . Proceed in a symmetric fashion for B . The simplicial vertices thus chosen will not be connected, since S separates A and B . \square

Thus, given a decomposable graph, we can find some simplicial vertex s to remove. We need to show that the remaining graph is also decomposable, so as to apply the induction hypothesis. In particular, we prove that \mathcal{G} decomposable implies that any vertex-induced subgraph $\mathcal{G}[U]$ is also decomposable. We prove this induction on $|U|$. Trivial for $|U| = 1$. Trivially true if \mathcal{G} is complete; otherwise, break into $A \cup S \cup B$. Removing a node from S leaves $S \setminus \{s\}$ complete, and $A \cup S$ and $B \cup S$ decomposable by the induction hypothesis. Removing a node from A does not change $B \cup S$, and either leaves A empty (in which case remainder $B \cup S$ is decomposable), or leaves $A \cup S$ decomposable by induction.

(R) \Rightarrow (J): Proof by induction on graph size N . Trivial for $N = 1$. Let s be a simplicial vertex, and consider subgraph \mathcal{G}' obtained by removing s . By induction, \mathcal{G}' has a junction tree \mathcal{T}' , which we will extend to a junction tree \mathcal{T} for \mathcal{G} . Let C' be a maximal clique in \mathcal{T}' that contains all the neighbors of s ; this must exist since s is simplicial. If C' is precisely the neighbors of s , then we can add s to C' so as to obtain \mathcal{T} , which is a junction tree for \mathcal{G} .

If not (i.e., if C' contains the neighbors of s as a proper subset), then we can add a new clique containing s and its neighbors to \mathcal{T}' , with an edge to C' . Since s is in no other clique of \mathcal{T} and $C \setminus \{s\}$ is a subset of C' , the tree \mathcal{T}' is a junction tree for \mathcal{G} .

(J) \Rightarrow (T): Proof by induction on number of vertices M in junction tree. For $M = 1$, \mathcal{G} is complete and hence triangulated. Consider a junction tree \mathcal{T} with $M + 1$ vertices. For a fixed leaf C of \mathcal{T} , let C' be the unique neighbor of C in \mathcal{T} , and let \mathcal{T}' be the tree that remains when C is removed.

Step 1: If $C \subseteq C'$, then \mathcal{T}' is a junction tree for \mathcal{G} , and result follows by induction.

Step 2: If $C \cap C' \subset C$ (in a *strict* sense), then consider the subgraph \mathcal{G}' formed by removing the non-empty set $R := C \setminus C'$ from \mathcal{V} . We claim that it is chordal. First, observe that R has an empty intersection with every clique in

\mathcal{T}' (using junction tree property). (That is, say $R \cap D \neq \emptyset$ for some clique node D in \mathcal{T}' . Then there exists $s \in C \cap D$, but $s \notin C'$, which violates running intersection.) Follows that \mathcal{T}' is a junction tree for \mathcal{G}' , and so \mathcal{G}' is chordal (by applying induction hypothesis).

Step 3: Now claim that \mathcal{G} is chordal. Any cycle entirely contained in \mathcal{G}' is chordless by induction. If the cycle is entirely within the complete subgraph $\mathcal{G}[C]$, it is also chordless. Any other cycle must intersect R , $C \cap C'$ and $\mathcal{V} \setminus C$. In particular, it must cross $C \cap C'$ twice, and since this set is complete, it has a chord.

References

1. S. Aji, R. McEliece, The generalized distributive law. *IEEE Trans. Inf. Theory* **46**, 325–343 (2000)
2. S. Amari, Differential geometry of curved exponential families—curvatures and information loss. *Ann. Stat.* **10**(2), 357–385 (1982)
3. S. Amari, *Differential-Geometrical Methods in Statistics* (Springer, New York, 1985)
4. P. Amestoy, T.A. Davis, I.S. Duff, An approximate minimum degree ordering algorithm. *SIAM J. Matrix Anal. Appl.* **17**, 886–905 (1996)
5. A. Anandkumar, V.Y.F. Tan, F. Huang, A.S. Willsky, High-dimensional structure learning of Ising models: local separation criterion. *Ann. Stat.* **40**(3), 1346–1375 (2012)
6. S. Arnborg, Complexity of finding embeddings in a k -tree. *SIAM J. Algebr. Discrete Math.* **3**(2), 277–284 (1987)
7. O. Banerjee, L.E. Ghaoui, A. d’Aspremont, Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *J. Mach. Learn. Res.* **9**, 485–516 (2008)
8. U. Bertele, F. Brioschi, *Nonserial Dynamic Programming* (Academic, New York, 1972)
9. D. Bertsekas, *Nonlinear Programming* (Athena Scientific, Belmont, MA, 1995)
10. J. Besag, Spatial interaction and the statistical analysis of lattice systems. *J. R. Stat. Soc. Ser. B* **36**, 192–236 (1974)
11. J. Besag, Efficiency of pseudolikelihood estimation for simple Gaussian fields. *Biometrika* **64**(3), 616–618 (1977)
12. J. Besag, On the statistical analysis of dirty pictures. *J. R. Stat. Soc. Ser. B* **48**(3), 259–279 (1986)
13. Y.M. Bishop, S.E. Fienberg, P.W. Holland, *Discrete Multivariate Analysis: Theory and Practice* (MIT Press, Boston, MA, 1975)
14. H.L. Bodlaender, Dynamic programming on graphs with bounded treewidth, in *Automata, Languages and Programming*, vol. 317 (Springer, Berlin, 1988), pp. 105–118
15. H. Bodlaender, A tourist guide through treewidth. *Acta Cybern.* **11**, 1–21 (1993)
16. H.L. Bodlaender, A linear-time algorithm for finding tree decompositions of small treewidth. *SIAM J. Comput.* **25**, 1305–1317 (1996)
17. B. Bollobás, *Graph Theory: An Introductory Course* (Springer, New York, 1979)
18. S. Boyd, L. Vandenberghe, *Convex Optimization* (Cambridge University Press, Cambridge, 2004)
19. L.D. Brown, *Fundamentals of Statistical Exponential Families* (Institute of Mathematical Statistics, Hayward, CA, 1986)
20. A. Cayley, A theorem on trees. *Q. J. Math.* **23**, 376–378 (1889)
21. C.K. Chow, C.N. Liu, Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inf. Theory* **IT-14**, 462–467 (1968)

22. P. Clifford, Markov random fields in statistics, in *Disorder in Physical Systems*, ed. by G. Grimmett, D.J.A. Welsh. Oxford Science Publications (Clarendon Press, Oxford, 1990)
23. T.H. Cormen, C.E. Leiserson, R.L. Rivest, *Introduction to Algorithms* (MIT Press, Cambridge, MA, 1990)
24. I. Csiszar, I-divergence geometry of probability distributions and minimization problems. *Ann. Probab.* **3**(1), 146–158 (1975)
25. I. Csiszar, Sanov property, generalized I-projection and a conditional limit theorem. *Ann. Probab.* **12**(3), 768–793 (1984)
26. I. Csiszar, A geometric interpretation of Darroch and Ratcliff's generalized iterative scaling. *Ann. Stat.* **17**(3), 1409–1413 (1989)
27. J.N. Darroch, D. Ratcliff, Generalized iterative scaling for log-linear models. *Ann. Math. Stat.* **43**, 1470–1480 (1972)
28. A.P. Dawid, Applications of a general propagation algorithm for probabilistic expert systems. *Stat. Comput.* **2**, 25–36 (1992)
29. R.L. Dykstra, An iterative procedure for obtaining I-projections onto the intersection of convex sets. *Ann. Probab.* **13**(3), 975–984 (1985)
30. B. Efron, The geometry of exponential families. *Ann. Stat.* **6**, 362–376 (1978)
31. S. Fienberg, An iterative procedure for estimation in contingency tables. *Ann. Math. Stat.* **41**(3), 907–917 (1970)
32. G.D. Forney Jr., The Viterbi algorithm. *Proc. IEEE* **61**, 268–277 (1973)
33. J. Friedman, T. Hastie, R. Tibshirani, Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **9**, 432–441 (2008)
34. R.G. Gallager, *Low-Density Parity Check Codes* (MIT Press, Cambridge, MA, 1963)
35. S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. PAMI* **6**, 721–741 (1984)
36. A. George, J.W.H. Liu, The evolution of the minimum degree ordering algorithm. *SIAM Rev.* **31**(1), 1–19 (1989)
37. G.R. Grimmett, A theorem about random fields. *Bull. Lond. Math. Soc.* **5**, 81–84 (1973)
38. S.J. Haberman, *The Analysis of Frequency Data* (University of Chicago Press, Chicago, IL, 1974)
39. E. Ising, Beitrag zur theorie der ferromagnetismus. *Zeitschrift für Physik* **31**(1), 253–258 (1925)
40. M. Kalisch, P. Bühlmann, Estimating high-dimensional directed acyclic graphs with the PC algorithm. *J. Mach. Learn. Res.* **8**, 613–636 (2007)
41. R. Kalman, A new approach to linear filtering and prediction problems. *Am. Soc. Mech. Eng.: Basic Eng. Ser. D* **82**, 35–45 (1960)
42. D. Karger, N. Srebro, Learning Markov networks: maximum bounded tree-width graphs, in *Symposium on Discrete Algorithms* (2001), pp. 392–401
43. F. Kschischang, B. Frey, H.A. Loeliger, Factor graphs and the sum-product algorithm. *IEEE Trans. Inf. Theory* **47**(2), 498–519 (2001)
44. S.L. Lauritzen, *Graphical Models* (Oxford University Press, Oxford, 1996)
45. S.L. Lauritzen, D.J. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *J. R. Stat. Soc. B* **50**, 155–224 (1988)
46. H.A. Loeliger, An introduction to factor graphs. *IEEE Signal Process. Mag.* **21**, 28–41 (2004)
47. N. Meinshausen, A note on the Lasso for graphical Gaussian model selection. *Stat. Probab. Lett.* **78**(7), 880–884 (2008)
48. N. Meinshausen, P. Bühlmann, High-dimensional graphs and variable selection with the Lasso. *Ann. Stat.* **34**, 1436–1462 (2006)
49. S.V. Parter, The use of linear graphs in Gaussian elimination. *SIAM Rev.* **3**, 119–130 (1961)
50. J. Pearl, *Probabilistic Reasoning in Intelligent Systems* (Morgan Kaufman, San Mateo, 1988)
51. L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **77**(2), 257–285 (1989)

52. P. Ravikumar, M.J. Wainwright, J. Lafferty, High-dimensional Ising model selection using ℓ_1 -regularized logistic regression. *Ann. Stat.* **38**(3), 1287–1319 (2010)
53. P. Ravikumar, M.J. Wainwright, G. Raskutti, B. Yu, High-dimensional covariance estimation by minimizing ℓ_1 -penalized log-determinant divergence. *Electron. J. Stat.* **5**, 935–980 (2011)
54. D.J. Rose, Triangulated graphs and the elimination process. *J. Math. Anal. Appl.* **32**, 597–609 (1970)
55. G.R. Shafer, P.P. Shenoy, Probability propagation. *Ann. Math. Artif. Intell.* **2**, 327–352 (1990)
56. P. Spirtes, C. Glymour, R. Scheines, *Causation, Prediction and Search* (MIT Press, Cambridge, 2000)
57. N. Srebro, Maximum likelihood Markov networks: an algorithmic approach. Master's thesis, MIT, 2000
58. F.F. Stephan, Iterative method of adjusting sample frequency tables when expected margins are known. *Ann. Math. Stat.* **13**, 166–178 (1942)
59. R.M. Tanner, A recursive approach to low complexity codes. *IEEE Trans. Inf. Theory* **IT-27**, 533–547 (1980)
60. R.E. Tarjan, M. Yannakakis, Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM J. Comput.* **13**(3), 566–579 (1984)
61. J.H. van Lint, R.M. Wilson, *A Course in Combinatorics* (Cambridge University Press, Cambridge, 1992)
62. S. Verdú, H.V. Poor, Abstract dynamic programming models under commutativity conditions. *SIAM J. Control Optim.* **25**(4), 990–1006 (1987)
63. A. Viterbi, Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Inf. Theory* **IT-13**, 260–269 (1967)
64. M.J. Wainwright, M.I. Jordan, Graphical models, exponential families and variational inference. *Found. Trends Mach. Learn.* **1**(1–2), 1–305 (2008)
65. M.J. Wainwright, T.S. Jaakkola, A.S. Willsky, Tree-based reparameterization framework for analysis of sum-product and related algorithms. *IEEE Trans. Inf. Theory* **49**(5), 1120–1146 (2003)
66. M.J. Wainwright, T.S. Jaakkola, A.S. Willsky, Tree consistency and bounds on the max-product algorithm and its generalizations. *Stat. Comput.* **14**, 143–166 (2004)
67. M. Yannakakis, Computing the minimum fill-in is NP-complete. *SIAM J. Algebr. Discrete Methods* **2**(1), 77–79 (1981)

Bridging the Gap Between Information Theory and Wireless Networking

P.R. Kumar

1 Introduction

In 1948 Shannon [1] addressed the fundamental problem of reliable communication over an unreliable channel. Shannon made three major contributions. First he provided the right formulation of the problem. Second, he provided the tools to study this problem. Third, he provided the answer.

Shannon also addressed the problem of communication between multiple senders and receivers, called multi-user information theory or more recently network information theory. However, in this case the answers were not as definitive. Over the ensuing five decades there has been progress on the subject of multi-user information theory for certain problems, but, by and large, definitive characterizations have continued to be elusive.

Meanwhile, about 40 years ago, the wireline Internet was born, and has now essentially become pervasive. Alongside, the pioneering efforts in ALOHA [2] led to the wireless Ethernet [3] and subsequently to the wireless WiFi technology [4]. Wireless networks interconnecting many sources and destinations have also attracted much attention. There have been implementations of ad hoc networks, some deployments in disaster scenarios, several deployments of sensor networks, and increasing interest in vehicular networks.

Thus, on one hand while theoretical efforts at developing a network information theory have been stymied, on the other hand technological interest and excitement has been increasing, with wireless networks possibly on the cusp of a takeoff. There was therefore a huge gap between information theory and networking practice [5].

P.R. Kumar (✉)

CSL and Department of ECE, University of Illinois, 1308 West Main St., Urbana,
IL 61801-2307, USA

e-mail: prkumar@illinois.edu

© Springer International Publishing Switzerland 2015

F. Fagnani et al. (eds.), *Mathematical Foundations of Complex Networked Information Systems*, Lecture Notes in Mathematics 2141,

DOI 10.1007/978-3-319-16967-5_4

This raised the question of what sort of theory can begin to address the kinds of networks that are being built and that may soon proliferate. This problem was addressed in [6], which studied a prevalent technological model of wireless communication. It attempted to shed light on two questions: how much information could be communicated over wireless networks with many sources and destinations, and secondly what sort of strategies were appropriate. A preliminary issue that needed to be confronted was how to formulate this problem in such a way that some key relevant issues could be captured. Along the way, it was necessary to address how one should even measure the information carrying capacity of a network with many sources and destinations. This work showed that simple models of current technology, more precisely the shared medium aspect of wireless, could be used to demonstrate that there were limitations to how much data could be carried. This work also addressed the issue of scaling laws as the number of nodes in the network increases.

The work in [6] however only studied the limits and potentials of a particular mode of utilizing the wireless medium, but left unaddressed and unanswered what are the fundamental limitations on the potential of the wireless medium. That is, it left unaddressed what are the limitations of wireless networks that are independent of technology, but that depend only on the properties of the *medium*. This problem was addressed in [7] which studied it in an information-theoretically rigorous manner. It showed fundamental connections between the attenuation properties of the wireless medium and the capability of the network to transport data irrespective of the technology used. In particular it showed that one sort of scaling law holds in the heavy attenuation regime, while other scaling regimes can occur in the low attenuation regime.

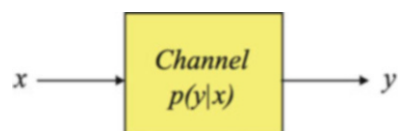
In this chapter we give a brief account of these results which have been obtained over the past decade, since 2000.

2 Shannon's Point to Point Results

It is useful to begin with an understanding of the fundamental results of Shannon since they provide the right background and context to understand the tradeoffs in the communication problem.

We begin by modeling a *noisy memoryless channel* by a conditional probability distribution $p(y|x)$, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. The transmitter chooses an element x from a set \mathcal{X} , called the alphabet, and transmits it. The receiver receives y with probability $p(y|x)$; see Fig. 1.

Fig. 1 Noisy memoryless channel



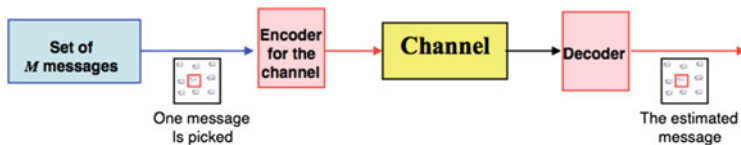


Fig. 2 Shannon’s model of communication problem

This is a noisy channel since there is not a deterministic one-to-one mapping from x to y . It is memoryless in that this conditional probability captures the behavior of the channel every time it is used. So if on T successive instants (we adopt a discrete-time model), the transmitter transmits the string $(x(1), x(2), \dots, x(T))$, then the receiver receives a string $(y(1), y(2), \dots, y(T))$, with probability $\prod_{t=1}^T p(y(t)|x(t))$. That is, there is no intersymbol interference in that $y(t)$ is not influenced by $(x(1), x(2), \dots, x(t - 1))$ or $(x(t + 1), x(t + 2), \dots, x(T))$.

The question that Shannon studied is: How much information can be *reliably* transmitted over such a noisy channel?

Let us suppose that there is a bag of M messages; see Fig. 2.

The transmitter wants to communicate one message from this set to the receiver. The meaning of the messages is not important, but only the index of the message is. That is, it suffices for the receiver to know that it was the m -th message from this set that was communicated. Thus the *semantics* of the message is determined by its index or label. For this to be an acceptable model of the communication problem, it is of course necessary for both the transmitter and receiver to agree beforehand on the set of all messages as well as how they are labeled. (In fact, in the (not so) old days, there were some canned messages such “Arrived safely,” “Best wishes for your wedding,” “Happy Birthday,” that were assigned numbers such as 1, 2, 3, respectively, and it was only the number that was transmitted over the telegraph wire. At the receiving end, when “2” was received the message “Best wishes for your wedding” was printed out and delivered to the recipient.)

Let us suppose that the transmitter uses the channel T times to send a message from this bag of M messages. Given a message m that is intended to be sent, the transmitter transmits a string $(x_m(1), x_m(2), \dots, x_m(T))$ where each $x_m(t) \in \mathcal{X}$. For this purpose, there is a mapping from the set $\{1, 2, \dots, M - 1, M\}$ of messages to the set of \mathcal{X}^T of transmittable strings of length T . This mapping is called an *encoder*, and we denote it by $E : \{1, 2, \dots, M\} \rightarrow \mathcal{X}^T$.

The receiver receives a string $(y(1), y(2), \dots, y(T))$, and, based on that, estimates which message was sent. For this purpose, the receiver has a mapping from the set of received strings of length T , \mathcal{Y}^T , to the set of messages $\{1, 2, \dots, M - 1, M\}$. We call this map $D : \mathcal{Y}^T \rightarrow \{1, 2, \dots, M\}$ the *decoder*.

What the receiver estimates and what the transmitter sends need not be the same. When they differ we say that there is an *error*. In the above context this happens when $D((y(1), y(2), \dots, y(T))) \neq m$. Let $p_m(\text{error})$ be the probability of

error when the string corresponding to message m is sent. Also, let $p(\text{error}) := \text{Max}_{1 \leq m \leq M} p_m(\text{error})$ denote the maximum probability of error over all messages.

Clearly, in general it will be impossible make $p(\text{error})$ equal to zero. So how then is reliable communication to be achieved?

Shannon's brilliant idea was to study whether the probability of error could be made asymptotically zero as $M \rightarrow +\infty$. Of course, when M is increased to infinity, so will T need to be increased too. At what rate should T be increased?

To understand this, it is necessary to recognize that in order to represent the set of M messages, one needs $\log_2 M$ bits. That is, every message can be represented by a unique string of $\log_2 M$ bits. (We ignore the fact that $\log_2 M$ may not be an integer.) Suppose that we use $T = R \log_2 M$ transmissions to transmit the index of the message. Let us suppose this is how T increases with M .

Let us now fix R , and suppose that for every M there is an encoder $E : \{1, 2, \dots, M\} \rightarrow \mathcal{X}^{R \log_2 M}$, and decoder $D : \mathcal{Y}^{R \log_2 M} \rightarrow \{1, 2, \dots, M\}$, such that the probability of error goes to zero as $M \rightarrow +\infty$. Then we can say that the transmitter and receiver can *reliably communicate at rate R* .

The reason for calling R as the "rate" is just because it effectively takes $R \log_2 M$ transmissions to transmits $\log_2 M$ bits. The reason for saying this can be done "reliably" is only because the probability of error can be made small by increasing M .

The *capacity of the channel* is simply defined as the supremum of the set of rates at which the transmitter and receiver can reliably communicate:

$$C := \sup\{R : \text{The transmitter and receiver can reliably communicate at rate } R\}.$$

Shannon also determined the capacity of the discrete memoryless channel modeled by $p(y|x)$. To state his answer, it is necessary to define the notions of entropy and mutual information.

Definition 1 The *entropy* of the discrete-valued random variable X with probability distribution $p(x)$ for $x \in \mathcal{X}$, is

$$H(X) := - \sum_{x \in \mathcal{X}} p(x) \log_2 p(x) = E[-\log p(X)].$$

It is measured in "bits."

The *conditional entropy* of a random variable X given a random variable Y is

$$H(X|Y) := - \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x, y) \log_2 p(x|y) = E[-\log p(X|Y)].$$

(continued)

Definition 1 (continued)

The *mutual information* between two jointly distributed random variables (X, Y) is the difference between the (unconditional) entropy of X and the conditional entropy of X given Y :

$$I(X; Y) := H(X) - H(X|Y).$$

Consider now three jointly distributed random variables (X, Y, Z) . The *conditional mutual information* between X and Y given the random variable Z is

$$I(X; Y|Z) = H(X|Z) - H(X|Y, Z).$$

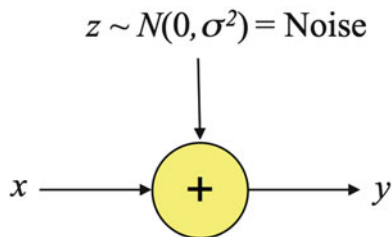
Theorem 1 *The capacity of the discrete memoryless channel is*

$$C = \sup_{\text{Probability distributions on } X} I(X; Y) \text{ bits/transmission.}$$

To summarize, Shannon’s contribution consists of precisely formulating the problem, defining the notion of capacity, and then determining what it is.

A very important channel in practice is the *additive white Gaussian noise channel*; see Fig. 3. Here a scalar x is transmitted, and it is received after corruption by additive Gaussian noise. So $y = x + e$, where e is $N(0, \sigma^2)$. The transmitter is restricted to power level P . Here x and y are scalars, so that this is not a discrete channel. Moreover there is a power constraint on the transmitter. Nevertheless the above theorem can be generalized:

Fig. 3 The additive white Gaussian noise channel



Theorem 2 *The capacity of the AWGN channel with power constraint P and noise variance σ^2 is*

$$C = \frac{1}{2} \log\left(1 + \frac{P}{\sigma^2}\right) \text{ bits/transmission.}$$

This result shows that the capacity grows logarithmically in the signal-to-noise ratio (SNR) $\frac{P}{\sigma^2}$.

3 The Multiple-Access and Gaussian Broadcast Channels

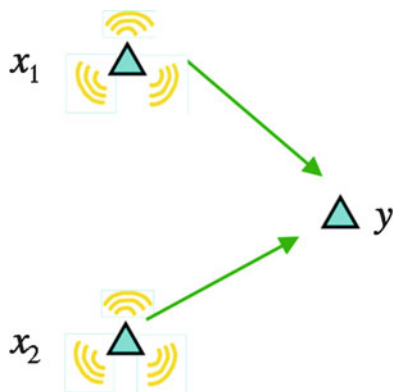
Since the work of Shannon there has been much attention devoted to the problem of determining the capacity of *networks* which consist of more than two nodes.

Success in characterizing the capacity has only been achieved in a few rare cases. Two notable examples are the *multiple access channel* and the *Gaussian broadcast channel*.

The multiple access channel consists of several users, let us say two users for simplicity, communicating simultaneously to a common receiver; see Fig. 4.

The channel is a *shared medium*. When users 1 and 2 transmit $x_1 \in \mathcal{X}_1$ and $x_2 \in \mathcal{X}_2$ at the same time, the receiver receives $y \in \mathcal{Y}$ with probability $p(y|x_1, x_2)$. Suppose each user i wants to communicate at a rate R_i . What *rate vectors* $R = (R_1, R_2)$ are (reliably, as above) feasible?

Fig. 4 The multiple access channel



This has been answered by Ahlswede [8] and Liao [9]:

Theorem 3 *The closure of the set of achievable rate vectors is the closed convex hull of the set of the set of rate vectors that satisfy*

$$R_1 \leq I(X_1, Y|X_2),$$

$$R_2 \leq I(X_2, Y|X_1),$$

and

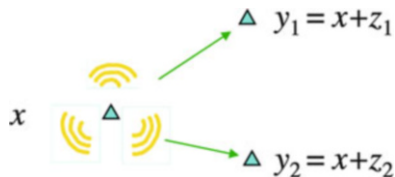
$$R_1 + R_2 \leq I((X_1, X_2); Y).$$

over all joint distributions where X_1 and X_2 are independent random variables.

The opposite of the multiple-access channel is the *broadcast channel*. In this case there is a single transmitter, and, say, two receivers. When the transmitter transmits $x \in \mathcal{X}$, the two receivers receive $y_1 \in \mathcal{Y}_1$ and $y_2 \in \mathcal{Y}_2$ with probability $p(y_1, y_2|x)$. Let R_1 and R_2 denote the rates at which the transmitter wants to convey independent information to the two receivers. What rate vectors $R = (R_1, R_2)$ are reliably feasible?

This problem has not been solved for general memoryless channels. One special case that has been solved is the Gaussian broadcast channel; see Fig. 5. Suppose that x is a scalar real number, and that the two receivers obtain versions of x that are corrupted by additive Gaussian noise. (The case of vector x has been solved in [10].) That is $y_i = x + z_i$ where z_i is a $N(0, \sigma_i^2)$ random variable. Here the random variables are not discrete random variables, but the theory can be extended to this case.

Fig. 5 The scalar Gaussian broadcast channel



Theorem 4 Suppose $\sigma_2 \geq \sigma_1$. Then the closure of the set of achievable rate vectors is the closed convex hull of the set of rate vectors that satisfy

$$R_1 \leq \frac{1}{2} \log\left(1 + \frac{\lambda P}{\sigma_1^2}\right),$$

$$R_2 \leq \frac{1}{2} \log\left(1 + \frac{(1 - \lambda)P}{\lambda P + \sigma_2^2}\right),$$

for some

$$0 \leq \lambda \leq 1.$$

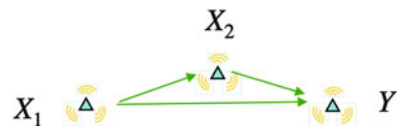
The interpretation of this expression is that the transmitter dedicates a fraction λP of its power P to receiver 1, and the remaining fraction $(1 - \lambda)P$ to receiver 2. Receiver 2 treats the signal intended for receiver 1 as noise, and decodes its own signal. Receiver 1 first decodes the signal meant for receiver two. It can do this because its noise is less than receiver 2's. After this it subtracts the portion of the receiver signal that was meant for receiver 2, and decodes its own signal.

Both these situations correspond to bottleneck scenarios. In the case of the multiple-access channel the receiver is the bottleneck, while in the case of the broadcast channel the transmitter is the bottleneck.

A very important networking scenario of great interest is the *relay channel*; see Fig. 6. Here there is a source x_1 that wants to send information to a destination y . To help the source, there is a relay x_2 . The relay receives a noisy version of x_1 . After a delay of one time unit it transmits x_2 . The destination receives a noisy combination of what source and relay transmit. This is possibly the simplest generalization of the point-to-point channel that was solved by Shannon. Nevertheless, after several decades of research, the capacity of the relay channel is still unknown in general; see [11].

In general, the capacity of most networks, even simple networks, is not precisely known. This raises the question of how to make progress in this area, more so how can we even begin to understand more complex scenarios?

Fig. 6 The relay channel



4 A Spatial Model of a Wireless Network

Let us eschew information theory for a while.

Let us formulate a network with a physical character by supposing that the nodes of the network are located on a two-dimensional plane. Let us suppose that nodes directly choose a *range* for their transmissions. This can be indirectly done by a node choosing a power level for its transmission.

A key aspect of a wireless network is that it is a shared medium. Let us simply suppose that a receiver cannot recover a message when it is subjected to *strong interference*.

We can capture the above considerations in a very simple model of communication, as follows. Let us suppose that there are n nodes located inside a disk of area $A \subset \mathbb{R}^2$; see Fig. 7. The spatial nature of the network can be used to model when the interference from a nearby transmission is strong and when it is not. Let us suppose that when a transmitter wishes to communicate with a receiver at a distance r from it, then it chooses a power level that gives it a range r . This transmission however also causes interference to other nodes. We will suppose that the *interference region* is a disk of radius $(1 + \Delta)r$ around the transmitter; see Fig. 8. Within this region, we will suppose that no other receiver can receive any other transmission successfully.

To illustrate this, consider the case of two concurrent transmissions, as shown in Fig. 9. Transmitter T_1 transmits to its intended receiver at a range of r_1 . Concurrently, transmitter T_2 transmits to its intended receiver R_2 at a range of r_2 . The interference region created by the transmission of T_1 is a disk of radius $(1 + \Delta)r_1$ centered at T_1 , while the interference region created by the transmission of T_2 is a disk of radius $(1 + \Delta)r_2$ centered at T_2 . Then the transmission from T_1 to R_1 is successful only if R_1 is outside the interference region created by T_2 . Similarly, the transmission from T_2 is successful only if R_2 is outside the interference region created by T_1 .

To summarize, a reception from a transmitter to its intended receiver is successful if and only if the intender receiver is not in the interference region of any other transmission.

Fig. 7 A geographic model: n nodes in a disk of area A

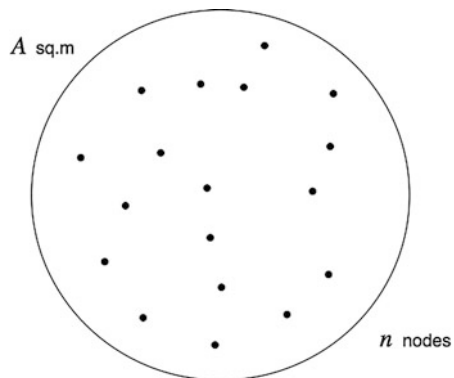


Fig. 8 The interference region caused by a transmission of range r

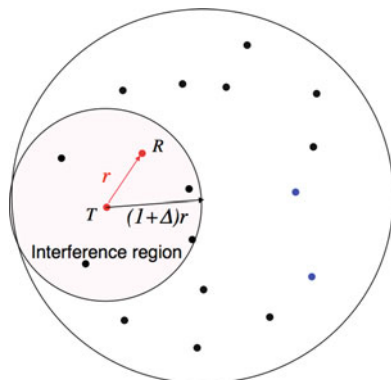
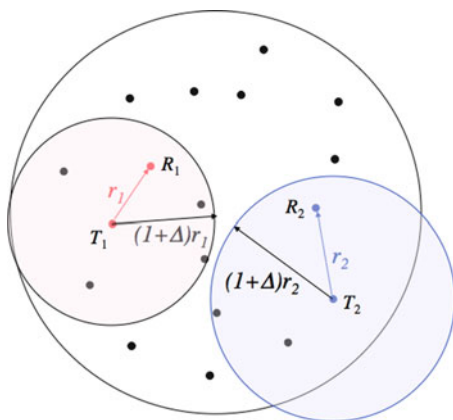


Fig. 9 Two concurrent transmissions



Let us suppose that when any transmission is successful it transfers data at a rate of W bits per second from the transmitter to its intended receiver.

We note this is model does not capture more sophisticated notions such as signal-to-noise-plus-interference ratio. However, as we will see, the results obtained from this simple model are remarkably robust with respect to the accuracy and details of the model.

5 Multi-Hop Transport

Suppose that data is to be transferred from *source* S to its *destination* D , at a rate of λ_i bits/second. We will allow for *multi-hop transport*. That is, a path $(S = A_1, A_2, A_3, \dots, A_m = D)$ is chosen, which connects the source node S to its destination node D . Then packets are transferred by node A_j broadcasting to node A_{j+1} , for each $j = 1, 2, \dots, m - 1$. Each such transmission from node A_j to node A_{j+1} is successful under the conditions outlined in Sect. 4. That is, each node A_{j+1}

has to be outside the interference regions of all other concurrent transmissions when it is receiving a packet from node A_j .

6 The Transport Capacity

In a network there are usually many source-destination pairs. One question that arises is: How should we measure the capacity of the network? One answer, which we have already encountered in Sect. 3, is to simply describe the entire capacity region, which is defined as the (closure of) the set of all rate vectors that are (reliably) feasible, where each source-destination's rate is one element of this rate vector.

There are several difficulties in pursuing this very ambitious goal. A primary difficulty is that, in fact, even when n is small the capacity region has stymied efforts at precise characterization. An example is the relay channel which has only three nodes, for which one still does not know the supremal rate at which data can be transferred from one of the three nodes to another. Moreover, the problem can be even worse. If there are n nodes in the network, then there are $n(n - 1)$ source-destination pairs. The rate vector is then potentially of dimension $n(n - 1)$, which is a huge number when n is large.

So we pursue a dramatic simplification. Can we describe the capacity roughly by just a single number? Is there a gross measure of the data pumping rate of a network? For this purpose we use the performance measure of *bit-meters/second*. That is, we multiply the throughput between a source and a destination by the distance between the source and destination, and sum this over all source-destination pairs.

Specifically, suppose the set of source-destination pairs is $\{(S_1, D_1), (S_2, D_2), \dots, (S_k, D_k)\}$. Suppose that data is to be transferred from source S_i to its destination D_i , at a rate of λ_i bits/second. Let ρ_i be the distance between nodes S_i and D_i . Then the above calculation yields $\sum_{i=1}^k \lambda_i \rho_i$ bit-meters/second. We note that this is reminiscent of how airlines, for example, measure their size in terms of man-miles flown per year. Our measure is the information-theoretic analog of such a performance measure.

We will call the supremal bit-meters/second that can be carried by a network as its *transport capacity*.

It should be noted that it is a single aggregate number which can capture what we are after—a gross measure of what a network can achieve.

We will occasionally abuse notation and call the performance measure itself as the transport capacity; referring, for example, to a particular policy as realizing so much transport capacity.

7 Best Case Transport Capacity and Scaling Laws

In the definition of transport capacity, we are required to take the supremum of the bit meters/second that can be carried over the entire network. This can be a formidable task since it involves optimization over (1) all spatio-temporal transmission strategies, since these strategies determine successful passing of information from one node another, (2) all multi-path routing strategies, since more than one path can be used to transfer information from a source node to its destination node, (3) all choices of sets of source-destination pairs, and (4) all feasible rate vectors for the set of source-destination pairs.

However, we will see that we can simplify this by adding another layer of optimization: over the locations of the nodes. That is, rather than determining the transport capacity of a given network with n nodes located in given positions, we will consider the supremum transport capacity that can be achieved over all networks of n nodes. We will call this the *best case transport capacity*.

We will see that one can obtain an upper bound on the best case transport capacity that is a function of the number n of nodes.

Actually, we take a further step. We will study the behavior of the best transport capacity as a function of n . We call this a *scaling law*.

8 An Upper Bound on Transport Capacity

It turns out that one can obtain an upper bound on the best case transport capacity rather easily. From Fig. 9 it follows that

$$\begin{aligned} |R_1 R_2| &\geq |T_1 R_2| - |T_1 R_1| \text{ (from the triangle inequality)} \\ &\geq (1 + \Delta)r_1 - r_1 \\ &= \Delta r_1. \end{aligned}$$

Similarly, interchanging the roles of the links $T_1 R_1$ and $T_2 R_2$, it also follows that

$$\begin{aligned} |R_1 R_2| &\geq |T_2 R_1| - |T_2 R_2| \text{ (from the triangle inequality)} \\ &\geq (1 + \Delta)r_2 - r_2 \\ &= \Delta r_2. \end{aligned}$$

Hence, in particular,

$$|R_1 R_2| \geq \frac{1}{2}(\Delta r_1 + \Delta r_2).$$

This inequality is illustrated in Fig. 10.

Fig. 10 Two disks of radius $\frac{1}{2}\Delta r_1$ around R_1 and radius $\frac{1}{2}\Delta r_2$ around R_2 are disjoint

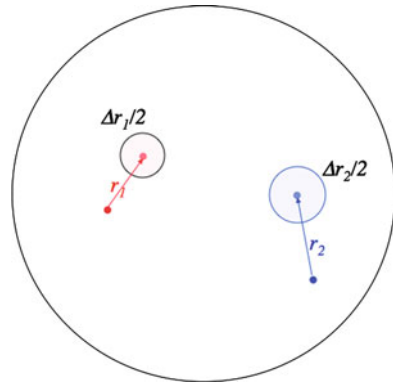
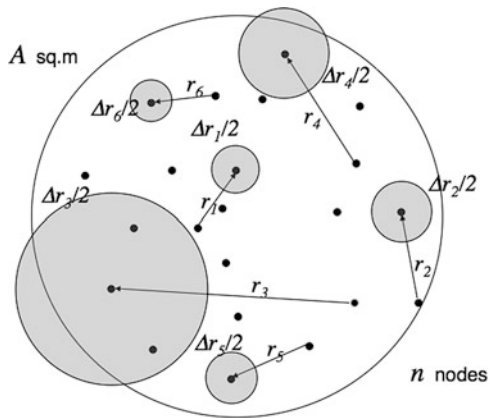


Fig. 11 Disks of radius $\frac{1}{2}\Delta$ *Range of transmission around receivers are disjoint



Since the above holds for every pair of concurrent transmissions, it follows that around each receiver there is a disk of radius $\frac{1}{2}(\Delta \cdot \text{Range of transmission})$ that is disjoint of such disks around all other concurrent receivers of successful transmissions. This is illustrated in Fig. 11. We can regard this statement as saying “every transmission consumes area proportional to the square of the range of the transmission.”

Now, we see in Fig. 11, a portion of the disk may lie outside the domain of area A square meters. However, each of these disks must have at least one-quarter of its area within the domain of area A square meters. Hence we see that

$$\sum_{i=1}^{n/2} \frac{\pi \Delta^2 r_i^2}{16} \leq A. \tag{1}$$

In the above summation we allowed for a maximum of $n/2$ concurrent transmissions, because every transmission must have a corresponding receiver.

Now we note that the function $f(x) := x^2$ is a convex function. For convex functions we know that if $\sum_{i=1}^m \lambda_i x_i$ is a convex combination of $\{x_1, x_2, \dots, x_m\}$, with $\sum_{i=1}^m \lambda_i = 1$, $\lambda_i \geq 0$, then

$$f\left(\sum_{i=1}^m \lambda_i x_i\right) \leq \sum_{i=1}^m \lambda_i f(x_i).$$

Applied to our context, using $m = \frac{n}{2}$, $\lambda_i \equiv \frac{2}{n}$, and $x_i = r_i$, this results in

$$f\left(\frac{2}{n} \sum_{i=1}^{n/2} r_i\right) \leq \sum_{i=1}^{n/2} \frac{2}{n} f(r_i),$$

i.e.,

$$\left(\frac{2}{n} \sum_{i=1}^{n/2} r_i\right)^2 \leq \sum_{i=1}^{n/2} \frac{2}{n} r_i^2.$$

Substituting $\sum_{i=1}^{n/2} r_i^2 \leq \frac{16A}{\pi\Delta^2}$ from (1), we obtain

$$\left(\frac{2}{n} \sum_{i=1}^{n/2} r_i\right)^2 \leq \frac{32A}{\pi\Delta^2 n},$$

i.e.,

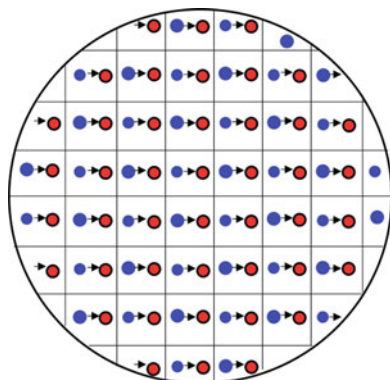
$$\left(\sum_{i=1}^{n/2} r_i\right)^2 \leq \frac{8An}{\pi\Delta^2}.$$

Taking square roots, and multiplying by the data rate W of a successful transmission gives,

$$\sum_{i=1}^{n/2} W r_i \leq \sqrt{\frac{8}{\pi\Delta^2}} W \sqrt{An}. \quad (2)$$

However, the left hand side above is just the number of bit-meters/second provided by all the concurrent transmissions. Since the right hand side is an upper bound on any set of concurrent transmissions, we obtain the following result.

Fig. 12 A layout that achieves $\Omega(\sqrt{n})$ bit-meters/second



Theorem 5 *The transport capacity of the wireless network is bounded as follows:*

$$\text{Transport capacity} \leq cW \sqrt{An}, \text{ for some constant } c. \tag{3}$$

Theorem 5 provides an upper bound on the best case. But is it sharp? In fact the above square-root law is sharp. Consider the layout shown in Fig. 12. There are $n/2$ cells. Each cell is a square that has an area about $2A/n$ square-meters. The width of the side of each cell is about $\sqrt{2A/n}$ meters. Within each such cell there is a transmitter and a receiver for its transmission, which are separated by about $\frac{c}{\sqrt{n}}$ meters. Suppose that the value of Δ is such that all these pairs of nodes can concurrently transmit to their intended receivers at W bits/second. Thus we have $n/2$ nodes, each transmitting at W bits/second to a neighbor that is $\frac{c}{\sqrt{n}}$ meters away. Hence the network as a whole achieves a transport capacity of $c' \sqrt{n}$ bit-meters/second. This shows that the square-root law is sharp for the best case of a wireless network, under this model of interference.

9 Implication of Square-Root Law for Transport Capacity

Let us examine the result of Theorem 5 a bit more closely. First we see that the bound on transport capacity scales linearly with W . This is to be expected because W is the transmission rate, and so scaling it by a constant should scale the transport capacity by the same constant. Next we see that the bound on transport capacity scales as the square-root of the area. This also is to be expected since the right hand side is measured in meters while area is measured in square-meters.

Therefore the only possibly non-obvious feature of the bound is that it scales like the *square-root* of the number of nodes, *not* linearly in the number of nodes.

This square-root scaling law has the following implication. The transport capacity measures the aggregate pumping capacity of the entire network. If we now apportion it equally among all the nodes we see that each node can obtain no more than

$$\text{Per-node transport capacity} \leq cW \sqrt{\frac{A}{n}} \text{ bit-meters/second.} \quad (4)$$

Hence as the number of nodes increases (in a domain of fixed area A), the per-node share decreases like the square-root of the number of nodes.

Above, we have held the area of the domain fixed, while increasing the number of nodes located in it. This however will result in the distance between nodes shrinking, i.e., a *dense* network.

Instead, if we hold constant the average distance between neighboring nodes, at, say 1 m, then the area has to grow linearly in the number of nodes, i.e.,

$$A = c'n.$$

This is called an *expanding* network. Substituting this value for A into (2) gives

$$\text{Per-node transport capacity} \leq c''W \text{ bit-meters/second.} \quad (5)$$

Of course, this corresponds to a total transport capacity over all nodes that grows linearly with n :

$$\text{Transport capacity} \leq c''Wn \text{ bit-meters/second.} \quad (6)$$

The central content of the above result (4) can be interpreted as follows. The bit-meters/second that each node gets is a constant, no more than $c''W$ bit-meters/second. Now suppose that each node only originates traffic for one of its *nearest* neighbors. The distance to such a nearest neighbor is about 1 m. Dividing the $c''W$ bit-meters/second by 1 m, we get $c''W$ bits/second. Hence each node can send only a constant bits/second, in particular no more than $c''W$ bits/second, to its nearest neighbor.

However, suppose that in the above expanding network each node has traffic destined for a remote node at the other end of the network. Such a node is about \sqrt{n} meters away since the area of the domain is n square meters. In this case, dividing $c''W$ bit-meters/second by \sqrt{n} meters, we get $\frac{c''W}{\sqrt{n}}$ bits/second. Hence each node can only send no more than $O(\frac{1}{\sqrt{n}})$ bits/second to such a far away node. In this case the node consumes *more* meters for its traffic, and so obtains *less* bits/second. The point to note is that since it is bit-meters/second that is fixed, a node can either send

traffic over a short distance and get more bits/second, or it can send traffic over a longer distance and get lesser bits/second.

One way to understand the above result is as follows. Consider a node that is \sqrt{n} meters away. Since the distance between nearest neighbors is about 1 meter, a node can send its traffic to its destination by relaying it over multiple hops, actually about \sqrt{n} hops. But in this case the node is consuming some data rate, $O(\frac{1}{\sqrt{n}})$ bits/second, from *each* of \sqrt{n} nearest neighbor transmissions.

Essentially, we see that under this interference model, wireless networks can only support nearest neighbor traffic if the rate is to not diminish with the total number of nodes in the network.

10 The Need for an Information-Theoretic Analysis

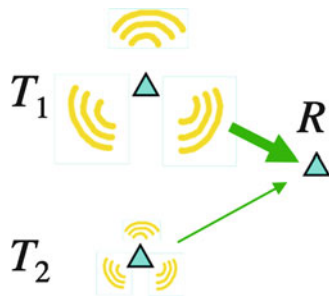
It should be remembered that the above results are all contingent on the interference model, the basic premise of which is that concurrent transmissions cause interference at a receiver. However, this may not be true.

Consider for example the situation shown in Fig. 13. There are two transmitters T_1 and T_2 transmitting concurrently to a common receiver R . Transmitter T_1 is using a large power level, while transmitter T_2 is using a low power level. Receiver R can decode transmitter T_1 's transmission since the interfering transmission from T_2 is of low power. However, after first decoding T_1 , the receiver can then subtract the component of T_1 's signal from its received signal, if it knows the channel gain accurately. Once it does that, the message from T_2 's transmission can also be decoded! This procedure is called “successive interference cancellation.”

In fact, interestingly, the greater the power level of T_1 's transmission, the easier it is to decode T_1 and subtract it (provided of course that the channel gain from T_1 to R is known very accurately). Therefore T_1 does not destructively interfere with T_2 's transmission even though it is considerably more powerful than T_2 's transmission. This shows that interference need *not* be destructive.

In fact this suggests that the wireless medium can be more creatively utilized than the wired medium. Two important characteristics of the wireless medium are

Fig. 13 A strong transmission need not result in destructive interference to a weak transmission



(1) transmissions are broadcasts, and (2) receptions are superpositions of several attenuated transmissions. Both these features are characteristic of a *shared medium*. Above we have seen the technique of successive interference cancellation. In fact there are several ways of using the wireless medium and therefore several ways of operating a wireless network. For example, instead of decoding a message at each hop, and then re-encoding it and re-transmitting it, a node can simply amplify whatever it received and rebroadcast it. This technique is called *amplify-and-forward*. So one could perform relaying by amplify-and-forward rather than *decode-and-forward*. There are other even more novel possibilities. A node could *actively* cancel interference to help a remote node. This is akin to the strategy used in noise canceling headsets, where a signal that is effectively the opposite of noise is transmitted so that it actively cancels the effect of noise at the receiver.

All these possibilities suggest that the technological model used in Sect. 4 needs to re-examined to see if the results that are obtained based on using technology in that way are unnecessarily restrictive. That model presumed that information is conveyed by relaying packets from one node to another. At each such hop, the information is completely decoded, and then re-encoded and transmitted, i.e., it is a decode-and-forward strategy. Not only that, the model of interference that was used assumes that interference is akin to noise. That is, there is no valuable information content in interference. As a consequence it presumes that interference is destructive. This set of choices is arbitrary and is only one set of choices in a much larger infinite-dimensional space of strategies that one can conceive of for operating wireless networks.

Thus we somehow need to determine what are the fundamental limitations to information transfer in wireless networks that are not merely self-imposed limitations. To answer this we need to consider wireless networks from an information-theoretic perspective.

11 Wireless Network Information Theory

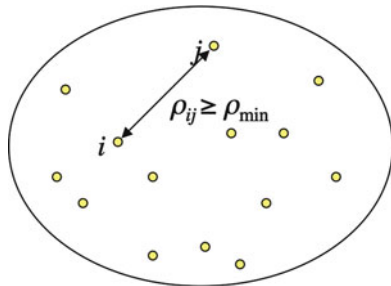
We need to model the fact that radio transmissions are *broadcast* in nature, and also that what a node receives is a *superposition* of several suitably *attenuated* transmissions.

We will adopt a more geographic model to describe the physical layout of nodes and the attenuation of signals. In fact the notion of “distance” will play a fundamental role not only in modeling the system, but also in the performance measure we continue to adopt—transport capacity.

Consider n nodes on a two-dimensional plane, as shown in Fig. 14. The distance between nodes i and j is $\rho_{ij} \geq \rho_{\min} > 0$.

The attenuation that takes place over a distance ρ is $\frac{\exp(-\gamma\rho)}{\rho_{ij}^\delta}$. The numerator represents *absorption* by the medium. It is inevitably present except in vacuum; see

Fig. 14 n nodes in a two-dimensional plane with separation distance greater than ρ_{\min}



[12], resulting in a value of $\gamma > 0$. The absorption leads to exponential decrease in the strength of the received signal. The constant γ is called the *absorption constant*.

Now let us turn to the denominator. When the constant α in the denominator is 1, then the denominator represents the familiar *inverse-square law*. The reason is that power is proportional to the *square* of the amplitude of the signal, so if the power decreases by a factor $\frac{1}{\rho^2}$, it corresponds to the amplitude decreasing by a factor $\frac{1}{\rho}$. The constant $\alpha \geq 1$ is called the *path-loss exponent*. A value greater than one results, for example, from a two-ray propagation model that captures the combination of a direct path as well as a ground reflection path; see [13].

One other physical restriction that we need to impose is some form of transmit power constraint, for we see from Theorem 2 that when power is unrestricted the rate is unbounded. Let us entertain two possibilities:

1. Each node is given an individual power allocation P_{ind} . So if P_i is the power used by node i , then

$$P_i \leq P_{ind} \text{ for all } 1 \leq i \leq n. \tag{7}$$

We will call this the *individual power constraint*.

2. There is total power allocation P_{total} that is to be shared by all the nodes:

$$\sum_{i=1}^n P_i \leq P_{total}. \tag{8}$$

We will call this the *total power constraint*.

We will consider one or the other of these two power constraints.

Consider a discrete-time operation of the network. Suppose that at time t , node i broadcasts $x_i(t)$. Then receiver j receives a superposition $y_j(t)$ of all the broadcasts of all the nodes, suitably attenuated, in the presence of noise:

$$y_j(t) = \sum_{i \neq j} \frac{\exp(-\gamma \rho_{ij})}{\rho_{ij}^\alpha} x_i(t) + z_i(t). \tag{9}$$

Here $z_i(t)$ is assumed to be additive white Gaussian noise of variance σ^2 .

With the benefit of this quite physical model of the network, we now pursue an information-theoretic operation. Suppose that each node i wants to send information to node j at a rate of R_{ij} bits/time unit. The vector

$$R := \{R_{ij} : 1 \leq i, j \leq n, \text{ with } i \neq j\}$$

is called the *rate vector*.

As in the point-to-point model of Shannon described in Sect. 2, let T denote a large block length. In T time units, node i therefore wishes to communicate TR_{ij} bits to node j over the network. Suppose therefore a bag of $2^{TR_{ij}}$ messages, from which node i picks a particular message m_{ij} that it wishes to communicate to node j . Note that node i therefore may have a set of messages $\{m_{ij} : j \neq i\}$ that it wishes to send to all the other nodes in the network.

For this purpose, node i chooses an *encoder* E_i^T . At each time unit $1 \leq t \leq T$, the encoder chooses a symbol $x_i(t)$ to transmit which is based on *two* sets of information that it possesses at time t :

1. The set of messages $\{m_{ij} : j \neq i\}$ that it wishes to communicate to the other nodes. This is the *private* information that it alone possesses at the beginning of the block.
2. All the *causal* information $\{y_i(s) : 1 \leq s \leq t - 1\}$, that it has acquired from its receptions up to time $t - 1$.

Thus

$$x_i(t) = E_i^T(t, \{m_{ij} : j \neq i\}, \{y_i(s) : 1 \leq s \leq t - 1\}). \quad (10)$$

If P_i is the power allocated or to be used by node i , then

$$\frac{1}{T} \sum_{t=1}^T x_i^2(t) \leq P_i.$$

The transmissions $\{x_i(t) : 1 \leq t \leq T\}$ have to satisfy whichever of the power constraints, (7) or (8), that we wish to impose.

At the end of the block of T time-units, each node j has the following knowledge:

1. Its own set of messages $\{m_{jk} : k \neq j\}$ that it wished to communicate to the other nodes. This is of course the *private* information that it possessed at the beginning of the block.
2. All the *causal* information $\{y_j(s) : 1 \leq s \leq T\}$, that it acquired from all its receptions up to time T .

Based on this, node j wishes to decode all the messages $\{m_{ij} : i \neq j\}$. For this purpose it chooses a joint decoder D_j^T which produces estimates of all the messages

from all the other nodes that were destined for it:

$$\hat{m}_{ij}(t) = D_j^T(i, \{m_{jk} : k \neq j\}, \{y_j(s) : 1 \leq s \leq T\}) \text{ for all } i \neq j. \quad (11)$$

Let P_{error}^T be the average error probability:

$$\text{Prob}(\hat{m}_{ij} = m_{ij} \text{ for all } 1 \leq i, j \leq n, i \neq j) = 1 - P_{error}^T.$$

when the messages are chosen uniformly and independently from their respective sets of allowed messages.

We will say that a rate vector R is *feasible* if for each T there is an encoding strategy $\{E_i^T : 1 \leq i \leq n\}$, and a decoding strategy $\{D_j^T : 1 \leq j \leq n\}$, such that $\lim_{T \rightarrow +\infty} P_{error}^T = 0$.

The *capacity region* C is the closure of the set of all feasible rate vectors.

The allowable dependency in (10) of each transmission of each node on *all the information* that it initially possessed, or causally acquired up to that time based on all received signals up to that time, as well as the allowable dependency of the decoding strategy (11) on all information that is known to a node, are the reasons for the conclusiveness of information-theoretic results. By allowing for all such strategies as in (10,11), we do not impose unnecessary restrictions on the operation of the system. Thus strategies for amplify-and-forward/decode-and-forward or active noise-cancellation, or successive interference cancellation, and not just these, other strategies too that we may not even have conceived of as yet, are all allowed.

12 Information-Theoretic Definition of Transport Capacity

The holy grail is to determine the capacity region C . However, that continues to elude us.

We can make progress by trying to focus on just the *transport capacity* C_T , defined as

$$C_T := \sup_{\text{Feasible rate vectors } R} \sum_{\{(i,j):i \neq j\}} \rho_{ij} R_{ij} \text{ bit-meters/time-unit.}$$

Note that ρ_{ij} is the distance between nodes i and j . So, just as in Sect. 6, we are weighting the rates achieved between source-destination pairs by the distances between the source and destination nodes.

It should be noted that the rate-vector R is an $n(n-1)$ -dimensional vector. Hence the capacity region C is a subset of a rather high dimensional space. On the other hand the transport capacity is just a scalar. Clearly the transport capacity does not describe every rate vector that the network is capable of supporting. However it

provides a quick assessment of what the network is capable of supporting in terms of its ability to pump information over distance.

13 Information-Theoretic Bounds

The key question is how to obtain an upper bound on transport capacity. To do this we need to convert it to a more tractable problem.

One of the few general results that has been used to obtain upper bounds (also called *outer bounds*) is the *cutset bound* of network information theory proved in the Ph.D. dissertation of El Gamal [14].

Consider a cutset that divides the set of all nodes into two subsets S and S^c , as shown in Fig. 15. The information flow across the cutset (from left to right) is $\sum_{i \in S, j \in S^c} R_{ij}$. Let $x_S := \{x_i : i \in S\}$ denote the transmissions of the nodes in S . The cutset bound says that if R is a feasible rate vector, then there exists a joint probability distribution $p(x_1, x_2, \dots, x_n)$ such that

$$\sum_{i \in S, j \in S^c} R_{ij} \leq I(x_S; y_{S^c} | x_{S^c}) \text{ for all subsets } S \subseteq \{1, 2, \dots, n\}. \quad (12)$$

So if we can somehow reduce the problem of determining an upper bound on transport capacity to a one of bounding information flow across cutsets, then we can obtain an upper bound on the transport capacity. For simplicity, let us suppose that $\rho_{min} = 2$; that is, nodes are no less than 2 m apart. A reduction to a problem involving cutsets can then be done by considering a large number of horizontal and vertical cutsets, separated by 1 m apart; see Fig. 16, due to the following reasons.

Note that each square of side 1 m can contain no more than 1 node since the minimum separation distance between any two nodes is no less than 2 m. Consider two nodes i and j separated by $\rho_{ij} \geq 2$ m; as in Fig. 16. Denote by (a_i, b_i) the coordinates of node i . We wish to study the quantity $R_{ij} \rho_{ij}$. Let us focus on the quantity ρ_{ij} . This is the length of the straight line connecting nodes i and j , which is the ℓ_2 or Euclidean norm. However, the ℓ_2 norm is bounded by a multiple of the

Fig. 15 A cutset and the information flow across it

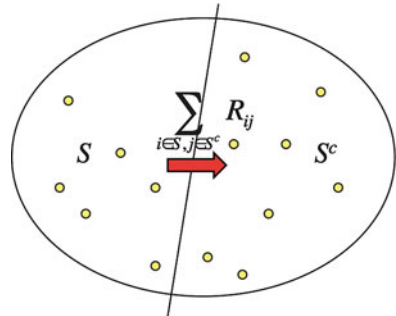
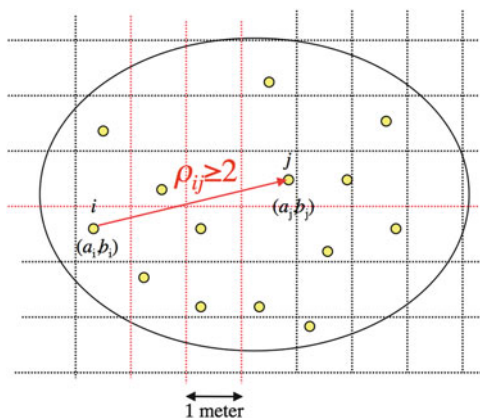


Fig. 16 Vertical and horizontal cuts separated by 1 m



ℓ_1 norm, i.e., there is a constant $c > 0$ such that

$$\rho_{ij} \leq c (|a_j - a_i| + |b_j - b_i|).$$

In turn,

$$|a_j - a_i| \leq \text{Number of vertical lines cut by the straight line from } i \text{ to } j,$$

and

$$|b_j - b_i| \leq \text{Number of horizontal lines cut by the straight line from } i \text{ to } j.$$

Hence

$$\begin{aligned} \rho_{ij} \leq & c (\text{Number of vertical lines cut by straight line from } i \text{ to node } j \\ & + \text{Number of horizontal lines cut by straight line from node } i \text{ to } j). \end{aligned}$$

From this it follows that

The nodes $i \in S$ use power P_i , and the receptions of nodes $j \in S^c$ have additive white Gaussian noise of variance σ^2 . Hence, in a manner similar to the calculations that lead to Theorem 2, the conditional mutual information $I(x_S; y_{S^c} | x_{S^c})$ can be computed.

This gives rise to the following bound on transport capacity in terms of the total transmit power used by all the nodes whenever there is absorption, which is the case except in a vacuum, or even otherwise when $\delta > 3$.

Theorem 6 When $\gamma > 0$, or $\delta > 3$, the transport capacity is bounded by

$$C_T \leq \frac{c_1(\gamma, \delta, \rho_{min})}{\sigma^2} P_{total},$$

where c_1 is a constant that depends only on the indicated quantities.

We can proceed to further study the dependence of transport capacity on the number of nodes n . To do this, simply consider the case of individual power constraint P_{ind} on all nodes. Then

$$P_{total} \leq n P_{ind}.$$

Substituting this in the above theorem gives the following *information-theoretic scaling law*.

Theorem 7 When $\gamma > 0$, or $\delta > 3$, the transport capacity is bounded by

$$C_T \leq \frac{c_1(\gamma, \delta, \rho_{min}) P_{ind}}{\sigma^2} n,$$

where c_1 is a constant that depends only on the indicated quantities.

14 Implication of Information-Theoretic Scaling Law

In the previous section addressing the information-theoretic scaling law we have considered the case of an expanding network since the minimum distance between nodes is lower bounded. We can therefore compare this scaling law with the scaling law (6) that results when we only use multi-hop relaying, with decode-and-forward, and treating all interference as noise. Both scale linearly with the number of nodes n . Therefore we can conclude that such restricted operation is order-optimal, i.e., optimal to within a multiplicative constant. Thus, through an information-theoretic analysis we have obtained an architectural conclusion concerning how to order-optimally operate wireless networks.

Moreover we have determined how transport capacity can scale as the number of nodes increases.

Both these results hold whenever there is absorption in the medium, which is generally the case, except in vacuum.

15 Extensions

The above results can be and have been extended in several ways.

The model of Sect. 4 was initiated in [6], and is called the *protocol model*. The sharpest best case results are obtained in [15]. An extension of the protocol model is the *physical model*, where the success and/or rate of a transmission depends on the signal-to-noise-plus-interference ratio. This model too was studied in [6], and the sharpest results on the best case are obtained in [16].

Instead of the best case, one can consider the *random case*. Here the nodes are randomly located in a domain, and source-destination pairs are randomly chosen. Such networks are called *random networks*. It is of interest to see how to best operate such randomly formed networks. The random case version of the protocol model is studied in [6], where the maximal per-node throughput is sharply characterized up to order. The random case under the physical model is also studied in [6].

The information-theoretic approach to wireless networks was initiated in [7]. The results presented in this chapter are from [7].

Subsequent work has focused on how to extend the results in [7]. There have been two types of attempts. One direction of work has been to determine to what extent the *heavy attenuation regime's* scaling behavior continues to hold [17–20]. This is the regime where the absorption constant $\gamma > 0$, i.e., there is absorption, or the path-loss exponent δ is large. (It should be noted that $\gamma > 0$ except in vacuum.) The results of [7] have been extended to the case of more general fading models in [17], and the same scaling laws are shown to hold. The validity of the linear scaling results for smaller δ than 3 is examined in [18] and [19], with the latter showing that the regime holds even for $\gamma = 0$ and $\delta > 2$. The common per-node throughput in a random network is examined in [20].

Another direction of work has been to address what sorts of cooperation strategies among the nodes are appropriate in the *low attenuation regime*. This was initiated in [7], where counterexamples to linear scaling were demonstrated when $\gamma = 0$ and δ is very small. It was shown that one could obtain unbounded transport capacity for bounded power, and superlinear scaling through other multi-node cooperation strategies. In [21] it was shown how to achieve superlinear scaling in a dense network. This was extended in [22] where a novel hierarchical cooperation strategy is proposed and the range of $1 < \delta < 2$ was addressed. The value of the pre-constant was more carefully examined in [23]. The case of arbitrarily spaced nodes was examined in [24].

Another avenue of work has been to proceed to obtain limitations on the capabilities of wireless networks by examining Maxwell's equations themselves [25].

The results of the protocol model are extended to the case of *function computation in networks*, and not just data transfer, in [26].

References

1. C.E. Shannon, A mathematical theory of communication. *Bell Syst. Tech. J.* **27**, 379–423, 623–656 (1948)
2. N. Abramson, The ALOHA system – another alternative for computer communications, in *AFIPS Conference Proceedings*, vol. 37 (1970), pp. 281–285
3. R.M. Metcalfe, D.R. Boggs, Ethernet: distributed packet switching for local computer networks. *Commun. ACM* **19**(7), 395–404 (1976)
4. IEEE 802 LAN/MAN Standards Committee: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE Standard 802.11, 1999 edition (1999)
5. A. Ephremides, B. Hajek, Information theory and communication networks: an unconsummated union. *IEEE Trans. Inf. Theory* **44**, 2416–2434 (1998)
6. P. Gupta, P.R. Kumar, The capacity of wireless networks. *IEEE Trans. Inf. Theory* **IT-46**, 388–404 (2000)
7. L. Xie, P.R. Kumar, A network information theory for wireless communication: scaling laws and optimal operation. *IEEE Trans. Inf. Theory* **50**, 748–767 (2004)
8. R. Ahlswede, Multiway communication channels, in *Proceedings of the Second International Symposium on Information Theory*, Tshaksador, Armenian S.S.R. (1971), pp. 23–52
9. H. Liao, Multiple access channels, Ph.D. thesis, University of Hawaii, 1972
10. H. Weingarten, Y. Steinberg, S. Shamai, The capacity region of the Gaussian MIMO broadcast channel, in *Proceedings IEEE International Symposium on Information Theory (ISIT)*, Chicago, IL (2004)
11. T.M. Cover, A. El Gamal, Capacity theorems for the relay channel. *IEEE Trans. Inf. Theory* **25**, 572–584 (1979)
12. M. Franceschetti, J. Bruck, L. Schulman, Microcellular systems, random walks and wave propagation, in *Proceedings of the IEEE Symposium on Antennas and Propagation Society (IEEE AP-S 2002)*, San Antonio (2002)
13. T.S. Rappaport, *Wireless Communications, Principles and Practice* (Prentice Hall, New York, 1996)
14. A.E. Gamal, Results in Multiple User Channel Capacity, Ph.D. thesis, Stanford University, 1978
15. A. Agarwal, P.R. Kumar, Improved capacity bounds for wireless networks. *Wirel. Commun. Mob. Comput.* **4**, 251–261 (2004)
16. A. Agarwal, P.R. Kumar, Capacity bound for ad-hoc and hybrid wireless networks. *ACM SIGCOMM Comput. Commun. Rev.* **34**, 71–81 (2004). Special issue on Science of Networking Design
17. F. Xue, L. Xie, P.R. Kumar, The transport capacity of wireless networks over fading channels. *IEEE Trans. Inf. Theory* **51**, 834–847 (2005)
18. A. Jovicic, P. Viswanath, S.R. Kulkarni, Upper bounds to transport capacity of wireless networks. *IEEE Trans. Inf. Theory* **50**(11), 2555–2565 (2004)
19. L.-L. Xie, P.R. Kumar, On the path-loss attenuation regime for positive cost and linear scaling of transport capacity in wireless networks. *Joint Special Issue IEEE Trans. Inf. Theory IEEE/ACM Trans. Netw. Netw. Inf. Theory* **52**(6), 2313–2328 (2006)
20. O. Leveque, I.E. Telatar, Information-theoretic upper bounds on the capacity of large extended ad hoc wireless networks. *IEEE Trans. Inf. Theory* **51**(3), 858–865 (2005)
21. S. Aeron, V. Saligrama, Wireless ad hoc networks: strategies and scaling laws for the fixed SNR regime. *IEEE Trans. Inf. Theory* **53**(6), 2044–2059 (2007)
22. A. Ozgur, O. Leveque, D.N.C. Tse, Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks. *IEEE Trans. Inf. Theory* **53**(10), 3549–3572 (2007)
23. L.-L. Xie, On information-theoretic scaling laws for wireless networks. *IEEE Trans. Inf. Theory* (2008). Revised, July 2009. <http://arxiv.org/abs/0809.1205>
24. U. Niesen, P. Gupta, D. Shah, The capacity region of large wireless networks. *IEEE Trans. Inf. Theory* (2008). Available online at <http://arxiv.org/abs/0809.1344>

25. M. Franceschetti, M.D. Migliore, P. Minero, The capacity of wireless networks: information-theoretic and physical limits. *IEEE Trans. Inf. Theory* **55**, 3413–3424 (2009)
26. A. Giridhar, P.R. Kumar, Computing and communicating functions over sensor networks. *IEEE J. Sel. Areas Commun.* **23**(4), 755–764 (2005)

Edited by J.-M. Morel, B. Teissier; P.K. Maini

Editorial Policy (for Multi-Author Publications: Summer Schools / Intensive Courses)

1. Lecture Notes aim to report new developments in all areas of mathematics and their applications - quickly, informally and at a high level. Mathematical texts analysing new developments in modelling and numerical simulation are welcome. Manuscripts should be reasonably self-contained and rounded off. Thus they may, and often will, present not only results of the author but also related work by other people. They should provide sufficient motivation, examples and applications. There should also be an introduction making the text comprehensible to a wider audience. This clearly distinguishes Lecture Notes from journal articles or technical reports which normally are very concise. Articles intended for a journal but too long to be accepted by most journals, usually do not have this "lecture notes" character.
2. In general SUMMER SCHOOLS and other similar INTENSIVE COURSES are held to present mathematical topics that are close to the frontiers of recent research to an audience at the beginning or intermediate graduate level, who may want to continue with this area of work, for a thesis or later. This makes demands on the didactic aspects of the presentation. Because the subjects of such schools are advanced, there often exists no textbook, and so ideally, the publication resulting from such a school could be a first approximation to such a textbook. Usually several authors are involved in the writing, so it is not always simple to obtain a unified approach to the presentation.
For prospective publication in LNM, the resulting manuscript should not be just a collection of course notes, each of which has been developed by an individual author with little or no coordination with the others, and with little or no common concept. The subject matter should dictate the structure of the book, and the authorship of each part or chapter should take secondary importance. Of course the choice of authors is crucial to the quality of the material at the school and in the book, and the intention here is not to belittle their impact, but simply to say that the book should be planned to be written by these authors jointly, and not just assembled as a result of what these authors happen to submit.
This represents considerable preparatory work (as it is imperative to ensure that the authors know these criteria before they invest work on a manuscript), and also considerable editing work afterwards, to get the book into final shape. Still it is the form that holds the most promise of a successful book that will be used by its intended audience, rather than yet another volume of proceedings for the library shelf.
3. Manuscripts should be submitted either online at www.editorialmanager.com/lnm/ to Springer's mathematics editorial, or to one of the series editors. Volume editors are expected to arrange for the refereeing, to the usual scientific standards, of the individual contributions. If the resulting reports can be forwarded to us (series editors or Springer) this is very helpful. If no reports are forwarded or if other questions remain unclear in respect of homogeneity etc, the series editors may wish to consult external referees for an overall evaluation of the volume. A final decision to publish can be made only on the basis of the complete manuscript; however a preliminary decision can be based on a pre-final or incomplete manuscript. The strict minimum amount of material that will be considered should include a detailed outline describing the planned contents of each chapter.
Volume editors and authors should be aware that incomplete or insufficiently close to final manuscripts almost always result in longer evaluation times. They should also be aware that parallel submission of their manuscript to another publisher while under consideration for LNM will in general lead to immediate rejection.

4. Manuscripts should in general be submitted in English. Final manuscripts should contain at least 100 pages of mathematical text and should always include
 - a general table of contents;
 - an informative introduction, with adequate motivation and perhaps some historical remarks: it should be accessible to a reader not intimately familiar with the topic treated;
 - a global subject index: as a rule this is genuinely helpful for the reader.

Lecture Notes volumes are, as a rule, printed digitally from the authors' files. We strongly recommend that all contributions in a volume be written in the same LaTeX version, preferably LaTeX2e. To ensure best results, authors are asked to use the LaTeX2e style files available from Springer's web-server at

<ftp://ftp.springer.de/pub/tex/latex/svmonot1/> (for monographs) and

<ftp://ftp.springer.de/pub/tex/latex/svmult1/> (for summer schools/tutorials).

Additional technical instructions, if necessary, are available on request from:

lnm@springer.com.

5. Careful preparation of the manuscripts will help keep production time short besides ensuring satisfactory appearance of the finished book in print and online. After acceptance of the manuscript authors will be asked to prepare the final LaTeX source files and also the corresponding dvi-, pdf- or zipped ps-file. The LaTeX source files are essential for producing the full-text online version of the book. For the existing online volumes of LNM see:
<http://www.springerlink.com/openurl.asp?genre=journal&issn=0075-8434>.
The actual production of a Lecture Notes volume takes approximately 12 weeks.

6. Volume editors receive a total of 50 free copies of their volume to be shared with the authors, but no royalties. They and the authors are entitled to a discount of 33.3 % on the price of Springer books purchased for their personal use, if ordering directly from Springer.

7. Commitment to publish is made by letter of intent rather than by signing a formal contract. Springer-Verlag secures the copyright for each volume. Authors are free to reuse material contained in their LNM volumes in later publications: a brief written (or e-mail) request for formal permission is sufficient.

Addresses:

Professor J.-M. Morel, CMLA,
École Normale Supérieure de Cachan,
61 Avenue du Président Wilson, 94235 Cachan Cedex, France
E-mail: morel@cmla.ens-cachan.fr

Professor B. Teissier, Institut Mathématique de Jussieu,
UMR 7586 du CNRS, Équipe "Géométrie et Dynamique",
175 rue du Chevaleret,
75013 Paris, France
E-mail: teissier@math.jussieu.fr

For the "Mathematical Biosciences Subseries" of LNM:

Professor P. K. Maini, Center for Mathematical Biology,
Mathematical Institute, 24-29 St Giles,
Oxford OX1 3LP, UK
E-mail: maini@maths.ox.ac.uk

Springer, Mathematics Editorial I,
Tiergartenstr. 17,
69121 Heidelberg, Germany,
Tel.: +49 (6221) 4876-8259
Fax: +49 (6221) 4876-8259
E-mail: lnm@springer.com