Correlation Decay and Decentralized Optimization in Graphical Models

by

Théophane Weber

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2010

Author .....
Sloan School of Management
January 16, 2010

Certified by .....
David Gamarnik
J. Spencer Standish Associate Professor of Operations Research
Thesis Supervisor

Certified by .....
John Tsitsiklis
Clarence J. Lebel Professor of Electrical Engineering
Thesis Supervisor

Accepted by .....
Dimitris Bertsimas
Boeing Professor of Operations Research
Co-Director, Operations Research Center

Correlation Decay and Decentralized Optimization in Graphical Models

by

Théophane Weber

Submitted to the Operations Research Center
on January 15, 2010, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

Many models of optimization, statistics, social organizations and machine learning capture local dependencies by means of a network that describes the interconnections and interactions of different components. However, in most cases, optimization or inference on these models is hard due to the dimensionality of the networks. This is so even when using algorithms that take advantage of the underlying graphical structure. Approximate methods are therefore needed. The aim of this thesis is to study such large-scale systems, focusing on the question of how randomness affects the complexity of optimizing in a graph; of particular interest is the study of a phenomenon known as correlation decay, namely, the phenomenon where the influence of a node on another node of the network decreases quickly as the distance between them grows.

In the first part of this thesis, we develop a new message-passing algorithm for optimization in graphical models. We formally prove a connection between the correlation decay property and (i) the near-optimality of this algorithm, as well as (ii) the decentralized nature of optimal solutions. In the context of discrete optimization with random costs, we develop a technique for establishing that a system exhibits correlation decay. We illustrate the applicability of the method by giving concrete results for the cases of uniform and Gaussian distributed cost coefficients in networks with bounded connectivity.

In the second part, we pursue similar questions in a combinatorial optimization setting: we consider the problem of finding a maximum weight independent set in a bounded degree graph, when the node weights are i.i.d. random variables. Surprisingly, we discover that the problem becomes tractable for certain distributions. Specifically, we construct a PTAS for the case of exponentially distributed weights and arbitrary graphs with degree at most 3, and obtain generalizations for higher degrees and different distributions. At the same time we prove that no PTAS exists for the case of exponentially distributed weights for graphs with sufficiently large but bounded degree, unless P=NP.

Next, we shift our focus to graphical games, which are a game-theoretic analog of graphical models. We establish a connection between the problem of finding an approximate Nash equilibrium in a graphical game and the problem of optimization in graphical

models. We use this connection to re-derive NashProp, a message-passing algorithm which computes Nash equilibria for graphical games on trees; we also suggest several new search algorithms for graphical games in general networks. Finally, we propose a definition of correlation decay in graphical games, and establish that the property holds in a restricted family of graphical games.

The last part of the thesis is devoted to a particular application of graphical models and message-passing algorithms to the problem of early prediction of Alzheimer's disease. To this end, we develop a new measure of synchronicity between different parts of the brain, and apply it to electroencephalogram data. We show that the resulting prediction method outperforms a vast number of other EEG-based measures in the task of predicting the onset of Alzheimer's disease.

Thesis Supervisor: David Gamarnik

Title: J. Spencer Standish Associate Professor of Operations Research

Thesis Supervisor: John Tsitsiklis

Title: Clarence J. Lebel Professor of Electrical Engineering

Acknowledgments[1]

First and foremost, I would like to extend my deepest gratitude to my advisors, David Gamarnik and John Tsitsiklis, for their tremendous help, patience, support, and guidance throughout these past years. Their high research standards, wisdom, and great human qualities are a few of the many reasons I feel extremely lucky to have worked with and learnt from them during my time at the ORC. I am also very thankful for all the time they spent making me a better writer.

I would like to thank Professor Alan Willsky, for taking the time to serve on my thesis committee, for offering great feedback and advice on my work, and for being so enthusiastic about research in general.

I am indebted to my former advisor Daniela Pucci de Farias, for her unconditional support and friendship even as she made major changes in her life path. I also wish to thank again my Master's thesis advisor, Jérémie Gallien, who taught me through the first steps of research.

While at MIT, I have benefited a lot from interacting with many faculty members, and thank all of them for making MIT such a great place. I would like to thank in particular Professor Devavrat Shah, for always being so friendly, and from whom I learnt a lot about message-passing algorithms, and Professor Robert Freund, for answering many questions I had about optimization, and for sharing his experiences.

Thanks to the administrative staff at the ORC, Laura, Paulette, and Andrew, for getting me out of trouble more times than I can really count, and to all the staff at LIDS.

The last chapter of this thesis is based on joint work from my time at the Riken Brain Science Institute in Japan, and I am deeply grateful to Francois Vialatte, Andrzej Cichocki and Justin Dauwels for the wonderful welcome they extended to me, and for working with me on such an interesting problem.

Thanks also to friends at Lyric - especially Ben, Bill, Jeff and Shawn, for welcoming me in the next phase of my research career.

I have great memories of times spent with friends from the ORC. Yann, Guillaume, Katy K., Margrét, Susan, Carol, Pavithra, Nelson, Hamed, David Cz., Juliane, Dan, Ilan, Kostas, Ruben, Bernardo, Ross, Yehua, David G., and many others all made the ORC a truly special place I will be sad to leave. Special thanks to Yann, Nelson, Hamed and David Goldberg for many good times and discussions about life and/or research. I am also grateful to David for collaborating with me on many results of this thesis.

I also want to thank friends outside of the ORC for making life in Boston so great: fellow party-goers (Francois, Srini and Lauren), 109-ers and roommates extraordinaires (Meg, Shaun, Quirin, Arti, Kristen, Francois and Celia, Ilan and Inna, Paul, and many others), Wellesley friends (Laure-Anne, Meg, Pau); fellow Japan travelers (Mike, Aniket, Chris, Margaret, Annie, special thanks to Helena for being such a great and patient friend);

[1]This thesis was supported in part by NSF grant DMI-0447766

Suki and her family for always being so nice to me.

Many friends from France I miss very much, and I am glad we were able to keep in touch through all these years (in anticipation for deserved flak: I do feel bad for not giving news often enough!) – thanks to Sarah, Vaness, Eglantine, Jessica and too many others to list.

I feel deeply indebted to many dear friends, for they have taught me much and made me a better person. Alex, for this loyalty and thoughtfulness. Aurélien, for his great sense of humor, and for always making me look forward to visiting Paris. Paolo 'True Man' Morra is an inspiration to this day for us all. Justin and Shokho for their enthusiasm in all things, their unending supply of cheerfulness, and for being amazing brunch companions. The Mysterious and Magnificient Mohamed Mostagir, for listening to me complain more than anyone else has, for always finding the right word to make me laugh and cheer me up, and for knowing me better than I probably know myself. Hadil, for her contagious spirit. Khaldoun, for being a kindred spirit, for so much help in the last years, and for all the good times ("elle rentre quand, Naima?"). To all of you, thanks for being so great :)

Thanks to my family for their unwavering support. Coming home every year to spend time with my brother Jean-Guillaume and my sister Margot was the event I most looked forward to, and the thought of these times kept me going for the months we were apart. My mom dedicated her life to her children, and I will be eternally grateful to her for all the values she taught us, for making so many opportunities available to us, and for her courage and selflessness.

Finally, none of this would have been possible without the love of Katy, who always brings happiness into my life. Her encouragements and uplifting spirits brightened the hard times, her sweet smile and loving kindness made every day better. No words can truly express my love for her. This thesis is for you.

Contents

1 Message-passing schemes and Belief Propagation **29**

1.1 Introduction and literature review 29

1.2 Message-passing schemes and framework 32

1.3 The Belief Propagation algorithm 34

1.4 Variations of Belief Propagation 39

1.5 Conclusions 42

2 The Cavity Expansion algorithm **45**

2.1 Introduction 45

2.2 The cavity recursion 46

2.2.1 The SAW tree construction 46

2.2.2 Extension to factor graphs 50

2.3 The Cavity Expansion algorithm for graphs or factor graphs 53

2.3.1 The CE algorithm 53

2.3.2 Properties and computational complexity 55

2.3.3 Message-passing version of the CE algorithm 58

2.4 Conclusions 59

3 Correlation decay and efficient decentralized optimization in decision networks with random objective functions **61**

3.1 Introduction 61
3.2 Model description and results 63
3.3 Correlation decay and decentralized optimization 66
3.4 Establishing the correlation decay property 71
3.4.1 Coupling technique 73
3.4.2 Establishing coupling bounds 78
3.5 Decentralization 86
3.6 Regularization technique 88
3.7 Conclusions 90

4 Correlation decay and average-case complexity of the Maximum Weight Independent Set problem 93

4.1 Introduction 93
4.2 Model description and results 94
4.3 Cavity expansion and the algorithm 97
4.4 Correlation decay for the MWIS problem 101
4.5 Hardness result 106
4.6 Generalization to phase-type distribution 108
4.6.1 Mixture of exponentials 108
4.6.2 Phase-type distribution 111
4.7 Conclusions 113

5 Graphical games 115

5.1 Introduction 115
5.2 Game theory, Nash equilibria, and approximate Nash equilibria 118
5.3 Graphical games 120
5.4 Computation of Nash equilibrium 122
5.4.1 Nash cavity function 122

5.4.2 From Nash Cavity functions to Nash equilibria 123
5.4.3 Existence of approximate Nash equilibria 125
5.5 Message-passing algorithms for graphical games 128
5.5.1 TreeProp and NashProp 128
5.5.2 A framework for deriving message-passing algorithms for graphical games 130
5.5.3 Search algorithms 133
5.6 Correlation decay and local Nash equilibrium 137
5.6.1 Results 139
5.6.2 Branching argument 140
5.6.3 Dobrushin trick 143
5.7 Conclusions 146

6 Application of graphical models and message-passing techniques to the early diagnosis of Alzheimer's disease 149

6.1 Introduction 149
6.2 Basic principle 151
6.2.1 Measures of synchronicity 151
6.2.2 Stochastic Event Synchrony 152
6.3 A class of statistical model measuring similarity between two point processes 159
6.3.1 Bivariate SES 159
6.3.2 Statistical inference for bivariate SES 165
6.4 Comparing multiple point processes at the same time 169
6.4.1 Principle of multivariate SES 169
6.4.2 Stochastic model for multivariate SES 170
6.4.3 Statistical inference for multivariate SES 173
6.5 Application to early diagnosis of Alzheimer's disease 176
6.5.1 EEG Data 177

6.5.2 Results and Discussion 179
6.5.3 Classification 181
6.6 Conclusions 182

A Glossary **185**

B Notions in complexity theory and approximation algorithms **191**
B.1 The P and NP classes, approximation algorithms 191
B.2 The PPAD class 194

C Preprocessing of brain data: Wavelet Transform and Bump Modeling of EEG data **197**

D Complements on multivariate SES and additional graphs **205**
D.1 Computational hardness of multivariate SES 205
D.2 Extension of multivariate SES to the multinomial prior 207
D.3 Extra figures 209

List of Figures

0-1 Equivalence between hypergraphs and factor graphs 16

1-1 Message-passing schemes 33
1-2 Dynamic optimization recursion on a tree 35
1-3 Tree splitting 37

2-1 First step: building the telescoping sum 47
2-2 Second step: building the modified subnetworks 49

6-1 Bump modeling of EEG data 153
6-2 Stochastic Event Synchrony: principle 155
6-3 Bump modeling 156
6-4 Multivariate SES: principle 158
6-5 Generative model 160
6-6 Classification with bivariate SES 183
6-7 Classification with multivariate SES 183
6-8 AD Classification with three features 184

C-1 Bump modeling: principle 201
C-2 EEG electrodes placement 204

D-1 Box plots for the most discriminating classical measures 210

D-2 Box plots for multivariate SES 211

List of Tables

6.1 p-values of different synchronicity measures for MCI and control population 180

Definitions

We use the following conventions and notations in this thesis: key concepts, when they are first introduced, are italicized. We will use the symbol $\triangleq$ for equations which define symbols. The notation of single elements v of a set V uses a regular typeface, while the bold typeface $\mathbf{v}$ denotes vectors or arrays. For any set V, 2^V denotes the set of all subsets of V.

A *graph* $\mathcal{G} = (V, E)$ is an object composed of a finite set of nodes or vertices V and a set E of unordered pairs of elements of V. $\vec{E}$ denotes the set of oriented edges of (V, E) whose unoriented version belongs to E. More generally, a *hypergraph* (V, C) is composed of a set of nodes V and a set of hyperedges $C \subset 2^V$, where each hyperedge $\mathcal{C} \in C$ is a nonempty subset of V. Finally, a network consists of a graph or hypergraph, along with a collection of functions indexed by the elements (nodes, edges, or hyperedges) of the graph or hypergraph.

Hypergraphs can be seen as being equivalent to *factor graphs*. A factor graph is a bipartite graph $\mathcal{G} = (G, E)$, with $G = V \cup A$, and where the vertices of V are called *variable nodes* and the vertices of A are called *factor nodes*. There is a trivial bijection between hypergraphs and graphs, where for any hypergraph (V, C), we create a factor graph (V, A, E), where each hyperedge $\mathcal{C}$ is mapped to a single factor node $a(\mathcal{C}) \in A$, and we create edges in the factor graph according to the rule $v \in \mathcal{C}$ if and only if $(v, a(\mathcal{C})) \in E$.

Given a graph (V, E) (resp. hypergraph (V, A)) and a subset U of V, the graph (resp. hypergraph) induced by U is the graph (U, E') (resp. (U, C')), where $E' = \{(u, v) \in E : u, v \in U\}$ ($C' = \{\mathcal{C} \in C : \mathcal{C} \subset U\}$. Induced networks, which we call *subnetworks*, are defined similarly.

For any graph $\mathcal{G} = (V, E)$, and any two nodes (u, v) in V, let $d(u, v)$ be the length of

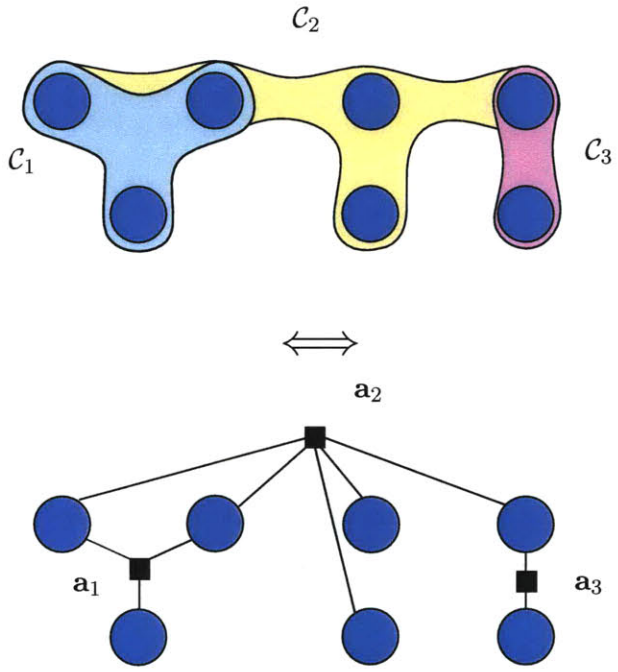


Figure 0-1: Equivalence between hypergraphs and factor graphs

a shortest path (in number of edges) between u and v. Given a node u and integer $r \geq 0$, let $\mathcal{B}_{\mathcal{G}}(u,r) \triangleq \{v \in V : d(u,v) \leq r\}$. Let also $\mathcal{N}_{\mathcal{G}}(u) \triangleq \mathcal{B}(u,1)\backslash\{u\}$. The extended set of neighbors $\mathcal{N}(u)^e$ is $\mathcal{B}_{\mathcal{G}}(u,1) = \mathcal{N}(u) \cup \{u\}$. For any $r > 0$, let $\mathcal{N}_G^r(v)$ be the subnetwork induced by $\mathcal{B}_{\mathcal{G}}(u,r)$. For any node u, $\Delta_{\mathcal{G}}(u) \triangleq |\mathcal{N}_{\mathcal{G}}(u)|$ is the number of neighbors of u in $\mathcal{G}$. Let $\Delta_{\mathcal{G}}$ be the maximum degree of graph (V,E); namely, $\Delta_{\mathcal{G}} = \max_v |\mathcal{N}_{\mathcal{G}}(v)|$. Often we will omit the reference to the network $\mathcal{G}$ when obvious from context.

We will often consider sets and sequences (i.e., ordered sets) of elements indexed by the vertices of a network (V,E). For any set or sequence of elements $\mathbf{x} = (x_v)_{v\in V}$ indexed by the elements of a set V, and any subset $U = (v_1, v_2, \ldots, v_{|U|})$ of V, $\mathbf{x}_U = (x_{v_1}, x_{v_2}, \ldots, v_{v_{|U|}})$ denotes the sequence of corresponding elements. For any set or sequence $\mathbf{x}$ and element $u \in V$, we denote $\mathbf{x}_{-u} \triangleq \mathbf{x}_{V\backslash\{u\}} = (x_v)_{v\in V, v\neq u}$ the set of elements for all nodes other than u. Finally, for any set or sequence $\mathbf{x}$, and element $u \in V$, we denote $\mathbf{x}_{\sim u} \triangleq \mathbf{x}_{v\in\mathcal{N}(u)}$ the set of elements that are neighbors of u.

We will assume an underlying probability space, denoted as $(\Omega, \mathcal{B}, \mathbb{P})$. For a set of discrete random variables $\mathbf{X} = (X_1, \ldots, X_n)$ and possible outcome $\mathbf{x} = (x_1, \ldots, x_n)$, we

denote $\mathbb{P}(\mathbf{X} = \mathbf{x})$ the probability that the random vector $\mathbf{X}$ takes the value $\mathbf{x}$. If the X_i are jointly continuous, we denote by $d\mathbb{P}(\mathbf{X} = \mathbf{x})$ the density of $\mathbf{X}$. Finally, for any random variable X, $\mathbb{E}[X]$ denotes the expected value of X, and for any sub-σ algebra $\mathcal{A}$ of $\mathcal{B}$, $\mathbb{E}[X \mid \mathcal{A}]$ is the conditional expectation of X given $\mathcal{A}$.

For any finite set χ, we denote $\mathcal{S}(\chi)$ the simplex over χ, or alternatively, the set of probability distributions over elements of χ: $\mathcal{S}(\chi) = \{x \in [0,1]^\chi \mid \forall a \in \chi,\ x(a) \geq 0, \text{ and } \sum_{a \in \chi} x(a) = 1\}$. Elements a of χ can be viewed as elements of $\mathcal{S}(\chi)$ (with unit mass at a). For any $\delta = 1/n > 0$ for some positive integer n, let $\mathcal{S}_\delta(\chi)$ be the set of probability distributions over χ whose components are integer multiples of δ ($\forall \mathbf{s} \in \mathcal{S}_\delta(\chi), \exists \mathbf{k} = (k_1, \ldots, k_{|\chi|}) \in \mathbb{N}^{|\chi|}$ such that $\mathbf{s} = \delta\mathbf{k}$ and $\sum_i k_i = 1/\delta$).

Introduction

Graphical Models

Many models of optimization, statistics, control, and learning capture local interactions and sparse dependencies by means of a network in which different components are connected and interacting. Originating in statistical physics under the name *Ising model* in the first half of the 20th century (see [Isi24, Ons39, Ons44]), these classes of models have since flourished in a number of different fields, most prominently, statistics and artificial intelligence [Lau96, Jor98, WJ08], theoretical computer science and combinatorial optimization [AS04, BSS05, GNS06, BG06, BGK+07, GG09], coding theory [BGT93, MU07, RU08], game theory [KLS01a, OK03, DP06], and decentralized optimization and control [KP00, GDKV03, RR03, CRVRL06]. While the problems these different communities consider and the questions they aim to answer are at first glance different, research in recent years has uncovered previously unknown connections between these fields. At a high level, research in graphical models has aimed to understand how macroscopic phenomena of interest arise from local properties, and efforts to answer this question have led to the fast development of new classes of distributed algorithms designed to compute critical physical parameters of the system, perform optimization, or carry statistical inference on the network.

The basic model

The basic model we will be considering throughout this thesis is the following: we consider a team of agents working in a networked structure given as a hypergraph (V, A), where V is the set of agents, and where each hyperedge $\mathbf{a} \in A$ represents local interaction within a subteam of agents. Each agent $u \in V$ makes a decision x_u in a finite set $\chi \triangleq \{0, 1, \ldots, T -$

1}. For every $v \in V$ (resp. every hyperedge $\mathbf{a}$), a function $\phi_v : \chi \to \mathbb{R}$ (resp. $\phi_{\mathbf{a}} : \chi^{|\mathbf{a}|} \to \mathbb{R}$) is given. Functions ϕ_v and $\phi_{\mathbf{a}}$ will be called *potential functions* and *interaction functions* respectively. Let $\Phi = ((\phi_v)_{v \in V}, (\phi_{\mathbf{a}})_{\mathbf{a} \in A})$. A vector $\mathbf{x} = (x_1, x_2, \ldots, x_{|V|})$ of actions is called a solutions for the decision network (individual components x_v will be called *decisions*). The set $\mathcal{G} = (V, A, \Phi, \chi)$ is called a *decision network* or *graphical model*. The set of potential and interaction functions Φ defines a network function (often called the *energy function*) $F_{\mathcal{G}}$ which associates with each solution $\mathbf{x}$ the value

$$F_{\mathcal{G}}(\mathbf{x}) = \sum_{\mathbf{a} \in A} \phi_{\mathbf{a}}(x_{\mathbf{a}}) + \sum_{v} \phi_v(x_v) \tag{1}$$

In an optimization context, the goal is to find a decision vector $\mathbf{x}^*$ which minimizes or maximizes the function $F_{\mathcal{G}}(\mathbf{x})$: $\mathbf{x}^* \in \operatorname{argmax}_{\mathbf{x}} F_{\mathcal{G}}(\mathbf{x})$.

In the context of statistics, the function $F_{\mathcal{G}}$ defines a family of probability distributions on decision vectors $\mathbf{x}$. These distributions are called *exponential family distributions*, and are indexed by real parameters $\theta = (\theta_{\mathbf{a}})_{\mathbf{a} \in A}$ called *exponential parameters* or *canonical parameters*; the exponential family distribution with parameters θ assigns to each solution $\mathbf{x}$ a probability

$$\mathbb{P}(\mathbf{x}) = \frac{1}{Z(\theta)} \exp\Big(\sum_{\mathbf{a} \in A} \theta_{\mathbf{a}} \phi_{\mathbf{a}}(x_{\mathbf{a}})\Big) \tag{2}$$

where

$$Z(\theta) \triangleq \sum_{\mathbf{x}} \exp\Big(\sum_{\mathbf{a} \in A} \theta_{\mathbf{a}} \phi_{\mathbf{a}}(x_{\mathbf{a}})\Big) \tag{3}$$

The quantity $Z(\theta)$ is called the *partition function* of the network $\mathcal{G}$, and $\mathbf{F}(\theta) \triangleq \log(Z(\theta))$ is called the *log-partition function* or *cumulant function*.

An important special case is when $\theta_{\mathbf{a}} = -\beta$ for all $\mathbf{a} \in A$. The resulting distribution is called the *Gibbs distribution* (also known as *Boltzmann distribution*) and the parameter β is then called *inverse temperature*, in analogy with the Boltzmann equation from statistical physics. For a Gibbs distribution, the log-partition function $F(\beta)$ is also called the *free energy* of the system.

Frameworks, examples, and applications

Let us mention a few example of models from different fields which can be cast as special cases of graphical models, and consider the key problems each field focuses on.

As an example, common models in the area of statistical inference are *Bayesian networks* and *Markov Random Fields* (MRF) (see [Lau96, WJ08] for a more in-depth presentation of these topics). A Markov Random Field is a probability distribution on a set of random variables $(X_1, X_2, \ldots, X_n) \in \chi^n$ such that the joint probability distribution takes the form

$$\mathbb{P}(\mathbf{X}) = \frac{1}{Z} \prod_{\mathcal{C} \in \mathrm{cl}(V,E)} \psi_{\mathcal{C}}(x_{\mathcal{C}}) \tag{4}$$

where (V, E) is a graph, $\mathrm{cl}(V, E)$ is the set of cliques (complete subgraphs) of (V, E), and for each clique $\mathcal{C}$, $\psi_{\mathcal{C}}$ is a nonnegative function. Markov Random Fields constitute a generalization of Markov chains, which can be seen as one-dimensional, directed MRFs. In a similar fashion to Markov chains, Markov Random Fields satisfy a collection of conditional independence relations, and can be proven to model any set of consistent independence relations, explaining their power as modeling tools. A Bayesian network is defined on a *directed acyclic graph* (DAG) (V, E) and defines a joint probability distribution

$$\mathbb{P}(\mathbf{X}) = \prod_{v \in V} p(x_v \mid x_{\mathcal{P}(v)}) \tag{5}$$

where V is the set of nodes of the DAG (V, E), and for each V, $\mathcal{P}(v)$ denotes the set of parents of v in V, E. Morever, for any assignment of variables $x_{\mathcal{P}(v)}$, $p(x_v \mid x_{\mathcal{P}(v)})$ is a discrete probability distribution on x_v. Because they are defined on a directed graph, Bayesian networks are extensively used in systems where causality links between variables are of interest. As mentioned, one can easily show [WJ08] that both Markov Random Fields and Bayesian nets can be modeled as graphical models. The main research problems pertaining to MRFs and Bayesian nets are as follows. In many cases, the variables of an MRF or Bayesian net can be divided in three categories: variables which are observed (*observed variables*), variables which are unobserved and which are not of direct interest (*structural variables*), and variables which we would like to predict given the information provided by the observed variables (*target variables*). The key mathematical step then consists in marginalizing the structural variables in the models described by equations (4) and (5),

in order to obtain the conditional distribution of the target variables given the observed variables. Another key object in such a model is the state which achieves the mode of the density, namely, the state which maximizes the a priori likelihood.

Other examples can be found in the field of combinatorial optimization, where the problem of interest can often be described as that of finding subsets of nodes or edges which satisfy various constraints supported by the underlying graphs. Examples include independent sets, matchings, k-SAT, graph coloring problems, and many others. Often, the goal is to identify which of these objects minimizes or maximizes some objective function. The search space is typically exponentially large, and so is the number of local minima, making such an optimization task hard. Again, it can be shown that many such optimization problems can be converted into an optimization problem in a graphical model (see Chapter 4 for the concrete example of Maximum Independent Set). Other problems of interest involve counting the number of solutions which satisfy the constraints, or at least estimating the rate at which this number grows when the size of some structured graph grows as well. Both of these problems can be directly related to the issue of computing the partition function of a graphical model.

Finally, in statistical physics, the Ising model [Tal03] with interaction energy J is described by set of particles $\mathbf{s}$ positioned on a lattice (V, E), each of which can be in one of two spin states ($s_v \in \{-1, +1\}$). In this case, the total energy of the system is given by $F_{\mathcal{G}}(\mathbf{s}) = -\sum_{(u,v)\in E} J s_u s_v$, and the corresponding Boltzmann distribution is

$$\mathbb{P}(\mathbf{s}) = \exp\left(\beta \sum_{(u,v)\in E} J s_u s_v\right) \tag{6}$$

where β is the inverse temperature. An important object of interest in this model is the ground state – a state which achieves the minimum possible energy, i.e., corresponds to the minimum of $F_{\mathcal{G}}(\mathbf{s})$. For $J < 0$, this is equivalent to the problem of finding the so-called max-cut of a graph.

Computational complexity and randomness

Looking at all the examples above, research in graphical models can be seen as trying to address two categories of problems.

The first category includes *counting* and *sampling* problems. Counting involves computing (exactly or approximately) the partition function $Z(\beta)$, or equivalently, the free energy $\mathbf{F}(\beta)$. Sampling involves sampling a set of variables according to the Gibbs distribution for a given β. A related task include computing or sampling from the marginal Gibbs distribution for a given variable (or a small number of variables). In a wide number of frameworks, counting and sampling can be shown to be problems of equivalent computational complexity [JVV86].

The second category regards *optimization*. Here the objective is to identify a vector $\mathbf{x}$ which minimizes or maximizes the energy function $F_{\mathcal{G}}(\mathbf{x})$. Many hard constraints can be modeled by making infeasible configurations have infinite positive (or negative) energy.

In many cases of interest, the combinatorial nature of the problems considered implies that optimization or inference on these models is hard, even when using algorithms that take advantage of the underlying graphical structure [Coo90, Rot96, CSH08]. Exact inference in most graphical models is NP-hard, counting the number of solutions of many combinatorial problems on a graph is ♯P-hard [Jer03], and finding an optimal policy for a Markov Decision Process takes exponential time and space in the dimension of the state-space, even for very simple factored Markov Decision Processes [PT87, BT00]. Thus, approximate methods to find solutions that theoretically or empirically achieve proximity to optimality are needed.

Optimization methods and message-passing schemes

The search for approximate methods typically differs from field to field. In combinatorial optimization the focus has been on developing methods that achieve some provably guaranteed approximation level using a variety of approaches, including linear programming, semi-definite relaxations and purely combinatorial methods [Hoc97]. In the area of graphical models, researchers have been developing new families of inference algorithms, one of the most prominent being *message-passing* algorithms.

At a high level, message-passing schemes function as follows. For each directed edge $e = (u \to v)$ of the graph, a message $\mu_{u \to v}$ is defined, usually a real number, a vector of

real numbers, or a function. Each node of the graph receives messages from its neighbors, combines them in some particular way, and computes new messages that it sends back to its neighbors. The passing of messages in the network is either performed synchronously or asynchronously, and upon convergence (if the scheme converges), all incoming messages to a node u are combined in order to compute either a decision x_u or a marginal distribution $\mathbb{P}(X_u)$.

One of the most studied message-passing algorithms is the *Belief Propagation* (BP) algorithm, [Lau96, Jor04, YFW00]. The BP algorithm is designed both for solving the problem of finding the optimal state using the max-product version, as well as for the problem of computing the partition function, using the sum-product version. The BP algorithm is known to find an optimal solution x^* when the underlying graph is a tree, but may fail to converge, let alone produce optimal solutions, when the underlying graph contains cycles. Despite this fact, it often has excellent empirical performance [FM98, YMW06, WYM07]. Moreover, it is a distributed algorithm and easy to implement. This justifies the wide applicability of BP in practice and the intense focus on it by the researchers in the signal processing and artificial intelligence communities. Nevertheless, a major research effort has been devoted to developing corrected version of Belief Propagation, and to understanding the performance of message-passing schemes.

This thesis focuses on developing a new message-passing style algorithm, the *cavity expansion* algorithm, and to understand and study its performance in the context of large-scale graphical models, with emphasis on the question of how randomness affects the complexity of optimizing in a graph. In particular, we put ourselves in a framework where the potential and edge functions are *randomly generated*, and try to understand under which conditions a problem is computationally hard or easy. These conditions typically relate to the structure of the graph considered, along with the distribution of the cost functions. The connections which have been uncovered between statistical physics and optimization can be of much help in this respect. Of particular interest is the study of a statistical physics phenomenon known as the *correlation decay* property. At a high level, correlation decay indicates a situation in which the "influence" of a node on another node of the network decreases quickly as the distance between them grows. We show that, in many cases, the onset of correlation decay often implies that the optimization problem becomes easy on average.

Organization of the thesis and contributions

Chapter 1: Message-passing schemes and Belief Propagation

In the first chapter, we present our general optimization framework, and give a short introduction to message-passing algorithms, Belief Propagation, and some of the most prominent message-passing variations that were designed to address issues of correctness or convergence of BP.

Chapter 2: The Cavity Expansion algorithm

In the second chapter, we propose a new message-passing-like algorithm for the problem of finding $x^* \in \text{argmax}\, F_{\mathcal{G}}(x)$, which we call the *Cavity Expansion* (CE) algorithm. Our algorithm draws upon several recent ideas, and relies on a technique used in recent deterministic approximate counting algorithms. It was recently recently shown in [Wei06] and [BG06] that a counting problem on a general graph can be reduced to a counting problem on a related self-avoiding exponential size tree. Following a generalization of this technique later developed in [GK07b, BGK$^+$07], we extend the approach to general optimization problems. We do not explicitly use the self-avoiding tree construction, and opt instead for a simpler notion of recursive cavity approximation. The description of the CE algorithm begins by introducing a *cavity* $B_v(x)$ for each node/decision pair (v, x). $B_v(x)$ is defined as the difference between the optimal reward for the entire network when the action in v is x versus the optimal reward when the action in the same node is 0. It is easily shown that knowing $B_v(x)$ is equivalent to solving the original decision problem.

Our main contribution is to obtain a recursion expressing the cavity $B_v(x)$ in terms of cavities of the neighbors of v in suitably modified sub-networks of the underlying network.

From this recursion, we develop the CE algorithm, which proceeds by expanding this recursion in the breadth-first search manner for some designed number of steps t, thus constructing an associated computation tree with depth t. We analyze the computational effort and prove it is exponential in t. We therefore need conditions which guarantee that using the cavity recursion for small t results in near-optimal decisions, which is the object of the following chapters.

Chapter 3: Correlation decay and efficient decentralized optimization in decision networks with random objective functions

In the third chapter, we investigate the connection between a property of random systems, called the *correlation decay* property, and the existence of polynomial-time, decentralized algorithms for optimization in graphical models with discrete variables and random cost functions.

A key insight of this thesis is that in many cases, the dependence of the cavity $B_v(x)$ on cavities associated with other nodes in the computation tree dies out exponentially fast as a function of the distance between the nodes. This phenomenon is generally called *correlation decay* and was studied for regular, locally tree-like graphs in [GNS06].

It is then reasonable to expect that the Cavity Expansion algorithm and the correlation decay analysis can be merged in some way. Namely, optimization problems with general graphs and random costs can be solved approximately by constructing a computation tree and proving the correlation decay property on it. This is precisely our approach: we show that the correlation decay property is a sufficient condition which guarantees the near optimality of the CE algorithm. Thus, the main associated technical goal is establishing the correlation decay property for the associated computation tree.

We indeed establish that the correlation decay property holds for several classes of decision networks associated with random reward functions $\Phi =(\phi_v, \phi_{v,u})$. We provide a general technique to compute conditions on the parameters of families of distribution that ensure that the system exhibits the correlation decay property. We illustrate the applicability of the method by giving concrete results for the cases of uniform and Gaussian distributed functions in networks with bounded connectivity (i.e., bounded graph degree).

Chapter 4: Connections between correlation decay and computational hardness of probabilistic combinatorial optimization

In the fourth chapter, we look at similar questions for a specific combinatorial optimization problem, namely, the *Maximum Weight Independent Set* (MWIS). We show how the CE algorithm applies to the MWIS problem and provides conditions under which the CE algorithm finds an approximately optimal solution. The application of CE in a randomized setting has a particularly interesting implication for the theory of average case analysis of combinatorial optimization. Unlike some other NP-complete problems, finding a MWIS of a graph does not admit a constant factor approximation algorithm for general graphs.

We show that when the graph has maximum degree 3 and when the nodes are weighted independently with exponentially distributed weights, the problem of finding the maximum weighted independent set admits a polynomial time algorithm for approximating the optimal solution within $1 + \epsilon$ for every constant $\epsilon > 0$. We also provide a generalization for higher degrees, and detail a framework for analyzing the correlation decay property for arbitrary distributions, via a phase-type distribution approximation. Thus, surprisingly, introducing random weights translates a combinatorially intractable problem into a tractable one. We note that the sufficient condition pertains only to the weight distribution and the degree of the graph. As such, CE does not suffer from the loopiness of the graph being considered, a very uncommon feature for a message-passing style algorithm. We also provide partial converse results, showing that even under a random cost assumption, it is NP-hard to compute the MWIS of a graph with sufficiently large degree.

Chapter 5: Correlation Decay in graphical games

Graphical games [KLS01a] are a natural extension of the discrete optimization graphical models of Chapter 1, where each agent is assigned her own family of cost functions which she tries to optimize while taking into account other agents' potentially conflicting objectives. It is well known that computing the Nash equilibrium of a game is hard, (see Daskalakis *et al.* [DGP09]), even when considering sparse networks. These facts arguably make NE an unlikely explanation for people's or markets' behaviors. Thus, there has thus been an interest in adapting message-passing algorithms to the computation of Nash Equilibria, under the reasoning that simple distributed schemes might better represent social computation. In [KLS01a], and then [OK03], Kearns *et al.* develop Nash Propagation, an analog of Belief Propagation for the setting of graphical games. Like BP, NP is optimal or near-optimal for tree-structured games.

Our objective is two-fold. First, we develop a general framework for designing message-passing algorithms for graphical games. These message-passing algorithms aim to compute so-called *Nash cavity* functions, which are local constraints encoding as much of the global Nash equilibrium constraints as possible. With the help of this framework, we develop the *Nash Cavity Algorithm*, a general message-passing heuristic which aims to try to compute Nash cavity functions for general graphical games. In particular, we show that TreeProp is a special case of the Nash Cavity algorithm for graphical games on tree.

Second, we appropriately define the correlation decay property for particular graphical

games on trees, and show that, under appropriate conditions, the Nash cavity functions of games exhibiting the correlation decay property can be computed locally.

Chapter 6: Application of graphical models and message-passing techniques to the early diagnosis of Alzheimer's disease

In the last chapter, we consider a particular application of graphical models and message-passing algorithms. The problem in question is a statistical signal processing problem, specifically, measuring the similarity of a collection of N point processes in $\mathbb{R}^M$. The work is motivated by the following application: developing a new measure of synchronicity between different parts of the brain, as recorded by electroencephalogram electrodes, and using said measure to give an early prediction of Alzheimer's disease. We show that the resulting measure of synchronicity outperforms a vast number of other EEG-based measures in the task of predicting the onset of Alzheimer's disease.

Chapter 1

Message-passing schemes and Belief Propagation

1.1 Introduction and literature review

Foundations of message-passing algorithms

In this chapter, we present the Belief Propagation (BP) algorithm, arguably the first, simplest, and most commonly used form of message-passing algorithms. Introduced by Pearl in the context of inference in probabilistic AI [Pea82, PS88, Pea00], the *sum-product* algorithm enabled distributed computation of marginal probabilities in belief networks, and its success encouraged the shift from classical to probabilistic AI. BP was later extended to *max-product* (*min-sum* in the log domain), a version of BP which computes the mode of the distribution underlying a Bayesian Network. Belief Propagation was then proven to output the correct solution when the graph is a tree (or a forest), and initially, most research effort in message-passing for AI focused on algebraic, exact generalizations of BP to graphs with cycles [Lau96].

Eventually, it was discovered that Belief Propagation, even when applied to graphs with cycles (the idea is referred to as "loopy Belief Propagation"), often had excellent empirical performance [FM98]. This surprising discovery fostered much research activity in the domain of message-passing algorithms, leading to the development of several new distributed computation techniques, and revealing connections between classical optimization (most prominently convex optimization and linear programming relaxations) and message-passing

methods [BBCZ07, WYM07, YMW06, WJ08, SSW08].

Much earlier, Gallager, in his 1960 PhD thesis [Gal60, Gal63], invented a new class of error-correcting techniques called *Low Density Parity Check* (LDPC) codes, along with an iterative algorithm to perform decoding of LDPC codes. This early algorithm was later found to be an early version of Belief Propagation. Similar iterative algorithms performing on graphs were later investigated by researchers in coding theory, in particular Forney [FJ70], Tanner [Tan81], Battail [Bat89], Hagenauer and Hoeher [HH89], and finally Berrou and Glavieux [BGT93], whose BP-like turbocodes nearly attained the Shannon capacity. The performance of turbocodes sparked great interest in message-passing algorithms in the coding theory community, see for instance later work by Kschischang, Frey, Loeliger, Vontobel, Richardson, Urbanke, and many others [DMU04, LETB04, VK05, LDH+07, MU07, RU08].

Finally, a new class of models for magnetized particles with frustrated interations, *spin glasses*, generated a lot of interest in the statistical physics community in the late 1980s, especially after Parisi solved a particular Ising model proposed by Sherrington and Kirkpatrick [SK75] a decade before. Parisi's technique (see [MPV87]), the *cavity method*, bore a lot of similarities to Belief Propagation, and was found to have connections to combinatorial optimization problems such as k-SAT or graph coloring. This result was one example of the convergence of interests between statistical physics, mathematics (combinatorics especially), computer science (computational complexity) and AI.

On the one hand, physicists started to study combinatorial optimization problem in order to both understand better the relations between computational hardness and randomness (in particular, through the study of phase transitions), and to develop stronger algorithms for solving constraint satisfaction problems. Some of the algorithms developed, such as survey propagation, proved to solve very efficiently large instances of hard problems (see for instance [BMZ05]).

On the other hand, mathematicians investigated, formalized, and made rigorous techniques and problems from statistical physics. Of particular interest is the solution of the $\zeta(2)$ Parisi conjecture for the random minimal assignment problem (see [Ald92, Ald01, AS03]). One of the key ideas was the study of fixed points of recursive distributional equations (RDE) (see Chapter 4). Again, the existence and convergence to a fixed point of a RDE can also be understood in terms of asymptotic convergence of the Belief Propagation algorithm in an infinite random graph.

Research on graphical models, message-passing algorithms, and Belief Propagation in

particular, can therefore truly be seen to be at the intersection of many different fields: AI, coding and information theory, statistical physics, probabilistic combinatorics, and computational complexity (see [WJ08] for an overview of inference techniques in graphical models and [HW05, MM08] for comprehensive studies of the relations between statistical physics, statistical inference, and combinatorial optimization).

Each of these brings a different point of view on the mechanics and performance of Belief Propagation, and, based on those particular insights, offers particular generalizations or corrections of Belief Propagation.

Modern work on Belief Propagation

The empirical success of Belief Propagation, despite the fact that BP is a nonexact recursion, prompts the following two questions. First, can one identify problems and conditions under which BP is provably optimal, and second, can one design a "corrected" version of BP, which will achieve greater theoretical and practical performance, specifically for the cases where BP is proven not to work?

Regarding the first question, researchers have recently identified a number of frameworks in which BP converges to the optimal solution, even if the underlying graph is not a tree. In a framework similar to ours, Moallemi and Van Roy [MR09] show that BP converges and produces an optimal solution when the action space is continuous and the cost functions are quadratic, convex. More generally, when these functions are simply convex, the authors exhibit in [MR07] a sufficient condition (more specifically, a certain diagonal dominance condition) for the convergence and optimality of BP. Other cases where BP produces optimal solutions include Maximum Weighted Matchings [San07, BBCZ08, BSS08], Maximum Weighted Independent Sets if the LP relaxation is tight [SSW08], network flows [GSW09], and, more generally, optimization problems with totally unimodular constraint matrices [Che08]. Furthermore, in the case of Gaussian Markov Random Fields, sufficient conditions for convergence and correctness of Belief Propagation were studied in [RR01, CJW08, JMW06, MJW06]. Finally, a number of researchers have investigated sufficient conditions for BP to converge (to potentially suboptimal solutions), and then tried to quantify the resulting error of the solution obtained; see for instance [Wei00, TJ02, MK05, IFW06].

Regarding the second question, over the last few years, many corrected or improved versions of BP have been proposed, most notably the junction tree algorithm [Lau96], survey propagation [MMW07], Kikuchi approximation-based BP and generalized Belief Propagation (GBP) [YFW00], tree-reweighted Belief Propagation [WJW03b, WJW05a, KW05, Kol06], loop-corrected Belief Propagation [MWKR07], loop calculus [CC06a, CC06b], and dual LP-based Belief Propagation algorithms [SMG+08b, SGJ08, SJ09]. Each of these algorithms differs in its conditions for convergence or optimality, running time, or the type of bounds provided on the free energy of the system considered.

In this chapter, we present the Belief Propagation algorithm (optimization version), along with some of the most prominent message-passing algorithms which aim to correct BP for the problems in which it performs poorly.

1.2 Message-passing schemes and framework

Let us for convenience restate the model we previously introduced. We consider a pairwise decision network $\mathcal{G} = (V, E, \Phi, \chi)$. Here (V, E) is an undirected graph without repeated edges, in which each node $u \in V$ represents an agent, and edges $e \in E$ represent a possible interaction between two agents. Each agent makes a decision $x_u \in \chi \triangleq \{0, 1, \ldots, T-1\}$. For every $v \in V$, a function $\phi_v : \chi \to \mathbb{R}$ is given. Also for every edge $e = (u, v)$ a function $\phi_e : \chi^2 \to \mathbb{R}$ is given. Functions ϕ_v and ϕ_e will be called *potential functions* and *interaction functions* respectively. Note that in general, we don't require the presence of potential functions, as these can be absorbed into the interaction functions as follows: for any potential function ϕ_u corresponding to a node $u \in V$, choose an arbitrary neighbor v of u, and update ϕ_u and $\phi_{u,v}$ into ϕ'_u and $\phi'_{u,v}$ as follows: $\phi'_u(x) = 0$ for all x, and $\phi'_{u,v}(x_u, x_v) = \phi_{u,v}(x_u, x_v) + \phi_u(x_u)$ for all x_u, x_v. Let $\Phi = ((\phi_v)_{v \in V}, (\phi_e)_{e \in E})$. The object $\mathcal{G} = (V, E, \Phi, \chi)$ will be called a *pairwise decision network*, or *pairwise graphical model*. A vector $\mathbf{x} = (x_1, x_2, \ldots, x_{|V|})$ of actions is called a solution for the decision network. The value of solution $\mathbf{x}$ is defined to be $F_{\mathcal{G}}(\mathbf{x}) = \sum_{(u,v) \in E} \phi_{u,v}(x_u, x_v) + \sum_v \phi_v(x_v)$. The quantity $J_{\mathcal{G}} \triangleq \max_{\mathbf{x}} F_{\mathcal{G}}(\mathbf{x})$ is called the (optimal) value of the network $\mathcal{G}$. A decision $\mathbf{x}$ is optimal if $F_{\mathcal{G}}(\mathbf{x}) = J_{\mathcal{G}}$. Our objective is to compute an optimal (or near-optimal) solution for the network:

Problem (Maximization in decision network)**.**

Given a network $\mathcal{G} = (V, E, (\phi_v)_{v \in V}, (\phi_{u,v})_{(u,v) \in E}), \chi)$, *find* $\mathbf{x}^* \in \chi^V$ *such that*

$$\mathbf{x}^* \in \text{argmax}_{\mathbf{x}} \left(\sum_v \phi_v(x_v) + \sum_{u,v} \phi_{u,v}(x_u, x_v) \right)$$

Message-passing schemes are a simple, naturally distributed, and modular class of algorithms for performing optimization in graphical models. They function as follows: we define a vector of messages $\boldsymbol{M} \in S^{\vec{E}}$, where S is the space of messages (often, $S = \mathbb{R}$). Given some understanding of the optimization problem at hand, we design for each oriented edge $e = (u \to v)$ a function $F_{u \to v} : S^{|\mathcal{N}_u|} \to S$, and we iteratively update the vector $\mathbf{M}$ by the following operation:

$$\forall (u \to v) \in E, \quad M_{u \to v} = F_{u \to v}(\mathbf{M}_{\mathcal{N}_u \to u}) \tag{1.1}$$

In other words, messages outgoing from u are functions of message incoming to u (see Fig. 1.2). Assuming the scheme converges, we set the variable x_u to $g_u(\boldsymbol{M}_{\mathcal{N}_u \to u})$ for some carefully chosen function g_u: the decision x_u of u is a function of messages incoming to u in steady state.

Because of the recursions we consider have their root in dynamic programming, in the following, it will be more natural for us to denote $\mu_{u \to v}$, the message sent from u to v, by $\mu_{v \leftarrow u}$, message received by v from u (these two notions being identical in our context).

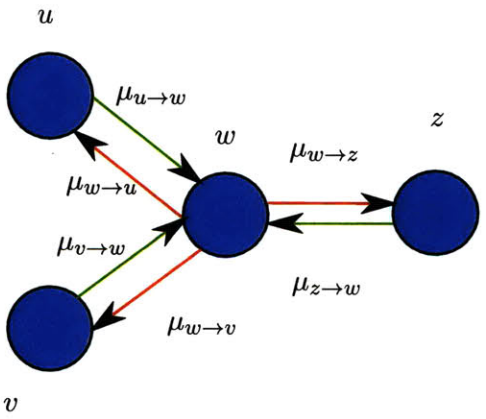


Figure 1-1: A message-passing scheme

1.3 The Belief Propagation algorithm

In this section, we derive the Belief Propagation equations by proceeding in two steps. First, we develop a recursion where variables are located on the nodes of our network $\mathcal{G}$. This recursion is natural and based on a simple dynamic optimization principle, but is not easily parallelizable, nor easily applied to general (non tree) graphs. In the second step, we show how the dynamic optimization equations can be converted into a new set of recursive equations, where this time variables dwell on the edges of the graph. This new set of equations constituting the Belief Propagation equations, are naturally parallelizable, and are easily applied to arbitrary graphs (albeit non optimally). We begin by introducing useful notations.

Given a subset of nodes $\mathbf{v} = (v_1, \ldots, v_k)$, and $\mathbf{x} = (x_1, \ldots, x_k) \in \chi^k$, let $J_{\mathcal{G},\mathbf{v}}(\mathbf{x})$ be the optimal value when the actions of nodes $v_1, \ldots, v_k$ are fixed to be $x_1, \ldots, x_k$ respectively: $J_{\mathcal{G},\mathbf{v}}(\mathbf{x}) = \max_{\mathbf{x}: x_{v_i} = x_i, 1 \leq i \leq k} F_{\mathcal{G}}(\mathbf{x})$. Given $v \in V$ and $x \in \chi$, the quantity $B_{\mathcal{G},v}(x) \stackrel{\Delta}{=} J_{\mathcal{G},v}(x) - J_{\mathcal{G},v}(0)$ is called the *cavity* of action x at node v. Namely it is the difference of optimal values when the decision at node v is set to x and 0 respectively (the choice of 0 is arbitrary). The cavity function of v is $B_{\mathcal{G},v} = (B_{\mathcal{G},v}(x))_{x \in \chi}$. Since $B_{\mathcal{G},v}(0) = 0$, $B_{\mathcal{G},v}$ can be thought of as element of $\mathbb{R}^{T-1}$. In the important special case $\chi = \{0, 1\}$, the cavity function is a scalar $B_{\mathcal{G},v} = J_{\mathcal{G},v}(1) - J_{\mathcal{G},v}(0)$. In this case, if $B_{\mathcal{G},v} > 0$ (resp. $B_{\mathcal{G},v} < 0$) then $J_{\mathcal{G},v}(1) > J_{\mathcal{G},v}(0)$ and action 1 (resp. action 0) is optimal for v. When $B_{\mathcal{G},v} = 0$ there are optimal decisions consistent both with $x_v = 0$ and $x_v = 1$. When $\mathcal{G}$ is obvious from the context, it will be omitted from the notation.

Vertex-based dynamic optimization

Given a decision network $\mathcal{G} = (V, E, \Phi, \chi)$ suppose that (V, E) is a tree $\mathcal{T}$, and arbitrarily root the tree at a given node u. Using the graph orientation induced by the choice of u as a root (i.e., children of a node v are further from u than v is), let $\mathcal{K}_u(v)$ denote the set of children of any node v in (V, E), and let $\mathcal{T}_u(v)$ be defined as the subtree rooted in node v. In particular, $\mathcal{G} = \mathcal{T}_u(u)$. Given any two neighbors $v, w \in V$, and an arbitrary vector $B = (B(x), x \in \chi)$, define

$$\mu_{v \leftarrow w}(x, B) = \max_y(\phi_{v,w}(x, y) + B(y)) - \max_y(\phi_{v,w}(0, y) + B(y)) \tag{1.2}$$

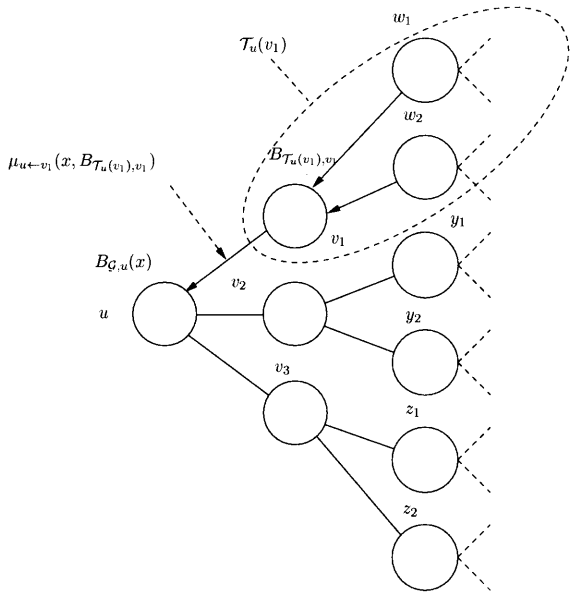


Figure 1-2: Dynamic optimization recursion on a tree

for every action $x \in \chi$. μ is called the partial cavity function.

Proposition 1 (Cavity recursion for trees). *For every $v \in V$ and $x \in \chi$,*

$$B_{\mathcal{T}_u(v),v}(x) = \phi_v(x) - \phi_v(0) + \sum_{w \in \mathcal{K}_u(v)} \mu_{v\leftarrow w}(x, B_{\mathcal{T}_u(w),w}) \tag{1.3}$$

Proof. Suppose $\mathcal{K}_u(v) = \{w_1, \ldots, w_d\}$. Observe that the subtrees $\mathcal{T}_u(w_i), 1 \le i \le d$, are disconnected (see Fig. 1.3). Thus,

$$\begin{aligned}
B_{\mathcal{T}_u(v),v}(x) &= \phi_v(x) + \max_{x_1,\ldots,x_d} \Big\{ \sum_{j=1}^{d} \phi_{v,w_j}(x, x_j) + J_{\mathcal{T}_u(w_j),w_j}(x_j) \Big\} \\
&\quad - \phi_v(0) - \max_{x_1,\ldots,x_d} \Big\{ \sum_{j=1}^{d} \phi_{v,w_j}(0, x_j) + J_{\mathcal{T}_u(w_j),w_j}(x_j)) \Big\} \\
&= \phi_v(x) - \phi_v(0) \\
&\quad + \sum_{j=1}^{d} \Big\{ \max_y \left(\phi_{v,w_j}(x, y) + J_{\mathcal{T}_u(w_j),w_j}(y)\right) - \max_y \left(\phi_{v,w_j}(0, y) + J_{\mathcal{T}_u(w_j),w_j}(y)\right) \Big\}
\end{aligned}$$

For every j,

$$\max_y \left(\phi_{v,w_j}(x,y) + J_{\mathcal{T}_u(w_j),w_j}(y)\right) - \max_y \left(\phi_{v,w_j}(0,y) + J_{\mathcal{T}_u(w_j),w_j}(y)\right) =$$
$$\max_y \left(\phi_{v,w_j}(x,y) + J_{\mathcal{T}_u(w_j),w_j}(y) - J_{\mathcal{T}_u(w_j),w_j}(0)\right) - \max_y \left(\phi_{u,v_j}(0,y) + J_{\mathcal{T}_u(w_j),w_j}(y) - J_{\mathcal{T}_u(w_j),w_j}(0)\right)$$

The quantity above is exactly $\mu_{v\leftarrow w_j}(x, B_{\mathcal{T}_u(w_j),w_j})$. □

By analogy with the algorithm we develop in the next chapter, Equation 1.3 will be called the *cavity recursion* for trees. It is based on nonserial dynamic optimization [BB72], which computes value functions of subtrees of the original graph. In the equations above, the variables of interest are cavities, and they are computed at the nodes of the graph.

Edge-based Belief Propagation

We now transform equation 1.3 into an equivalent system of equations where the variables are computed on oriented edges of the graph. Recall that since $\mathcal{G}$ is a tree, for any two nodes v, w, removing the edge (v, w) from the graph (V, E) separates it into two trees, one containing v, the other containing w. We denote the one which contains v (resp. w) $\mathcal{T}_{w\leftarrow v}$ (resp. $\mathcal{T}_{v\leftarrow w}$). For any edge (v, w), let $\mathcal{G}_{v\leftarrow w}$ be the network induced by $\mathcal{T}_{v\leftarrow w} \cup \{v, w\}$, with the additional modification that the potential ϕ_v is removed from that network. Finally, let $M_{v\leftarrow w}(x_v)$ denote $B_{\mathcal{G}_{v\leftarrow w},v}(x_v)$. Since v has a unique neighbor w in $\mathcal{G}_{v\leftarrow w}$, it is easy to check that we have

$$M_{v\leftarrow w}(x_v) = \mu_{v\leftarrow w}(x_v, B_{\mathcal{T}_{v\leftarrow w},v}) \tag{1.4}$$

Proposition 2 (Belief Propagation). *For all $u \in \mathcal{G}$,*

$$B_{\mathcal{G},u}(x_u) = \phi_u(x_u) - \phi_u(0) + \sum_{v\in\mathcal{N}(u)} M_{u\leftarrow v}(x_u) \tag{1.5}$$

For all $(u, v) \in E$,

$$M_{u\leftarrow v}(x_u) = \mu_{u\leftarrow v}\left(x_u, \sum_{w\in\mathcal{N}(v)\setminus\{u\}} \left(\phi_v + M_{v\leftarrow w_i}\right)\right) \tag{1.6}$$

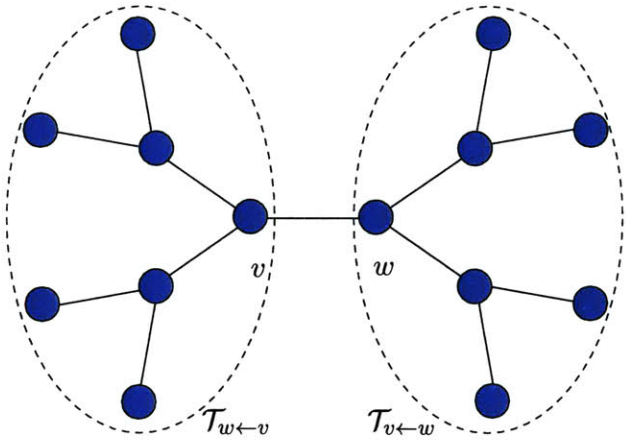


Figure 1-3: Tree splitting
Tree $\mathcal{T}$ split into $\mathcal{T}_{v\leftarrow w}$ and $\mathcal{T}_{w\leftarrow v}$

Proof. For the first equation, consider Equation (1.3) for $v = u$, and note that $\mathcal{K}_u(u) = \mathcal{N}(u)$, and as noted previously, $\mathcal{T}_u(u) = \mathcal{G}$. We obtain

$$B_{\mathcal{G},u}(x_u) = \phi_u(x) - \phi_u(0) + \sum_{v\in\mathcal{N}(u)} \mu_{u\leftarrow v}(x, B_{\mathcal{T}_u(v),v})$$

Since clearly $\mathcal{T}_{u\leftarrow v} = \mathcal{T}_u(v)$, Equation (1.4) implies that $B_{\mathcal{G},u}(x_u) = \phi_u(x) - \phi_u(0) + \sum_{v\in\mathcal{N}(u)} M_{u\leftarrow v}(x_u)$, and thus finishes the proof of the first equation. The second equation follows from the exact same principles and the observation that for a node v with neighbors $\{u, w_1, \ldots, w_d\}$, $\mathcal{T}_u(w_i)$ is equal to $\mathcal{T}_{v\leftarrow w_i}$ for all i. □

Equations (1.5) and (1.6) are called *belief propagation equations*, and make the practicality of the BP algorithm apparent. Indeed, Equation (1.6) can be seen to be iterative in nature, since it writes the set of all messages $(M_{u\leftarrow v})_{(u\leftarrow v)\in\vec{E}}$ as a function of itself. This naturally suggests the following iterative scheme to compute the $M_{u\leftarrow v}$: for any $(u \leftarrow v) \in \vec{E}$, $x_u \in \chi$, and $r > 0$

$$M^r_{u\leftarrow v}(x_u) = \mu_{u\leftarrow v}\left(x_u, \sum_{w\in\mathcal{N}(v)\setminus\{u\}} M^{r-1}_{v\leftarrow w_i}(x_v)\right) \tag{1.7}$$

where the values $M^0_{u\leftarrow v}$ are initialized to arbitrary values. While it is not necessarily obvious that this scheme converges for trees, it can easily be shown by induction that for r greater

than the depth of tree, the messages $M^r_{u\leftarrow v}$ are in fact stationary and equal to their correct values $M_{u\leftarrow v}$. This subsequently allows the use of Equation (1.5) to compute the cavities, and therefore the optimal solution. In that sense, Proposition 2 is the restatement of the well-known fact that BP finds an optimal solution on a tree [MM08]. More importantly, unlike Equation (1.3), the algebraic structure of Equation (1.7) does not require that the graph be a tree, and can therefore be applied to any general graph. The resulting algorithm is called *loopy Belief Propagation.* As mentioned previously, it may now not converge, and even if it does, plugging the obtained messages into Equation (1.5) may result in an arbitrarily poor solution.

Generalization to factor graphs

We now generalize the Belief Propagation equations to hypergraphs. As for the pairwise case, we will first derive dynamic optimization equations, and then rewrite them as message-passing equations in a factor graph. Consider a factor graph (V, A, E, Φ, χ), for which the underlying network (V, A, E) is a tree $\mathcal{T}$, and arbitrarily root the tree at a given node $u \in V$. Again, denote $\mathcal{T}_u(v)$ the subtree rooted at $v \in V \cup A$ when using the orientation induced by u as root, and $\mathcal{K}_u(v)$ the children of v. Note that for any $v \in V$, $\mathcal{K}_u(v) \subset A$, and for any $a \in A$, $\mathcal{K}_u(a) \subset V$. Finally, consider any $v \in V$ and $a \in \mathcal{K}_u(v)$, and denote $k_a = \mathcal{K}_u(a)$ the number of children of a; for any $x_v \in \chi$, and an arbitrary function M from χ^{k_a} to $\mathbb{R}$, define the partial cavity function (for factor graphs) $\mu_{u\leftarrow a}$ as

$$\begin{aligned}\mu_{v\leftarrow a}(x, B) = &\max_{y_i \in \chi} \left(\phi_a(x, y_1, y_2, \ldots, y_{k_a}) + M(y_1, \ldots, y_{k_a})\right) \\ &- \max_{y_i \in \chi} \left(\phi_a(0, y_1, y_2, \ldots, y_{k_a}) + M(y_1, \ldots, y_{k_a})\right)\end{aligned} \tag{1.8}$$

The analog of the recursion (1.3) for factor graphs is as follows (the proof is essentially identical to that of Proposition 1):

Proposition 3. *For every $v \in V$ and $x \in \chi$,*

$$B_{\mathcal{T}_u(v),v}(x) = \phi_v(x) - \phi_v(0) + \sum_{a \in \mathcal{K}_u(v)} \mu_{v\leftarrow a}(x, \sum_{w \in \mathcal{K}_u(a)} B_{\mathcal{T}_u(w),w}) \tag{1.9}$$

We now proceed to convert Equation (1.9) to a set of recursive equations on messages in the factor graphs. Once again, we use similar notations to the bipartite case: for any two neighbors $v \in V, a \in A$, removing the edge (v, a) separates $\mathcal{T}$ into two trees. $\mathcal{T}_{a\leftarrow v}$

will denote the one containing v, and $\mathcal{T}_{v\leftarrow a}$ the one containing a. Let $\mathcal{G}_{a\leftarrow u}$ be the network induced by $\mathcal{T}_{a\leftarrow u}$, and $\mathcal{G}_{u\leftarrow a} \cup (u,a)$ be the one induced by $\mathcal{T}_{u\leftarrow a}$, with again potential ϕ_u removed from that network. Note that (u,a) is not included in $\mathcal{G}_{u\leftarrow a}$). Finally, let $M_{v\leftarrow a}(x_v)$ be $B_{\mathcal{G}_{v\leftarrow a},v}(x_v)$ and $M_{a\leftarrow u}(x_v)$ be $B_{\mathcal{G}_{a\leftarrow v},v}(x_v)$. The Belief Propagation algorithm for factor graphs is given by the following proposition:

Proposition 4. *For all* $u \in \mathcal{G}$

$$B_{\mathcal{G},u}(x_u) = \phi_u(x) - \phi_u(0) + \sum_{a\in\mathcal{N}(u)} M_{u\leftarrow a}(x_u) \tag{1.10}$$

For all $(u,a) \in E$,

$$M_{u\leftarrow a}(x_u) = \mu_{u\leftarrow a}\left(x_u, (M_{a\leftarrow w})_{w\in\mathcal{N}(a)\setminus\{u\}}\right) \tag{1.11}$$

and

$$M_{a\leftarrow u}(x_u) = \phi_u(x_u) + \sum_{a'\in\mathcal{N}(u)\setminus\{a\}} M_{u\leftarrow a'}(x_u) \tag{1.12}$$

1.4 Variations of Belief Propagation

In this section, we present just a few generalizations of the BP algorithm, drawing our examples from a variety of different fields, and intending to illustrate the fact that Belief Propagation, as a technique developed independently in different research areas, may be understood from many different points of view. Each of these points of view provides some understanding of why Belief Propagation may fail to give an optimal solution, and suggest an appropriate modification to help improve the message-passing algorithm.

Junction tree algorithm

The junction tree algorithm is probably the oldest improvement to the BP algorithm, and actually predates the use of loopy BP on general graphical models. Its development stemmed from the study of Belief Propagation as an algebraic operation aimed at operating a marginalization of variables of a probability distribution factored through a graphical structure. Using tools of graph theory, Pearl [Pea00] proposed a more complex algorithm for computing marginals of the random variables defined through a graphical model (the

optimization version of the junction tree algorithm was only later developed, see [Daw92]). At a high level, the junction tree algorithm consists of converting our graphical model $\mathcal{G} = (V, A, E, \Phi, \chi)$ into a new graphical model $\mathcal{G}' = (V', A', E', \Phi', \chi')$ with the following properties:

- For each $v' \in V'$, v' is a subset of V
- For each $v' \in V'$, the set of decisions for x'_v is the product space $\chi^{v'}$
- (V', A', E') is a tree for which each $a' \in A'$ has at most two neighbors v'_1 and v'_2.
- For any $a' \in A'$ with neighbors (v'_1, v'_2), $\phi_{a'}$ is a function which depends only on the variables in $v'_1 \cap v'_2$. Any node in $v'_1 \cap v'_2$ will be said to belong to a'.
- For any two v', w' in V', and any factor a' on the unique path between v' and w', $v' \cap w'$ must be a subset of a'.
- $\mathcal{G}$ and $\mathcal{G}'$ have the same optimal value function

Such a $\mathcal{G}'$ will be called a *junction tree*; the proof of its existence relies on tools in graph theory. The *junction tree algorithm* simply consists in running Belief Propagation on the junction tree. The complexity of doing so is exponential in the size of the largest subset v' (since the corresponding decision space for that node is $\chi^{v'}$). There may exist many junction trees $\mathcal{G}'$ for a given graphical model $\mathcal{G}$; the minimum of $\max_{v'} |v'|$ over all such transformations is called the treewidth of the hypergraph. However, computing the treewidth of a graph is in general a hard problem [Bod06], and for a large number of graphs, the treewidth grows linearly with the size of the graph, limiting the applicability of the junction tree to fairly simple structures.

Variational inference and convex relaxations

The next class of algorithms we consider are message-passing algorithms derived from linear and convex relaxations of the optimization problems we consider. Consider a pairwise graphical model $\mathcal{G} = (V, E, \Phi, \chi)$, and for simplicity assume there are only interaction functions $\phi_{u,v}$. For any family $\theta = (\theta_{u,v})_{(u,v)\in E} \in \mathbb{R}$ of nonnegative real numbers, we define a new graphical model $\mathcal{G}(\theta) = (V, E, \Phi \cdot \theta, \chi)$, such that for any $(u, v) \in E$, the interaction function for edge (u, v) in $\mathcal{G}(\theta)$ is equal to $\theta_{u,v}\phi_{u,v}$. Since $J_{\mathcal{G}(\theta)} =$

$\max_{\mathbf{x}\in\mathcal{X}^V}(\sum_{u,v}\theta_{u,v}\phi_{u,v}(x_u,x_v))$ is the maximum of linear function of θ, it is therefore convex in θ. Consider a set $(\theta^1,\theta^2,\ldots,\theta^k)$ of θ functions, along with a probability distribution ρ on the elements of $\{1,2,\ldots,k\}$). Suppose that

$$\forall(u,v)\in E,\quad \sum_i \rho(i)\,\theta^i_{u,v}=1$$

In other words, for each u,v, we have $\mathbb{E}_\rho[\theta_{u,v}]=1_{(u,v)\in E}$. By applying Jensen's inequality, we obtain that

$$J_{\mathcal{G}}=J_{\mathcal{G}(\mathbb{E}_\rho[\theta])}\leq\sum_i \rho_i\,J_{\mathcal{G}(\theta^i)} \tag{1.13}$$

Therefore, if for each i, $J_{\mathcal{G}(\theta^i)}$ is easy to compute, we readily obtain an upper bound for the value function of $\mathcal{G}$. One possible idea is to consider a distribution ρ on spanning trees of E, and choose each θ^i such that $\mathcal{G}(\theta^i)$ is a tree, in which case each $J_{\mathcal{G}(\theta^i)}$ is easy to compute through Belief Propagation. This idea is the basis of *tree-reweighted belief propagation* (TRBP), introduced by Wainwright, Jaakkola and Willsky in [WJW03a], and later extended in [KW05, WJW05b, Kol06]. For instance, for any distribution ρ on a collection of trees $(\mathcal{T}_1,\mathcal{T}_2,\ldots,\mathcal{T}_k)$, denote $\rho_{u,v}=\sum_i\rho(i)1_{(u,v)\in\mathcal{T}_i}$, and consider the function $\theta^i_{u,v}=\frac{1_{(u,v)\in\mathcal{T}_i}}{\rho_{u,v}}$. It is easy to check that this set of θ functions satisfies the conditions $\sum_i\rho(i)\theta^i=1$ and $\mathcal{G}(\theta^i)$ is a tree for each i. It happens that the BP iterations for each tree can be combined into one global iteration, given by the following TRBP equations:

$$M_{u\leftarrow v}(x_u)=\max_{x_v}\left(\begin{matrix}\phi_{u,v}(x_v)\\ \rho_{u,v}\end{matrix}+\sum_{w\in\mathcal{N}(v)\setminus\{u\}}\rho_{u,w}M_{v\leftarrow w}(x_v)-(1-\rho_{u,v})M_{v\leftarrow u}(x_v)\right) \tag{1.14}$$

Upon convergence, the final step consists of combining the messages in order to properly upper bound the value function $J_{\mathcal{G}}$ (see [WJW03a] for more details).

A similar idea is explored in [GJ07], where the authors exhibit a message-passing scheme which converges to the same value as a particular LP relaxation of our optimization problem. Denote $p(x_1,\ldots,x_n)$ a probability distribution over all decision variables of $\mathcal{G}$. Using

Jensen's inequality, we obtain:

$$J_{\mathcal{G}} = \max_x \left(\sum_v \phi_v(x_v) + \sum_{u,v} \phi_{u,v}(x_u, x_v) \right) = \max_p \mathbb{E}_{\mathbf{x}\sim p} \left[\sum_v \phi_v(x_v) + \sum_{u,v} \phi_{u,v}(x_u, x_v) \right]$$
$$= \max_p \left(\sum_{(u,v),x_u,x_v} p_{u,v}(x_u, x_v)\phi_{u,v}(x_u, x_v) + \sum_{v,x_v} p_v(x_v)\phi_v(x_v) \right)$$

Clearly, any distribution p on the joint variables $(x_1, \ldots, x_n)$ satisfies for any u, v the three consistency conditions $\sum_{x_u,x_v} p_{u,v}(x_u, x_v) = 1$, $\sum_{x_v} p(x_u, x_v) = p(x_u)$, and $\sum_{x_u} p(x_u, x_v) = p(x_v)$. Let $\mathcal{L}oc$ denote the set of probability distributions which satisfy these three conditions. $\mathcal{L}oc$ is a polyhedron defined by a polynomial number (in $|V| + |E|$) of inequalities, and it defines a natural LP relaxation for upper-bounding the value of $J_{\mathcal{G}}$:

$$J_{\mathcal{G}} \leq \max_{p\in\mathcal{L}oc} \left(\sum_{(u,v),x_u,x_v} p_{u,v}(x_u, x_v)\phi_{u,v}(x_u, x_v) + \sum_{v,x_v} p_v(x_v)\phi_v(x_v) \right)$$

The main result of [GJ07] is to establish that by considering a relaxation scheme following the argument of Equation (1.13), one can construct a convergent message-passing scheme, MPLP, which computes exactly the value of the LP relaxation constructed above. This is done by considering a collection of simple "star" trees (each star tree consisting of one node and its neighbors), and optimizing over the values of θ^i. The MPLP equations they obtain (edge-version) are as follows:

$$M_{u\leftarrow v}(x_u) = -\frac{1}{2} \sum_{k\in\mathcal{N}(u)\setminus\{v\}} M_{u\leftarrow k}(x_u) + \frac{1}{2} \max_{x_v} \left[\sum_{k'\in\mathcal{N}v\setminus\{u\}} M_{v\leftarrow k'}(x_v) + \phi_{u,v}(x_u, x_v) \right]$$

The connection between convex relaxations and message-passing algorithms were later refined, for instance by designing message-passing algorithms which compute tighter LP relaxations of the underlying optimization problem (see for instance [SGJ08, SMG+08a]).

1.5 Conclusions

In this chapter, we presented the Belief Propagation algorithm, and showed how for trees, the BP algorithm is an iterative version of a natural recursive dynamic programming al-

gorithm. Since this recursion is only correct for trees, we cannot expect loopy Belief Propagation to be optimal for general graphs. This observation leads to the following question: can the cavity recursion (1.3) be corrected for general graphs, and can we derive new message-passing-like algorithms from this corrected recursion? This question will be the focus of our next Chapter.

Chapter 2

The Cavity Expansion algorithm

2.1 Introduction

In this chapter, we introduce a new, exact recursion for computing the objective values of optimization problems, and, based on that recursion, we propose a new message-passing-like scheme called the *Cavity Expansion algorithm* (CE). This algorithm will be at the center of many of the results of this thesis. Our construction relies on a technique recently used for constructing approximate counting algorithms. Specifically, Bandyopadhyay and Gamarnik [BG06] and Weitz [Wei06] proposed approximate counting algorithms which are based on local (in the graph-theoretic sense) computation. A crucial idea of Weitz [Wei06] was to establish that certain counting problems in general graphs are equivalent to counting problems on specially designed trees called *self-avoiding walks* trees (SAW trees). Similar SAW constructions were later used for optimization in specific settings [JS07], decoding [LMM07], and statistical physics and inference [Moo08]. The self-avoiding walk approach was later extended in Gamarnik and Katz [GK07b, GK07a] and in Bayati *et al.* [BGK+07], where it was shown that rather than constructing the SAW tree, it was possible to write the recursion on the associated computation tree as a recursion on the original graph which can be directly used for computations. The present chapter develops a similar recursive approach for general optimization problems on arbitrary factor graphs.

2.2 The cavity recursion

2.2.1 The SAW tree construction

In this section, we construct a generalization of identity (1.3), in the same pairwise graphical model framework introduced in the previous chapter. This generalization can be achieved by building a sequence of certain auxiliary decision networks $\mathcal{G}(u,k,x)$ constructed as follows.

Fix any node u and action x and let $\mathcal{N}(u) = \{v_1,\ldots,v_d\}$. For every $k = 1,\ldots,d$ let $\mathcal{G}(u,k,x)$ be the decision network (V',E',Φ',χ) on the same decision set χ constructed as follows. (V',E') is the subgraph induced by $V' = V \setminus \{u\}$. Namely, $E' = E \setminus \{(u,v_1),\ldots,(u,v_d)\}$. Also $\phi'_e = \phi_e$ for all e in E' and the potential functions ϕ'_v are defined as follows. For any $v \in V\setminus\{u,v_1,\ldots,v_{k-1},v_{k+1},\ldots,v_d\}$, $\phi'_v = \phi_v$, and

$$\phi'_v(y) = \begin{cases} \phi_v(y) + \phi_{u,v}(x,y), & \text{for } v \in \{v_1,\ldots,v_{k-1}\} \\ \phi_v(y) + \phi_{u,v}(0,y), & \text{for } v \in \{v_{k+1},\ldots,v_d\} \end{cases} \tag{2.1}$$

(2.2)

Theorem 1 (Cavity Recursion). *Given a network $\mathcal{G}$ and $u \in V$, let $\mathcal{N}(u) = (v_1,\ldots,v_d)$. For every $x \in \chi$,*

$$B_u(x) = \phi_u(x) - \phi_u(0) + \sum_{k=1}^{d} \mu_{u\leftarrow v_k}(x, B_{\mathcal{G}(u,k,x),v_k}) \tag{2.3}$$

Though we we will not prove it here (see [Wei06, JS07] for more details), it can be shown that carrying out the cavity recursion for the binary decisions, pairwise costs case is equivalent to carrying out the cavity tree recursion (1.3) on a self-avoiding walk tree, i.e., a tree for which each branch is a path in $\mathcal{G}$ which does not intersect itself. Our result is however more general, as it can be generalized to multi-spin, general factor graph model. We now proceed to prove Theorem 1.

Proof. Let $x_{j,k} = x$ when $j \le k$ and $= 0$ otherwise. Let $\mathbf{v} = (v_1,\ldots,v_d)$, and $\mathbf{z} =$

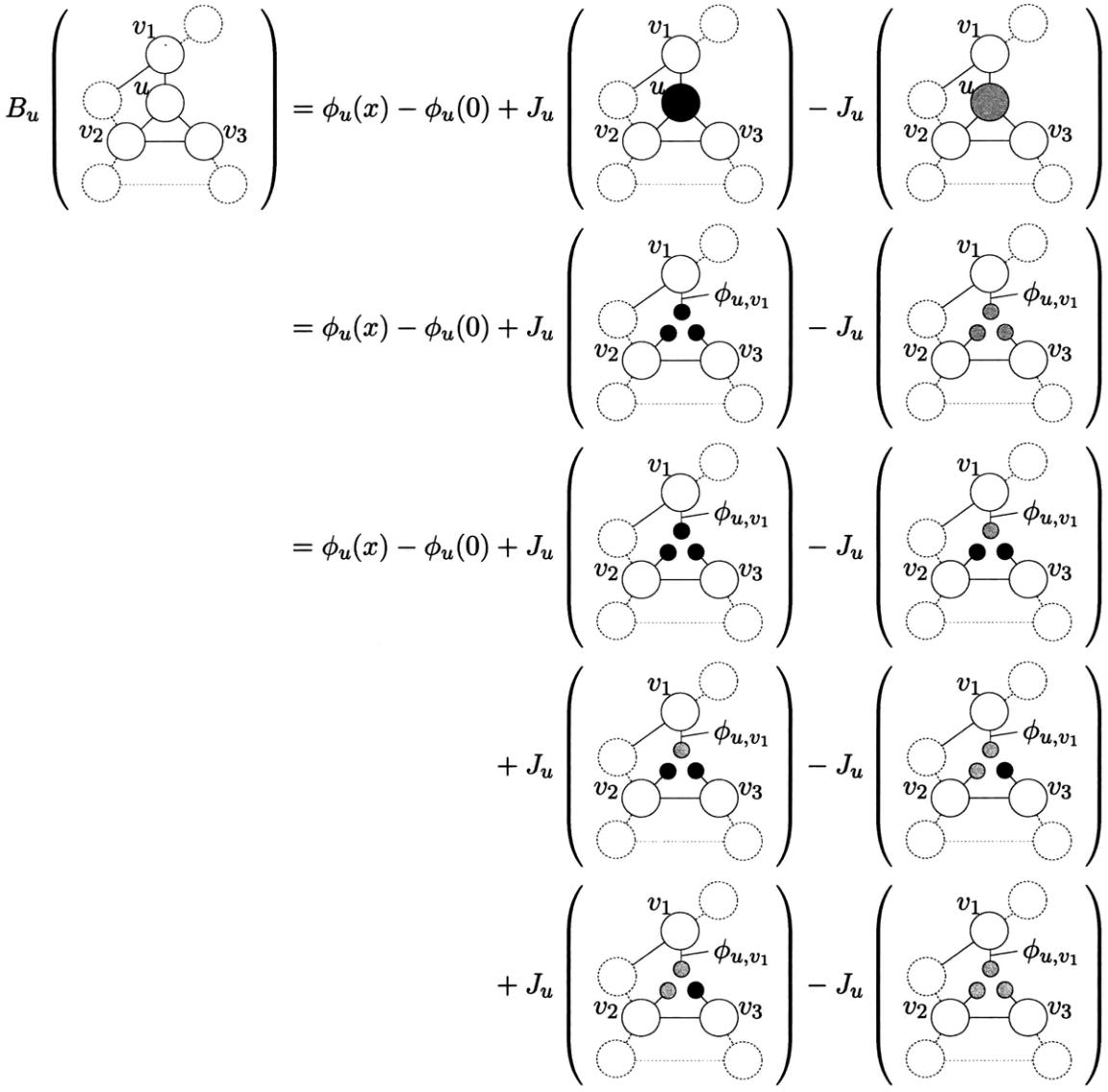


Figure 2-1: First step: building the telescoping sum; black nodes indicate decision x, gray node decision 0

$(z_1, \ldots, z_d) \in \chi^d$. We have

$$B_{\mathcal{G},u}(x) = \phi_u(x) - \phi_u(0) + \max_{\mathbf{z}} \Big\{ \sum_{j=1}^{d} \phi_{u,v_j}(x, z_j) + J_{\mathcal{G}\setminus\{u\},\mathbf{v}}(\mathbf{z}) \Big\} - \max_{\mathbf{z}} \Big\{ \sum_{j=1}^{d} \phi_{u,v_j}(0, z_j) + J_{\mathcal{G}\setminus\{u\},\mathbf{v}}(\mathbf{z}) \Big\}.$$

Consider a telescoping sum,

$$B_{\mathcal{G},u}(x) = \phi_u(x) - \phi_u(0) + \sum_{k=1}^{d} \Bigg[\max_{\mathbf{z}} \Big\{ \sum_{j=1}^{d} \phi_{u,v_j}(x_{j,k}, z_j) + J_{\mathcal{G}\setminus\{u\},\mathbf{v}}(\mathbf{z}) \Big\} - \max_{\mathbf{z}} \Big\{ \sum_{j=1}^{d} \phi_{u,v_j}(x_{j,k-1}, z_j) + J_{\mathcal{G}\setminus\{u\},\mathbf{v}}(\mathbf{z}) \Big\} \Bigg] \tag{2.4}$$

and the k^{th} difference:

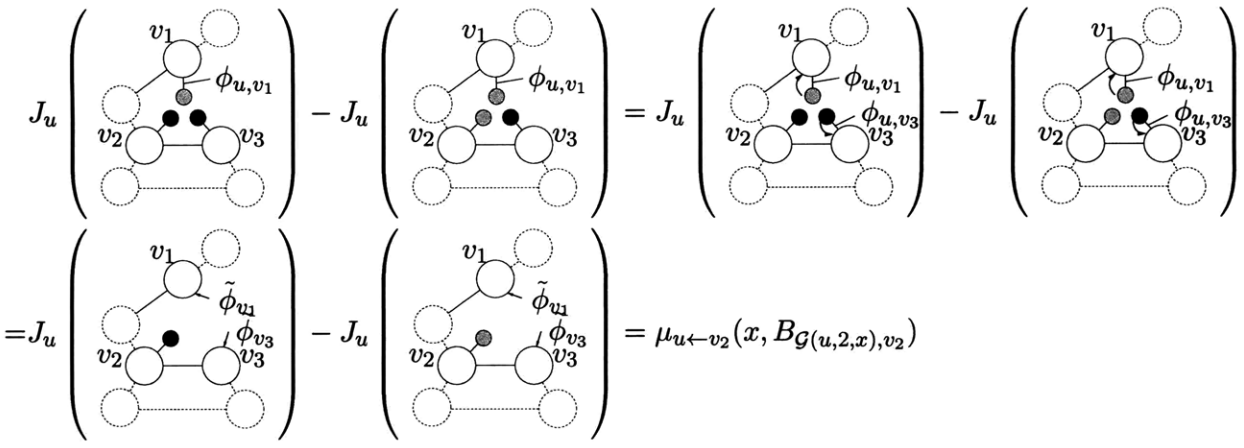
$$\max_{\mathbf{z}} \Big\{ \sum_{j=1}^{d} \phi_{u,v_j}(x_{j,k}, z_j) + J_{\mathcal{G}\setminus\{u\},\mathbf{v}}(\mathbf{z}) \Big\} - \max_{\mathbf{z}} \Big\{ \sum_{j=1}^{d} \phi_{u,v_j}(x_{j,k-1}, z_j) + J_{\mathcal{G}\setminus\{u\},\mathbf{v}}(\mathbf{z}) \Big\} \tag{2.5}$$

Let $\mathbf{z}_{-k} = (z_1, \ldots, z_{k-1}, z_{k+1}, \ldots, z_d)$. Then,

$$\max_{\mathbf{z}} \Big\{ \sum_{j=1}^{d} \phi_{u,v_j}(x_{j,k}, z_j) + J_{\mathcal{G}\setminus\{u\},\mathbf{v}}(\mathbf{z}) \Big\} =$$
$$\max_{z_k} \Big(\phi_{u,v_k}(x, z_k) + \max_{\mathbf{z}_{-k}} \Big\{ \sum_{j \le k-1} \phi_{u,v_j}(x, z_j) + \sum_{j \ge k+1} \phi_{u,v_j}(0, z_j) + J_{\mathcal{G}\setminus\{u\},\mathbf{v}}(\mathbf{z}) \Big\} \Big) \tag{2.6}$$

Similarly,

$$\max_{\mathbf{z}} \Big\{ \sum_{j=1}^{d} \phi_{u,v_j}(x_{j,k-1}, z_j) + J_{\mathcal{G}\setminus\{u\},\mathbf{v}}(\mathbf{z}) \Big\} =$$
$$\max_{z_k} \Big(\phi_{u,v_k}(0, z_k) + \max_{\mathbf{z}_{-k}} \Big\{ \sum_{j \le k-1} \phi_{u,v_j}(x, z_j) + \sum_{j \ge k+1} \phi_{u,v_j}(0, z_j) + J_{\mathcal{G}\setminus\{u\},\mathbf{v}}(\mathbf{z}) \Big\} \Big) \tag{2.7}$$



with

$$\tilde{\phi}_{v_1}(z) = \phi_{v_1}(z) + \phi_{u,v_1}(0, z)$$
$$\tilde{\phi}_{v_3}(z) = \phi_{v_3}(z) + \phi_{u,v_3}(x, z)$$

Figure 2-2: Second step: building the modified subnetworks (here $\mathcal{G}(u, 2, x)$)

For each z_k, we have

$$\max_{\mathbf{z}_{-k}} \Big\{ \sum_{j \leq k-1} \phi_{u,v_j}(x, z_j) + \sum_{j \geq k+1} \phi_{u,v_j}(0, z_j) + J_{\mathcal{G}\setminus\{u\},\mathbf{v}}(\mathbf{z}) \Big\} = J_{\mathcal{G}(u,k,x),v_k}(z_k)$$

By adding and substrating $J_{\mathcal{G}(u,k,x),v_k}(0)$, expression (2.5) can therefore be rewritten as

$$\max_y(\phi_{u,v_k}(x, y) + B_{\mathcal{G}(u,k,x)}(y)) - \max_y(\phi_{u,v_k}(0, y) + B_{\mathcal{G}(u,k,x)}(y))$$

which is exactly $\mu_{u \leftarrow v_k}(x, B_{\mathcal{G}(u,k,x)})$. Finally, we obtain:

$$B_u(x) = \phi_u(x) - \phi_u(0) + \sum_{k=1}^{d} \mu_{u \leftarrow v_k}(x, B_{\mathcal{G}(u,k,x),v_k})$$

□

2.2.2 Extension to factor graphs

We now turn to our most general result, the cavity recursion for factor graphs, which in principle allows computation of bonuses for arbitrary optimization problems. We will need to define a new family of modified subnetworks in order to derive the result. Consider a node u with neighbors $\mathcal{N}(u) = \{a_1, a_2, \ldots, a_d\}$, and an arbitrary action $x \in \chi$. Once again, we need the neighbors to be arbitrarily ordered (we will see in Chapter 5 this can have an importance). For every $k = 1, 2, \ldots, d$, let $\mathcal{G}(u, k, x)$ be the decision network $(V \setminus \{u\}, A' \setminus \{a_k\}, \Phi', \chi)$ where Φ' is constructed as follows. For any $a \in A \setminus \{a_1, \ldots, a_d\}$, $\phi'_a = \phi_a$. For each $j \neq k$, ϕ'_{a_j} is a function only of $\mathbf{x}_{a \setminus \{u\}}$, and the updated values are given by

$$\phi'_{a_j}(\mathbf{x}_{a\setminus\{u\}}) = \begin{cases} \phi_{a_j}(x, \mathbf{x}_{a\setminus\{u\}}), & j < k; \\ \phi_{a_j}(0, \mathbf{x}_{a\setminus\{u\}}), & j > k. \end{cases} \tag{2.8}$$

(2.9)

Theorem 2. *For every $u \in V$ with neighbors $\{a_1, \ldots, a_d\}$, and $x \in \chi$,*

$$B_{\mathcal{G},u}(x) = \phi_u(x) - \phi_u(0) + \sum_{1 \leq k \leq d} \mu_{u \leftarrow a_k}(x, M_{\mathcal{G}(u,k,x)}) \tag{2.10}$$

where $M_{\mathcal{G}(u,k,x)}$ is defined as follows: let $(u, w_1, \ldots, w_{n(k)})$ be the neighbors of a_j in $\mathcal{G}$.

$$M_{\mathcal{G}(u,k,x)}(x_1, \ldots, x_{n(k)}) = B_{\mathcal{G}(u,k,x),(w_1,\ldots,w_{n(k)})}(x_1, x_2, \ldots, x_{n(k)}) \tag{2.11}$$

Note that Equation (2.10), while similar to the factor graph Belief Propagation Equation (1.9), does not have the same exact same structural form (in the way the pairwise BP Equation (1.3) and the pairwise cavity Equation (2.3) have the same structural form). In particular, the function M in Equation (2.11) is not decomposed as a sum of bonus functions of the neighbors of a_k in different network. This issue can be partially addressed by our next result. For any network $\mathcal{G}$, vector of node $\mathbf{v} = (v_1, \ldots, v_n)$ and actions $\mathbf{x} = (x_1, \ldots, x_n)$, let $\mathcal{G}\{\mathbf{v} : \mathbf{x}\}$ be the subnetwork (V', E', Φ', χ), where (V', E') is induced by $V \setminus \{v_1, \ldots, v_n\}$, and all factors which depended on the action of some v_i are computed by fixing the action of v_i to x_i (while they have similar definitions, $\mathcal{G}(u, k, x)$ and $\mathcal{G}\{\mathbf{v} : \mathbf{x}\}$ differ in that no factors are removed from $\mathcal{G}\{\mathbf{v} : \mathbf{x}\}$).

Theorem 3. *Consider any u, action x, and a factor a_j such that $\mathcal{N}(a_k) = \{u, w_1, \ldots, w_{n(k)}\}$ and let $\mathcal{G}' = \mathcal{G}(u, k, x)$ for simplicity. For any $x_1, \ldots, x_{n(k)} \in \chi$, and for any $n \geq 0$, let $\mathbf{w}[n] = (w_1, \ldots, w_n)$ and $\mathbf{x}[n] = (x_1, \ldots, x_n)$.*

$$M_{\mathcal{G}(u,k,x)}(x_1, \ldots, x_{n(k)}) = \sum_{1 \leq n \leq n(k)} B_{\mathcal{G}'\{\mathbf{w}[n-1]:\mathbf{x}[n-1]\},w_n}(x_n) \tag{2.12}$$

We now prove Theorems 2 and 3.

Proof. Using the same telescoping sum used to prove Theorem 1, we have

$$\begin{aligned} B_{\mathcal{G},u}(x) = \phi_u(x) - \phi_u(0) &+ \max_{\mathbf{x}_{-u}} \left(\sum_{1 \leq j \leq d} \phi_{a_j}(x, x_{a_j \setminus \{u\}}) + \sum_{a \notin \mathcal{N}(u)} \phi_a(x_a) \right) \\ &- \max_{\mathbf{x}_{-u}} \left(\sum_{1 \leq j \leq d} \phi_{a_j}(0, x_{a_j \setminus \{u\}}) + \sum_{a \notin \mathcal{N}(u)} \phi_a(x_a) \right) \\ = \phi_u(x) - \phi_u(0) &+ \sum_{1 \leq k \leq d} \left(\max_{\mathbf{x}_{-u}} \left(\sum_{1 \leq j \leq d} \phi_{a_j}(x_{j,k}, x_{a_j \setminus \{u\}}) + \sum_{a \notin \mathcal{N}(u)} \phi_a(x_a) \right) \right. \\ &\left. - \max_{\mathbf{x}_{-u}} \left(\sum_{1 \leq j \leq d} \phi_{a_j}(x_{j,k-1}, x_{a_j \setminus \{u\}}) + \sum_{a \notin \mathcal{N}(u)} \phi_a(x_a) \right) \right) \end{aligned}$$

Consider the k^{th} difference:

$$\begin{aligned} &\max_{\mathbf{x}_{-u}} \left(\phi_{a_k}(x, x_{a_k \setminus \{u\}}) + \sum_{1 \leq j \leq k-1} \phi_{a_j}(x, x_{a_j \setminus \{u\}}) + \sum_{k+1 \leq j \leq d} \phi_{a_j}(0, x_{a_j \setminus \{u\}}) + \sum_{a \notin \mathcal{N}(u)} \phi_a(x_a) \right) \\ &- \max_{\mathbf{x}_{-u}} \left(\phi_{a_k}(0, x_{a_k \setminus \{u\}}) + \sum_{1 \leq j \leq k-1} \phi_{a_j}(x, x_{a_j \setminus \{u\}}) + \sum_{k+1 \leq j \leq d} \phi_{a_j}(0, x_{a_j \setminus \{u\}}) + \sum_{a \notin \mathcal{N}(u)} \phi_a(x_a) \right) \end{aligned}$$

The first term of the k^{th} difference can be decomposed as follows:

$$\max_{\mathbf{x}_{-u}} \left(\phi_{a_k}(x, x_{a_k \setminus \{u\}}) + \sum_{1 \leq j \leq k-1} \phi_{a_j}(x, x_{a_j \setminus \{u\}}) + \sum_{k+1 \leq j \leq d} \phi_{a_j}(0, x_{a_j \setminus \{u\}}) + \sum_{a \notin \mathcal{N}(u)} \phi_a(x_a) \right)$$

$$= \max_{x_{w_1}, x_{w_2}, \ldots, x_{w_{n(k)}}} \left(\phi_{a_k}(x, x_{a_k \setminus \{u\}}) + \max_{x_{V \setminus \mathcal{N}(a_k)}} \left(\sum_{1 \leq j \leq k-1} \phi_{a_j}(x, x_{a_j \setminus \{u\}}) \right. \right.$$

$$\left. \left. + \sum_{k+1 \leq j \leq d} \phi_{a_j}(0, x_{a_j \setminus \{u\}}) + \sum_{a \notin \mathcal{N}(u)} \phi_a(x_a) \right) \right)$$

And it is easy to check that

$$\max_{x_{V \setminus \mathcal{N}(a_k)}} \left(\sum_{1 \leq j \leq k-1} \phi_{a_j}(x, x_{a_j \setminus \{u\}}) + \sum_{k+1 \leq j \leq d} \phi_{a_j}(0, x_{a_j \setminus \{u\}}) + \sum_{a \notin \mathcal{N}(u)} \phi_a(x_a) \right) =$$

$$J_{\mathcal{G}(u,k,x),(w_1,\ldots,w_{n(k)})}(x_1, x_2, \ldots, x_{n(k)})$$

The same decomposition applies to the second term of the k^{th} difference, and we obtain that the k^{th} difference can be rewritten:

$$\max_{x_{w_1}, x_{w_2}, \ldots, x_{w_{n(k)}}} \left(\phi_{a_k}(x, x_{a_k \setminus \{u\}}) + J_{\mathcal{G}(u,k,x),(w_1,\ldots,w_{n(k)})}(x_1, x_2, \ldots, x_{n(k)}) \right)$$

$$- \max_{x_{w_1}, x_{w_2}, \ldots, x_{w_{n(k)}}} \left(\phi_{a_k}(0, x_{a_k \setminus \{u\}}) + J_{\mathcal{G}(u,k,x),(w_1,\ldots,w_{n(k)})}(x_1, x_2, \ldots, x_{n(k)}) \right)$$

which by substracting $J_{\mathcal{G}(u,k,x),(w_1,\ldots,w_{n(k)})}(0, 0, \ldots, 0)$ is seen to be equal to

$$\max_{x_{w_1}, x_{w_2}, \ldots, x_{w_{n(k)}}} \left(\phi_{a_k}(x, x_{a_k \setminus \{u\}}) + B_{\mathcal{G}(u,k,x),(w_1,\ldots,w_{n(k)})}(x_1, x_2, \ldots, x_{n(k)}) \right)$$

$$- \max_{x_{w_1}, x_{w_2}, \ldots, x_{w_{n(k)}}} \left(\phi_{a_k}(0, x_{a_k \setminus \{u\}}) + B_{\mathcal{G}(u,k,x),(w_1,\ldots,w_{n(k)})}(x_1, x_2, \ldots, x_{n(k)}) \right)$$

$$= \mu_{u \leftarrow a_k}(x, M_{\mathcal{G}(u,k,x)})$$

which finishes the proof of Theorem 2. Theorem 3 is a special case of a simple identity: for any network $\mathcal{G}$, collection of node $(v_1, \ldots, v_d)$ and decision vector $(x_1, \ldots, x_d)$, we have

$$B_{\mathcal{G},(v_1,\ldots,v_d)}(x_1, \ldots, x_d) = \sum_{1 \leq n \leq d} B_{\mathcal{G}\{\mathbf{v}[n-1]:\mathbf{x}[n-1]\},v_n}(x_n) \tag{2.13}$$

To prove the above, let $\mathbf{x}[n]$ denote $(x_1, \ldots, x_n, 0, \ldots, 0)$, and simply note that

$$\begin{aligned} B_{\mathcal{G},(v_1,\ldots,v_d)}(x_1,\ldots,x_d) =& J_{\mathcal{G},(v_1,\ldots,v_d)}(\mathbf{x}[d]) - J_{\mathcal{G},(v_1,\ldots,v_d)}(\mathbf{x}[0]) \\ =& \sum_{1\leq n\leq d} J_{\mathcal{G},(v_1,\ldots,v_d)}(\mathbf{x}[n]) - J_{\mathcal{G},(v_1,\ldots,v_d)}(\mathbf{x}[n-1]) \\ =& \sum_{1\leq n\leq d} B_{\mathcal{G}\{\mathbf{v}[n-1]:\mathbf{x}[n-1]\},v_n}(x_n) \end{aligned}$$

2.3 The Cavity Expansion algorithm for graphs or factor graphs

Armed with the cavity recursion, we are now in a position to develop new optimization algorithms for general graphical models. The main problem with using the cavity recursion for computing bonuses is that its computation time can be extremely large. As mentioned previously, carrying the recursion until it terminates is equivalent to carrying the computation on a corresponding SAW tree. In general, the SAW tree will have degree as large as the degree of our network $\mathcal{G}$, and depth equal to the length of the longest self-avoiding walk of the graph, which itself often grows linearly with the size of the graph. Consequently, in the vast majority of cases, using the cavity recursion to compute the network cavities will result in an exponential time algorithm. In order to remedy this, we design two main algorithms. The first, the cavity expansion algorithm, is a message-passing-type algorithm, whose main idea can be summarized as interrupting the computation after a predetermined depth. The second, the cavity propagation algorithm, is a message-passing version of the cavity expansion algorithm.

2.3.1 The CE algorithm

Given a decision network $\mathcal{G}$, a node $u \in V$ with $\mathcal{N}_u = \{v_1, \ldots, v_d\}$, and $r \in \mathbb{Z}_+$, introduce a vector $\mathrm{CE}[\mathcal{G}, u, r] = (\mathrm{CE}[\mathcal{G}, u, r, x], x \in \chi) \in \mathbb{R}^T$ defined recursively as follows.

1. $\mathrm{CE}[\mathcal{G}, u, 0, x] = 0$

2. For every $r = 1, 2, \ldots$, and every $x \in \chi$,

$$\mathrm{CE}[\mathcal{G}, u, r, x] = \phi_u(x) - \phi_u(0) + \sum_{j=1}^{d} \mu_{u \leftarrow v_j}\Big(x, \mathrm{CE}[\mathcal{G}(u, j, x), v_j, r-1]\Big), \qquad (2.14)$$

where $\mathcal{G}(u, k, x)$ is defined in Subsection 2.2.2, and the sum $\sum_{j=1}^{d}$ is equal to 0 when $\mathcal{N}_u = \emptyset$. Note that from the definition of $\mathcal{G}(u, k, x)$, the definition and output of $\mathrm{CE}[\mathcal{G}, u, r]$ depend on the order in which the neighbors v_j of u are considered. $\mathrm{CE}[\mathcal{G}, u, r]$ serves as an r-step approximation, in some appropriate sense to be explained in Chapter 3, of the cavity vector $B_{\mathcal{G},u}$. The motivation for this definition is relation (2.3) of Theorem 1. The local cavity approximation can be computed using an algorithm described below, which we call *Cavity Expansion (CE)* algorithm.

Cavity Expansion: $\mathrm{CE}[\mathcal{G}, u, r, x]$
INPUT: A network $\mathcal{G}$, a node u in $\mathcal{G}$, an action x and a computation depth $r \geq 0$
BEGIN
If $r = 0$ return 0
else do
Find neighbors $\mathcal{N}(u) = \{v_1, v_2, \ldots, v_d\}$ of u in $\mathcal{G}$.
If $\mathcal{N}(u) = \emptyset$, return $\phi_u(x) - \phi_u(0)$.
Else
For each $j = 1, \ldots, d$, construct the network $\mathcal{G}(u, j, x)$.
For each $j = 1, \ldots, d$, and $y \in \chi$, compute $\mathrm{CE}[\mathcal{G}(u, j, x), v_j, r-1, y]$
For each $j = 1, \ldots, d$, compute $\mu_{u \leftarrow v_j}(x, \mathrm{CE}[\mathcal{G}(u, j, x), v_j, r-1, y])$
Return $\phi_u(x) - \phi_u(0) + \sum_{1 \leq j \leq d} \mu_{u \leftarrow v_j}(x, \mathrm{CE}[\mathcal{G}(u, j, x), v_j, r-1, y])$ as $\mathrm{CE}[\mathcal{G}, u, r, x]$.

The algorithm above terminates because r decreases by one at each recursive call of the algorithm. As a result, an initial call to $\mathrm{CE}[\mathcal{G}, u, r, x]$ will result in a finite number of recursive calls to some $\mathrm{CE}[\mathcal{G}_j, u_j, k_j, x_j]$, where $k_j < r$. Let $(\mathcal{G}_i, v_i, x_i)_{1 \leq i \leq m}$ be the subset of

arguments for the calls used in computing CE$[\mathcal{G}, u, r, x]$ for which $k_i = 0$. In the algorithm above, the values returned for $r = 0$ are 0, but it can be generalized by choosing a value $\mathcal{C}_i$ for the call CE$[\mathcal{G}_i, v_i, 0, x_i]$.

The set of values $\mathcal{C} = (\mathcal{C}_i)_{1 \le i \le m}$ will be called a *boundary condition*. We denote by CE$[\mathcal{G}, u, r, x, \mathcal{C}]$ the output of the cavity algorithm with boundary condition $\mathcal{C}$. The interpretation of CE$[\mathcal{G}, u, r, x, \mathcal{C}]$ is that it is an estimate of the cavity $B_{\mathcal{G},u}(x)$ via r steps of recursion (1.3) when the recursion is initialized by setting CE$[\mathcal{G}_i, u_i, 0, x_i] = \mathcal{C}_i$ and is run r steps. We will sometimes omit $\mathcal{C}$ from the notation when such specification is not necessary. Call $\mathcal{C}^* = (\mathcal{C}_i^*) \stackrel{\Delta}{=} (B_{\mathcal{G}_i,v_i}(x_i))$ the "true boundary condition". The justification comes from the following proposition, the proof of which follows directly from Theorem 1.

Proposition 5. *Given node u and $\mathcal{N}(u) = \{v_1, \ldots, v_d\}$, suppose for every $j = 1, \ldots, d$ and $y \in \chi$, CE$[\mathcal{G}(u, j, x), v_j, r-1, y] = B_{\mathcal{G}(u,j,x),v_j}(y)$; then, CE$[\mathcal{G}, u, r, x] = B_{\mathcal{G},u}(x)$.*

As a result, if $\mathcal{C}$ is the "correct" boundary condition, then CE$[\mathcal{G}, u, r, x, \mathcal{C}] = B_{\mathcal{G},u}(x)$ for every u, r, x. The execution of the Cavity Expansion algorithm can be visualized as a computation on a tree, due to its recursive nature. This has some similarity with a computation tree associated with the performance of the Belief Propagation algorithm, [TJ02, SSW08, BSS08]. The important difference with [TJ02] is that the presence of cycles is incorporated via the construction $\mathcal{G}(u, j, x)$ (similarly to [Wei06, JS07, BGK$^+$07, GK07a, GK07b]. As a result, the computation tree of the CE is finite (though often extremely large), as opposed to the BP computation tree.

2.3.2 Properties and computational complexity

Independence Lemma

An important lemma, which we will use frequently in the rest of the thesis, states that in the computation tree of the cavity recursion, the cost function of an edge is statistically independent from the subtree below that edge.

Proposition 6. *Given u, x and $\mathcal{N}(v) = \{v_1, \ldots, v_d\}$, for every $r, j = 1, \ldots, d$ and $y \in \chi$, CE$[\mathcal{G}(u, j, x), v_j, r-1, y]$ and ϕ_{u,v_j} are independent.*

Note, however, that ϕ_{u,v_j} and $\mathcal{G}(u, k, x)$ are generally dependent when $j \neq k$

Proof. The proposition follows from the fact that for any j, the interaction function ϕ_{u,v_j}

does not appear in $\mathcal{G}(u,j,x)$, because node u does not belong to $\mathcal{G}(u,j,x)$), and does not modify the potential functions of $\mathcal{G}(u,j,x)$ in the step (2.1). □

Bounds on the cavities

Another strength of the CE algorithm is that it in the binary case ($\chi = \{0,1\}$, it provides upper and lower bounds on the cavities (this is contrast to convex relaxation, which provides upper bound on the value functions instead, but the technique works only for binary networks), through appropriate simple choice of the boundary condition. Consider the following modified algorithm, which for any network $\mathcal{G}$, node u, and integer $r \geq 0$, computes two cavity approximations $\mathrm{CE}^+[\mathcal{G},u,r]$ and $\mathrm{CE}^-[\mathcal{G},u,r]$(recall that by design, the cavity of 0 is always 0).

Cavity Expansion with bound $\mathrm{CE}^+[\mathcal{G},u,r], \mathrm{CE}^-[\mathcal{G},u,r]$
INPUT: A network $\mathcal{G}$, a node u in $\mathcal{G}$, and a computation depth $r \geq 0$
BEGIN
If $r = 0$ return $\mathrm{CE}^+[\mathcal{G},u,r] = +\infty$ and $\mathrm{CE}^-[\mathcal{G},u,r] = -\infty$
else do
Find neighbors $\mathcal{N}(u) = \{v_1, v_2, \ldots, v_d\}$ of u in $\mathcal{G}$.
If $\mathcal{N}(u) = \emptyset$, return $\phi_u(1) - \phi_u(0)$.
Else
For each $j = 1, \ldots, d$, construct the network $\mathcal{G}(u,j,1)$.
For each $j = 1, \ldots, d$, compute $\mathrm{CE}^-[\mathcal{G}(u,j,1), v_j, r-1]$ and
$\mathrm{CE}^+[\mathcal{G}(u,j,1), v_j, r-1]$
For each $j = 1, \ldots d$ and each $y \in \chi$, form the quantity $\epsilon_j = \phi_{u,v_j}(0,0) + \phi_{u,v_j}(1,1) - \phi_{u,v_j}(0,1) - \phi_{u,v_j}(1,0)$
For each $j = 1 \ldots d$, create two variables B_j^+ and B_j^- as follows: if $\epsilon_j \geq 0$, let $B_j^+ = \mathrm{CE}^+[\mathcal{G}(u,j,1), v_j, r-1]$ and $B_j^- = \mathrm{CE}^-[\mathcal{G}(u,j,1), v_j, r-1]$; otherwise, let $B_j^+ = \mathrm{CE}^-[\mathcal{G}(u,j,1), v_j, r-1]$ and $B_j^- = \mathrm{CE}^+[\mathcal{G}(u,j,1), v_j, r-1]$.
For each $j = 1, \ldots, d$, compute $\mu_{u \leftarrow v_j}(1, B_j^+)$ and $\mu_{u \leftarrow v_j}(1, B_j^-)$
Return $\phi_u(1) - \phi_u(0) + \sum_{1 \leq j \leq d} \mu_{u \leftarrow v_j}(1, B_j^+)$ as $\mathrm{CE}^+[\mathcal{G},u,r]$ and
$\phi_u(1) - \phi_u(0) + \sum_{1 \leq j \leq d} \mu_{u \leftarrow v_j}(1, B_j^-)$ as $\mathrm{CE}^-[\mathcal{G},u,r]$

Theorem 4. *For any network $\mathcal{G}$, node u, depth $r \in \mathbb{N}_+$, and action x,*

$$CE^-[\mathcal{G}, u, r, x] \leq B_{\mathcal{G},u}(x) \leq CE^+[\mathcal{G}, u, r, x]$$

Proof. By induction: the result is clearly correct if $r = 0$ or if $\mathcal{N}(u) = \emptyset$. The next step of the proof is the following lemma:

Lemma 1. *For any $(u, v) \in E$, $\mu_{u \leftarrow v}(1, B)$ is nondecreasing in B if $\phi_{u,v}(1,1) + \phi_{u,v}(0,0) - \phi_{u,v}(0,1) - \phi_{u,v}(1,0) \geq 0$, and nonincreasing otherwise.*

To see why why this is true, simply consider all possible cases for the inequalities between values of $\phi_{u,v}$. Next, assume that for all j,

$$\mathrm{CE}^-[\mathcal{G}(u, j, 1), v_j, r-1] \leq B_{\mathcal{G}(u,j,1),v_j}(1) \leq \mathrm{CE}^+[\mathcal{G}(u, j, 1), v_j, r-1]$$

From Lemma 1, this implies

$$\mu_{u \leftarrow v}(1, B_j^-) \leq \mu_{u \leftarrow v}(1, B_{\mathcal{G}(u,j,1),v_j}) \leq \mu_{u \leftarrow v}(1, B_j^+)$$

and we obtain the result by summing over j. □

Computational complexity

Our last proposition analyzes the complexity of running the Cavity Expansion algorithm.

Proposition 7. *For every $\mathcal{G}, u, r, x$, the value $CE[\mathcal{G}, u, r, x]$ can be computed in time $O\big(r(\Delta T)^r\big)$.*

Proof. The computation time required to construct the networks $\mathcal{G}(u, j, x)$, compute the messages $\mu_{u \leftarrow v_j}(x, B_{v_j})$, and return $\Phi_u(x) - \Phi_u(0) + \sum_{1 \leq j \leq d} \mu_{u \leftarrow v_j}(x, B_{v_j})$, is $O(\Delta T)$. Let us prove by induction that that for any subnetwork $\mathcal{G}'$ of $\mathcal{G}$, $\mathrm{CE}[\mathcal{G}', u, r, x]$ can be computed in time bounded by $O(r(\Delta T)^r)$. The computation time for $r = 0$ is constant. For $r > 1$, the computations of $\mathrm{CE}[\mathcal{G}', u, r, x]$ requires a fixed cost of $O(\Delta T)$, as well as (ΔT) calls to CE with depth $(r-1)$. The total cost is therefore bounded by $O(\Delta T + (\Delta T)\,(r-1)(\Delta T)^{r-1})$, which is $O(r(\Delta T)^r)$. □

2.3.3 Message-passing version of the CE algorithm

In this last section, we detail how to derive a new message-passing algorithm from the cavity recursion. We restrict ourselves to the pairwise case. For any network $\mathcal{G}$, neighbors u, v, and decision x, let

$$M_{\mathcal{G},u\leftarrow v}(x_u) \triangleq \max_{x_v} \left(\phi_{u,v}(x_u, x_v) + B_{\mathcal{G},v}(x_v)\right) - \max_{x_v} \left(\phi_{u,v}(0, x_v) + \phi_v(x_v) B_{\mathcal{G},v}(x_v)\right)$$

The following proposition is an analog of Proposition 2 for the cavity recursion:

Proposition 8.

$$\begin{aligned} M_{\mathcal{G},u\leftarrow v}(x_u) = &\max_{x_v} \left(\phi_{u,v}(x_u, x_v) + \phi_v(x_v) + \sum_{w\in\mathcal{N}(v)\setminus\{u\}} M_{\mathcal{G}(u,k,x),v\leftarrow v_k}(x_v) \right) \\ &- \max_{x_v} \left(\phi_{u,v}(0, x_v) + \phi_v(x_v) + \sum_{w\in\mathcal{N}(v)\setminus\{u\}} M_{\mathcal{G}(u,k,x),v\leftarrow v_k}(x_v) \right) \end{aligned} \tag{2.15}$$

Proposition 8 suggests a new message-passing algorithm for computing the cavities of a network $\mathcal{G}$. The algorithm, which we call *Cavity Propagation*, depends on a depth parameter r and a network $\mathcal{G}$, and at a high level, functions as follows. Cavity Propagation computes messages recursively by using Equation (2.15) while decreasing the depth parameter at each iteration, and whenever the depth reaches zero, resets the modified subnetwork to the original graph $\mathcal{G}$. Formally, consider some network $\mathcal{G}$ and depth parameter $r \in \mathbb{N}_+$. Initial calls to CE$[\mathcal{G}, u, r, x]$ for all u will result in a finite number of recursive calls to some CE$[\mathcal{G}_j, u_j, k_j, x_j]$, where $k_j < r$. For any $s \leq r$, let $\mathcal{R}_s = \{\mathcal{G}_j \mid k_j = s\}$ be the set of all subnetworks $\mathcal{H}$ which were called with a depth equal to s, and $\mathcal{R} = \bigcup \mathcal{R}_s$ the set of all subnetworks called recursively by the CE algorithm with depth r. For any $\mathcal{H} \in \mathcal{R}$ and $(u, v) \in \mathcal{H}$, and iteration time $t > 0$, we define a message $M_{\mathcal{H},u\leftarrow v}$ which is updated

according to the Cavity Propagation equations:

$$\begin{aligned}
\text{For } \mathcal{H} \in \mathcal{R}_0 \; M^t_{\mathcal{H},u\leftarrow v}(x_u) = &\max_{x_v} \left(\phi_{u,v}(x_u, x_v) + \phi_v(x_v) + \sum_{w \in \mathcal{N}(v)\setminus\{u\}} M^{t-1}_{\mathcal{G},v\leftarrow v_k}(x_v) \right) \\
&- \max_{x_v} \left(\phi_{u,v}(0, x_v) + \phi_v(x_v) + \sum_{w \in \mathcal{N}(v)\setminus\{u\}} M^{t-1}_{\mathcal{G},v\leftarrow v_k}(x_v) \right) \qquad (2.16) \\
\text{For } \mathcal{H} \notin \mathcal{R}_0, \; M^t_{\mathcal{H},u\leftarrow v}(x_u) = &\max_{x_v} \left(\phi_{u,v}(x_u, x_v) + \phi_v(x_v) + \sum_{w \in \mathcal{N}(v)\setminus\{u\}} M^{t-1}_{\mathcal{H}(u,k,x),v\leftarrow v_k}(x_v) \right) \\
&- \max_{x_v} \left(\phi_{u,v}(0, x_v) + \phi_v(x_v) + \sum_{w \in \mathcal{N}(v)\setminus\{u\}} M^{r,t-1}_{\mathcal{H}(u,k,x)^{t-1},v\leftarrow v_k}(x_v) \right)
\end{aligned}$$

It is easy to see that the algorithm for $r = 0$ corresponds to Belief Propagation, and for r greater than the length L of longest self-avoiding walk of the graph, it provides the exact cavity in each node v of the original network $\mathcal{G}$. We can therefore expect that for $0 < r < L$, Cavity Propagation is an increasingly powerful family of message-passing algorithms.

2.4 Conclusions

Starting from an exact but computationally intensive recursion to compute cavities in arbitrary graphical models, we developed a new message-passing-type algorithm, the Cavity Expansion algorithm. At a high level, the CE algorithm works by locally computing cavities as a function of neighboring cavities. The algorithm then proceeds by expanding the cavity recursion in the breadth-first search manner for some designed number of steps t, thus constructing an associated computation tree with depth t. At the initialization point the cavity values are assigned some default value. The approximation value $\hat{B}_v(x)$ is then computed using this computation tree. If this computation was conducted for t equalling roughly the length L of the longest self-avoiding path of the graph, it would result in exact computation of the cavity values $B_v(x)$. Yet the computation effort associated with this scheme is exponential in L, which itself often grows linearly with the size of the graph. The CE algorithm interrupts the expansion after a fixed number of steps $t << L$. As such, the CE constructs cavity approximations which are only based on information local to each

node. We are therefore led to wonder whether conditions exist which can guarantee that the resulting approximations are very close to the correct cavities. We will address this question in the following two chapters.

Chapter 3

Correlation decay and efficient decentralized optimization in decision networks with random objective functions

3.1 Introduction

In this chapter, we begin our investigation of the connections between a property of random systems called *correlation decay*, and the near-optimality of the CE algorithm we introduced in Chapter 2. Here, we will focus on optimization in graphical models with discrete variables and random cost functions.

The concept of correlation decay was introduced by Dobrushin [Dob68a, Dob68b] in the context of infinite Markov Random Fields (see also [Spi71, Geo88] for monographs on related topics). The purpose was to identify sufficient conditions for the uniqueness of a distribution on an infinite Markov Random Field, when given only local conditional distributions. Dobrushin identified a simple sufficient condition for uniqueness which stated, at a high level, that if local correlations were weak enough, the local conditional distributions could only correspond to one distribution on the infinite field. This condition was later found to have applications in computer science, as it was shown that finite Markov Random Fields which satisfied conditions similar to Dobrushin's exhibited fast mixing of the

corresponding Markov Chain Monte Carlo dynamics [JS97]. In addition, further connections were found between Dobrushin's condition and convergence of the Belief Propagation algorithm [TJ02].

Thus, correlation decay as introduced by Dobrushin is well adapted to counting and sampling problems in Markov Random Fields. However, in order to apply these ideas to optimization, a different concept of correlation decay is therefore needed. Such a concept was introduced and studied in the context of probabilistic combinatorial optimization [Ald92, Ald01, AS03, GNS06, GG09], where it was shown that some optimization problem on regular and random locally tree-like graphs with random costs are tractable as they exhibit the correlation decay property. Moreover, it was shown that under the correlation decay property, such optimization problems exhibit a phenomenon known as *long-range independence*: intuitively, this means that the optimal decision taken by a node in a network is asymptotically independent from that of nodes faraway from it.

This idea is reminiscent of the approach taken by the CE algorithm, which, as mentioned in Chapter 2, uses only local network information to compute a decision for each node. It is then reasonable to expect that the SAW tree construction of our last chapter and the correlation decay analysis can be merged in some way. Namely, optimization problems with general graphs and random costs can be solved approximately by constructing a computation tree and proving the correlation decay property on it. This is precisely our approach: we show that if we compute $B_v(x)$ based on the computation tree with only constant depth t, the resulting error $\hat{B}_v(x) - B_v(x)$ is exponentially small in t. By taking $t = O(\log(1/\epsilon))$ for any target accuracy ϵ, this approach leads to an ϵ-approximation scheme for computing the optimal reward $\max_x F(x)$.

In this chapter, we provide a general technique to compute conditions on the parameters of families of distribution that ensure the system exhibits the correlation decay property. We illustrate the applicability of the method by giving concrete results for the cases of uniform and Gaussian distributed functions in networks with bounded connectivity (graph degree) Δ.

Another implication of correlation decay concerns decentralization of the decisions. Define the local neighborhood $\mathcal{N}_v^r$ of radius r for node v in $\mathcal{G}$ as the subnetwork induced by $\mathcal{B}_v(r)$. Intuitively, $\mathcal{N}_v^r$ is the subnetwork node v "sees" if its horizon has length r. A decentralized solution x_v^r of radius r for node v is a decision of χ which is built only using knowledge of $\mathcal{N}_v^r$. A vector of decisions taken with only partial (local) information is likely to be suboptimal, and precisely how much is lost by discarding nonlocal information can

for instance be measured by the quantity $F(\mathbf{x}) - F(\mathbf{x}^r)$. The tradeoff between decentralization and suboptimality was investigated by Van Roy and Rusmevichientong in [RR03], but the analysis was restricted to line graphs. Our analysis generalizes their approach, casting their results in the light of the correlation decay phenomenon; we find that if correlation decay occurs, we can accurately quantify the decentralization-optimality tradeoff.

The chapter is organized as follows. In Section 3.2, we describe the general model, examples, and main results. In Section 3.3, we prove our main result, the fact that correlation decay implies optimality of the cavity recursion and local optimality of the solution. The rest of the chapter is devoted to the analysis of a general coupling technique used to identify sufficient conditions for correlation decay (and hence, optimality of the CE algorithm). Concluding thoughts are offered in Section 3.7.

3.2 Model description and results

Recall the pairwise graphical model of Chapter 1: we consider a decision network $\mathcal{G} = (V, E, \Phi, \chi)$. Here (V, E) is an undirected simple graph in which each node $u \in V$ represents an agent, and edges $e \in E$ represent a possible interaction between two agents. Each agent makes a decision $x_u \in \chi \triangleq \{0, 1, \ldots, T-1\}$. For every $v \in V$, a function $\phi_v : \chi \to \mathbb{R}$ is given. Also for every edge $e = (u, v)$ a function $\phi_e : \chi^2 \to \mathbb{R}$ is given. Let Δ denote the maximum degree of the network $\mathcal{G}$.

Our objective is to compute an optimal (or near-optimal) solution for the network, and the main focus of this chapter will be on the case where $\phi_v(x), \phi_e(x, y)$ are random variables (however, the actual realizations of the random variables are observed by the agents, and their decisions depend on the values taken by $\phi_v(x)$ and $\phi_e(x, y)$). While we will usually assume independence of these random variables when v and e vary, we will allow dependence for the same v and e when we vary the decisions x, y. The details will be discussed when we proceed to concrete examples.

Examples

Graph Coloring

An assignment ϕ of nodes V to colors $\{1, \ldots, q\}$ is defined to be proper coloring if no monochromatic edges are created. Namely, for every edge (v, u), we want $\phi(v) \neq \phi(u)$. Sup-

pose each node/color pair $(v, x) \in V \times \{1, \ldots, q\}$ is equipped with a weight $W_{v,x} \geq 0$. The (weighted) coloring problem is the problem of finding a proper coloring ϕ with maximum total weight $\sum_v W_{v,\phi(v)}$. In terms of a decision network framework, we have $\phi_{v,u}(x, x) = -\infty, \phi_{v,u}(x, y) = 0, \forall x \neq y \in \chi = \{1, \ldots, q\}, (v, u) \in E$ and $\phi_v(x) = W_{v,x}, \forall v \in V, x \in \chi$.

MAX 2-SAT

Let $(Z_1, \ldots, Z_n)$ be a set of boolean variables. Let $(C_1, \ldots, C_m)$ be a list of clauses of the form $(Z_i \vee Z_j)$, $(Z_i \vee \overline{Z_j})$, $(Z_i \vee Z_j)$ or $(Z_i \vee Z_j)$. The MAX-2SAT problem consists of finding an assignment for binary variables Z_i which maximizes the number of satisfied constraints C_j. In terms of a decision network, take $V = \{1, \ldots, n\}$, $E = \{(i, j) : Z_i \text{ and } Z_j \text{ appear in a common clause}\}$, and for any k, let $\phi_k(x, y)$ to be 1 if the clause C_k is satisfied when $(Z_i, Z_j) = (x, y)$ and 0 otherwise. Let $\phi_v(x) = 0$ for all v, x.

MAP estimation

We note in that in the graphical model and message-passing literature, the term MAP estimation is often used to refer to MLE estimation of the graphical model $\mathcal{G}$, or in other words, to the task of finding $\max F_{\mathcal{G}}(x)$. *We consider here a problem which is properly "a-posteriori".* In this example, we see a situation in which the reward functions are naturally randomized.

Consider a graph (V, E) with $|V| = n$ and $|E| = m$, a set of real numbers $\mathbf{p} = (p_1, \ldots, p_n) \in [0, 1]^n$, and a family $(f_1, \ldots, f_m)$ of functions such that for each $(i, j) \in E$, $f_{i,j}$ is a function $f_{i,j}(o, x, y)$ where o is real and $x, y \in \{0, 1\}^2$. Assume that for each (x, y), $f_{i,j}(o, x, y)$ is a probability density for o. Consider two sets $\mathbf{C} = (C_i)_{1 \leq i \leq n}$ and $\mathbf{O} = (O_j)_{1 \leq j \leq m}$ of random variables, with joint probability density

$$P(\mathbf{O}, \mathbf{C}) = \prod_i {p_i}^{c_i} (1 - p_i)^{1 - c_i} \prod_{(i,j) \in E} f_{i,j}(o_{i,j}, c_i, c_j)$$

$\mathbf{C}$ is a set of Bernoulli random variables ("causes") with probability $P(C_i = 1) = p_i$, and $\mathbf{O}$ is a set of continuous "observation" random variables. Conditional on the cause variables $\mathbf{C}$, the observation variables $\mathbf{O}$ are independent, and each $O_{i,j}$ has density $f_{i,j}(o, c_i, c_j)$. Assume the variables $\mathbf{O}$ represent observed measurements used to infer on hidden causes $\mathbf{C}$. Using Bayes's formula, given observations $\mathbf{O}$, the log posterior probability of the cause

variables $\mathbf{C}$ is equal to:

$$\log P(\mathbf{C} = \mathbf{c} \mid \mathbf{O} = \mathbf{o}) = K + \sum_i \phi_i(c_i) + \sum_{i,j \in E} \phi_{i,j}(c_i, c_j)$$

where

$$\phi_i(c_i) = \log(p_i/(1-p_i))c_i$$
$$\phi_{i,j}(c_i, c_j) = \log(f_{i,j}(o_{i,j}, c_i, c_j))$$

where K is a random number which does not depend on $\mathbf{c}$. Finding the maximum a posteriori values of $\mathbf{C}$ given $\mathbf{O}$ is equivalent to finding the optimal solution of the decision network $\mathcal{G} = (V, E, \phi, \{0,1\})$. Note that the interaction functions $\phi_{i,j}$ are naturally randomized, since $\phi_{i,j}(x,y)$ is a continuous random variable with distribution

$$d\mathbb{P}(\phi_{i,j}(x,y) = t) = \mathrm{e}^t \sum_{x',y' \in \{0,1\}} d\mathbb{P}(f_{i,j}(o,x',y') = \mathrm{e}^t)$$

Main results

At a high level, all our results stem from a combined approach. The algorithm mentioned in the following results is always the CE algorithm introduced in Chapter 2. Correlation decay provides the framework used to prove optimality and polynomiality of the methods. The proof that a correlation decay condition holds is framework-specific; in this chapter, all results stem from the coupling technique detailed in Section 3.4, which applies to graphical models with random costs and no combinatorial constraints. We present detailed results from the uniform distribution and for the Gaussian distribution. It is important to note that while we limit our exposition to these two families, nothing prevents us for applying the methodology to a larger number of distributions. Finally, we refer the readers to Appendix B for the definition of additive FPTAS with high probability.

Uniform Distribution

Suppose that for all $u \in V$, $\phi_u(1)$ is uniformly distributed on $[-I_1, I_1]$, $\phi_u(0) = 0$, and that for every $e \in E$, $\phi_e(0,0), \phi_e(1,0), \phi_e(0,1)$ and $\phi_e(1,1)$ are all independent and uniformly distributed on $[-I_2, I_2]$, for some $I_1, I_2 > 0$. Intuitively, I_1 quantifies the 'bias' each agent has towards one action or another, while I_2 quantifies the strength of interactions between

agents.

Theorem 5. *Let* $\beta = \frac{5I_2}{2I_1}$. *If* $\beta(\Delta-1)^2 < 1$, *then there exists an additive FPTAS for finding* $J_{\mathcal{G}}$ *with high probability.*

Gaussian distribution

Here we consider the case of Gaussian distributed reward functions: assume that for any edge $e = (u, v)$ and any pair of action $(x, y) \in \{0, 1\}^2$, $\phi_{u,v}(x, y)$ is a Gaussian random variable with mean 0 and standard deviation σ_e. For every node $v \in V$, suppose $\phi_v(1) = 0$ and that $\phi_v(0)$ is a Gaussian random variable with mean 0 and standard deviation σ_p. Assume that all rewards $\phi_e(x, y)$ and $\phi_v(x)$ are independent.

Theorem 6. *Let* $\beta = \sqrt{\frac{\sigma_e^2}{\sigma_e^2+\sigma_p^2}}$ *If* $\beta(\Delta-1)+\sqrt{\beta(\Delta-1)^3} < 1$, *then there exists an additive FPTAS for finding* $J_{\mathcal{G}}$ *with high probability.*

3.3 Correlation decay and decentralized optimization

In this section, we investigate the relations between the correlation decay phenomenon and the existence of near-optimal decentralized decisions. When a network exhibits the correlation decay property, the cavity functions of faraway nodes are weakly related, implying a weak dependence between their optimal decisions as well. Thus, one can expect that good decentralized decisions exist. We will show that this is indeed the case.

Definition 1. *Given a function* $\rho(r) \geq 0, r \in \mathbb{Z}_+$ *such that* $\lim_{r\to\infty} \rho(r) = 0$, *a decision network* $\mathcal{G}$ *is said to satisfy the correlation decay property with rate* ρ *if for every two boundary conditions* $\mathcal{C}$, $\mathcal{C}'$

$$\max_{u,x} \mathbb{E}\left|CE[\mathcal{G}, u, r, x, \mathcal{C}] - CE[\mathcal{G}, u, r, x, \mathcal{C}']\right| \leq \rho(r).$$

If there exists $K_c > 0$ *and* $\alpha_c < 1$ *independent from the network topology such that* $\rho(r) \leq K_c\alpha_c^r$ *for all* r, *then we say that* $\mathcal{G}$ *satisfies the exponential correlation decay property with rate* α_c.

The correlation decay property implies that for every u, x,

$$\mathbb{E}\left|CE[\mathcal{G}, u, r, x] - B_{\mathcal{G},u}(x)\right| \leq \rho(r).$$

The following assumptions will be frequently used in the subsequent analysis.

Assumption 1. *For all $v \in V, x \neq y \in \chi$, $B_v(x) - B_v(y)$ is a continuous random variable with density bounded above by a constant $g > 0$.*

We will also assume the costs functions are bounded in L_2 norm:

Assumption 2. *There exists K_Φ such that for any $e \in E$, $\left(\sum_{x,y\in\chi} \mathbb{E}|\phi_e(x,y)|^2\right)^{1/2} \leq K_\Phi$ and for any $v \in V$, $\left(\sum_{x\in\chi} \mathbb{E}|\phi_v(x)|^2\right)^{1/2} \leq K_\Phi$*

Assumption 1 is designed to lead to the following two properties:
(a) There is a unique optimal action in every node with probability 1.
(b) The suboptimality gap between the optimal action and the second best action is large enough so that there is a "clear winner" among actions.

Correlation decay implies near-optimal decentralized decisions

Under Assumption 1 let $\mathbf{x} = (x_v)_{v\in V}$ be the unique (with probability one) optimal solution for the network $\mathcal{G}$. For every $v \in V$, $x \in \chi$, let $x_v^r = \text{argmax}_x CE[\mathcal{G}, v, r, x]$, ties broken arbitrarily, and $\mathbf{x}^r = (x_v^r)$. The main relation between the correlation decay property, the Cavity Expansion algorithm and the optimization problem is given by the following result.

Proposition 9. *Suppose $\mathcal{G}$ exhibits the correlation decay property with rate $\rho(r)$ and that Assumption 1 holds. Then,*

$$\mathbb{P}(x_u^r \neq x_u) \leq 2T^2\sqrt{2g\rho(r)}, \qquad \forall u \in V, r \geq 1. \tag{3.1}$$

Proof. For simplicity, let $B_u^r(x)$ denote $CE[\mathcal{G}, u, r, x]$. We will first prove that

$$\mathbb{P}(x_u^r \neq x_u) \leq T^2(g\epsilon + \frac{2\rho(r)}{\epsilon}) \tag{3.2}$$

The proposition will follow by choosing $\epsilon = \sqrt{2\rho(r)g^{-1}}$. Consider a node u, and notice that if

$$(B_u(x) - B_u(y))(B_u^r(x) - B_u^r(y)) > 0, \qquad \forall x \neq y,$$

then $x_u^r = x_u$. Indeed, since $B_u(x_u) - B_u(y) > 0$ for all $y \neq x_u$, the property implies the

same for B_u^r, and the assertion holds. Thus, the event $\{x_u^r \neq x_u\}$ implies the event

$$\{\exists(x,y), y \neq x : (B_u(x) - B_u(y))(B_u^r(x) - B_u^r(y)) \leq 0\}$$

Fix $\epsilon > 0$ and note that for two real numbers r and s, if $|r| > \epsilon$ and $|r - s| \leq \epsilon$, then $rs > 0$. Applying this to $r = B_u(x) - B_u(y)$ and $s = B_u^r(x) - B_u^r(y)$, we find that the events $|B_u(x) - B_u(y)| > \epsilon$ and

$$(|B_u(x) - B_u^r(x)| < \epsilon/2) \cap (|B_u(y) - B_u^r(y)| < \epsilon/2)$$

jointly imply

$$(B_u(x) - B_u(y))(B_u^r(x) - B_u^r(y)) > 0$$

Therefore, the event $(B_u(x) - B_u(y))(B_u^r(x) - B_u^r(y)) \leq 0$ implies

$$\{|B_u(x) - B_u(y)| \leq \epsilon\} \cup \{|B_u(x) - B_u^r(x)| \geq \epsilon/2\} \cup \{|B_u(y) - B_u^r(y)| \geq \epsilon/2\}$$

Applying the union bound, for any two actions $x \neq y$,

$$\begin{aligned}\mathbb{P}\Big((B_u(x) - B_u(y))(B_u^r(x) - B_u^r(y)) \leq 0\Big) \leq\ & \mathbb{P}(|B_u(x) - B_u(y)| \leq \epsilon) + \mathbb{P}(|B_u(x) - B_u^r(x)| \geq \epsilon/2) \\ & + \mathbb{P}(|B_u(y) - B_u^r(y)| \geq \epsilon/2). \qquad (3.3)\end{aligned}$$

Now $\mathbb{P}(|B_u(x) - B_u(y)| \leq \epsilon)$ is at most $2g\epsilon$ by Assumption 1. Using the Markov inequality, we find that the second summand in (3.3) is at most $2\mathbb{E}|B_u(x) - B_u^r(x)|/\epsilon \leq 2\rho(r)/\epsilon$. The same bound applies to the third summand. Finally, noting there are $T(T-1)/2$ different pairs (x, y) with $x \neq y$ and applying the union bound, we obtain:

$$\begin{aligned}\mathbb{P}(x_u^r \neq x_u) &\leq (T(T-1)/2)(2g\epsilon + 4\rho(r)/\epsilon) \\ &\leq T^2(g\epsilon + \frac{2\rho(r)}{\epsilon}).\end{aligned}$$

□

For the special case of exponential correlation decay, we obtain the following result, the proof of which immediately follows from Proposition 9.

Corollary 1. *Suppose $\mathcal{G}$ exhibits the exponential correlation decay property with rate α_c*

and constant K_c, and suppose Assumption 1 holds. Then

$$\mathbb{P}(x_u^r \neq x_u) \leq 2T^2\sqrt{2gK_c}\alpha_c^{r/2}, \qquad \forall u \in V, r \geq 1.$$

In particular, for any $\epsilon > 0$, if

$$r \geq 2^{\dfrac{|\log K_c'| + |\log \epsilon|}{|\log(\alpha_c)|}}$$

then

$$\mathbb{P}(x_u^r \neq x_u) \leq \epsilon$$

where $K_c' = 2T^2\sqrt{2gK_c}$

In summary, correlation decay - and in particular fast (i.e., exponential) correlation decay - implies that the optimal action in a node depends with high probability only on the structure of the network in a small radius around the node. As in [RR03], we call such a property *decentralization* of optimal actions. Note that the radius required to achieve an ϵ error does not depend on the size of the entire network; moreover, for exponential correlation decay, it grows only as a logarithm of the accepted error.

The main caveat of Proposition 9 is that Assumption 1 does not necessarily hold. For instance, it definitely does not apply to models with discrete random variables ϕ_u and $\phi_{u,v}$. In fact, Assumption 1 is not really necessary, and it can be shown that a regularization technique allows to relax this assumption. Note that Assumption 2 is not needed for Proposition 9 to hold.

Correlation decay and efficient optimization

Proposition 9 illustrates how optimal actions are decentralized under the correlation decay property. In this section, we use this result to show that the resulting optimization algorithm is both near-optimal and computationally efficient.

As before, let before $\mathbf{x} = (x_u)$ denote the optimal solution for the network $\mathcal{G}$, and let $\mathbf{x}^r = (x_u^r)$ be the decisions resulting from the Cavity Expansion algorithm with depth r. Let $\tilde{\mathbf{x}} = (\tilde{x}_u)$ denote (any) optimal solution for the perturbed network $\tilde{\mathcal{G}}$. Let $K_1 = 10K_\Phi T(|V| + |E|)$, and $K_2 = K_1 (g\, K_c)^{1/4}$, where K_c is defined in the assumption of exponential correlation decay.

Theorem 7. *Suppose a decision network $\mathcal{G}$ satisfies correlation decay property with rate $\rho(r)$. Then, for all $r > 0$*

$$\mathbb{E}[F(\mathbf{x}) - F(\mathbf{x}^{r,\delta})] \leq K_1(g\rho(r))^{1/4} \tag{3.4}$$

Corollary 2. *Suppose $\mathcal{G}$ exhibits exponential correlation decay property with rate α_c and constant K_c. Then, for any $\epsilon > 0$, if*

$$r \geq \big(8|\log\epsilon| + 4|\log(K_2)|\big)|\log(\alpha_c)|^{-1}$$

then

$$\mathbb{P}\big(F(\mathbf{x}) - F(\mathbf{x}^r) > \epsilon\big) \leq \epsilon$$

and $\mathbf{x}^r$ can be computed in time polynomial in $|V|$, $1/\epsilon$.

Proof. By applying the union bound on Proposition 9, for every (u, v), we have: $\mathbb{P}\big((x_u^r, x_v^r) \neq (x_u, x_v)\big) \leq 4T^2\sqrt{2g\rho(r)}$. We have

$$\mathbb{E}|F(\mathbf{x}) - F(\mathbf{x}^r)| \leq \sum_{u\in V} \mathbb{E}|\phi_u(x_u) - \phi_u(x_u^r)| + \sum_{(u,v)\in E} \mathbb{E}|\phi_{u,v}(x_u, x_v) - \phi_{u,v}(x_u^r, x_v^r)|$$

For any $u, v \in V$,

$$\begin{aligned}
\mathbb{E}[\phi_{u,v}(x_u, x_v) - \phi_{u,v}(x_u^r, x_v^r)] &\leq \mathbb{E}\Big[1_{(x_u^r, x_v^r)\neq(x_u,x_v)}\Big(\big|\phi_{u,v}(x_u, x_v)\big| + \big|\phi_{u,v}(x_u^r, x_v^r)\big|\Big)\Big] \\
&\leq 2K_\Phi\,\mathbb{P}\big((x_u^r, x_v^r) \neq (x_u, x_v)\big)^{1/2} \\
&\leq 4K_\Phi T\,(2g\rho(r))^{1/4}
\end{aligned}$$

where the second inequality follows from Cauchy-Schwarz. Similarly, for any u we have

$$\mathbb{E}|\phi_u(x_u) - \phi_u(x_u^r)| \leq 4K_\Phi T\,(2g\rho(r))^{1/4}$$

By summing over all nodes and edges, we get: $\mathbb{E}[F(\mathbf{x}) - F(\mathbf{x}^r) \leq 8K_\Phi\,T(2g\rho(r))^{1/4} \leq K_1(g\rho(r))^{1/4}$, and Equation (3.4) follows. The corollary is then proved using the Markov Inequality; injecting the definition of exponential correlation decay into Equation (3.4), we obtain

$$\mathbb{P}(J_\mathcal{G} - F(\hat{x}) \geq \epsilon) \leq E[J_\mathcal{G} - F(\hat{x})]/\epsilon \leq K_2\alpha^{r/4}/\epsilon$$

Since $r \geq (4|\log(K_2)| + 8|\log(\epsilon)|)|\log(\alpha)|^{-1}$, we have $K_2\alpha^{r/4} \leq \epsilon^2$ and the result follows.

3.4 Establishing the correlation decay property

The previous section motivates the search for conditions implying the correlation decay property. This section is devoted to the study of a coupling argument which can be used to show that correlation decay holds. Results in this section are for the case $|\chi| = 2$. They can be extended to the case $|\chi| \geq 2$ at the expense of heavier notations, but without providing much additional insight. For this special case $\chi = \{0, 1\}$, we introduce a set of simplifying notations as follows.

Notation

Given $\mathcal{G} = (V, E, \Phi, \{0, 1\})$ and $u \in V$, let $v_1, \ldots, v_d$ be the neighbors of u in V. For any $r > 0$ and boundary conditions $\mathcal{C}$, $\mathcal{C}'$, define:

1. $B(r) \triangleq \mathrm{CE}[\mathcal{G}, u, r, 1, \mathcal{C}]$ and $B'(r) \triangleq \mathrm{CE}[\mathcal{G}, u, r, 1, \mathcal{C}']$

2. For $j = 1, \ldots d$, let $\mathcal{G}_j = \mathcal{G}(u, j, 1)$, and let $B_j(r-1) \triangleq \mathrm{CE}[\mathcal{G}_j, v_j, r-1, 1, \mathcal{C}]$ and $B'_j(r-1) \triangleq \mathrm{CE}[\mathcal{G}_j, v_j, r-1, 1, \mathcal{C}']$. Also let $\mathbf{B}(r-1) = (B_j(r-1))_{1 \leq j \leq d}$ and $\mathbf{B}'(r-1) = (B'_j(r-1))_{1 \leq j \leq d}$

3. For $k = 1, \ldots n_j$, let $(v_{j1}, \ldots, v_{jn_j})$ be the neighbors of v_j in $\mathcal{G}_j$, and let $B_{jk}(r-2) = \mathrm{CE}[\mathcal{G}_j(v_j, k, 1), v_{jk}, r-2, 1, \mathcal{C}]$ and $B'_{jk}(r-2) = \mathrm{CE}[\mathcal{G}_j(v_j, k, 1), v_j, r-2, 1, \mathcal{C}']$ for all $k = 1 \ldots n_j$. Also let $\mathbf{B_j}(r-2) = (B_{jk}(r-2))_{1 \leq k \leq n_j}$ and $\mathbf{B'_j}(r-2) = (B'_{jk}(r-2))_{1 \leq k \leq n_j}$.

4. For simplicity, since 1 is the only action different from the reference action 0, we denote $\mu_{u \leftarrow v_j}(z) \triangleq \mu_{u \leftarrow v_j}(1, z)$.
   From Equation (1.2), note the following alternative expression for $\mu_{u \leftarrow v_j}(z)$

$$\begin{aligned} \mu_{u \leftarrow v_j}(z) = \phi_{u,v_j}(1,1) - \phi_{u,v_j}(0,1) + \max(\phi_{u,v_j}(1,0) - \phi_{u,v_j}(1,1), z) \\ - \max(\phi_{u,v_j}(0,0) - \phi_{u,v_j}(0,1), z) \end{aligned} \tag{3.5}$$

5. Similarly, for any $j = 1 \ldots d$ and $k = 1 \ldots n_j$, let $\mu_{v_j \leftarrow v_{jk}}(z) \triangleq \mu_{v_j \leftarrow v_{jk}}(1, z)$.

6. For any $\mathbf{z} = (z_1, \ldots, z_d)$, let $\mu_u(\mathbf{z}) = \sum_j \mu_{u \leftarrow v_j}(z_j)$. Also, for any j, and any $\mathbf{z} = (z_1, \ldots, z_{n_j})$, let $\mu_{v_j}(\mathbf{z}) = \sum_{1 \le k \le n_j} \mu_{v_j \leftarrow v_{jk}}(z_k)$.

7. For any directed edge $e = (u \leftarrow v)$, denote

$$\begin{aligned}
\phi_e^1 &\triangleq \phi_{u,v}(1,0) - \phi_{u,v}(1,1) \\
\phi_e^2 &\triangleq \phi_{u,v}(0,0) - \phi_{u,v}(0,1) \\
\phi_e^3 &\triangleq \phi_{u,v}(1,1) - \phi_{u,v}(0,1) \\
X_e &\triangleq \phi_e^1 + \phi_e^2 \\
Y_e &\triangleq \phi_e^2 - \phi_e^1 = \phi_{u,v}(1,1) - \phi_{u,v}(1,0) - \phi_{u,v}(0,1) + \phi_{u,v}(0,0)
\end{aligned}$$

Note that $Y_{u \leftarrow v} = Y_{v \leftarrow u}$, so we simply denote it $Y_{u,v}$.

Note that for any e, $\mathbb{E}|Y_e| \le K_\Phi$ (see Assumption 2). Equation (2.14) can be rewritten as

$$B(r) = \mu_u(\mathbf{B}(r-1)) + \phi_u(1) - \phi_u(0) \tag{3.6}$$

$$B'(r) = \mu_u(\mathbf{B}'(r-1)) + \phi_u(1) - \phi_u(0) \tag{3.7}$$

Similarly, we have

$$B_j(r-1) = \mu_{v_j}(\mathbf{B_j}(r-2)) + \phi_{v_j}(1) - \phi_{v_j}(0) \tag{3.8}$$

$$B'_j(r-1) = \mu_{v_j}(\mathbf{B'_j}(r-2)) + \phi_{v_j}(1) - \phi_{v_j}(0) \tag{3.9}$$

Finally, Equation (3.5) can be rewritten

$$\mu_{u \leftarrow v}(z) = \phi^3_{u \leftarrow v} + \max(\phi^1_{u \leftarrow v}, z) - \max(\phi^2_{u \leftarrow v}, z) \tag{3.10}$$

Y_e represents how strongly the interaction function $\phi_{u,v}(x_u, x_v)$ is "coupling" the variables x_u and x_v. In particular, if Y_e is zero, the interaction function $\phi_{u,v}(x_u, x_v)$ can be decomposed into a sum of two potential functions $\phi_u(x_u) + \phi_v(x_v)$, that is, the edge between u and v is then be superfluous and can be removed. To see why this is the case, take $\phi_u(0) = 0$, $\phi_u(1) = \phi_{u,v}(1,0) - \phi_{u,v}(0,0)$, $\phi_v(0) = \phi_{u,v}(0,0)$ and $\phi_v(1) = \phi_{u,v}(0,1)$, which is also equal to $\phi_{u,v}(1,1) - \phi_{u,v}(1,0) + \phi_{u,v}(0,0)$, since $Y_e = 0$.

3.4.1 Coupling technique

In this section, we present a sufficient condition for correlation decay. The condition depends on the parameters of a particular form of coupling: For any neighbor v_j of u, it is possible that the partial cavities $\mu_{u\leftarrow v_j}$ and μ'_j depending on two different boundary conditions $\mathcal{C}$ and $\mathcal{C}'$ be equal even when $B_j \neq B'_j$. The probability that this coupling occurs decreases as the distance between B_j and B'_j grows bigger.

Definition 2. *A network $\mathcal{G}$ is said to exhibit (a,b)-coupling with parameters (a,b) if for every edge $e=(u,v)$, and every two real values x, x':*

$$\mathbb{P}\Big(\mu_{u\leftarrow v}(x+\phi_v(1)-\phi_v(0)) = \mu_{u\leftarrow v}(x'+\phi_v(1)-\phi_v(0))\Big) \geq (1-a)-b|x-x'| \quad (3.11)$$

The probability above, and hence the coupling parameters, depends on both the distribution of $\phi_v(1)-\phi_v(0)$ and the distribution of the values $\phi_{u,v}(x,y)$. Note that if for all x, x'

$$\mathbb{P}\Big(\mu_{u\leftarrow v}(x) = \mu_{u\leftarrow v}(x')\Big) \geq (1-a)-b|x-x'| \quad (3.12)$$

then $\mathcal{G}$ exhibits (a,b) coupling, but in general the tightest coupling values found for Equation (3.12) are much weaker than the ones we would find by analyzing condition (3.11). This form of distance dependent coupling is a useful tool in proving that correlation decay occurs, as illustrated by the following theorem:

Theorem 8. *Suppose $\mathcal{G}$ exhibits (a,b)-coupling. If*

$$a(\Delta-1)+\sqrt{bK_\Phi}(\Delta-1)^{3/2} < 1 \quad (3.13)$$

then the exponential correlation decay property holds with $K=\Delta^2 K_\Phi$ and $\alpha = a(\Delta-1)+\sqrt{bK_\Phi}(\Delta-1)^{3/2}$.

Suppose $\mathcal{G}$ exhibits (a,b)-coupling and that there exists $K_Y > 0$ such that $|Y_e| \leq K_Y$ with probability 1. If

$$a(\Delta-1)+bK_Y(\Delta-1)^2 < 1 \quad (3.14)$$

then the exponential correlation decay property holds with $\alpha = a(\Delta-1)+bK_Y(\Delta-1)^2$

Proof of Theorem 8

We begin by proving several useful lemmas.

Lemma 2. *For every (u, v), and every two real values x, x'*

$$|\mu_{u\leftarrow v}(x) - \mu_{u\leftarrow v}(x')| \leq |x - x'|. \tag{3.15}$$

Proof. From (3.5) we obtain

$$\begin{aligned}\mu_{u\leftarrow v}(x) - \mu_{u\leftarrow v}(x') = \max\Big(\phi_{u,v}(1,0) - \phi_{u,v}(1,1), x\Big) - \max\Big(\phi_{u,v}(0,0) - \phi_{u,v}(0,1), x\Big) \\ - \max\Big(\phi_{u,v}(1,0) - \phi_{u,v}(1,1), x'\Big) + \max\Big(\phi_{u,v}(0,0) - \phi_{u,v}(0,1), x'\Big).\end{aligned}$$

Using twice the relation $\max_x f(x) - \max_x g(x) \leq \max_x(f(x) - g(x))$, we obtain:

$$\begin{aligned}\mu_{u\leftarrow v}(x) - \mu_{u\leftarrow v}(x') &\leq \max(0, x - x') + \max(0, x' - x) \\ &= |x - x'|\end{aligned}$$

The other inequality is proved similarly. □

Lemma 3. *For every $u, v \in V$ and every two real values x, x'*

$$|\mu_{u\leftarrow v}(x) - \mu_{u\leftarrow v}(x')| \leq |Y_{u,v}| \tag{3.16}$$

Proof. Using (3.5) and (3.7), we have

$$\begin{aligned}\mu_{u\leftarrow v}(x) - (\phi_{u,v}(1,1) - \phi_{u,v}(0,1)) = \max(\phi_{u,v}(1,0) - \phi_{u,v}(1,1), x) \\ - \max(\phi_{u,vx}(0,0) - \phi_{u,v}(0,1), x).\end{aligned}$$

By using the relation $\max_x f(x) - \max_x g(x) \leq \max_x(f(x) - g(x))$ on the right hand side, we obtain

$$\mu_{u\leftarrow v}(x) - (\phi_{u,v}(1,1) - \phi_{u,v}(0,1)) \leq \max(0, -Y_{u,v}).$$

Similarly

$$-\mu_{u\leftarrow v}(x') + (\phi_{u,v}(1,1) - \phi_{u,v}(0,1)) \leq \max(0, Y_{u,v}).$$

Adding up

$$\mu_{u\leftarrow v}(x) - \mu_{u\leftarrow v}(x') \leq |Y_{u,v}|.$$

The other inequality is also proven similarly.

Lemma 4. *Suppose (a,b)-coupling holds. Then,*

$$\mathbb{E}|B(r) - B'(r)| \leq a \sum_{1\leq j\leq d} \mathbb{E}|B_j(r-1) - B'_j(r-1)| + b \sum_{1\leq j\leq d} \mathbb{E}\big[|B_j(r-1) - B'_j(r-1)|^2\big]. \tag{3.17}$$

Proof. Using (2.14), we obtain:

$$\begin{aligned}
\mathbb{E}|B(r) - B'(r)| &= \mathbb{E}\Big[\big|\phi_u(1) - \phi_u(0) + \sum_j \mu_{u\leftarrow v_j}(B_j(r-1)) - (\phi_u(1) - \phi_u(0)) - \sum_j \mu_{u\leftarrow v_j}(B'_j(r-1))\big|\Big] \\
&\leq \sum_j \mathbb{E}\big|\mu_{u\leftarrow v_j}(B_j(r-1)) - \mu_{u\leftarrow v_j}(B'_j(r-1))\big| \\
&= \sum_j \mathbb{E}\Big[\mathbb{E}\big[|\mu_{u\leftarrow v_j}(B_j(r-1)) - \mu_{u\leftarrow v_j}(B'_j(r-1))|\big|\mu_{v_j}(\mathbf{B_j}(r-2), \mu_{v_j}(\mathbf{B'_j}(r-2)\big]\Big]
\end{aligned}$$

By Lemma 2, we have $|\mu_{u\leftarrow v_j}(B_j(r-1)) - \mu_{u\leftarrow v_j}(B'_j(r-1))| \leq |B_j(r-1) - B'_j(r-1)|$. Also note from that from Equation (3.8) and (3.9), $|B_j(r-1) - B'_j(r-1)| = |\mu_{v_j}(\mathbf{B_j}(r-2)) - \mu_{v_j}(\mathbf{B'_j}(r-2))|$; hence conditional on both $\mu_{v_j}(\mathbf{B_j}(r-2)$ and $\mu_{v_j}(\mathbf{B'_j}(r-2)$, $|B_j(r-1) - B'_j(r-1)|$ is a constant. Therefore,

$$\begin{aligned}
&\mathbb{E}\Big[\big|\mu_{u\leftarrow v_j}(B_j(r-1)) - \mu_{u\leftarrow v_j}(B'_j(r-1))\big| \ \Big|\mu_{v_j}(\mathbf{B_j}(r-2), \mu_{v_j}(\mathbf{B'_j}(r-2)\Big] \qquad (3.18)\\
&\leq |B_j(r-1) - B'_j(r-1)|\, \mathbb{P}(\mu_{u\leftarrow v_j}(B_j(r-1)) \neq \mu_{u\leftarrow v_j}(B'_j(r-1)) \mid \mu_{v_j}(\mathbf{B_j}(r-2), \mu_{v_j}(\mathbf{B'_j}(r-2))
\end{aligned}$$

Note that in the (a,b) coupling definition, the probability is over the values of the functions ϕ_{u,v_j}, and ϕ_v. By Proposition 6, these are independent from $\mu_{v_j}(\mathbf{B_j}(r-2)$ and $\mu_{v_j}(\mathbf{B'_j}(r-2))$. Thus, by the (a,b) coupling assumption, $\mathbb{P}(\mu_{u\leftarrow v_j}(B_j(r-1)) \neq \mu_{u\leftarrow v_j}(B'_j(r-1)) \,|\, \mu_{v_j}(\mathbf{B_j}(r-2), \mu_{v_j}(\mathbf{B'_j}(r-2)) \leq a + b|B_j(r-1) - B'_j(r-1)|$. The result then follows. □

Fix an arbitrary node u in $\mathcal{G}$. Let $\mathcal{N}(u) = \{v_1, \ldots, v_d\}$. Let $d_j = |\mathcal{N}(v_j)| - 1$ be the number of neighbors of v_j in $\mathcal{G}$ other than u for $j = 1, \ldots, d$. We need to establish that for

every two boundary condition $\mathcal{C}, \mathcal{C}'$

$$\mathbb{E}|\mathrm{CE}(\mathcal{G}, u, r, \mathcal{C}) - \mathrm{CE}(\mathcal{G}, u, r, \mathcal{C}')| \leq K\alpha^r \tag{3.19}$$

We first establish the bound inductively for the case $d \leq \Delta - 1$. Let e_d denote the supremum of the left-hand side of (3.19), where the supremum is over all networks $\mathcal{G}'$ with degree at most Δ, such that the corresponding constant $K_{\Phi'} \leq K_\Phi$, over all nodes u in $\mathcal{G}$ with degree $|\mathcal{N}(u)| \leq \Delta - 1$ and all over all choices of boundary conditions $\mathcal{C}, \mathcal{C}'$. Each condition corresponds to a different recursive inequality for e_r

Condition (3.13)

Under (3.13), we claim that

$$e_r \leq a(\Delta - 1)e_{r-1} + b(\Delta - 1)^3 K_\Phi e_{r-2} \tag{3.20}$$

Applying (3.8) and (3.9), we have

$$|B_j(r-1) - B'_j(r-1)| \leq \sum_{1 \leq k \leq d_j} |\mu_{v_j \leftarrow v_{jk}}(B_{jk}(r-2)) - \mu_{v_j \leftarrow v_{jk}}(B'_{jk}(r-2))|$$

Thus,

$$\begin{aligned} |B_j(r-1) - B'_j(r-1)|^2 &\leq \Big(\sum_{1 \leq k \leq d_j} |\mu_{v_j \leftarrow v_{jk}}(B_{jk}(r-2)) - \mu_{v_j \leftarrow v_{jk}}(B'_{jk}(r-2))| \Big)^2 \\ &\leq d_j \sum_{1 \leq k \leq d_j} |\mu_{v_j \leftarrow v_{jk}}(B_{jk}(r-2)) - \mu_{v_j \leftarrow v_{jk}}(B'_{jk}(r-2))|^2 \end{aligned}$$

By Lemmas 2 and 3 we have $|\mu_{v_j \leftarrow v_{jk}}(B_{jk}(r-2)) - \mu_{v_j \leftarrow v_{jk}}(B'_{jk}(r-2))| \leq |B_{jk}(r-2) - B'_{jk}(r-2)|$ and $|\mu_{v_j \leftarrow v_{jk}}(B_{jk}(r-2)) - \mu_{v_j \leftarrow v_{jk}}(B'_{jk}(r-2))| \leq |Y_{jk}|$. Also, $d_j \leq \Delta - 1$.Therefore,

$$|B_j(r-1) - B'_j(r-1)|^2 \leq (\Delta - 1) \sum_{1 \leq k \leq d_j} |B_{jk}(r-2) - B'_{jk}(r-2)| \,.\, |Y_{jk}| \tag{3.21}$$

By Proposition 6, the random variables $|B_{jk}(r-2) - B'_{jk}(r-2)|$ and $|Y_{jk}|$ are independent.

We obtain:

$$\begin{aligned}\mathbb{E}|B_j(r-1) - B'_j(r-1)|^2 \leq &(\Delta-1)\sum_{1\leq k\leq d_j}\mathbb{E}|B_{jk}(r-2) - B'_{jk}(r-2)|\,.\,\mathbb{E}|Y_{jk}| \qquad (3.22)\\ \leq &(\Delta-1)K_\Phi(\sum_{1\leq k\leq d_j}\mathbb{E}|B_{jk}(r-2) - B'_{jk}(r-2)|)\\ \leq &(\Delta-1)^2K_\Phi e_{r-2}\end{aligned}$$

where the second inequality follows from the definition of K_Φ and the third inequality follows from the definition of e_r and the fact that the neighbors v_{jk}, $1 \leq k \leq d_j$ of v_j have degrees at most $\Delta - 1$ in the corresponding networks for which $B_{jk}(r-2)$ and $B'_{jk}(r-2)$ were defined. Applying Lemma 4 and the definition of e_r, we obtain

$$\begin{aligned}\mathbb{E}|B(r) - B'(r)| &\leq a\sum_{1\leq j\leq d}\mathbb{E}|B_j(r-1) - B'_j(r-1)| + b\sum_{1\leq j\leq d}\mathbb{E}\left[|B_j(r-1) - B'_j(r-1)|^2\right]\\ &\leq a(\Delta-1)e_{r-1} + b(\Delta-1)^3K_\Phi e_{r-2}\end{aligned}$$

This implies (3.20).

From (3.20) we obtain that $e_r \leq K\alpha^r$ for $K = \Delta K_\Phi$ and α given as the largest in absolute value root of the quadratic equation $\alpha^2 = a(\Delta-1)\alpha + b(\Delta-1)^3K_\Phi$. We find this root to be

$$\begin{aligned}a &= \frac{1}{2}(a(\Delta-1) + \sqrt{a^2(\Delta-1)^2 + 4b(\Delta-1)^3K_\Phi})\\ &\leq a(\Delta-1) + \sqrt{b(\Delta-1)^3K_\Phi}\\ &< 1\end{aligned}$$

where the last inequality follows from assumption (3.13). This completes the proof for the case where the degree d of u is at most $\Delta - 1$.

Now suppose $d = |\mathcal{N}(u)| = \Delta$. Applying (3.6) and (3.7) we have

$$|B(r) - B'(r)| \leq \sum_{1\leq j\leq d}|\mu_{u\leftarrow v_j}(B_j(r-1) - \mu_{u\leftarrow v_j}(B'_j(r-1))|$$

Applying again Lemma 2, the right-hand side is at most

$$\sum_{1\leq j\leq d} |B_j(r-1) - B'_j(r-1)| \leq \Delta e_{r-1}$$

since $B_j(r-1)$ and $B'_j(r-1)$ are defined for v_j in a subnetwork $\mathcal{G}_j = \mathcal{G}(u, j, 1)$, where v_j has degree at most $\Delta - 1$. Thus, the correlation decay property again holds for u with ΔK replacing K.

Condition (3.14) Recall from Lemma 4 that for all r, we have:

$$\mathbb{E}|B(r) - B'(r)| \leq a \sum_{1\leq j\leq d} \mathbb{E}|B_j(r-1) - B'_j(r-1)| + b \sum_{1\leq j\leq d} \mathbb{E}\big[|B_j(r-1) - B'_j(r-1)|^2\big].$$

For all j, $|B_j(r-1) - B'_j(r-1)| = |\sum_k (\mu_{v_j \leftarrow v_{jk}}(B_{jk}) - \mu_{v_j \leftarrow v_{jk}}(B'_{jk}))|$. Moreover, for each j, k, $|\mu_{v_j \leftarrow v_{jk}}(B_{jk}) - \mu_{v_j \leftarrow v_{jk}}(B'_{jk})| \leq |Y_{jk}| \leq K_Y$ (the second inequality follows from Lemma 3, the third by assumption). As a result,

$$|B_j(r-1) - B'_j(r-1)|^2 \leq (\Delta - 1) K_Y |B_j(r-1) - B_j(r-1)|$$

We obtain:

$$e_r \leq (a + bK_Y(\Delta - 1))\,(\Delta - 1) e_{r-1}$$

Since $a(\Delta - 1) + bK_Y(\Delta - 1)^2 < 1$, e_r goes to zero exponentially fast. The same reasoning as previously shows that this property implies correlation decay.

3.4.2 Establishing coupling bounds

Coupling Lemma

Theorem 8 details sufficient condition under which the distance-dependent coupling induces correlation decay (and thus efficient decentralized algorithms, vis-à-vis Proposition 7 and Theorem 7). It remains to show how can we prove coupling bounds. The following simple observation can be used to achieve this goal.

For any edge $(u, v) \in \mathcal{G}$, and any two real numbers x, x', consider the following events

$$E^+_{u\leftarrow v}(x, x') = \{\min(x, x') + \phi_v(1) - \phi_v(0) \geq \max(\phi^1_{u\leftarrow v}, \phi^2_{u\leftarrow v})\}$$

$$E^-_{u\leftarrow v}(x,x') = \{\max(x,x') + \phi_v(1) - \phi_v(0) \leq \min(\phi^1_{u\leftarrow v}, \phi^2_{u\leftarrow v})\}$$
$$E_{u\leftarrow v}(x,x') = E^+_{u,v}(x,x') \cup E^-_{u,v}(x,x')$$

Lemma 5. *If $E_{u\leftarrow v}(x,x')$ occurs, then $\mu_{u\leftarrow v}(x+\phi_v(1)-\phi_v(0)) = \mu_{u\leftarrow v}(x'+\phi_v(1)-\phi_v(0))$. Therefore*

$$P(\mu_{u\leftarrow v}(x + \phi_v(1) - \phi_v(0)) = \mu_{u\leftarrow v}(x' + \phi_v(1) - \phi_v(0)) \geq P(E_{u\leftarrow v}(x,x'))$$

Proof. From representation (3.10), we have $\mu_{u\leftarrow v}(x) = \phi^3_{u\leftarrow v}+\max(\phi^1_{u\leftarrow v}, z)-\max(\phi^2_{u\leftarrow v}, z)$; let x, x' be any two reals. If both x and x' are greater than both $\phi^1_{u\leftarrow v}$ and $Phi^2_{u\leftarrow v}$, then $\mu_{u\leftarrow v}(x) = \phi^3_{u\leftarrow v} = \mu_{u\leftarrow v}(x')$. If both x and x' are smaller than both $\phi^1_{u\leftarrow v}$ and $Phi^2_{u\leftarrow v}$, then $\mu_{u\leftarrow v}(x) = \phi^3_{u\leftarrow v} + \phi^1_{u\leftarrow v} - \phi^2_{u\leftarrow v} = \mu_{u\leftarrow v}(x')$. The result follows from applying the above observation to $x + \phi_v(1) - \phi_v(0)$ and $x' + \phi_v(1) - \phi_v(0)$. □

Note that Lemma 5 implies that the probability that coupling does not occur $P(\mu_{u\leftarrow v}(x+\phi_v(1) - \phi_v(0)) \neq \mu_{u\leftarrow v}(x' + \phi_v(1) - \phi_v(0)))$ is upper bounded by the probability of $(E_{u\leftarrow v}(x,x'))^c$. When obvious from context, we drop the subscript $u \leftarrow v$. We will often use the following description of $(E_(x,x'))^c$: for two real values $x \geq x'$,

$$(E(x,x'))^c =$$
$$\{\min(\phi^1,\phi^2) + \phi_v(0) - \phi_v(1) < x < \max(\phi^1,\phi^2) + \phi_v(0) - \phi_v(1) + x - x'\} \qquad (3.23)$$

Uniform Distribution: Proof of Theorem 5

In order to prove Theorem 5, we compute the coupling parameters a, b for this distribution and apply the second form of Theorem 8.

Lemma 6. *The network with uniformly distributed rewards described in section 3.2 exhibits (a,b) coupling with $a = \frac{I_2}{2I_1}$ and $b = \frac{1}{2I_1}$.*

Proof. For any fixed edge $(u,v) \in \mathcal{G}$, $\phi^1_{u\leftarrow v}$ and $\phi^2_{u\leftarrow v}$ are i.i.d. random variables with a triangular distribution with support $[-2I_2, 2I_2]$ and mode 0. Because $\phi^1_{u\leftarrow v}$ and $\phi^2_{u\leftarrow v}$ are

i.i.d., by symmetry we obtain:

$$\mathbb{P}((E(x,x'))^c) =$$
$$2\int_{-2I_2}^{2I_2} d\mathbb{P}_{\phi^1}(a_1)\int_{a_1}^{2I_2} d\mathbb{P}_{\phi^2}(a_2)\, P(a_1+\phi_v(0)-\phi_v(1)<x<\phi_v(0)-\phi_v(1)+a_2+x-x') =$$
$$2\int_{-2I_2}^{2I_2} d\mathbb{P}_{\phi^1}(a_1)\int_{a_1}^{2I_2} d\mathbb{P}_{\phi^2}(a_2)\, P(x'-a_2<\phi_v(0)-\phi_v(1)<x-a_1)$$

The quantity $P(x'-a_2<\phi_v(0)-\phi_v(1)<x-a_1)$ can be upper bounded by $\frac{a_2-a_1+x-x'}{2I_1}$, and we obtain:

$$P(E(x,x')^c) \;\leq\; \frac{x-x'}{2I_1}+\frac{1}{I_1}\int_{-2I_2}^{2I_2} d\mathbb{P}_{\phi^1}(a_1)\int_{a_1}^{2I_2} d\mathbb{P}_{\phi^2}(a_2)(a_2-a_1)$$

Note that $d\mathbb{P}_{\phi^2}(a_2)=\frac{1}{4I_2^2}(a_2+2I_2)d(a_2)$ for $a_2\leq 0$, and $d\mathbb{P}_{\phi^2}(a_2)=\frac{1}{4I_2^2}(2I_2-a_2)d(a_2)$ for $a_2\geq 0$; identical expressions hold for $d\mathbb{P}_{\phi^1}(a_1)$. Therefore, for $a_1\geq 0$,

$$\begin{aligned}\int_{a_1}^{2I_2} d\mathbb{P}_{\phi^2}(a_2)(a_2-a_1) &= \frac{1}{4I_2^2}\int_{a_1}^{2I_2}(2I_2-a_2)(a_2-a_1)\,d(a_2)\\ &= \frac{1}{4I_2^2}\Big(-\int_{a_1}^{2I_2}(2I_2-a_2)^2 d(a_2)+(2I_2-a_1)\int_{a_1}^{2I_2}(2I_2-a_2)d(a_2)\Big)\\ &= \frac{1}{4I_2^2}\Big(-\frac{1}{3}(2I_2-a_1)^3+\frac{1}{2}(2I_2-a_1)^3\Big)=\frac{1}{24I_2^2}(2I_2-a_1)^3\end{aligned}$$

Similarly, for $a_1\leq 0$,

$$\int_{a_1}^{2I_2} d\mathbb{P}_{\phi^2}(a_2)(a_2-a_1) = -a_1+\frac{1}{24I_2^2}(a_1+2I_2)^3$$

The final integral is therefore equal to:

$$\begin{aligned}&\int_{-2I_2}^{2I_2} d\mathbb{P}_{\phi^1}(a_1)\int_{a_1}^{2I_2} d\mathbb{P}_{\phi^2}(a_2)(a_2-a_1)\\ =\;& \frac{1}{4I_2^2}\Big(\int_{-2I_2}^{0}\big((a_1+2I_2)(-a_1+\frac{1}{24I_2^2}(a_1+2I_2)^3\big)d(a_1)+\int_0^{2I_2}\frac{1}{24I_2^2}(2I_2-a_1)^4 d(a_1)\Big)\\ =\;& \frac{1}{4I_2^2}\Big(\frac{24}{15}I_2^3+\frac{4}{15}I_2^3\Big)=\frac{7}{15}I_2\end{aligned}$$

Finally,

$$P((E(x,x')^c) \leq \frac{7I_2}{15I_1} + \frac{|x-x'|}{2I_1} \leq \frac{I_2}{2I_1} + \frac{|x-x'|}{2I_1}$$

Therefore, the system exhibits coupling with parameters $(\frac{I_2}{2I_1}, \frac{1}{2I_1})$. □

We can now finish the proof of Theorem 5. For all $(u,v) \in E$ and $x, y \in \chi$, $|\phi_{u,v}(x,y)| \leq I_2$. Therefore, for any (u,v), $|Y_{u,v}| = |\phi_{u,v}(1,1) - \phi_{u,v}(0,1) - \phi_{u,v}(1,0) + \phi_{u,v}(0,0)| \leq 4I_2$.

Note that for all edges, $|Y_e| \leq 4I_2$, so that the condition $\beta(\Delta-1)^2 < 1$ implies $\frac{I_2}{2I_1}(\Delta - 1) + \frac{4I_2}{2I_1}(\Delta-1)^2 < 1$. Since $(\Delta-1) \leq (\Delta-1)^2$, if $\beta(\Delta-1)^2 < 1$ we also have $\frac{I_2}{2I_1}(\Delta - 1) + \frac{4I_2}{2I_1}(\Delta-1)^2 < 1$. This is exactly condition (3.14) with a, b as given by Lemma 6 and $K_Y = 4I_2$. It follows that $\mathcal{G}$ exhibits exponential correlation decay, and since Assumptions 1 and 2 hold, all conditions of Corollary 2 are satisfied, and there exists an additive FPTAS for computing $J_{\mathcal{G}}$.

Gaussian distribution: Proof of Theorem 6

In this section, we compute the coupling parameters for Gaussian distributed reward functions. Rather than considering only the assumptions of Theorem 6, we adopt a more general framework. The proof will then follow from the application of Theorem 8 (first condition) and a special case of the computation detailed below (see Corollary 3). Assume that for every edge $e = (u,v)$ the value functions $(\phi_{u,v}(0,0), \phi_{u,v}(0,1), \phi_{u,v}(1,0), \phi_{u,v}(1,1))$ are independent, identically distributed four-dimensional Gaussian random variables, with mean $\mu = (\mu_i)_{i \in \{00,01,10,11\}}$, and covariance matrix $S = (S_{ij})_{i,j \in \{00,01,10,11\}}$. For every node $v \in V$, suppose $\phi_v(1) = 0$ and that $\phi_v(0)$ is a Gaussian random variable with mean μ_p and standard deviation σ_p. Moreover, suppose all the ϕ_v and ϕ_e are independent for $v \in V$, $e \in E$. Let

$$\sigma_1^2 = S_{10,10} - 2S_{10,11} + S_{11,11} + \sigma_p^2 \qquad \sigma_2^2 = S_{00,00} - 2S_{00,01} + S_{01,01} + \sigma_p^2$$

$$\rho = (\sigma_1\sigma_2)^{-1}(S_{00,10} - S_{00,11} - S_{01,10} + S_{01,11} + \sigma_p^2) \qquad C = \frac{\sigma_2^2 - \sigma_1^2}{\sqrt{(\sigma_1^2+\sigma_2^2)^2 - 4\rho^2\sigma_1^2\sigma_2^2}}$$

$$\sigma_X^2 = \sigma_1^2 + \sigma_2^2 + 2\rho\sigma_1\sigma_2 \qquad \sigma_Y^2 = \sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2$$

Proposition 10. *Assume $C < 1$. Then the network exhibits coupling with parameters*

(a, b) *equal to:*

$$a = \frac{1}{\pi} \arctan\left(\sqrt{\frac{1}{1-C^2}}\frac{\sigma_Y}{\sigma_X}\right) + \sqrt{\frac{2}{\pi}}\frac{|\mu_{00} + \mu_{11} - \mu_{10} - \mu_{01}|}{\sigma_X}$$
$$b = \sqrt{\frac{2}{\pi}}\frac{1}{\sigma_X}$$

Corollary 3. *Suppose that for each* $e, (\phi_e(0,0), \phi_e(0,1), \phi_e(1,0), \phi_e(1,1))$ *are i.i.d. Gaussian variables with mean* 0 *and standard deviation* σ_e. *Let* $\beta = \sqrt{\frac{\sigma_e^2}{\sigma_e^2+\sigma_p^2}}$ *Then* $a \leq \beta$ *and* $bK_\phi \leq \beta$.

Proof. Under the conditions of corollary 3, we have $\sigma_Y^2 = 4\sigma_e^2$, $\sigma_X^2 = 4\sigma_p^2 + 4\sigma_e^2$, and $C = 0$. Note also that $K_\Phi \leq 2\sigma_e$ By Proposition 10, the network exhibits coupling with parameters

$$a = \frac{1}{\pi} \arctan\left(\sqrt{\frac{\sigma_e^2}{\sigma_e^2 + \sigma_p^2}}\right) \leq \frac{1}{\pi}\beta \leq \beta$$
$$b = \sqrt{\frac{1}{2\pi}}\frac{1}{\sqrt{\sigma_e^2 + \sigma_p^2}} \text{ and so, } bK_\Phi \leq \sqrt{\frac{2}{\pi}}\beta \leq \beta$$

□

Note that if $\sigma_e \to 0$, then $\beta \to 0$ and correlation decay takes place; moreover, combining Corollary 3 and Theorem 8 (condition (3.13)) directly yields Theorem 6.

Proof of Proposition 10 . Fix an edge (u, v) in E; for simplicity, in the rest of this section denote $\bar{\phi}^1 = \phi^1_{u \leftarrow v} + \phi_v(0) - \phi_v(1)$ and $\bar{\phi}^2 = \phi^2_{u \leftarrow v} + \phi_v(0) - \phi_v(1)$. It follows that $(\bar{\phi}^1, \bar{\phi}^2)$ follows a bivariate Gaussian distribution with mean (μ_1, μ_2):

$$\mu_1 = \mu_{10} - \mu_{11} + \mu_p \text{ and } \mu_2 = \mu_{00} - \mu_{01} + \mu_p$$

and covariance matrix

$$S_A = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

Let $X = \overline{\phi}^1 + \overline{\phi}^2$, $Y = \overline{\phi}^2 - \overline{\phi}^1$. Then, (X, Y) is a bivariate Gaussian vector with means $\mathbb{E}[X] = \mu_1 + \mu_2$ and $\mathbb{E}[Y] = \mu_2 - \mu_1$, standard deviations σ_X, σ_Y and correlation C as defined previously. Denote also $\overline{X} \triangleq X - E[X]$ and $\overline{Y} \triangleq Y - E[Y]$, the centered versions

of X and Y. Consider two real numbers $x \geq x'$, and let (b, t) be the two real numbers such that $x = b + t/2$, $x' = b - t/2$. From Equation (3.23), we have

$$(E(x, x'))^c = \{\min(\overline{\phi}^1, \overline{\phi}^2) - t/2 < b < \max(\overline{\phi}^1, \overline{\phi}^2) + t/2\}$$

The first step of the proof consists in rewriting the event $(E(x, x'))^c$ in terms of the variables X, Y:

Lemma 7.

$$(E(x, x'))^c = \{|Y| \geq |X - 2b| - t\}$$

Proof.

$$\begin{aligned}
(E(x, x'))^c =& \{\min(\overline{\phi}^1, \overline{\phi}^2) - t/2 < b < \max(\overline{\phi}^1, \overline{\phi}^2) + t/2\} \\
=& \{\overline{\phi}^1 - t/2 < b < \overline{\phi}^2 + t/2, \overline{\phi}^1 \leq \overline{\phi}^2\} \cup \{\overline{\phi}^2 - t/2 < b < \overline{\phi}^1 + t/2, Y \leq 0, \overline{\phi}^2 \leq \overline{\phi}^1\} \\
=& \{2\overline{\phi}^1 - t < 2b < 2\overline{\phi}^2 + t, \overline{\phi}^1 \leq \overline{\phi}^2\} \cup \{2\overline{\phi}^2 - t < 2b < 2\overline{\phi}^1 + t, \overline{\phi}^2 \leq \overline{\phi}^1\} \\
=& \{X - Y - t < 2b < X + Y + t, Y \geq 0\} \cup \{X + Y - t < 2b < X - Y + t, Y \leq 0\} \\
=& \{(X - 2b) - |Y| - t < 0 < (X - 2b) + |Y| + t\} \\
=& \{|Y| \geq (X - 2b - t)\} \cap \{|Y| \geq (2b - X - t)\} \\
=& \{|Y| \geq |X - 2b| - t\}
\end{aligned}$$

□

For any b and $t \geq 0$, let $S(t) = \{x, y : |y| \geq |x| - t\}$, and for any real x, let $S(t, y) = \{x : |y| \geq |x| - t\}$. Note that $S(t, y)$ is symmetric and convex in x for all y. Using the lemma, we obtain:

$$\begin{aligned}
\mathbb{P}((E)^c(x, x')) =& \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1 - C^2}} \int_{S(t)} \exp(-\frac{1}{2(1 - C^2)}(\frac{(x - \mu_1 - \mu_2 + 2b)^2}{\sigma_x^2} + \frac{(y - \mu_2 + \mu_1)^2}{\sigma_y^2} \\
& - 2C\frac{(x - \mu_1 - \mu_2 + 2b)(y + \mu_2 - \mu_1)}{\sigma_x\sigma_y}))dxdy \\
=& \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1 - C^2}} \int_y \exp(-\frac{1}{2(1 - C^2)}\frac{(y - \mu_2 + \mu_1)^2}{\sigma_y^2})\, g(y)dy \qquad (3.24)
\end{aligned}$$

where:

$$g(y) = \int_{x \in S(t,y)} \exp(-\frac{1}{2(1-C^2)}(\frac{(x-\mu_1-\mu_2+2b)^2}{\sigma_x^2} - 2C\frac{(x-\mu_1-\mu_2+2b)(y-\mu_2+\mu_1)}{\sigma_x\sigma_y}))dx$$

Let $\tilde{x_b} = \frac{(x-\mu_1-\mu_2+2b)}{\sigma_x}$ and $\tilde{y} = \frac{(y-\mu_2+\mu_1)}{\sigma_y}$. Then:

$$g(y) = \exp(\frac{C^2}{2(1-C^2)}\tilde{y}^2) \int_{x \in S(t,y)} \exp(-\frac{1}{2(1-C^2)}(\tilde{x_b} - C\tilde{y})^2)dx$$

Now, $\tilde{x_b} - C\tilde{y} = \frac{x-\mu_1-\mu_2+2b-\frac{C\sigma_x(y-\mu_2+\mu_1)}{\sigma_y}}{\sigma_x}$. Recall Anderson's inequality [Dud99]: let γ be a centered Gaussian measure on $\mathbb{R}^k$, and S be a convex, symmetric subset of $\mathbb{R}^k$. Then, for all z, $\gamma(S) \geq \gamma(S+z)$. Since $S(t,y)$ is a convex symmetric subset, by setting $2b = \mu_1 + \mu_2 + \frac{C\sigma_x(y-\mu_2+\mu_1)}{\sigma_y}$, it follows that

$$g(y) \leq \exp(\frac{C^2}{2(1-C^2)}\tilde{y}^2) \int_{x \in S(t,y)} \exp(-\frac{1}{2\sigma_x^2(1-C^2)}x^2)dx$$

Injecting that bound in Equation (3.24), we obtain:

$$\begin{aligned}
\mathbb{P}((E)^c(x,x')) \leq & \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-C^2}} \int_y \exp(-\frac{1}{2(1-C^2)}\frac{(y-\mu_2+\mu_1)^2}{\sigma_y^2}) \\
& \Big(\exp(\frac{C^2}{2(1-C^2)}\frac{(y-\mu_2+\mu_1)^2}{\sigma_y^2}) \int_{x \in S(t,y)} \exp(-\frac{1}{2\sigma_x^2(1-C^2)}x^2)dx\Big)dy \\
\leq & \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-C^2}} \int_{S(t)} \exp(-\frac{1}{2(1-C^2)}(\frac{x^2}{\sigma_x^2} + (1-C^2)\frac{(y-\mu_2+\mu_1)^2}{\sigma_y^2}))dxdy
\end{aligned}$$

Finally, note that the triangular inequality, for any α we have $S(t) \subset S_\alpha(t) \triangleq \{(x,y) : |y-\alpha| \geq |x| - t - |\alpha|\}$. We obtain:

$$\begin{aligned}
\mathbb{P}((E)^c(x,x')) \leq & \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-C^2}} \int_{S_{\mu_2-\mu_1}(t)} \exp(-\frac{1}{2(1-C^2)}(\frac{x^2}{\sigma_x^2} + (1-C^2)\frac{(y-\mu_2+\mu_1)^2}{\sigma_y^2}))dxdy \\
\leq & \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-C^2}} \int_{S(t+|\mu_2-\mu_1|)} \exp(-\frac{1}{2(1-C^2)}(\frac{x^2}{\sigma_x^2} + (1-C^2)\frac{y^2}{\sigma_y^2}))dxdy
\end{aligned}$$

where the second inequality follows from a simple change of variable. Let $t' = t + |\mu_2 - \mu_1|$

Finally, we decompose $S(t')$ as the union of two sets: $S(t) = S_{\text{int}}(T) \cup S_{\text{out}}(t)$, where:

$$\begin{aligned} S_{\text{int}}(t') =& \{(X,Y) : |X| < t'\} \\ S_{\text{out}}(t') =& \{(X,Y) : |X| \geq t' \text{ and } |Y| \geq (|X| - t')\}, \end{aligned}$$

and note that $S_{\text{int}}(t') \cap S_{\text{out}}(t') = \emptyset$. We have:

$$\mathbb{P}(S_{\text{int}}(t')) \leq \frac{2t'}{\sqrt{2\pi(1-C^2)}\sigma_x}$$

and, by symmetry of $S_{\text{out}}(t')$ in X and Y,

$$\begin{aligned} \mathbb{P}(S_{\text{out}}(t')) =& 4\mathbb{P}(\{(x,y) : x \geq t, y \geq 0, y \geq x - t\}) \\ =& \frac{2}{\pi\sigma_x\sigma_y\sqrt{1-C^2}} \int_{\{(x,y):x\geq t, y\geq 0, y\geq x-t\}} \exp(-\frac{1}{2(1-C^2)}(\frac{x^2}{\sigma_x^2} + (1-C^2)\frac{y^2}{\sigma_y^2}))\, dxdy \end{aligned}$$

Using the change of variables $(x', y') = (\frac{x-t}{\sqrt{1-C^2}\sigma_x}, \frac{y}{\sigma_y})$,we get:

$$\mathbb{P}(S_{\text{out}}(t')) = \frac{2}{\pi} \int_{\{(x',y'):x'>0, y'>0, y' \geq \frac{\sigma_x\sqrt{1-C^2}}{\sigma_y}x'\}} \Big(\exp(-(x' + \frac{t'}{\sqrt{1-C^2}\,\sigma_x})^2 - y'^2) \Big) dx'dy'$$

Since $(x' + \frac{t'}{\sqrt{1-C^2}\,\sigma_x})^2 \geq x'^2$, it follows that:

$$\mathbb{P}(S_{\text{out}}(t')) \leq \frac{2}{\pi} \int_{\{(x',y'):x'>0, y'>0, y' \geq \frac{\sigma_x\sqrt{1-C^2}}{\sigma_y}x'\}} \Big(\exp(-x'^2 - y'^2) \Big) dxdy$$

By using a radial change of variables $(x', y') = (r\cos(\theta), r\sin(\theta))$ we can compute exactly the expression above, and find:

$$\begin{aligned} \mathbb{P}(S_{\text{out}}(t')) \leq& \frac{2}{\pi} \int_{\{(r,\theta):r>0, \arctan(\frac{\sigma_x\sqrt{1-C^2}}{\sigma_y}) \leq \theta \leq \frac{\pi}{2}\}} \exp(-r^2) r dr d\theta \\ =& \frac{1}{\pi} \arctan(\frac{\sigma_y}{\sigma_x\sqrt{1-C^2}}) \end{aligned}$$

$$\mathbb{P}((E)^c(x,x')) \leq \left(\frac{1}{\pi}\arctan(\frac{\sigma_y}{\sigma_x\sqrt{1-C^2}}) + \sqrt{\frac{2}{\pi(1-C^2)}}\frac{|\mu_2-\mu_1|}{\sigma_x}\right) + \sqrt{\frac{2}{\pi(1-C^2)}}\frac{t}{\sigma_x} \tag{3.25}$$

which gives us the desired bounds on (a,b). □

3.5 Decentralization

In this section, we consider another property obtained as a result of the correlation decay property, specifically, the decentralization of optimal solutions. In economics, the field of team theory [Mar55, Rad62, MR72] tries to quantify the minimal suboptimality losses incurred by a team of agents when each of them takes a decision with limited information. For instance, we may consider a set of decision networks $\mathcal{D}$ and a function i from $\{(\mathcal{G},v), \mathcal{G} \in \mathcal{D}, v \in \mathcal{G}\}$ to some set called information set $\mathcal{I}$. We explicitly assume that i is not one-to-one. Then, we consider some scheme $\psi : \mathcal{I} \mapsto \chi$ and assume that each agent v in a network $\mathcal{G}$ of $\mathcal{D}$ uses the scheme ψ to choose its decision: $\forall \mathcal{G} \in \mathcal{D}, \forall v \in \mathcal{G}, x_v = \psi(f(\mathcal{G}))$. The main research question is to devise a scheme ψ which minimizes some measure of the suboptimality loss $F_{\mathcal{G}}(x) - F_{\mathcal{G}}(\psi(f(\mathcal{G})))$.

An interesting question raised in [RR03] asks what the cost of decentralization is for a team of agents. In other words, if we assume that each node only receives local information on the network topology and costs, what kind of performance can the team attain? For any node v and integer $r \geq 0$, we recall that $\mathcal{N}_v^r$ denotes the subnetwork induced by $\mathcal{B}_v^r$. We call decentralized algorithm ψ^r of radius r a function that takes as input a local neighborhood $\mathcal{N}_v^r$ and outputs a decision $x \in \chi$. The corresponding decentralized solution is $\mathbf{x}^r$, defined by $\mathbf{x}^r(v) = \psi^r(\mathcal{N}_v^r)$ for any $v \in V$. A decision made with only partial information is likely to be suboptimal, and precisely how much is lost by discarding nonlocal information is measured by the following quantity:

$$\frac{1}{|V|+|E|}E\left[F_{\mathcal{G}}(x)\right] - E\left[F_{\mathcal{G}}(x^r)\right]$$

It should be clear at this point that the CE algorithm in fact provides a decentralized solution, that is, $x_v^r = \text{argmax}_x \text{CE}[\mathcal{G},u,r,x]$ is a decentralized decision of radius r for node v. Therefore, another way to interpret Theorem 7 is that the decentralization suboptimality loss is essentially upper-bounded by the rate of correlation decay.

The main result of [RR03] states that for a chain of agents (i.e., a decision network for which the graph is a line), the cost of decentralization using randomized decentralized algorithms can be upper bounded by $\frac{1}{r^\alpha}$ for any $0 < \alpha < 1$. We will show that their result is a special case of our coupling theorem. Assume that the functions $\phi_{u,v}$ and ϕ_v are deterministic and bounded by K_Φ, and that $\Delta = 2$, so that (V, E) is a disjoint union of path and cycles. For given $r \in \mathbb{N}_+$ and $\delta > 0$, construct $\mathbf{x}^{r,\delta}$ as follows:

1. For each node $v \in V$, force x_v to 0 with probability δ, and leave x_v undecided otherwise

2. For each undecided node $v \in V$, run the cavity algorithm with depth r and choose the action which maximizes the approximate cavity function obtained by the cavity algorithm.

Proposition 11. *If $\Delta = 2$ and if for all $(u, v) \in E$, $x_u, x_v \in \chi$, we have $|\phi_{u,v}(x_u, x_v)| \leq K_\Phi$ a.s. and $|\phi_u(x_u)| \leq K_\Phi$ a.s., then for any $r > 0$, there exists $\delta > 0$ such that the suboptimality gap of $\mathbf{x}^{r,\delta}$ is bounded by $L(|V| + |E|)\frac{\log r}{r}$, where L is a constant which depends only on K_Φ.*

Proof. Let $\delta > 0$, and let $(h_u)_{u \in V}$ be a family of i.i.d. Bernoulli random variables with probability δ. For each $u \in V$, the action x_u is forced to 0 if $h_u = 1$.

Consider the modified network $\mathcal{G}^\delta = (V, E, \phi^\delta, \chi)$, where for any u, if $h_u = 1$, the potential function $\phi_u(1)$ is changed to $-\infty$ ($\phi_u(0)$ is unchanged), and for any (u, v), if $h_u = 1$, the interaction function $\phi_{u,v}(x_u, x_v)$ is changed to $\phi_{u,v}(0, x_v)$ (and becomes a function of x_v only). Let $\mathbf{x}^\delta$ be the optimal solution of $\mathcal{G}^\delta$, and $\mathbf{x}$ be the optimal solution of $\mathcal{G}$. Let $H = \{u \in V : h_u = 1\}$, and let $E' = \{(u, v) : u \notin H, v \notin H\}$.

$$\begin{aligned} F(\mathbf{x}^\delta) =& F^\delta(\mathbf{x}^\delta) \\ \geq& \sum_{(u,v) \in E'} \phi_{u,v}(x_u, x_v) + \sum_{u \notin H} \phi_u(x_u) + \sum_{(u,v) \in E \setminus E'} \phi_{u,v}(x_u^\delta, x_v^\delta) + \sum_{u \in H} \phi_u(x_u^\delta) \end{aligned}$$

Subtracting this quantity from $F(\mathbf{x})$, we obtain:

$$\begin{aligned} F(\mathbf{x}) - F(\mathbf{x}^\delta) \leq& \sum_{(u,v) \in E \setminus E'} |\phi_{u,v}(x_u^\delta, x_v^\delta) - \phi_{u,v}(x_u, x_v)| + \sum_{u \in H} |\phi_u(x_u^\delta) - \phi_u(x)| \\ \leq& 6K_\Phi |H| \end{aligned}$$

We have $\mathbb{E}|H| = |V|\delta$. By taking expectations, we obtain

$$\mathbb{E}[F(\mathbf{x}) - F(x^{\delta})] \leq 6K_{\Phi}\,\delta|V|$$

Let us now prove that forcing some variables to 0 induces coupling in the network with value function F^{δ}: remember that for any message sent on an edge (u,v), $|\mu_{u\leftarrow v}(B) - \mu_{u\leftarrow v}(B')| \leq |Y_{u,v}|$. But if $h_v = 1$, then $\phi_{u,v}(x_u, x_v)$ is actually $\phi_{u,v}(x_u, 0)$ and it immediately follows in that case that $Y_{u,v} = 0$. Therefore, for any two messages μ, μ' sent on the computation tree and started with different boundary conditions, we have

$$\mathbb{P}(\mu = \mu') \geq \delta$$

It follows the system has distance-dependent coupling with parameters (a,b), with $a = (1-\delta)$ and $b = 0$. Since $\Delta = 2$, $a(1-\Delta) = (1-\delta) < 1$ and by Proposition 8, the network exhibits exponential correlation decay. By Equation (3.4), there exist constants K_1, K_2 which depend only on K_{Φ} such that

$$\mathbb{E}[F(\mathbf{x}) - F(\mathbf{x}^{r,\delta})] \leq (|V| + |E|)K_1\delta + K_2(1-\delta)^r$$

By choosing $\delta = \frac{\log r}{r}$, it follows that

$$\begin{aligned}
\mathbb{E}[f(\mathbf{x}) - f(\mathbf{x}^{r,\delta})] \leq & (|V| + |E|)K_1\frac{\log r}{r} + K_2\exp(r\log(1-\delta)) \\
\leq & (|V| + |E|)K_1\frac{\log r}{r} + K_2\exp(-r\frac{\log r}{r}) \\
\leq & (|V| + |E|)K_1\frac{\log r}{r} + K_2\frac{1}{r} \\
\leq & (|V| + |E|)L\frac{\log r}{r}
\end{aligned}$$

□

3.6 Regularization technique

The main caveat of Proposition 9 is that Assumption 1 does not necessarily hold. In fact, it definitely does not apply to models with discrete random variables Φ_u and $\Phi_{u,v}$. We now introduce an idea of regularization via a small perturbation of $\Phi_u, \Phi_{u,v}$.

Let $Z_{v,x}, v \in V, x \in \chi$ be a collection of independent standard Gaussian random variables. Fix $\delta > 0$ and consider $\tilde{\Phi}_v(x) = \Phi_v(x) + \delta Z_{v,x}$. Let $\tilde{\mathcal{G}} = (V, E, \tilde{\Phi}_v, \Phi_{v,u})$. In other words $\tilde{\mathcal{G}}$ is obtained from $\mathcal{G}$ by perturbing the potential functions Φ_v with $\delta Z_{v,x}$. Also let $\mathbf{x}$ and $\tilde{\mathbf{x}}$ be (any) optimal decisions for the networks $\mathcal{G}$ and $\tilde{\mathcal{G}}$, respectively. Denote by $\tilde{B}_v(x)$ the corresponding bonus function. Then we find that the bonuses in the regularized network $\tilde{\mathcal{G}}$ have bounded density, and that the functions F and $\tilde{F}$ reach similar values

Proposition 12. *For every $x \neq y$, $\tilde{B}_v(x) - \tilde{B}_v(y)$ is a continuous random variable with density bounded above by $\frac{1}{\sqrt{4\pi\delta}}$. Moreover, for any random vector $\mathbf{z}$, we have:*

$$\mathbb{E}|F(\mathbf{z}) - \tilde{F}(\mathbf{z})] \leq T\sqrt{\frac{2}{\pi}}|V|\delta. \tag{3.26}$$

While we will not do it here, it is easy to use Equation (3.26) to show that a near-optimal solution for the regularized network $\tilde{\mathcal{G}}$ is also a near-optimal solution for $\mathcal{G}$.

Proof. We have:

$$\tilde{B}_u(x) - \tilde{B}_u(y) = B_u(x) - B_u(y) + \delta Z_{u,x} - \delta Z_{u,y}.$$

Let $D = B_u(x) - B_u(y)$ and $\tilde{D} = \tilde{B}_u(x) - \tilde{B}_v(y)$. Since $Z_{u,x} - Z_{u,y}$ is a zero mean Gaussian random variable with variance 2, then for every $t \in \mathbb{R}$ and $h > 0$, by conditioning on $Z_{u,x} - Z_{u,y}$, we obtain:

$$\begin{aligned}
\mathbb{P}(t \leq \tilde{D} < t + h) &= \int_{-\infty}^{+\infty} \frac{1}{\sqrt{4\pi\delta}} e^{-\frac{u^2}{4\delta^2}} \mathbb{P}(t - u \leq D < t + h - u) du \\
&\leq \frac{1}{\sqrt{4\pi\delta}} \int_{-\infty}^{+\infty} \mathbb{P}(t - u \leq D < t + h - u) du \leq \frac{h}{\sqrt{4\pi\delta}},
\end{aligned}$$

where the last inequality follows from the fact that, for any random variable X with $\mathbb{E}|X| < +\infty$,

$$\int_{-\infty}^{\infty} \mathbb{P}(x \leq X < x + h) dx \leq h$$

Taking the limit for $h \to 0$, we conclude that D has a density and which is bounded by $\frac{1}{\sqrt{4\pi\delta}}$. Finally, we have:

$$|F(\mathbf{z}) - \tilde{F}(\mathbf{z})| \leq \delta \sum_v \sum_y |Z_{v,y}|$$

which implies

$$\mathbb{E}|F(\mathbf{z}) - \tilde{F}(\mathbf{z})| \leq T\sqrt{\frac{2}{\pi}}\,|V|\,\delta$$

□

Finally, we will show that the regularization technique can only improve the coupling technique of section 3.4:

Lemma 8. *If $\mathcal{G}$ exhibits (a, b) coupling, then for any $\delta \geq 0$, $\tilde{\mathcal{G}}$ also exhibits (a, b) coupling.*

Proof. Let $Z = Z_{v,1} - Z_{v,0}$. Then $\tilde{\Phi}_v(1) - \tilde{\Phi}_v(0) = \Phi_v(1) - \Phi_v(0) + Z$. For any x, x'

$$\mathbb{P}\Big(\mu_{u\leftarrow v}(x + \tilde{\Phi}_v(1) - \tilde{\Phi}_v(0)) = \mu_{u\leftarrow v}(x' + \tilde{\Phi}_v(1) - \tilde{\Phi}_v(0))\Big) =$$
$$\int_z dP_Z(z)\mathbb{P}\Big(\mu_{u\leftarrow v}(x + z + \Phi_v(1) - \Phi_v(0)) = \mu_{u\leftarrow v}(x' + z + \Phi_v(1) - \Phi_v(0))\Big)$$

Applying definition (3.11) to $x + z$ and $x' + z$, we obtain

$$\begin{aligned}\mathbb{P}\Big(\mu_{u\leftarrow v}(x + \tilde{\Phi}_v(1) - \tilde{\Phi}_v(0)) = \mu_{u\leftarrow v}(x' + \tilde{\Phi}_v(1) - \tilde{\Phi}_v(0))\Big) &\geq \int_z dP_Z(z)\big((1-a) - b|x - x'|\big) \\ &\geq (1-a) - b|x - x'|\end{aligned}$$

□

3.7 Conclusions

In this chapter, we introduced a new definition of correlation decay adapted to optimization problems in arbitrary graphical models, and sought out the connections between the correlation decay property and the near-optimality of the Cavity Expansion algorithm. We have shown that such connections do indeed exist: graphical models with random costs often exhibit the correlation decay property, and therefore admit near-optimal approximation algorithms. We have identified a variety of models which exhibit the correlation decay property and we have proposed a general-purpose coupling technique which demonstrates when the property holds. However, this technique is limited to settings where all costs are bounded, and is thus restricted to settings without any hard constraints. This limitation prompts the question of whether the correlation decay property can be proven to hold

for constrained optimization problems. In Chapter 4, we will study the correlation decay phenomenon for a combinatorial optimization problem with constraints.

Chapter 4

Correlation decay and average-case complexity of the Maximum Weight Independent Set problem

4.1 Introduction

In this chapter, we investigate whether the correlation decay analysis of Chapter 3 can be extended to combinatorial optimization problems, in particular, the Maximum Weighted Independent Set (MWIS) problem.

The problem of finding the largest independent set of a graph is a well-known NP-complete problem. Moreover, unlike some other NP-complete problems, it does not admit a constant factor approximation algorithm for general graphs: Hastad [Has96] has shown that for every $0 < \delta < 1$ no $n^{1-\delta}$ approximation algorithm can exist for this problem unless $P = NP$, where n is the number of nodes. Even for the class of graphs with largest degree at most 3, no factor 1.0071 approximation algorithm can exist, under the same complexity-theoretic assumption; see Berman and Karpinski [BK98]. Similar results are established in the same paper for the cases of graphs with maximum degrees 4 and 5 with slightly larger constants. Thus, the problem does not admit a PTAS (Polynomial Time Approximation Scheme) even in the least non-trivial class of degree-3 graphs.

The study of correlation decay in the context of combinatorial optimization was introduced by Aldous [Ald92],[Ald01],[AS03] in the context of solving the well-known $\zeta(2)$

conjecture for the random minimal assignment problem. More recently, a different average case model was considered in Gamarnik *et al.* [GNS06]: the nodes of an Erdos-Rényi graph (with average degree c) are equipped with random weights distributed exponentially. The authors show that when the problem exhibits correlation decay, they are able to compute the limiting expression for the maximum weight independent set. Correlation decay was proven to hold in the regime $c \leq 2e$, and similar results were established for r-regular graphs with girth diverging to infinity for the cases $r = 3, 4$. They also show that the correlation decay property does not hold when $r > 4$. The local-weak convergence/cavity method thus was used extensively, but only in the setting of random graphs, which are known to have a locally tree-like structure.

In this chapter, we extend the correlation decay analysis to general graphs. The application of the Cavity Expansion algorithm in a randomized setting has a particularly interesting implication for the theory of average case analysis of combinatorial optimization. We consider an arbitrary graph with largest degree at most 3, where the nodes are equipped with random weights, generated i.i.d. from an exponential distribution with parameter 1. Surprisingly, we discover that this is a tractable problem — we construct a randomized PTAS, even though the unit weight version of this problem (maximum cardinality independent set) does not admit any PTAS, as mentioned above. We extend this result to more general graphs but for distributions which are mixtures of exponential distributions.

Furthermore, we show that the setting with random weights hits a complexity-theoretic barrier just as the classical cardinality problem does. Specifically, we show that for graphs with sufficiently large degree the problem of finding with high probability the largest-weight independent set with i.i.d. exponentially distributed weights does not admit a PTAS. This negative result is proven by showing that for large degree graphs, largest-weighted independent sets are dominated by independents sets with cardinality close to largest possible. Since the latter does not admit a constant factor approximation up to $O(\Delta/2^{\sqrt{O(\Delta)}})$ multiplicative factor [Tre01], the same will apply to the former case.

4.2 Model description and results

Consider a simple undirected graph $\mathcal{G} = (V, E)$, $V = [n] = \{1, 2, \ldots, n\}$. A set of nodes $I \subset V$ is an independent set if $(u, v) \notin E$ for every $u, v \in I$. The quantity $\alpha = \alpha(\mathcal{G}) = \max_I |I|$ is called the independence number of the graph, where the maximization is over all

independent sets. Let $I^s = I^s_{\mathcal{G}}$ denote the independent set with the largest size: $|I^s| = \alpha$. In cases where we have several such independent sets, let I^s be any such independent set.

Suppose the nodes of the graph are equipped with weights $W_i \geq 0, i \in V$. The weight of an (independent) set I is $\sum_{u \in I} W_u$. The maximum weight independent set problem is the problem of finding an independent set I with maximum weight. It can be recast as a decision network problem $\mathcal{G} = (V, E, \Phi, \{0,1\}$ by setting $\phi_e(0,0) = \phi_e(0,1) = \phi_e(1,0) = 0, \phi_e(1,1) = -\infty$ for al edges $e \in E$, and $\phi_v(1) = W_v, \phi_v(0) = 0$ for all $v \in V$.

In this chapter we consider a variation of the MWIS problem, where the nodes of the graph are equipped with random weights $W_i, i \in V$, drawn independently from a common distribution $F(t) = \mathbb{P}(W \leq t), t \geq 0$. The goal is again to find an independent set I with the largest total weight $W(I) \triangleq \sum_{i \in I} W_i$. Naturally, this problem includes the problem of computing $\alpha(\mathcal{G})$ as special case when $F(t)$ is the deterministic distribution concentrated on 1. Our main result shows that, surprisingly, the problem of finding maximum weight independent set becomes tractable for certain distributions F, specifically when F is an exponential distribution with parameter 1, $F(t) = 1 - \exp(-t)$, and the graph has degree $\Delta \leq 3$. Let $I^* = I^*(\mathcal{G})$ be the largest weighted independent set, when it is unique, and let $W(I^*)$ be its weight. In our setting it is a random variable. Observe that I^* is indeed unique when F is a continuous distribution, which is our case.

We now state our first main result:

Theorem 9. *There exists an algorithm which for every $\mathcal{G} = (V, E)$ with $\Delta_{\mathcal{G}} \leq 3$, and every $\epsilon > 0$, produces a (random) independent set $\mathcal{I}$ such that*

$$\mathbb{P}\left(\frac{W(I^*)}{W(\mathcal{I})} > 1 + \epsilon\right) < \epsilon, \tag{4.1}$$

when the node weights are independently and exponentially distributed with parameter 1. The algorithm runs in time $O\left(n 2^{O(\epsilon^{-2}\log(1/\epsilon))}\right)$, namely it is an EPRAS.

Remarks:

1. Our algorithm, as we shall see, uses randomization, independent of the underlying randomness of the instance. Thus, the probabilistic statement (4.1) is with respect to two sources of randomness: randomness of weights and randomization of the algorithm.

2. The choice of parameter 1 in the distribution is without the loss of generality, of course: any common parameter leads to the same result.

3. Observe that the running time of the algorithm is actually *linear* in the number of nodes n. The dependence on the approximation and accuracy parameter ϵ is exponential, but the exponent does not involve n. In fact our algorithm is local in nature and, as a result, it can be run in a distributed fashion.

The exponential distribution is not the only distribution which can be analyzed in this framework. It is natural to ask if the above result can be generalized, and in particular to wonder if it is possible to find for each Δ a distribution which guarantees correlation decay holds for graphs with degree bounded by Δ. It is indeed possible, as we extend Theorem 9, albeit to the case of mixtures of exponential distributions. Let $\rho > 25$ be an arbitrary constant and let $\alpha_j = \rho^j, j \geq 1$.

Theorem 10. *There exists an algorithm which for every $\mathcal{G} = (V, E)$ with $\Delta_{\mathcal{G}} \leq \Delta$ and $\epsilon > 0$ produces a (random) independent set $\mathcal{I}$ such that*

$$\mathbb{P}\left(\frac{W(I^*)}{W(\mathcal{I})} > 1 + \epsilon\right) < \epsilon, \tag{4.2}$$

when the nodes weights are distributed according to $P(W > t) = \frac{1}{\Delta}\sum_{1 \leq j \leq \Delta} \exp(-\alpha_j t)$. The algorithm runs in time $O\left(n(\frac{1}{\epsilon})^{\Delta}\right)$, namely it is an FPTAS.

Note that for the case of a mixture of exponential distributions described above, our algorithm is in fact an F(fully)PTAS as opposed to an EPRAS for Theorem 9. The reason for this (rather the reason for the weaker EPRAS result) is that in order to establish the correlation decay property for the case of exponential distributions we need, for technical reasons, that the average degree is strictly less than two. Thus, our algorithm is preempted by preprocessing consisting of deleting each node with small probability $\delta = \delta(\epsilon)$ independently for all nodes. This makes the correlation decay rate dependent on δ and ultimately leads to an exponential dependence on ϵ. On the other hand, for the case of a mixture of exponential distributions, we will show a correlation decay rate which holds for every degree (by adjusting the weights in the mixture).

Our last result is a partial converse to the results above; one could conjecture that randomizing the weights makes the problem essentially easy to solve, and that perhaps

being able to solve the randomized version does not tell much about the deterministic version. We show that this is not the case, and that the setting with random weights hits a complexity-theoretic barrier just as the classical cardinality problem does. Specifically, we show that for graphs with sufficiently large degree the problem of finding with high probability the largest-weighted independent set with i.i.d. exponentially distributed weights does not admit any PTAS. We need to keep in mind that since we are dealing with instances which are random (in terms of weights) and worst-case (in terms of the underlying graph) at the same time, we need to be careful as to the notion of hardness we use.

Assuming that the results of Theorems 9 and 10 hold (which we call "finding the MWIS with high probability"), it can easily be proven that there exists a PTAS for computing the deterministic number $E[W(I)^*]$, the expected weight of the MWIS in the graph $\mathcal{G}$ considered. However, we show that if the maximum degree of the graph is increased, it is impossible to approximate the quantity $E[W(I)^*]$ arbitrarily closely, unless P=NP.

Theorem 11. *There exist Δ_0 and c_1^*, c_2^* such that for all $\Delta \geq \Delta_0$ the problem of computing $\mathbb{E}[W(I)^*]$ to within a multiplicative factor $\rho = \Delta/(c_1^* \log \Delta 2^{c_2^*\sqrt{\log \Delta}})$ for graphs with degree at most Δ is NP-complete.*

In principle, we could compute a concrete Δ_0 such that for all $\Delta \geq \Delta_0$ the claim of the theorem holds. But computing such Δ_0 explicitly does not seem to offer much insight. We note that in the related work by Trevisan [Tre01], no attempt is made to compute a similar bound either.

The main idea of the proof is to show that the difference between the largest weighted independent set and the largest independent set measured by cardinality is diminishing in Δ. A similar proof idea was used in [LV97] for proving the hardness of approximately counting independent sets in sparse graphs.

4.3 Cavity expansion and the algorithm

We begin by establishing the cavity recursion CE in the special case of MWIS. In this section we consider a general graph $\mathcal{G}$, whose nodes are equipped with arbitrary non-negative weights $W_i, i \in V$. Thus, no probabilistic assumption on W_i is adopted yet. Note that for the Independent Set problem, we have $J_{\mathcal{G}} = W(I^*)$, and for any $(i_1, \ldots, i_r)$, $J_{\mathcal{G},(i_1,\ldots,i_r)}(\mathbf{0}) = J_{\mathcal{G}\setminus\{i_1,\ldots,i_r\}}$, where $\mathcal{G} \setminus \{i_1, \ldots, i_r\}$ is the subgraph induced by nodes $V \setminus \{i_1, \ldots, i_r\}$. For

any node i, we define the quantity $C_{\mathcal{G}}(i) = J_{\mathcal{G}} - J_{\mathcal{G}\setminus\{i\}}$, which will be called censored cavity at node i.

Proposition 13. *Given $i \in V$, let $N(i) = \{i_1, \ldots, i_r\}$. Then*

$$C_{\mathcal{G}}(i) = \max\Big(0, W_i - \sum_{1\le l\le r} C_{\mathcal{G}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l)\Big), \tag{4.3}$$

where $\sum_{1\le l\le r} = 0$ when $N(i) = \emptyset$. Moreover, if $W_i - \sum_{1\le l\le r} C_{\mathcal{G}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l) > 0$, namely $C_{\mathcal{G}}(i) > 0$ then every largest weight independent set must contain i. Similarly if $W_i - \sum_{1\le l\le r} C_{\mathcal{G}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l) < 0$, implying $C_{\mathcal{G}}(i) = 0$, then every largest weight independent set does not contain i.

Remark: The proposition leaves out a "fuzzy" case $W_i - \sum_{1\le l\le r} C_{\mathcal{G}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l) = 0$. This will not be a problem in our setting since, due to the continuity of the weight distribution, the probability of this event is zero. Modulo this tie, the event $C_{\mathcal{G}}(i) > 0$ ($C_{\mathcal{G}}(i) = 0$) determines whether i must (must not) belong to the largest-weighted independent set.

Proof. Observe that

$$J_{\mathcal{G}} = \max\Big(J_{\mathcal{G}\setminus\{i\}}, W_i + J_{\mathcal{G}\setminus\{i,i_1,\ldots,i_r\}}\Big)$$

Subtracting $J_{\mathcal{G}\setminus\{i\}}$ from both sides, we obtain

$$C_{\mathcal{G}}(i) = \max\Big(0, W_i - (J_{\mathcal{G}\setminus\{i\}} - J_{\mathcal{G}\setminus\{i,i_1,\ldots,i_r\}})\Big)$$

Observe further,

$$\begin{aligned} J_{\mathcal{G}\setminus\{i\}} - J_{\mathcal{G}\setminus\{i,i_1,\ldots,i_r\}} &= \sum_{1\le l\le r} J_{\mathcal{G}\setminus\{i,i_1,\ldots,i_{l-1}\}} - J_{\mathcal{G}\setminus\{i,i_1,\ldots,i_l\}} \\ &= \sum_{1\le l\le r} C_{\mathcal{G}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l) \end{aligned}$$

The proof of the second part follows directly from the analysis above. □

Let us now relate the above theorem and notations in terms of the more general result of Chapter 2. It is easy to see that for independent sets, for $\mathbf{i} = (i_1, i_2, \ldots, i_l)$,we have

$J_{\mathcal{G}}\backslash\{i_1, i_2, \ldots, i_l\} = J_{\mathcal{G},\mathbf{i}}(0, 0, \ldots, 0)$. From Theorem 1, we have

$$B_{\mathcal{G},i} = J_{\mathcal{G},i}(1) - J_{\mathcal{G},i}(0) = W_i + \sum_l \mu_{i\leftarrow i_l}(1, C_{\mathcal{G}(i,l),i_l})$$

In the decision network formulation of MWIS, we have $\phi_e(x, y)$ equal to $-\infty$ for $(x, y) = (1, 1)$ and 0, otherwise. Therefore, by definition of $\mu_{i\leftarrow i_l}$, we have

$$\begin{aligned}\mu_{i\leftarrow i_l}(1, B_{\mathcal{G}(i,l),i_l}) &= \max(-\infty + B_{\mathcal{G}(i,l)}, 0) - \max(B_{\mathcal{G}(i,l)}, 0) \\ &= -\max(B_{\mathcal{G}(i,l)}, 0)\end{aligned}$$

Thus, we have

$$B_{\mathcal{G},i} = W_i - \sum_l \max(B_{\mathcal{G}(i,l)}, 0)$$

which gives

$$\max(B_{\mathcal{G},i}, 0) = \max(0, W_i - \sum_l \max(B_{\mathcal{G}(i,l)}, 0))$$

This is exactly the specialized cavity recursion (4.3), when setting $C_{\mathcal{G}}(i) = \max(B_{\mathcal{G},i}, 0)$, and supposing $\mathcal{G}(i, l) = \mathcal{G} \setminus \{i, i_1, \ldots, i_{l-1}\}$. By carefully looking at the definition of $\mathcal{G}(i, l)$, we can see this is indeed the case, and therefore Equation (4.3) is a special case of Equation (2.3) when applied to the "censored cavities" or $C = \max(B, 0)$, for the problem of MWIS.

We now construct quantities which provide bounds on the cavity C. For every induced subgraph $\mathcal{H}$ of $\mathcal{G}$ every $t = 0, 1, 2, \ldots$ and every $i \in \hat{V}$ define $C^-_{\mathcal{H}}(i, t)$ recursively as follows. Let $N_{\mathcal{H}}(i) = \{i_1, \ldots, i_r\}$. Then,

$$C^-_{\mathcal{H}}(i,t) = \begin{cases} 0, & t = 0; \\ \max\left(0, W_i - \sum_{1\le l\le r} C^+_{\mathcal{H}\backslash\{i,i_1,\ldots,i_{l-1}\}}(i_l, t-1)\right), & t \ge 1. \end{cases} \tag{4.4}$$

$$C^+_{\mathcal{H}}(i,t) = \begin{cases} W_i, & t = 0; \\ \max\left(0, W_i - \sum_{1\le l\le r} C^-_{\mathcal{H}\backslash\{i,i_1,\ldots,i_{l-1}\}}(i_l, t-1)\right), & t \ge 1. \end{cases} \tag{4.5}$$

By Proposition 13, if it was the case that $C^-_{\mathcal{H}\backslash\{i,i_1,\ldots,i_{l-1}\}}(i_l, t-1) = C_{\mathcal{H}\backslash\{i,i_1,\ldots,i_{l-1}\}}(i_l)$ for all l, then $C^-_{\mathcal{H}}(i, t) = C_{\mathcal{H}}(i, t)$. The same applies to $C^+_{\mathcal{H}}(i, t)$. However, this is generally not the case due to our "incorrect" initialization $C^-_{\mathcal{H}}(i, 0) = 0$. Our goal is to show using

correlation decay that when t is sufficiently large, $C^-_{\mathcal{H}}(i,t)$ is *approximately* correct. Showing this is the subject of the next section. We now close this section by giving two helpful lemma, before describing the modified Cavity Expansion algorithm used for the proof of Theorem 9.

Lemma 9. *For every $\mathcal{G}$ with degree Δ, $i \in V(\mathcal{G})$ and t, the quantities $C^-_{\mathcal{G}}(i,t), C^+_{\mathcal{G}}(i,t)$ can be computed in time $O(t\Delta^t)$.*

The proof follows directly from Proposition 7.
It turns out that $C^-_{\mathcal{H}}(i,t)$ and $C^+_{\mathcal{H}}(i,t)$ provide valid bounds on the true cavities $C_{\mathcal{H}}(i)$.

Lemma 10. *For every t,*

$$C^-_{\mathcal{H}}(i,t) \leq C_{\mathcal{H}}(i) \leq C^+_{\mathcal{H}}(i,t),$$

Proof. The proof is by induction in t, and is a special case of Theorem 4. The assertion holds by definition of C^-, C^+ for $t = 0$. The induction follows from (4.3), definitions of C^-, C^+ and since the function $x \to \max(0, W - x)$ is non-increasing. □

We now describe our algorithm for producing a large weighted independent set. Our algorithm runs in two stages. Fix $\epsilon > 0$. In the first stage we take an input graph $\mathcal{G} = (V, E)$ and delete every node (and incident edges) with probability $\epsilon^2/2$, independently for all nodes. We denote the resulting (random) subgraph by $\mathcal{G}(\epsilon)$. In the second stage we compute $C^-_{\mathcal{G}(\epsilon)}(i,r)$ for every node i for the graph $\mathcal{G}(\epsilon)$ for some target even number of steps r. We set $\mathcal{I}(r,\epsilon) = \{i : C^-_{\mathcal{G}(\epsilon)}(i,r) > 0\}$. Let I^*_ϵ be the largest weighted independent set of $\mathcal{G}(\epsilon)$.

Lemma 11. *$\mathcal{I}(r,\epsilon)$ is an independent set.*

Proof. By Lemma 10, if $C^-_{\mathcal{G}(\epsilon)}(i,r) > 0$ then $C_{\mathcal{G}(\epsilon)} > 0$, and therefore $\mathcal{I} \subset I^*_\epsilon$. Thus our algorithm produces an independent set in $\mathcal{G}(\epsilon)$ and therefore in $\mathcal{G}$. □

Due to Lemma 9, the complexity of running three stages of $CA(t,\epsilon)$ is $O(n\,t\,\Delta^t)$. As it will be apparent from the analysis, we could take $B^-_{\mathcal{G}_0}$ instead of $B^+_{\mathcal{G}_0}$ and arrive at the same result. We now proceed to the analysis of the Cavity Expansion algorithm $CA(t,\epsilon)$.

4.4 Correlation decay for the MWIS problem

In this section, we will prove Theorem 9. The main bulk of the proof will be to show that $\mathcal{I}(r,\epsilon)$ is close to I_ϵ^* in the set-theoretic sense. We will use this fact to show that $W(\mathcal{I}(r,\epsilon))$ is close to $W(I_\epsilon^*)$. It will be then straightforward to show that $W(I_\epsilon^*)$ is close to $W(I^*)$, which will finally give us the desired result, Theorem 9. The key step therefore consists in proving that the correlation decay property holds. It is the object of our next proposition.

Correlation decay property

We first need introduce for any arbitrary induced subgraph $\mathcal{H}$ of $\mathcal{G}(\epsilon)$, and any node i in $\mathcal{H}$, introduce $M_{\mathcal{H}}(i) = \mathbb{E}[\exp(-C_{\mathcal{H}}(i))], M_{\mathcal{H}}^-(i,r) = \mathbb{E}[\exp(-C_{\mathcal{H}}^-(i,r))], M_{\mathcal{H}}^+(i,r) = \mathbb{E}[\exp(-C_{\mathcal{H}}^+(i,r))]$.

Proposition 14. *Let $\mathcal{G}(\epsilon) = (V_\epsilon, E_\epsilon)$ be the graph obtained from the original underlying graph as a result of the first phase of the algorithm (namely deleting every node with probability $\delta = \epsilon^2/2$ independently for all nodes). Then, for every node i in $\mathcal{G}(\epsilon)$ and every r*

$$\mathbb{P}(C_{\mathcal{G}(\epsilon)}(i) = 0, C_{\mathcal{G}(\epsilon)}^+(i,r) > 0) \leq 3(1-\epsilon^2/2)^r, \tag{4.6}$$

and

$$\mathbb{P}(C_{\mathcal{G}(\epsilon)}(i) > 0, C_{\mathcal{G}(\epsilon)}^-(i,r) = 0) \leq 3(1-\epsilon^2/2)^r. \tag{4.7}$$

Proof. Consider a subgraph $\mathcal{H}$ of $\mathcal{G}$, node $i \in \mathcal{H}$ with neighbors $\mathcal{N}_{\mathcal{H}}(i) = \{i_1, \ldots, i_d\}$, and suppose for now that the number of neighbors of i in $\mathcal{G}$ is less than 2.

Examine the recursion (4.3) and observe that all the randomness in terms $C_{\mathcal{H}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l)$ comes from the subgraph $\mathcal{H} \setminus \{i, i_1, \ldots, i_{l-1}\}$, and thus W_j is independent from the vector $(C_{\mathcal{H}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l), 1 \leq l \leq d)$. A similar assertion applies when we replace $C_{\mathcal{H}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l)$ with $C^-_{\mathcal{H}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l,r)$ and $C^+_{\mathcal{H}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l,r)$ for every r. Using the memoryless property of the exponential distribution, denoting W a standard exponential random variable,

we obtain:

$$\begin{aligned}\mathbb{E}[\exp(-C_{\mathcal{H}}(i))|\sum_{1\le l\le d} C_{\mathcal{H}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l)=x] =&\mathbb{P}(W_i\le x)\mathbb{E}[\exp(0)]+\\ &\mathbb{E}[\exp(-(W_i-x))\,|\,W_i>x]\mathbb{P}(W_i>x)\\ =&(1-\mathbb{P}(W_i>x))+\mathbb{E}[\exp(-W)]\mathbb{P}(W_i>x)\\ =&(1-\mathbb{P}(W_i>x))+(1/2)\mathbb{P}(W_i>x)\\ =&1-(1/2)\mathbb{P}(W_i>x)\\ =&1-(1/2)\exp(-x) \qquad (4.8)\end{aligned}$$

It follows that

$$\mathbb{E}[\exp(-C_{\mathcal{H}}(i))]=1-(1/2)\mathbb{E}\exp\left(-\sum_{1\le l\le d} C_{\mathcal{H}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l)\right)$$

Similarly, we obtain

$$\begin{aligned}\mathbb{E}\left[\exp\left(-C^-_{\mathcal{H}}(i,r)\right)\right]&=1-(1/2)\mathbb{E}\exp\Big(-\sum_{1\le l\le d} C^+_{\mathcal{H}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l,r-1)\Big)\\ \mathbb{E}\left[\exp\left(-C^+_{\mathcal{H}}(i,r)\right)\right]&=1-(1/2)\mathbb{E}\exp\Big(-\sum_{1\le l\le d} C^-_{\mathcal{H}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l,r-1)\Big)\end{aligned}$$

Since i had two neighbors or less $\mathcal{G}$, it also has two neighbors or less in $\mathcal{H}$. For $d=0$, we have trivially $M_{\mathcal{H}}(i)=M^-_{\mathcal{H}}(i)=M^+_{\mathcal{H}}(i)$. Suppose $d=1: N_{\mathcal{H}}(i)=\{i_1\}$. Then,

$$\begin{aligned}M^-_{\mathcal{H}}(i,r)-M^+_{\mathcal{H}}(i,r)&=(1/2)\Big(\mathbb{E}\left[\exp(-C^-_{\mathcal{H}\setminus\{i\}}(i_1,r-1))\right]-\mathbb{E}[\exp(-C^+_{\mathcal{H}\setminus\{i\}}(i_1,r-1))]\Big)\\ &=(1/2)\Big(M^-_{\mathcal{H}\setminus\{i\}}(i_1,r-1)-M^+_{\mathcal{H}\setminus\{i\}}(i_1,r-1)\Big) \qquad (4.9)\end{aligned}$$

Finally, suppose $d=2$: $N(i)=\{i_1,i_2\}$. Then

$$\begin{aligned}M^-_{\mathcal{H}}(i,r)-M^+_{\mathcal{H}}(i,r)=&(1/2)\mathbb{E}\left[\exp(-C^-_{\mathcal{H}\setminus\{i\}}(i_1,r-1)-C^-_{\mathcal{H}\setminus\{i,i_1\}}(i_2,r-1))\right]\\ &-(1/2)\mathbb{E}\left[\exp(-C^+_{\mathcal{H}\setminus\{i\}}(i_1,r-1)-C^+_{\mathcal{H}\setminus\{i,i_1\}}(i_2,r-1))\right]\end{aligned}$$

$$= (1/2)\mathbb{E}\left[\exp(-C^-_{\mathcal{H}\setminus\{i\}}(i_1, r-1))(\exp(-C^-_{\mathcal{H}\setminus\{i,i_1\}}(i_2, r-1)) - \exp(-C^+_{\mathcal{H}\setminus\{i,i_1\}}(i_2, r-1))\right]$$
$$+ (1/2)\mathbb{E}\left[\exp(-C^+_{\mathcal{H}\setminus\{i,i_1\}}(i_2, r-1))(\exp(-C^-_{\mathcal{H}\setminus\{i\}}(i_1, r-1)) - \exp(-C^+_{\mathcal{H}\setminus\{i\}}(i_1, r-1))\right]$$

Using the non-negativity of C^-, C^+ and applying Lemma 10 we obtain

$$0 \leq M^-_{\mathcal{H}}(i,r) - M^+_{\mathcal{H}}(i,r) \leq (1/2)\big(M^-_{\mathcal{H}\setminus\{i,i_1\}}(i_2, r-1) - M^+_{\mathcal{H}\setminus\{i,i_1\}}(i_2, r-1)\big)$$
$$+ (1/2)\big(M^-_{\mathcal{H}\setminus\{i\}}(i_1, r-1)] - M^+_{\mathcal{H}\setminus\{i\}}(i_1, r-1)\big) \tag{4.10}$$

Summarizing the three cases we conclude

$$|M^+_{\mathcal{H}}(i,r) - M^-_{\mathcal{H}}(i,r)| \leq (d/2) \max_{\mathcal{H}',j} \left|M^+_{\mathcal{H}'}(j, r-1) - M^-_{\mathcal{H}}(j, r-1)\right|, \tag{4.11}$$

where the maximum is over subgraphs $\mathcal{H}'$ of $\mathcal{G}$ and nodes $j \in \mathcal{H}$ with degree less than 2 in $\mathcal{H}$. The reason for this is that in Equations (4.9) and (4.10); the moments $M^+_{\mathcal{H}'}(j, r-1)$ in the right hand side are always computed in a node j which has lost at least one of its neighbors (namely, i) in graph $\mathcal{H}$. Since the degree of j was at most 3 in $\mathcal{G}$ and one neighbor at least is removed, j has at most two neighbors in $\mathcal{H}$. By considering $\mathcal{H} \cap \mathcal{G}(\epsilon)$ in all previous equations, Equation (4.11) implies

$$|M^+_{\mathcal{H}\cap\mathcal{G}(\epsilon)}(i,r) - M^-_{\mathcal{H}\cap\mathcal{G}(\epsilon)}(i,r)| \leq (d(\epsilon)/2) \max_{\mathcal{H}',j} \left|M^+_{\mathcal{H}'\cap\mathcal{G}(\epsilon)}(j, r-1) - M^-_{\mathcal{H}\cap\mathcal{G}(\epsilon)}(j, r-1)\right|, \tag{4.12}$$

where $d(\epsilon)$ denotes the number of neighbors of i in $\mathcal{H} \cap \mathcal{G}(\epsilon)$. By definition of $\mathcal{G}(\epsilon)$, $d(\epsilon)$ is a binomial random variables with d trials and probability of success $(1 - \epsilon^2/2)$, where d is the deree of i in $\mathcal{H}$. Since $d \leq 2$, $E[d(\epsilon)] \leq 2(1 - \epsilon^2/2)$. Moreover, this randomness is independent from the randomness of the random weights of $\mathcal{H}$. Therefore,

$$\mathbb{E}\left|M^+_{\mathcal{H}\cap\mathcal{G}(\epsilon)}(i,r) - M^-_{\mathcal{H}\cap\mathcal{G}(\epsilon)}(i,r)\right| \leq (1 - \epsilon^2/2) \max_{\mathcal{H},j} \mathbb{E}\left|M^+_{\mathcal{H}\cap\mathcal{G}(\epsilon)}(j, r-1) - M^-_{\mathcal{H}\cap\mathcal{G}(\epsilon)}(j, r-1)\right| \tag{4.13}$$

where the external expectation is wrt randomness of the first phase of the algorithm (deleted nodes). Let e_{r-1} the right-hand side of (4.13). By taking the max of the left-hand side of (4.13) over all $(\mathcal{H}, j)$ where j has degree less than or equal to 2 in $\mathcal{H}$, we obtain the inequality $e_r \leq (1 - \epsilon^2/2)e_{r-1}$. Iterating on r and using $0 \leq M \leq 1$, this inequality implies

that $e_r \leq (1-\epsilon^2/2)^r$ for all $r \geq 0$. Finally, it is easy to show using the same techniques that Equation (4.11) holds for $r = 3$ as well. This finally implies that for an arbitrary node i in $\mathcal{G}(\epsilon)$,

$$\mathbb{E}\left|M^+_{\mathcal{G}(\epsilon)}(i,r) - M^-_{\mathcal{G}(\epsilon)}(i,r)\right| \leq 3/2(1-\epsilon^2/2)^r$$

Applying Lemma 10, we conclude for every r

$$0 \leq \mathbb{E}\left[\exp(-C^-_{\mathcal{G}(\epsilon)}(i,r)) - \exp(-C^+_{\mathcal{G}(\epsilon)}(i,r))\right] \leq 3/2(1-\epsilon^2/2)^r$$

Recalling (4.8) we have

$$\mathbb{E}[\exp(-C_{\mathcal{G}(\epsilon)}(i))] = 1 - (1/2)\mathbb{P}(W > \sum_{1\leq l\leq d} C_{\mathcal{G}(\epsilon)\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l)) = 1 - (1/2)\mathbb{P}(C_{\mathcal{G}(\epsilon)}(i) > 0),$$

Similar expressions are valid for $C^-_{\mathcal{G}(\epsilon)}(i,r)), C^+_{\mathcal{G}(\epsilon)}(i,r))$. We obtain

$$0 \leq \mathbb{P}(C^+_{\mathcal{G}(\epsilon)}(i,r) = 0) - \mathbb{P}(C^-_{\mathcal{G}(\epsilon)}(i,r) = 0) \leq 3(1-\epsilon^2/2)^r$$

Again applying Lemma 10, we obtain

$$\mathbb{P}(C_{\mathcal{G}(\epsilon)}(i) = 0, C^+_{\mathcal{G}(\epsilon)}(i,r) > 0) \leq \mathbb{P}(C^-_{\mathcal{G}(\epsilon)}(i,r) = 0, C^+_{\mathcal{G}(\epsilon)}(i,r) > 0) \leq 3(1-\epsilon^2/2)^r$$

and

$$\mathbb{P}(C_{\mathcal{G}(\epsilon)}(i) > 0, C^-_{\mathcal{G}(\epsilon)}(i,2r) = 0) \leq \mathbb{P}(C^-_{\mathcal{G}(\epsilon)}(i,2r) = 0, C^+_{\mathcal{G}(\epsilon)}(i,2r) > 0) \leq 3(1-\epsilon^2/2)^{2r}$$

This completes the proof of the proposition. □

Concentration argument

We can now complete the proof of Theorem 9. We need to bound $|W(I^*) - W(I^*_\epsilon)|$ and $W(I^*_\epsilon \setminus \mathcal{I}(r,\epsilon))$ and show that both quantities are small.

Let ΔV_ϵ be the set of nodes in $\mathcal{G}$ which are not in $\mathcal{G}(\epsilon)$. Trivially, $|W(I^*) - W(I^*_\epsilon)| \leq W(\Delta V_\epsilon)$. We have $\mathbb{E}[\Delta V_\epsilon] = \epsilon^2/2n$, and since the nodes were deleted irrespective of their weights, then $\mathbb{E}[W(\Delta V_\epsilon)] = \epsilon^2/2n$.

To analyze $W(I^*_\epsilon \setminus \mathcal{I}(r,\epsilon))$, observe that by (second part of) Proposition 14, for every

node $i, \mathbb{P}(i \in I_\epsilon^* \setminus \mathcal{I}(r,\epsilon)) \leq 3(1-\epsilon^2/2)^r \triangleq \delta_1$. Thus, $\mathbb{E}|I_\epsilon^* \setminus \mathcal{I}(r,\epsilon)| \leq \delta_1 n$. In order to obtain a bound on $W(I_\epsilon^* \setminus \mathcal{I}(r,\epsilon))$ we derive a crude bound on the largest weight of a subset with cardinality $\delta_1 n$. Fix a constant C and consider the set V_C of all nodes in $\mathcal{G}(\epsilon)$ with weights greater than C. We have $\mathbb{E}\left[W(V_C)\right] \leq (C + \mathbb{E}[W - C|W > C])\exp(-C)n = (C+1)\exp(-C)n$. The remaining nodes have a weight at most C. Therefore,

$$\begin{aligned}\mathbb{E}\left[W(I_\epsilon^* \setminus \mathcal{I}(r,\epsilon))\right] &\leq \mathbb{E}\left[W\Big(\big((I_\epsilon^* \setminus \mathcal{I}(r,\epsilon)\big) \cap V_C) \cup V_C^c\Big)\right] \leq C\mathbb{E}\left[|I_\epsilon^* \setminus \mathcal{I}(r,\epsilon)|\right] + \mathbb{E}\left[V_C\right] \\ &\leq C\delta_1 n + (C+1)\exp(-C)n.\end{aligned}$$

We conclude

$$\mathbb{E}[|W(I^*) - W(\mathcal{I}(r,\epsilon))|] \leq \epsilon^2/2n + C\delta_1 n + (C+1)\exp(-C)n. \tag{4.14}$$

Now we obtain a lower bound on $W(I^*)$. Consider the standard greedy algorithm for generating an independent set: take arbitrary node, remove neighbors, and repeat. It is well known and simple to see that this algorithm produces an independent set with cardinality at least $n/4$, since the largest degree is at most 3. Since the algorithm ignores the weights, then also the expected weight of this set is at least $n/4$. The variance of that weight is upper bounded by n. By Chebyshev's inequality

$$\mathbb{P}(W(I^*) < n/8) \leq \frac{n}{(n/4 - n/8)^2} = 64/n.$$

We now summarize the results.

$$\begin{aligned}\mathbb{P}(\frac{W(\mathcal{I}(r,\epsilon))}{W(I^*)} \leq 1-\epsilon) &\leq \mathbb{P}(\frac{W(\mathcal{I}(r,\epsilon))}{W(I^*)} \leq 1-\epsilon, W(I^*) \geq n/8) + \mathbb{P}(W(I^*) < n/8) \\ &\leq \mathbb{P}(\frac{|W(I^*) - W(\mathcal{I}(r,\epsilon))|}{W(I^*)} \geq \epsilon, W(I^*) \geq n/8) + 64/n \\ &\leq \mathbb{P}(\frac{|W(I^*) - W(\mathcal{I}(r,\epsilon))|}{n/8} \geq \epsilon) + 64/n \\ &\leq \frac{\epsilon^2/2 + 4C(1-\epsilon^2/2)^r + (C+1)\exp(-C)}{\epsilon/8} + 64/n,\end{aligned}$$

where we have used Markov's inequality in the last step and $\delta_1 = 4(1-\delta)^r$. Thus, it suffices to arrange δ and C so that the first ratio is at most $\epsilon/2$ and assuming, without the loss of generality, that $n \geq 128/\epsilon$, we will obtain that the sum is at most ϵ. It is a simple exercise

to show that by taking $r = O(\log(1/\epsilon)/\epsilon^2)$ and $C = O(\log(1/\epsilon))$, we obtain the required result. This completes the proof of Theorem 9. □

4.5 Hardness result

In this section, we prove Theorem 11.

Proof of Theorem 11. The main idea of the proof is to show that the weight of a maximum weighted independent set is close to the cardinality of a maximum independent set. A similar proof idea was used in [LV97] for proving the hardness of approximately counting independent sets in sparse graphs.

Given a graph $\mathbb{G}$ with degree bounded by Δ, let I^s denote (any) maximum cardinality independent set, and let I^* denote the unique maximum weight independent set corresponding to i.i.d. weights with exp(1) distribution. We make use of the following result due to Trevisan [Tre01].

Theorem 12. *There exist Δ_0 and c^* such that for all $\Delta \geq \Delta_0$ the problem of approximating the largest independent set in graphs with degree at most Δ to within a factor $\rho = \Delta/2^{c^*\sqrt{\log \Delta}}$ is NP-complete.*

Our main technical result is the following proposition. It states that the ratio of the expected weight of a maximum weight independent set to the cardinality of a maximum independent set grows as the logarithm of the maximum degree of the graph.

Proposition 15. *Suppose $\Delta \geq 2$. For every graph $\mathbb{G}$ with maximum degree Δ and n large enough, we have:*

$$1 \leq \frac{E[W(I^*)]}{|I^s|} \leq 10 \log \Delta.$$

This in combination with Theorem 12 leads to the desired result.

Proof. Let $W(1) < W(2) < \cdots < W(n)$ be the ordered weights associated with our graph

$\mathbb{G}$. Observe that

$$\begin{aligned} E[W(I^*)] &= E[\sum_{v \in I^*} W_v] \\ &\leq E[\sum_{n-|I^*|+1}^{n} W(i)] \\ &\leq E[\sum_{n-|I^s|+1}^{n} W(i)] \end{aligned}$$

The exponential distribution implies $E[W(j)] = H(n) - H(n-j)$, where $H(k)$ is the harmonic sum $\sum_{1 \leq i \leq k} 1/i$. Thus

$$\begin{aligned} \sum_{j=n-|I^s|+1}^{n} E[W(j)] &= \sum_{n-|I^s|+1 \leq j \leq n} (H(n) - H(n-j)) \\ &= |I^s| H(n) - \sum_{j \leq |I^s|-1} H(j). \end{aligned}$$

We use the bound $\log(k) \leq H(k) - \gamma \leq \log(k) + 1$, where $\gamma \approx .57$ is Euler's constant. Then

$$\begin{aligned} \sum_{j=n-|I^s|+1}^{n} E[W(j)] &\leq |I^s|(H(n) - \gamma) + \log(|I^s|) + 2 - \sum_{1 \leq j \leq |I^s|} \log(j) \\ &\leq |I^s|(H(n) - \gamma) + \log(|I^s|) + 2 - \int_1^{|I^s|} \log(t) dt \\ &\leq |I^s| \log(n) + |I^s| + \log(|I^s|) + 2 - |I^s| \log(|I^s|) + |I^s| \\ &\leq (|I^s| + 1)(\log \frac{n}{|I^s|} + 2 + \log(|I^s|)/|I^s|) \\ &\leq |I^s|(\log(\Delta + 1) + 3) + (\log(\Delta + 1) + 3), \end{aligned}$$

where the bound $|I^s| \geq n/(\Delta + 1)$ (obtained by using the greedy algorithm, see Section 4.4) is used. Again using the bound $|I^s| \geq n/(\Delta + 1)$, we find that $\frac{E[W(I^*)]}{|I^s|} \leq \log(\Delta + 1) + 3 + o(1)$. Since $E[W(I^*)] \geq E[W(I^s)] = |I^s|$, it follows that for all sufficiently large n, $1 \leq \frac{E[W(I^*)]}{|I^s|} \leq \log(\Delta + 1) + 4$. The proposition follows since for all $\Delta \geq 2$ we have $\log(\Delta + 1) + 4 \leq 10 \log \Delta$. $\square$

4.6 Generalization to phase-type distribution

4.6.1 Mixture of exponentials

In this section we present the proof of Theorem 10. The proof follows two steps. First, we show how to analyze correlations for the MWIS problem when the distribution of weights is a mixture of exponentials (as opposed to simply exponential, as in Theorem 9). Then we show that the correlation decay property holds for the parameters chosen in Theorem 10.

The mixture of Δ exponential distributions with rates $\alpha_j, 1 \leq j \leq \Delta$ and equal weights $1/\Delta$ can be viewed as first randomly generating a rate α with the probability law $\mathbb{P}(\alpha = \alpha_j) = 1/\Delta$ and then randomly generating exponentially distributed random variable with rate α_j, conditional on the rate being α_j.

For every subgraph $\mathcal{H}$ of $\mathcal{G}$, node i in $\mathcal{H}$ and $j = 1, \ldots, \Delta$, define $M^j_{\mathcal{H}}(i) = \mathbb{E}[\exp(-\alpha_j C_{\mathcal{H}}(i))]$, $M^{-,j}_{\mathcal{H}}(i,r) = \mathbb{E}[\exp(-\alpha_j C^-_{\mathcal{H}}(i,r))]$ and $M^{+,j}_{\mathcal{H}}(i,r) = \mathbb{E}[\exp(-\alpha_j C^+_{\mathcal{H}}(i,r))]$, where $C_{\mathcal{H}}(i)), C^+_{\mathcal{H}}(i,r))$ and $C^-_{\mathcal{H}}(i,r))$ are defined as in Section 4.3.

Lemma 12. *Fix any subgraph $\mathcal{H}$, node $i \in \mathcal{H}$ with $N_{\mathcal{H}}(i) = \{i_1, \ldots, i_d\}$. Then*

$$\begin{aligned}
\mathbb{E}[\exp(-\alpha_j C_{\mathcal{H}}(i))] &= 1 - \frac{1}{\Delta} \sum_{1 \leq k \leq m} \frac{\alpha_j}{\alpha_j + \alpha_k} \mathbb{E}[\exp(- \sum_{1 \leq l \leq d} \alpha_k C_{\mathcal{H} \setminus \{i, i_1, \ldots, i_{l-1}\}}(i_l))] \\
\mathbb{E}[\exp(-\alpha_j C^+_{\mathcal{H}}(i,r))] &= 1 - \frac{1}{\Delta} \sum_{1 \leq k \leq m} \frac{\alpha_j}{\alpha_j + \alpha_k} \mathbb{E}[\exp(- \sum_{1 \leq l \leq d} \alpha_k C^-_{\mathcal{H} \setminus \{i, i_1, \ldots, i_{l-1}\}}(i_l, r-1))] \\
\mathbb{E}[\exp(-\alpha_j C^-_{\mathcal{H}}(i,r))] &= 1 - \frac{1}{\Delta} \sum_{1 \leq k \leq m} \frac{\alpha_j}{\alpha_j + \alpha_k} \mathbb{E}[\exp(- \sum_{1 \leq l \leq d} \alpha_k C^+_{\mathcal{H} \setminus \{i, i_1, \ldots, i_{l-1}\}}(i_l, r-1))]
\end{aligned}$$

Proof. Let $\alpha(i)$ be the random rate associated with node i. Namely, $\mathbb{P}(\alpha(i) = \alpha_j) = 1/\Delta$. We condition on the event $\sum_{1 \leq l \leq d} C_{\mathcal{H} \setminus \{i, i_1, \ldots, i_{l-1}\}}(i_l) = x$. As $C_{\mathcal{H}}(i) = \max(0, W_i - x)$, we

obtain:

$$\begin{aligned}
\mathbb{E}[-\alpha_j C_{\mathcal{H}}(i)|x] &= \frac{1}{\Delta}\sum_k \mathbb{E}[-\alpha_j C_{\mathcal{H}}(i)|x, \alpha(i)=\alpha_k] \\
&= \frac{1}{\Delta}\sum_k \Big(\mathbb{P}(W_i \le x|\alpha(i)=\alpha_k) \\
&\quad +\mathbb{P}(W_i > x|\alpha(i)=\alpha_k)\mathbb{E}[\exp(-\alpha_j(W_i-x))|W_i>x, \alpha(i)=\alpha_k]\Big) \\
&= \frac{1}{\Delta}\sum_k \left(1-\exp(-\alpha_k x)+\exp(-\alpha_k x)\frac{\alpha_k}{\alpha_j+\alpha_k}\right) \\
&= 1-\frac{1}{\Delta}\sum_k \frac{\alpha_j}{\alpha_j+\alpha_k}\exp(-\alpha_k x)
\end{aligned}$$

Thus,

$$\mathbb{E}[-\alpha_j C_{\mathcal{H}}(i)] = 1-\frac{1}{\Delta}\sum_k \frac{\alpha_j}{\alpha_j+\alpha_k}\mathbb{E}[\exp(-\sum_{1\le l\le d}\alpha_k C_{\mathcal{H}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l))]$$

The other equalities follow identically. □

By taking differences, we obtain

$$\begin{aligned}
&M_{\mathcal{H}}^{-,j}(i,r)-M_{\mathcal{H}}^{+,j}(i,r) = \\
&\frac{1}{\Delta}\sum_k \frac{\alpha_j}{\alpha_j+\alpha_k}\left(\mathbb{E}[\prod_{1\le l\le d}\exp(-\alpha_k C^{+}_{\mathcal{H}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l,r-1))]-\mathbb{E}[\prod_{1\le l\le d}\exp(-\alpha_k C^{-}_{\mathcal{H}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l,r-1))]\right)
\end{aligned}$$

We now use the identity

$$\prod_{1\le l\le r}x_l-\prod_{1\le l\le r}y_l=\sum_{1\le l\le r}\Big((\prod_{1\le k\le l-1}x_k)(x_l-y_l)(\prod_{l+1\le k\le r}y_k)\Big),$$

which further implies

$$\Big|\prod_{1\le l\le r}x_l-\prod_{1\le l\le r}y_l\Big|\le\sum_{1\le l\le r}|x_l-y_l|,$$

when $\max_l |x_l|, |y_l| < 1$. By applying this inequality with $x_l = \exp(-\alpha_k C^{+}_{\mathcal{H}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l, r-$

1)) and $y_l = \exp(-\alpha_k C^-_{\mathcal{H}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l, r-1))$, we obtain

$$\left|M^{-,j}_{\mathcal{H}}(i,r) - M^{+,j}_{\mathcal{H}}(i,r)\right|$$
$$\leq \frac{1}{\Delta}\sum_{1\leq k\leq m}\frac{\alpha_j}{\alpha_j+\alpha_k}\sum_{1\leq l\leq d}\left|M^{-,k}_{\mathcal{H}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l, r-1) - M^{+,k}_{\mathcal{H}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l, r-1)\right|$$

This implies

$$\left|M^{-,j}_{\mathcal{H}}(i,r) - M^{+,j}_{\mathcal{H}}(i,r)\right| \quad (4.15)$$
$$\leq \frac{r}{\Delta}\sum_{1\leq k\leq m}\frac{\alpha_j}{\alpha_j+\alpha_k}\max_{1\leq l\leq d}\left|M^{-,k}_{\mathcal{H}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l, r-1) - M^{+,k}_{\mathcal{H}\setminus\{i,i_1,\ldots,i_{l-1}\}}(i_l, r-1)\right| \quad (4.16)$$

For any $t \geq 0$ and j, define $e_{r,j}$ as follows

$$e_{r,j} = \sup_{\mathcal{H}\subset\mathcal{G}, i\in\mathcal{H}}\left|M^{-,j}_{\mathcal{H}}(i,r) - M^{+,j}_{\mathcal{H}}(i,r)\right| \quad (4.17)$$

By taking maximum on the right- and left- hand side successively, inequality (4.15) implies

$$e_{r,j} \leq \frac{r}{\Delta}\sum_{1\leq k\leq m}\frac{\alpha_j}{\alpha_j+\alpha_k}e_{r-1,k}$$

For any $t \geq 0$, denote $\mathbf{e_r}$ the vector of $(e_{r,1}, \ldots, e_{r,m})$. Denote $\mathbf{M}$ the matrix such that for all (j,k), $M_{j,k} = \frac{r}{\Delta}\frac{\alpha_j}{\alpha_j+\alpha_k}$. We finally obtain

$$\mathbf{e_r} \leq M\mathbf{e_{r-1}}.$$

Therefore, if M^r converges to zero exponentially fast in each coordinate, then also $\mathbf{e_r}$ converges exponentially fast to 0. Following the same steps as the proof of Theorem 9, this will imply that for each node, the error of a decision made in $\mathcal{I}(r,0)$ is exponentially small in r . Note that $\frac{r}{\Delta} \leq 1$. Recall that $\alpha_j = \rho^j$. Therefore, for each j,k, we have $M_{j,k} \leq \frac{\rho^j}{\rho^j+\rho^k}$. Define M_Δ to be a $\Delta\times\Delta$ matrix defined by $M_{j,j} = 1/2, M_{j,k} = 1, j > k$ and $M_{j,k} = (1/\rho)^{k-j}, k > j$, for all $1 \leq j,k \leq \Delta$. Since $M \leq M_\Delta$, it suffices to show that M^r_Δ converges to zero exponentially fast. Proof of Theorem 10 will thus be completed with the proof of the following lemma:

Lemma 13. *Under the condition $\rho > 25$, there exists $\delta = \delta(\rho) < 1$ such that the absolute*

value of every entry of M_Δ^r *is at most* $\delta^r(\rho)$.

Proof. Let $\epsilon = 1/\rho$. Since elements of M are non-negative, it suffices to exhibit a strictly positive vector $x = x(\rho)$ and $0 < \theta = \theta(\rho) < 1$ such that $M'x \leq \theta x$, where M' is transpose of M. Let x be the vector defined by $x_k = \epsilon^{k/2}, 1 \leq k \leq \Delta$. We show that for any j,

$$(M'x)_j \leq (1/2 + 2\frac{\sqrt{\epsilon}}{1-\sqrt{\epsilon}})x_j$$

It is easy to verify that when $\rho > 25$, that is $\epsilon < 1/25$, $(1/2 + 2\frac{\sqrt{\epsilon}}{1-\sqrt{\epsilon}}) < 1$, and the proof would be complete. Fix $1 \leq j \leq \Delta$. Then,

$$\begin{aligned}(M'x)_j &= \sum_{1\leq k\leq j-1} M_{k,j}\, x_k + 1/2 x_j + \sum_{j+1\leq k\leq \Delta} M_{k,j}\, x_k \\ &= \sum_{1\leq k\leq j-1} \epsilon^{j-k}\epsilon^{k/2} + 1/2\epsilon^{j/2} + \sum_{j+1\leq k\leq \Delta} \epsilon^{k/2}\end{aligned}$$

Since $x_j = \epsilon^{j/2}$, we have

$$\begin{aligned}\frac{(Mx)_j}{x_j} &\leq \sum_{1\leq k\leq j-1} \epsilon^{(j-k)/2} + 1/2 + \sum_{j+1\leq k\leq \Delta} \epsilon^{(k-j)/2} \\ &= 1/2 + \sum_{1\leq k\leq j-1} \epsilon^{k/2} + \sum_{1\leq k\leq \Delta-j} \epsilon^{k/2} \leq 1/2 + \frac{2\epsilon^{1/2}}{1-\epsilon^{1/2}}\end{aligned}$$

This completes the proof of the lemma and of the theorem. □

4.6.2 Phase-type distribution

In this section, we generalize the correlation analysis of section 4.6.1 to general phase-type distributions. It is well known that phase-type distributions are dense in the space of all distributions, and this approach therefore allows us, in some sense, to study the correlation decay property for all distributions. Consider a triplet (m, Q, ν), where:

- m is a number of states.
- Q is a $m \times m$ generator for a continuous-time Markov process for which states $\{1, 2, \ldots, m-1\}$ are transient and state m is absorbing.
- ν is a distribution for the starting state.

Note that for all u, $Q(u,m) = Q(m,u) = Q(m,m) = 0$. Let $X(t)$ denote the state of that Markov process at time t (conditional on $X(0)$ being distributed according to μ). Since m is absorbing, we know that $T = \inf\{t : X(t) = m \mid X(0) \sim \mu\}$ is a finite stopping time. The distribution of T is called the *phase-type distribution* with parameters (Q, μ). Furthermore, for any state i, let $T_i = \inf\{t : X(t) = m \mid X(0) = i\}$, and for any (j,k), let $M_{[j,k]}$ be a $m \times 1$ vector with components $M_{[j,k]}(i) \triangleq \mathbb{E}\left[\left(\exp(QT_i)\right)_{j,k}\right]$. Finally, for every subgraph $\mathcal{H}$ of $\mathcal{G}$ and any v, w, define $M_{\mathcal{H}}^{v,w}(u) \triangleq \mathbb{E}\left[\left(\exp\left(QC_{\mathcal{H}}(u)\right)\right)_{v,w}\right]$. Similarly, define $M_{\mathcal{H}}^{-,v,w}(u,r) \triangleq \mathbb{E}\left[\left(\exp\left(QC_{\mathcal{H}}^{-}(u,r)\right)\right)_{j,k}\right]$ and $M_{\mathcal{H}}^{+,v,w}(u) \triangleq \mathbb{E}\left[\left(\exp\left(QC_{\mathcal{H}}^{+}(u)\right)\right)_{v,w}\right]$. We begin our analysis by the following lemma:

Lemma 14. *Let $x \geq 0$, and suppose W follows a (Q,μ) phase-type distribution. Let $C = \max(0, W - x)$. Then, for any (j,k)*

$$\mathbb{E}\left[\left(\exp\left(QC\right)\right)_{j,k}\right] = \nu^T \exp(Qx) M_{[j,k]} \tag{4.18}$$

Proof. Let $X(t)$ denote the state of the Markov process at time t, with starting state distributed according to μ. We are interested in the state of the process at time x. If $X(x) = m$, then the process has reached state m, the corresponding variable W is less than x, and $(W-x)^+ = 0$. However, if $X(x) = i < m$, then the remaining time until X reaches m is T_i (by definition of T_u). In this case, we have $W = x + T_i$, and therefore $(W-x)^+ = T_i$. Since $T_m = 0$, we can conclude in general that $(W-x)^+$ has the same distribution as $T_{X(x)}$. Finally, by classical theory of Markov processes, $\mathbb{P}(X(x) = i \mid X(0) \sim \nu) = (\nu^T \exp(Qx))_i$. Combining all these observations, we obtain:

$$\begin{aligned}\mathbb{E}\left[\left(\exp\left(QC\right)\right)_{j,k}\right] &= \sum_i \mathbb{P}(X(x) = i)\, \mathbb{E}\left[\left(\exp(QT_i)\right)_{j,k}\right] \\ &= \sum_i (\nu^T \exp(Qx))_i M_{[j,k]}(i) = \nu^T \exp(Qx) M_{[j,k]}\end{aligned}$$

□

Now, consider any subgraph $\mathcal{H}$, node $u \in \mathcal{H}$, with $N_{\mathcal{H}}(u) = \{v_1, \ldots, v_d\}$. Then, from $C_{\mathcal{H}}(u) = \max(0, W_u - \sum_{1 \leq l \leq d} C_{\mathcal{H} \setminus \{u, v_1, \ldots, v_{l-1}\}}(v_l))$ and Lemma 14, we obtain

Lemma 15.

$$
\begin{aligned}
M_{\mathcal{H}}^{j,k}(u) &= \nu^T \mathbb{E}\left[\exp\left(Q \cdot \Big(\sum_{1\le l\le d} C_{\mathcal{H}\setminus\{u,v_1,\ldots,v_{l-1}\}}(v_l)\Big)\right)\right] M_{[j,k]} \\
M_{\mathcal{H}}^{-,j,k}(u,r) &= \nu^T \mathbb{E}\left[\exp\left(Q \cdot \Big(\sum_{1\le l\le d} C^{+}_{\mathcal{H}\setminus\{u,v_1,\ldots,v_{l-1}\}}(v_l, r-1)\Big)\right)\right] M_{[j,k]} \\
M_{\mathcal{H}}^{+,j,k}(u,r) &= \nu^T \mathbb{E}\left[\exp\left(Q \cdot \Big(\sum_{1\le l\le d} C^{-}_{\mathcal{H}\setminus\{u,v_1,\ldots,v_{l-1}\}}(v_l, r-1)\Big)\right)\right] M_{[j,k]}
\end{aligned}
$$

Proof. The proof is essentially the same as that of Lemma 12. To obtain the result, we use the tower property by conditioning on the value x of $\sum_{1\le l\le d} C_{\mathcal{H}\setminus\{u,v_1,\ldots,v_{l-1}(v_l))}$, and invoke Lemma 14. |

4.7 Conclusions

In this chapter, we considered a combinatorial problem with hard constraints, and showed that, once again, the correlation decay property proved to be a sufficient condition for the existence of local, near-optimal algorithms. Our results highlight interesting and intriguing connections between the fields of complexity of algorithms for combinatorial optimization problems and statistical physics, specifically the cavity method and the issues of long-range independence. For example, in the special case of the MWIS problem, we showed that the problem admits a PTAS, provided by the CE algorithm, for certain node weight distribution, even though the maximum cardinality version of the same problem is known to be non-approximable unless P=NP. It would be interesting to see what weight distributions are amenable to the approach proposed in this paper. For example, one could consider the case of Bernoulli weights and see whether the correlation decay property breaks down precisely when the approximation becomes NP-hard. It might also be useful to inquire whether random weights assumptions for general decision networks can be substituted with deterministic weights which have some random-like properties, in a fashion similar to the study of pseudo-random graphs. This would move our approach even closer to the worst-case combinatorial optimization setting.

Chapter 5

Graphical games

5.1 Introduction

Graphical games, introduced by Kearns, Littman and Singh in [KLS01a], are a class of models used to sparsely represent local interactions between selfish, rational agents. They are a natural extension of the discrete optimization models used in Chapters 3 and 4 to a game-theoretic setting. In a graphical game, each agent is assigned her own utility function, which depends on her own decision and the decisions of a few other players in the network. Ideally, each agent would choose an action which maximizes her expected utility; however, as her utility depends on the actions of other agents, in general, she cannot assume that other players will have aligned objectives. As a result, agents have to make their decisions while taking into account other agents' potentially conflicting objectives. The traditional concept used in game theory to predict the resulting outcome is to postulate that the agents' strategies will result in some kind of equilibrium, where the different forces at play balance out. The well known concept of *Nash equilibrium* [Nas50, Nas51] dictates that the game will result in a situation where each player cannot increase her utility by changing her decision, assuming the strategies of other players are fixed to those dictated by the equilibrium.

One of the main problems behind the assumption that rational players play according to a Nash equilibrium is that a Nash equilibrium, while guaranteed to exist in a large number of situations (see for instance [FT91]), is hard to find. At the heart of the computational issues behind notions of game-theoretic equilibrium are the following two questions: how complex is the computation of Nash equilibrium, and what are the best algorithms to find

one? In recent work at the intersection of computer science and economics, great progress has been made towards answering these questions. It has been shown that even under restrictive assumptions, finding a pure Nash equilibrium is NP-hard, while computing a mixed Nash equilibrium is PPAD-hard (see [CS02, CS03, GGS05, DFP06, BFH09, DGP09] for more details and appendix B for a primer on the PPAD complexity class). Other computational problems, such as finding a Nash equilibrium satisfying particular constraints [GS04], particular optimality conditions [CS08, CS03, EGG07, GGS05], and the related problem of finding *all* pure Nash (or finding a finite algebraic representation of all mixed Nash) [MM96], were also found to be generally hard to solve.

A potential way to address the complexity of computing a Nash equilibrium is to study graphical games. Indeed, the latter have generated a lot of interest in the past decade, as their sparser and more structured representation compared to traditional models has the potential to help design simpler algorithms for computing equilibria. They also may help identify nontrivial classes of games for which an equilibrium can be found in polynomial time. Finally, they build connections with the well-developed field of inference in graphical models. Following that reasoning, the objective of this chapter is twofold: first, we want to develop a framework for creating new message-passing schemes to compute Nash equilibria. Second, we want to identify sufficient conditions for the fast computation of Nash equilibria. In particular, we want to prove that under a suitably defined correlation decay condition, simple distributed schemes can compute Nash equilibria in polynomial time for particular graphical games. The fact that the scheme would be distributed, decentralized, efficient, and near-optimal would give credence to the notion of Nash equilibrium as a model for social behavior.

Literature review

The first models for graphical games can be found in [KLS01a] (for a game-theoretic analog of Markov Random Fields), and in a slightly different fashion, in [VK02, KM03] and [LM00] (for a game-theoretic analog of Bayesian networks, more adapted to studying the question of causality in games). In their original paper [KLS01a], Kearns *et al.* present a general framework for graphical games, and introduce *TreeProp*, a simple message-passing algorithm for computing exact and approximate Nash equilibria in trees (the exact algorithm runs in exponential time in the worst case, while the approximate algorithm is a FPTAS). An important feature of their algorithm is that it provides a representation of all Nash

equilibria (approximate or not). They also propose in [LKS02] a heuristic modification of TreeProp aimed at finding one Nash equilibrium in polynomial time for trees (while it does run in polynomial time, the algorithm, unlike TreeProp, is not guaranteed to find an equilibrium). On the same topic, Elkind *et al.* prove in [EGG06] that it is unlikely that algorithms similar to TreeProp would be able to find an exact Nash equilibrium for trees (the question of whether this is possible or not nevertheless remains open).

Later, Ortiz and Kearns generalize TreeProp into *NashProp*, a message-passing heuristic which tries to compute Nash equilibria for general graphs (in a very similar fashion to the way Belief Propagation is generalized to arbitrary networks). They prove that *NashProp* is a convergent search algorithm which correctly reduces the size of the search space (in the sense that it only removes bad solutions from the space of solutions, but not necessarily all of them). Kakade *et al.* [KKLO03] investigate the computational issues behind the more general notion of *correlated equilibrium* in graphical games; they discover a very fruitful connection with Markov Random Fields. They use this connection to show that under technical assumptions, the correlated equilibrium can be sparsely represented as well, and under the assumption that the graph is chordal, prove that the correlated equilibrium can be computed in polynomial time. Another generalization to TreeProp, this time for games of imperfect information, is developed in [SSW04].

Other techniques for computing Nash equilibria in graphs are developed in [KM03, SSW07], where connections between *constraint satisfaction problems* (CSPs) and graphical games are used to recast TreeProp as a constraint satisfaction algorithm; in [BSK06], a version of the classical homotopic algorithm for computing NE is specialized to the structure of graphical games.

In [DP06], Daskalakis and Papadimitriou shows that the computation of pure Nash equilibria in graphical games can be recast as a MAP problem in a related Markov Random field, thus formally establishing a connection between graphical game theory and graphical models. This connection is used to exhibit a first class of nontrivial graphs for which the Nash equilibrium can be efficiently computed (if it exists), namely, graphs with bounded treewidth. A similar result is established again in [JLB07] in the more specific case of *action graphical games* (games where payoffs depend only on the set formed by the action of all the neighbors of a node, and not on which neighbor took which action).

Approximation algorithms for graphical games are considered in a number of papers, a 0.5-approximation algorithm is first exhibited in [DMP09], later improved to 0.38 [DMP07], and finally to 0.34, the current state of the art, in [TS07].

Random graphical games have also been studied in a number of papers. Often, these papers investigate the number of pure Nash equilibria in games with either random graphs or random payoffs. Rinott and Scarsini look at a complete graph with arbitrary correlations between payoffs in [RS00]. Graphical games *per se* are studied in [DGS07] (the authors assume a variety of fixed graph topologies and symmetric i.i.d. payoffs), and [CDM08] (the authors prove more general results and provide bounds on the expected number of NE for arbitrary graph topology and symmetric, i.i.d. payoffs; they also consider random Erdos-Renyi graphical games, and show the existence of a double-phase transition for the existence of a pure Nash equilibrium). In a different direction, Barany, Vempala and Vetta [BVV07] show that Nash equilibria are in some sense easier to compute in random games, and provide FPRAS for two player games, in the cases of uniform and Gaussian distributions.

We conclude this section by mentioning that in more specific settings, Saberi and Montanari [MS09] and Chien and Sinclair [CS09] propose simple iterative algorithms on graphs and show their convergence to Nash equilibria.

5.2 Game theory, Nash equilibria, and approximate Nash equilibria

Games, strategies, solutions

A *normal form game* is defined as a triplet (V, χ, Φ), where V is a set of agents, $\chi = \{0, 1, \ldots, T-1\}$ is a finite set of decisions for each player, and $\Phi = (\phi_u)_{u \in V}$ is a set of *utility* or *payoff* functions, with a given utility function $\phi_u : \chi^n \to \mathbb{R}$ for each agent u. For any $\mathbf{x} = (x_u) \in \chi^n$, $\phi_u(\mathbf{x})$ is the utility of agent u when agent v plays x_v.

A *strategy* $\mathbf{s}_u$ for player u is a set of probabilities for each action in χ: let $\mathbf{s}_u = (s_u(1), \ldots, s_u(T-1))$, such that for each i, $s_u(i) \in [0,1]$ and $\sum_i s_u(i) = 1$, then, $s_u(i)$ represents the probability that player u plays action i. Recall that for any finite set χ, we denote by $\mathcal{S}(\chi)$ the set of (discrete) probability distributions on χ. Therefore, a strategy $\mathbf{s}_u$ is formally an element of $\mathcal{S}(\chi)$. A strategy is said to be *pure* if there exists i such that $s_u(i) = 1$ and $s_u(j) = 0$ for $j \neq i$. In this case, we abuse notation and write $s_u = i$.

A *discretized* strategy of stepsize δ is a strategy $\mathbf{s}_u$ such that for all i, $s_u(i) = k\delta$, for some nonnegative integer k. The set of all δ-discretized strategy is denoted by $\mathcal{S}_\delta(\chi)$.

A *solution* (also called *strategy profile*) $\mathbf{s} = (\mathbf{s}_u)_{u \in \{1,\ldots,n\}}$ is a set consisting of one

strategy per player – in other words, it is an element of $\mathcal{S}(\chi)^n$. A solution is said to be pure if the strategy is pure for each player. Similarly, a solution is discretized if each strategy composing it is. For any function f on χ^n, its expected value under solution $\mathbf{s}$ is the expected value of f under the assumption that each player u plays i with probability $s_u(i)$, independently of other players.

$$\mathbb{E}_{\mathbf{s}}[f(\mathbf{x})] = \sum_{(x_1,\ldots,x_n)\in\chi^n} \prod_{u\in V} s_u(x_u) f(x_1,\ldots,x_n)$$

For any strategy $\mathbf{s}$ and player u, we define $\phi_u(\mathbf{s}) = \mathbb{E}_{\mathbf{s}}[\phi_u(\mathbf{x})]$.

For any player u, let $\mathbf{s}_{-u}$ be a tuple $(s_v)_{v\neq u}$ of strategies for all players but u. We call such a set a *complement solution* to player u. $\phi_u(s_u, \mathbf{s}_{-u})$ represents the utility of player u when he plays strategy s_u and the other players follow the strategy $\mathbf{s}_{-u}$.

Nash equilibrium

A *Nash equilibrium* (NE) is a solution $\mathbf{s}$ such that for each u, and each $x \in \chi$,

$$\phi_u(\mathbf{s}) \geq \phi_u(x, \mathbf{s}_{-u}) \tag{5.1}$$

Thus, for each player u, if all other players $v \neq u$ keep their strategy fixed to s_v, then u has no incentive to deviate from strategy s_u. It is known [FT91] that any normal form game admits a Nash equilibrium. A *pure Nash equilibrium* (PNE) is a pure solution $\mathbf{x}$ which is a Nash equilibrium. In other terms, tt is a set of action $\mathbf{x} = (x_1, \ldots, x_N) \in \chi^N$ such that for all $y \in \chi$ and $u \in V$,

$$\phi_u(\mathbf{x}) \geq \phi_u(y, \mathbf{x}_{-u}) \tag{5.2}$$

Finally, we will need the concept of approximate Nash equilibrium. For any $\epsilon > 0$, an ϵ-approximate Nash equilibrium is a solution $\mathbf{s}$ such that for all u and all $y \in \chi$,

$$\phi_u(\mathbf{s}) + \epsilon \geq \phi_u(y, \mathbf{s}_{-u}) \tag{5.3}$$

For any $\nu = (\epsilon, \delta)$, a δ-discretized solution which is a ϵ-approximate Nash equilibrium will be called a ν-Nash equilibrium. A pair (ϵ, δ) for which an (ϵ, δ)-NE is guaranteed to exist is called a *valid pair*. We only consider valid pairs throughout this chapter (an easy check

for (ϵ, δ) to be a valid pair is given by Proposition 17).

Best reponse function

For any player u, the *best response* function is the function $\mathrm{BR}_u : \mathcal{S}(\chi)^{N-1} \to 2^{\chi}$ such that for any $\mathbf{s}_{-u} \in \mathcal{S}(\chi)^{N-1}$,

$$\mathrm{BR}_u(\mathbf{s}_{-u}) = \mathrm{argmax}_x \phi_u(x, \mathbf{s}_{-u}) \tag{5.4}$$

In other words, for any $x \in \chi$, $x \in \mathrm{BR}_u(\mathbf{s}_{-u})$ if and only if $\phi_u(x, \mathbf{s}_{-u}) \geq \phi_u(y, \mathbf{s}_{-u})$, for all $y \in \chi$. From this, we obtain the following alternative definition of a pure Nash equilibrium: $\mathbf{x}$ is a PNE if and only if for all u, $x_u \in \mathrm{BR}_u(\mathbf{x}_{-u})$.

By extension, for any player u, and any complement strategy $\mathbf{s}_{-u}$, we will say that a strategy s_u belongs to the best response of $\mathbf{s}_{-u}$ if and only if for any action $y \in \chi$, we have

$$\phi_u(s_u, \mathbf{s}_{-u}) \geq \phi_u(y, \mathbf{s}_{-u})$$

Note this is simply the same as requiring that $\mathrm{Supp}(s_u) \subset \mathrm{BR}_u(\mathbf{s}_{-u})$. For each player u, we can also define an ϵ-approximate best response to a complement strategy $\mathbf{s}_{-u}$ by $\mathrm{BR}^{\epsilon}(\mathbf{s}_{-u}) = \{s_u \in \mathcal{S}(\chi) \mid \forall y \in \chi,\ \phi_u(s_u, \mathbf{s}_{-u}) + \epsilon \geq \phi_u(y, \mathbf{s}_{-u})\}$ and conclude that $\mathbf{s}$ is an ϵ-approximate NE if and only if for all u, $s_u \in \mathrm{BR}^{\epsilon}(\mathbf{s}_{-u})$.

5.3 Graphical games

Basic Model

A graphical game $\mathcal{G} = (V, E, \chi, \Phi)$ is a normal form game, where for each $u \in V$, the utility function ϕ_u is only allowed to depend on the action x_u and the actions $\mathbf{x}_{\mathcal{N}(u)}$ of the neighbors of u. For any u, let $\mathcal{N}(u)^e = \{u\} \bigcup \mathcal{N}(u)$. Then, ϕ_u is a function from $\chi^{|\mathcal{N}(u)^e|}$ to $\mathbb{R}$. In particular, this implies that for each player u, the best response function BR_u depends only on the strategies of the neighbors of u.

Given a subset of vertices $\mathbf{v} = (v_1, \ldots, v_k)$ and strategies $\mathbf{s}_{\mathbf{v}} = (s_{v_1}, \ldots, s_{v_k})$, let $\mathcal{G}[\mathbf{v} : \mathbf{s}]$ be the game obtained by fixing the strategies of each node $v_i \in \mathbf{v}$ to s_i. Mathematically, we have $\mathcal{G}[\mathbf{v} : \mathbf{s}] = (V', E', \chi, \Phi)$, where $V' = V \setminus \{v_1, \ldots, v_k\}$, $E' = E \cap (V' \times V')$, and for

any node u with neighbors $(w_1, \ldots, w_d)$ in V', and any $x_u, x_{w_1}, \ldots, x_{w_d}$ in χ, we have

$$\phi'_u(x_u, x_{w_1}, \ldots, x_{w_d}) = \mathbb{E}_{\mathbf{s_v}} \left[\phi_u(x_u, x_{w_1}, \ldots, x_{w_d}, s_{v_1}, \ldots, s_{v_k})\right]$$

(where a variable s_{v_i} appears on the expression above only if v_i was a neighbor of u in (V, E))

Note the following important fact: by definition, for any Nash equilibrium in $\mathcal{G}[\mathbf{v} : \mathbf{s}]$, the agents $v \in \mathbf{v}$ do not have to be in best-response to their neighbors, they are excluded from the game and serve as fixed boundary conditions.

Decomposable graphical games

We will need a further simplifying assumption regarding the structure of graphical games. A *decomposable graphical game* is a set (V, E, χ, Φ), with $\Phi = ((\psi_u)_{u \in V}, (\phi_{u \leftarrow v})_{(u \leftarrow v) \in \overrightarrow{E}})$. For each $u \in V$, ψ_u is a function from χ to $\mathbb{R}$, and for each oriented edge $u \leftarrow v$, $\phi_{u \leftarrow v}$ is a function from χ^2 to $\mathbb{R}$. In a decomposable game, the utility function for agent u, given actions $\mathbf{x}_{\mathcal{N}(u)^e}$, is

$$\phi_u(\mathbf{x}_{\mathcal{N}(u)^e}) = \psi_u(x_u) + \sum_{v \in \mathcal{N}(u)} \phi_{u \leftarrow v}(x_u, x_v)$$

Directed tree game

Our final and simplest model is the *directed tree game*. A directed tree game is a game $\mathcal{G} = (V, E, o, \chi, \Phi)$, where $\mathcal{T} \triangleq (V, E)$ is an out-tree with root o, and for each node v in the the tree, its utility function only depends on the action of that player, and the actions of its children. This is in contrast with a graphical game on a tree, where the utility function of each player depends on its action and the action of all its neighbors (in other words, given an arbitrary orientation of the tree, the utility function depends on a player's action, the action of its children, but also the action of its parent). For each node v, denote by $\mathcal{K}(v)$ the set of children of v in $\mathcal{T}$ (with orientation given by the root o). Thus, for any node u, ϕ_u is a function of $\mathbf{s}_{\mathcal{K}(u)}$. Directed tree games have the additional property that they always admit a pure Nash equilibrium:

Property 1. *There exists a pure Nash equilibrium for any directed tree game. Moreover, if all the payoff functions are injective, the pure Nash equilibrium is unique.*

Proof. We prove the result by induction. Clearly, for any directed tree game with a single

node v and payoff function ϕ_v, $\text{argmax}(\phi_v)$ is the set of pure Nash equilibria for that game. Furthermore, if ϕ_v is injective, the maximizing decision is unique, and so is the PNE.

Now consider a general directed tree game $\mathcal{G} = (V, E, o, \chi, \Phi)$, and let $(v_1, \ldots, v_d)$ be the children of o in $\mathcal{G}$. For each v_i, let $\mathcal{G}_{v_i}$ denote the subgame induced by the subtree rooted at v_i, each $\mathcal{G}_{v_i}$ is a directed tree game. By the induction hypothesis, each $\mathcal{G}_{v_i}$ admits a pure Nash equilibrium which we denote $\mathbf{x}_i$, and let x_{v_i} be the decision of v_i in $\mathbf{x}_i$. Then, take any $x_o \in \text{argmax}_x(\phi_o(x, x_{v_1}, \ldots, x_{v_d}))$, and let $\mathbf{x} = (x_o, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d)$. By construction, $\mathbf{x}$ is a pure Nash equilibrium for $\mathcal{G}$. Furthermore, if the payoff functions were injective, $\mathbf{x}_i$ was unique for each i, and so was x_o, and we conclude that the pure Nash equilibrium is unique as well. □

5.4 Computation of Nash equilibrium

5.4.1 Nash cavity function

Our general aim is to identify algorithms for computing approximate Nash equilibria in graphical games. There are three different problems of interest. The first one consists of finding any (depending on the problem, mixed, pure, or approximate) Nash equilibrium. The second consists of finding a Nash equilibrium with particular properties (maximizing total payoff, maximizing payoff of a single player, maximizing minimal payoff). The third consists of finding all Nash equilibria (note that traditional notions of computational complexity are harder to apply here, since it is not clear that the set of all Nash equilibrium can be succinctly represented). Clearly, the first problem is easier than the second, which is easier than the third; we a priori consider all three. We also wish to analyze spatial properties of the equilibria, and in particular, we wish to analyze the potential decentralization properties of Nash equilibria. Our main tool to compute Nash equilibria will be the following functions.

Given a graphical game $\mathcal{G}$ and node $v \in \mathcal{G}$, let $Z_{G,v}$ be a function from $\mathcal{S}(\chi)$ to R^+, such that $Z_{\mathcal{G},v}(s_v) > 0$ if and only if there exists a Nash equilibrium in $\mathcal{G}$ in which v plays s_v. We call these functions *Nash cavity functions*, they are the analog of the cavity functions of Chapters 1–4. We will focus in particular on two special kinds of Nash cavity functions. A *binary* Nash cavity function is a function $Z_{\mathcal{G},v}(s_v)$ which is equal to 1 if and only if there exists a Nash equilibrium in $\mathcal{G}$ in which v plays s_v, and 0 otherwise. A set $(Z_v)_{v \in V}$ of Nash cavity functions for each node in $\mathcal{G}$ is called a set of *local* Nash cavity functions if and

only if for each v, $\text{argmax}_{s_v} Z_v(s_v)$ is composed of a unique element s_v, and the solution $\mathbf{s} = (s_v)_{v \in V}$ is Nash equilibrium. Similar definitions for (ϵ, δ)-approximate (resp. pure) Nash equilibria are denoted Z_v^ν (resp. Z^p). Both local and binary cavity functions have their own interest: binary cavity functions are useful because they are more symmetrical (i.e., they do not differentiate between any of the Nash equilibrium), and are the most natural tool to search for the set of all Nash equilibria. Furthermore, as we will explain later, they are useful for the study of message-passing algorithms for graphical games. This is because message-passing algorithms that aim to compute binary Nash functions tend to have strong convergence properties. Another way to look at binary Nash cavity functions is that they are simply indicator functions of any other Nash cavity function. On the other hand, knowing local cavity functions can be extremely useful to locally compute Nash equilibria, as well as finding a Nash equilibrium with particular properties (for instance, finding a Nash equilibrium which maximizes total welfare).

5.4.2 From Nash Cavity functions to Nash equilibria

In this section, we will motivate further the problem of computing Nash cavity functions, by relating that problem to the problem of computing Nash equilibria. Given a set $\mathbf{w} = (w_1, \ldots, w_k)$, let $Z_{\mathcal{G},\mathbf{w}}$ be a function from $\mathcal{S}(\chi)^k$ to $[0, +\infty)$ such that $Z_{\mathcal{G},\mathbf{w}}(\mathbf{s_w}) > 0$ if and only if there exists a Nash equilibrium in $\mathcal{G}$ where for each $i = 1, \ldots, k$, w_i plays s_{w_i}. Similarly, let $Z^\nu_{\mathcal{G},\mathbf{w}}$ be a function from $\mathcal{S}_\delta(\chi)^k$ to $[0, +\infty)$ such that $Z^\nu_{\mathcal{G},\mathbf{w}}(\mathbf{s_w}) > 0$ if and only if there exists a ν-Nash equilibria in which w_i plays s_{w_i} for all i. Finally, let $Z^p_{\mathcal{G},\mathbf{w}}$ be a function from χ to $[0, \infty)$ such that $Z_{\mathcal{G},\mathbf{w}}(\mathbf{x_w}) > 0$ if and only if there exists a pure Nash equilibrium from $\mathcal{G}$ in which w_i plays x_{w_i}. The reason we are interested in these is because of the following reductions:

- Given an algorithm $\mathcal{A}$ which computes $Z^\nu_{\mathcal{G},\mathbf{w}}$ for any $(\mathcal{G}, \mathbf{w})$, there exists an algorithm $\mathcal{A}'$ which outputs an ν-Nash equilibrium for any $\mathcal{G}$ if there exists one. Conversely, given an algorithm $\mathcal{A}'$ which finds all ν-Nash equilibrium for $\mathcal{G}$, there exists an algorithm $\mathcal{A}$ which computes some function $Z^\nu_{\mathcal{G},\mathbf{w}}$ with the desired properties.

- Given an algorithm $\mathcal{A}$ which computes $Z_{\mathcal{G},\mathbf{w}}$ for any $(\mathcal{G}, \mathbf{w})$, there exists an algorithm $\mathcal{A}'$ which outputs a Nash equilibrium for any $\mathcal{G}$. Conversely, given an algorithm $\mathcal{A}'$ which finds all Nash equilibrium for $\mathcal{G}$, there exists an algorithm $\mathcal{A}$ which computes some Nash Cavity function $Z_{\mathcal{G},\mathbf{w}}$.

Note that the reductions above are trivial: a binary $Z_{\mathcal{G},\{v_1,\ldots,v_n\}}(\mathbf{s})$ is exactly the the indicator function $\mathbf{1}(\mathbf{s}$ is a Nash equilibrium for $\mathcal{G})$. Computing $Z_{\mathcal{G},\{v_1,\ldots,v_n\}}$ is however not a practical goal, as the very high dimension of the function makes it impossible to store it, let alone compute it. The hope is that in graphical games, computing $Z_{\mathcal{G},\{v_1,\ldots,v_k\}}$ for small values of k is sufficient (practically or theoretically) to find at least one Nash equilibrium. For instance, the main result of [KLS01a] shows that computing the $Z_{\mathcal{G},v}(s_v)$ and $Z_{\mathcal{G}[u:s_u],v}(s_v)$ for any u, v is enough to find all Nash equilibria on trees, and that the $Z_{\mathcal{G}[u:s_u],v}(s_v)$ can be computed recursively by a message-passing algorithm (for details on the algorithm, see section 5.5).

Proposition 16 (upstream phase of TreeProp [KLS01a]). *Given a game $\mathcal{G}$ defined on a tree, and Nash cavity functions $Z_{\mathcal{G},v}(s_v)$ and $Z_{\mathcal{G}[u:s_u],v}(s_v)$, the following algorithm finds a Nash equilibrium in polynomial time:*

TreeProp(upstream phase): TP$[\mathcal{G}, Z_{\mathcal{G},v}(s_v), Z_{\mathcal{G}[u:s_u],v}(s_v)]$
INPUT: A graphical game $\mathcal{G}$ defined on a tree, and Nash cavities
$Z_{\mathcal{G},v}(s_v), Z_{\mathcal{G}[u:s_u],v}(s_v)$
BEGIN
Choose an arbitrary node u, and choose s_u such that $Z_{\mathcal{G},u}(s_u) > 0$
Letting $(v_1, \ldots, v_d)$ be the neighbors of u, find $(s_{v_1}, \ldots, s_{v_d})$ such that $Z_{\mathcal{G}[u:s_u],v_i}(s_{v_i}) = 1$ for all i, and send a tag s_{v_i} to v_i.
FOR all vertices $w \neq u$ in $\mathcal{G}$ (in order of proximity to u) DO:
Receive a tag s_w from the parent node v . Fix decision to s_w.
Letting $(v, w_1, \ldots, w_d)$ be the neighbors of w, find $(s_{w_1}, \ldots, s_{w_d})$ such that for all i, $Z_{\mathcal{G}[v:s_v],w_i}(s_{w_i}) = 1$, and send tag s_{w_i} to each i.
END DO. END BEGIN.
OUTPUT: $\mathbf{s} = (s_v)_{v\in\mathcal{G}}$ such that s is a Nash equilibrium.

The algorithm can be proven to be correct by a simple induction. Extensive numerical evidence [VK02, OK03, KM03, SSW07] suggests that simpler versions of the search algo-

rithm, detailed below, are very efficient at computing Nash equilibria.

NashSearch: NS$[\mathcal{G}, (V_1, \ldots, V_k), (W_1, \ldots, W_k), Z_{\mathcal{G}[V_i:\cdot],W_i}(\cdot), Z_{\mathcal{G},v}]$
INPUT: A game $\mathcal{G}$, two sets $(V_1, \ldots, V_k)$ and $(W_1, \ldots, W_k)$, where for each i, V_i and W_i are subsets of V such that $V_i \cap W_i = \emptyset$. Assume we are given the function $Z_{\mathcal{G}[V_i:],W_i}(\cdot)$ for each i, as well as functions $Z_{\mathcal{G},v}$
BEGIN
Let $F = \emptyset$
WHILE $F \neq V$ DO
Pick a node in u in $V \setminus F$ such that $R_u \triangleq \{i : u \in V_i, V_i \subset F, W_i \setminus v \subset F\}$ is largest in cardinality. Find s_v such that $Z_v(s_v) > 0$ and such that for all $i \in R_v$,
$Z_{\mathcal{G}[S_i;s_{S_i}],v}(s_v, s_{T_i \setminus \{v\}}) > 0$
OUTPUT: $\mathbf{s} = (s_v)_{v \in \mathcal{G}}$ is a candidate for Nash equilibrium.

The set F represents the set of all nodes v for which the strategy s_v has been fixed already. The set R_v represents the set of all constraints which are checkable, meaning that all strategies in the boundary condition S_i are fixed, and all strategies the constraint depends on (T_i) are already fixed, except for the strategy of s_v. Intuitively speaking, the algorithm above simply tries to find a Nash equilibrium by sequentially looking for strategies s_v which satisfy currently checkable constraints (meaning a constraint for which all strategies but s_v have already been fixed by the algorithm).

5.4.3 Existence of approximate Nash equilibria

Graphical games are special cases of normal form games, and they therefore admit a Nash equilibrium. In general, the equilibrium will be mixed, and the corresponding solution will therefore be continuous. This creates a problem, since both message-passing algorithms and correlation decay methods are not well suited for continuous problems. The reason we are interested in ϵ-approximate Nash equilibrium is that there always exists a discretized

strategy (with δ small enough) which is an ϵ-Nash equilibrium (cf. the following proposition). This will allow us to view strategies as discrete rather than continuous objects, which will make the use of graphical models. Suppose $\chi = \{0, 1, \ldots, T-1\}$, and let Δ be the maximum degree of the graph. The following is an improved version of a lemma from Kearns [KLS01a]. For any normal form game, denote $||\Phi||_\infty$ be the maximum absolute utility over decisions and players: $||\Phi||_\infty \stackrel{\Delta}{=} \max_{u,\mathbf{x}\in\chi^N} |\phi_u(\mathbf{x})|$.

Proposition 17. *For any graphical game (V, E, χ, Φ), and for any $\epsilon > 0$, take an integer $n > 2T^{\Delta+1}(\Delta+1)\,||\Phi_u||_\infty$, and let $\delta = \frac{1}{n}$. Then there exists a δ-discretized solution $\mathbf{s}^\delta$ which is a ϵ-approximate NE.*

The following proof follows very closely that of Kearns *et al.*, using more general arguments and more careful bounds, and is included for completeness. We first need the following lemmas, the first of which is new, and the second of which is a generalization to more than two actions, and has a better dependency on n.

Lemma 16. *Let s_u be a strategy over T actions, and n be some positive integer. Let $\delta = \frac{1}{n} > 0$. Then, there exists a δ-discretized strategy s_u^δ such that for all i*

$$|s_u(i) - s_u^\delta(i)| \leq \delta$$

Proof. For any i, let $k_i^- = \lfloor n s_u(i) \rfloor$ and $k_i^+ = \lceil n s_u(i) \rceil$. For all i,

$$k_i^- \delta \leq s_u(i) \leq x_i^+ \delta \tag{5.5}$$

and

$$k_i^+ - k_i^- \leq 1 \tag{5.6}$$

By summing over i, we obtain

$$(\sum_i k_i^-)\delta \leq 1 \leq (\sum_i k_i^+)\delta$$

For any $0 \leq j \leq T$, consider the vector $\mathbf{k}^j$ defined by $\mathbf{k}^j(i) = k_i^-$ if $i > j$ and k_i^+ otherwise. In particular $\mathbf{k}^0 = \mathbf{k}^-$, and $\mathbf{k}^T = \mathbf{k}^+$. Note that $\sum_i \mathbf{k}^0(i) \leq n$ and $\sum_i \mathbf{k}^T(i) \geq n$. Since for any j, we have $\sum_i \mathbf{k}^{j+1}(i) - \sum_i \mathbf{k}^j(i) \leq 1$, there exists some j' such that $\sum_i \mathbf{k}^{j'}(i) = n$.

This implies that $s_u^\delta \triangleq \mathbf{k}^{j'}\delta$ is a δ-discretized strategy, and from Equations (5.5) and (5.6), we obtain $|s_u(i) - s_u^\delta(i)| \leq \delta$ for all i. □

Lemma 17. *Let* $\mathbf{x}$ *and* $\mathbf{y}$ *be two elements of the* m*-dimensional simplex, such that* $\sup_i |x_i - y_i| \leq \delta$. *Then,* $|\prod_i x_i - \prod_i y_i| \leq m\delta$

This is proven by simple application of the formula

$$\prod_i x_i - \prod_i y_i = \sum_{1\leq i\leq m} \Big(\prod_{1\leq j\leq i-1} x_j\Big)(x_i - y_i)\Big(\prod_{i+1\leq j\leq k} y_j\Big)$$

Lemma 18. *Let* u *be a player with* Δ_u *neighbors. Consider two solutions* $\mathbf{s}, \mathbf{s}^\delta$ *such that for any* $v \in V$ *and* $x_v \in \chi$, *we have* $|s_v(x_v) - s_v^\delta(x_v)| \leq \delta$. *Then,*

$$\left|\phi_u(\mathbf{s}) - \phi_u(\mathbf{s}^\delta)\right| \leq T^{\Delta_u+1}(\Delta_u + 1) \cdot ||\Phi||_\infty \cdot \delta$$

Proof. Recalling that $\mathcal{N}(u)^e = \mathcal{N}(u) \cup \{u\}$, and expanding the expectations over all possible outcomes, we obtain:

$$\begin{aligned}
\left|\phi_u(\mathbf{s}) - \phi_u(\mathbf{s}^\delta)\right| &= \left|\sum_{\mathbf{x}\in\chi^{|\mathcal{N}(u)'|}} \left(\prod_{v\in\mathcal{N}(u))'} s_v(x_v) - \prod_{v\in\mathcal{N}(u))'} s_v^\delta(x_v)\right) \phi_u(\mathbf{x})\right| \\
&\leq \sum_{\mathbf{x}\in\chi^{|\mathcal{N}(u)'|}} |\phi_u(\mathbf{x})| \cdot (\Delta_u + 1)\,\delta \\
&\leq T^{\Delta_u+1}\,(\Delta_u + 1)\,||\Phi||_\infty\,\delta
\end{aligned}$$

where the second inequality is from Lemma 17, and the third inequality comes from counting the number of elements in $\chi^{|\mathcal{N}(u)'|}$. □

Proof of Proposition 17. Since $\mathcal{G}$ is a normal form game, there exists a solution s which is a Nash equilibrium. For each v, consider the δ-discretized strategy $\mathbf{s}_v^\delta$ which approximates s_v as in Lemma 16, and let $\mathbf{s}^\delta = (s_v^\delta)_{v\in V}$. Consider an arbitrary player u and action $y \in \chi$.

Then,

$$\begin{aligned}\phi_u(y, \mathbf{s}^\delta_{-u}) - \phi_u(s^\delta_u, \mathbf{s}^\delta_{-u}) &= \phi_u(y, \mathbf{s}^\delta_{-u}) - \phi_u(y, \mathbf{s}_{-u}) \\ &+ \phi_u(y, \mathbf{s}_{-u}) - \phi_u(s_u, \mathbf{s}_{-u}) \\ &+ \phi_u(s_u, \mathbf{s}_{-u}) - \phi_u(s^\delta_u, \mathbf{s}^\delta_{-u}) \\ &\leq \epsilon/2 + 0 + \epsilon/2 \leq \epsilon\end{aligned}$$

The first and third summand are upper bounded by $\epsilon/2$ by application of Lemma 18 and the choice of δ. The second summand is upper bounded by 0 since $\mathbf{s}$ is a NE. □

5.5 Message-passing algorithms for graphical games

In this section, we explore the use of message-passing for graphical games. We first introduce the TreeProp and NashProp algorithms of [KLS01a], the canonical equivalents of the BP algorithm to the settings of graphical games. Then, in a spirit similar to the pure Nash-MRF reduction of Daskalakis *et al.* [DP06], we present a framework further establishing the connection between graphical games and optimization in graphical models. This allows in principle the derivation of a large number of new algorithms for graphical games. In particular, we prove that TreeProp is a special case of the cavity algorithm applied to the graphical model derived from a graphical game on a tree. In addition, we show how applying the cavity recursion to a general graphical game results in a new heuristic for finding Nash equilibria.

5.5.1 TreeProp and NashProp

Consider a graphical game $\mathcal{G} = (V, E, \chi, \Phi)$ such that (V, E) is a tree $\mathcal{T}$. For any two nodes u, v and strategies s_u, s_v, we are interested in computing binary Nash cavities $Z_{\mathcal{G}[u:s_u],v}(s_v)$. By analogy with the original paper, and in order to highlight the connection with a message-passing paradigm, we will also denote this quantity by $T_{u \leftarrow v}(s_u, s_v)$. The main result of [KLS01a] was the following:

Theorem 13 (Downstream pass of [KLS01a]). *Consider an arbitrary node $v \in V$, and let $(u, w_1, \ldots, w_d)$ be the neighbors of v. then, for all $(s_u, s_v) \in \mathcal{S}(\chi)^2$, $T_{u \leftarrow v}(s_u, s_v) = 1$ if and only if there exists $(s_{w_1}, s_{w_2}, \ldots, s_{w_d}) \in \mathcal{S}(\chi)$ such that*

1. *For all i, $T_{v \leftarrow w_i}(s_v, s_{w_i}) = 1$*

2. $s_v \in BR_v(s_u, s_{w_1}, \ldots, s_{w_d})$

Furthermore, for any node $u \in V$ with neighbors $\{v_1, \ldots, v_d\}$, $Z_u(s_u) = 1$ if and only if there exist $(s_{v_1}, \ldots, s_{v_d}) \in \mathcal{S}(\chi)$ such that

1. *For all i, $T_{u \leftarrow v_i}(s_u, s_{v_i}) = 1$*

2. $s_u \in BR_u(s_u, s_{v_1}, \ldots, s_{v_d})$

Proof. (included for completeness)
Recall that for any tree $\mathcal{T}$, removing an edge (u, v) separates $\mathcal{T}$ into $\mathcal{T}_{v \leftarrow u}$, which contains u, and $\mathcal{T}_{u \leftarrow v}$, which contains v. A Nash equilibrium for $\mathcal{G}[u : s_u]$ is composed of a Nash equilibrium for $\mathcal{T}_{u \leftarrow v}[u : s_u]$ and a Nash equilibrium for $\mathcal{T}_{v \leftarrow u}[u : s_u]$ (both conditional on u playing s_u). Since no payoff function of $\mathcal{T}_{u \leftarrow v}[u : s_u]$ depends on any node of $\mathcal{T}_{v \leftarrow u}[u : s_u]$ (and vice-versa), there exists a Nash equilibrium in $\mathcal{G}[u : s_u]$ in which v plays s_v if and only if there exists a Nash equilibrium in $\mathcal{T}_{u \leftarrow v}[u : s_u]$ in which v plays s_v.

Let us assume such an equilibrium exists, and let $\mathbf{s}_{u \leftarrow v}$ denote this Nash equilibrium, and let $(s_{w_1}, \ldots, s_{w_d})$ denote the actions of $(w_1, \ldots, w_d)$ in this equilibrium. Clearly, by projecting $\mathbf{s}_{u \leftarrow v}$ on each subtree $\mathcal{T}_{v \leftarrow w_i}$, there exists for each i a Nash equilibrium for each $\mathcal{T}_{v \leftarrow w_i}[v : s_v]$ where w_i plays s_{w_i}. Therefore, for each i, $T_{v \leftarrow w_i}(s_v, s_{w_i}) = 1$. Furthermore, since $(s_v, s_{w_1}, \ldots, s_{w_d})$ are the actions of $(v, w_1, \ldots, w_d)$ in a NE of $\mathcal{T}_{u \leftarrow v}[u : s_u]$, we obtain that $s_v \in \mathrm{BR}_v(s_u, s_{w_1}, \ldots, s_{w_d})$. This prove the "if" part.

For the only if part, we just reverse the argument. If there exist $(s_v, s_{w_1}, \ldots, s_{w_d})$ such that $T_{v \leftarrow w_i}(s_v, s_{w_i}) = 1$, then this means there exist Nash equilibria $\mathbf{s}_{v \leftarrow w_i}$ for each $\mathcal{T}_{w_i \leftarrow v}[v : s_v]$. Combining the equilibria into $s_{u \leftarrow v} = (s_v, \mathbf{s}_{v \leftarrow w_1}, \ldots, \mathbf{s}_{v \leftarrow w_d})$, we see that $\mathbf{s}_{u \leftarrow v}$ is a Nash equilibrium for $\mathcal{T}_{u \leftarrow v}[u : s_u]$, and therefore $T_{u \leftarrow v}(s_u, s_v) = 1$. □

By the exact same arguments, similar results are proven for (ϵ, δ)-approximate and pure Nash equilibria: for any node v with neighbors $\{u, w_1, \ldots, w_d\}$, δ-discretized strategies s_u, s_v, and decisions x_u, x_v, we let $T^{\nu}_{u \leftarrow v}(s_u, s_v) \triangleq Z^{\nu}_{\mathcal{G}[u:s_u],v}(s_v)$ and $T^{p}_{u \leftarrow v}(x_u, x_v) \triangleq Z^{p}_{\mathcal{G}[u:x_u],v}(x_v)$.

Theorem 14. *For all $(s_u, s_v) \in \mathcal{S}(\chi)^2$, $T_{u \leftarrow v}(s_u, s_v) = 1$ if and only if there exists $s_{w_1}, s_{w_2}, \ldots, s_{w_d} \in \mathcal{S}_\delta(\chi)$ such that*

1. *For all i, $T^{\nu}_{v \leftarrow w_i}(s_v, s_{w_i}) = 1$*

2. $s_v \in BR^{\epsilon}_v(s_u, s_{w_1}, \ldots, s_{w_d})$

Theorem 15. *For all* $(s_u, s_v) \in \mathcal{S}(\chi)^2$, $T^p_{u \leftarrow v}(s_u, s_v) = 1$ *if and only if there exist* $s_{w_1}, s_{w_2}, \ldots, s_{w_d} \in \chi$ *such that*

1. *For all* i, $T^p_{v \leftarrow w_i}(s_v, s_{w_i}) = 1$
2. $s_v \in BR_v(s_u, s_{w_1}, \ldots, s_{w_d})$

Note that while the proof of correctness of Theorem 13, 14 and 15 relies on the fact that (V, E) was a tree, the update equations themselves do not. It is therefore possible to use the exact same equations for any graphical game $\mathcal{G}$. The resulting algorithm is called Nash propagation. By initializing $T_{u \leftarrow v}$ to be identically 1 for all u, v, Ortiz and Kearns [OK03] show that Nash Propagation always converges, and that true Nash equilibria of the network $\mathcal{G}$ satisfy the constraints implied by Equations (13), (14) or (15). In that sense, NashProp is an algorithm which correctly reduces the search space (without necessarily reducing it to the set of all Nash equilibria). In the next section, we will see how this property is a special case of Nash search algorithms for binary Nash cavities.

Finally, observe that since TreeProp computes binary Nash cavities, even if one could show that these cavities can be computed locally (eg., through a correlation decay argument), it is an intrinsically nonlocal algorithm: in order to compute a Nash equilibrium from the Nash cavity function, TreeProp needs to use the iterative upstream algorithm mentioned in section 5.4.2 (this algorithm has the additional undesirable property of distinguishing a root node u for no apparent reason). We will see how to alleviate this assumption in the next section, and how to turn TreeProp into a truly local algorithm.

5.5.2 A framework for deriving message-passing algorithms for graphical games

From now on, we only consider (ϵ, δ)*-approximate NE, and will omit this fact from statements and notations in order to avoid repetitions.*

In this section, we present a general framework for establishing a connection between inference in graphical models and computation of approximate equilibria in graphical games. Our method is related to the reduction of Daskalakis *et al.* [DP06], with two differences: first, we generalize the approach from pure Nash equilibria to approximate, discretized Nash equilibria; second, by using the factor graph framework rather than Markov Random fields, the resulting graphical model is arguably simpler to construct and has fewer edges

(and is therefore more amenable to the use of message-passing algorithms). Furthermore, our focus will be on the Nash cavity functions, as opposed to directly optimizing the joint probability of the corresponding MRF.

The reduction

The reduction is very simple: for any graphical game $\mathcal{G} = (V, E, \Phi, \chi)$ and any $\nu = (\epsilon, \delta)$, we build a graphical model $\mathcal{H} = (V, A, E', \Phi', \chi)$ such that:

- A, the set of factor nodes, is a set indexed by the elements of V: $A = \{a_v\,, v \in V\}$
- For each $v \in V$, $(u, a_v) \in E'$ if and only if $(u, v) \in E$ or $u = v$.
- For each $v \in V$, ϕ_{a_v} is a function of $(s_v, \mathbf{s}_{-v})$ such that

$$\phi_{a_v}(s_v, \mathbf{s}_{-v}) \geq 0 \quad \text{if} \quad s_v \in \mathrm{BR}_v^{\epsilon}(\mathbf{s}_{-v}) \tag{5.7}$$

$$\phi_{a_v}(s_v, \mathbf{s}_{-v}) = -K \qquad \text{otherwise} \tag{5.8}$$

and K satisfies

$$K \geq n \max_{v, s_v, \mathbf{s}_{-v}} \phi_{a_v}(s_v, \mathbf{s}_{-v})$$

We denote by $F_{\mathcal{H}}(\mathbf{s})$ the cost function corresponding to the graphical model $\mathcal{H}$. Note that given the way (V, A, E') is constructed, $(u, a_v) \in E'$ if and only if $(v, a_u) \in E'$. In particular, no matter what the topology of (V, E) was, (V, A, E') is not a tree (unless $E = \emptyset$). Moreover, we can always take $K = \infty$, but it is sometimes desirable to keep K bounded, in particular to ensure that the cavities on $\mathcal{H}$ stay well defined.

From the definition of $\mathcal{H}$, we immediately obtain the following result, which links optimal solutions of $\mathcal{H}$ to ν-approximate Nash equilibria of $\mathcal{G}$, and is an analogous to Lemmas 3.1 and 3.2 in [DP06].

Theorem 16. *For any $\mathbf{s} \in \mathcal{S}_{\delta}(\chi)$, $F_{\mathcal{H}}(\mathbf{s}) \geq 0$ if and only if $\mathbf{s}$ is an (ϵ, δ)-approximate Nash equilibrium for $\mathcal{G}$.*

Proof. For any solution $\mathbf{s}$, if there exists a node v for which s_v is not an ϵ-best response to its neighbors, then the total cost includes at least one $-K$ term. The total contributions of all the other factors being at most $(n-1)\max_{v, s_v, \mathbf{s}_{-v}} \phi_{a_v}(s_v, \mathbf{s}_{-v})$, we obtain that the total cost is at most $-\max_{v, s_v, \mathbf{s}_{-v}} \phi_{a_v}(s_v, \mathbf{s}_{-v})$, and is therefore strictly negative. Conversely, it

is clear that if $\mathbf{s}$ is a (ϵ, δ)-approximate Nash equilibrium, then the contribution of each factor is nonnegative, and therefore the total cost is nonnegative as well. $\square$

This construction allows us to establish another connection between graphical games and graphical models, namely, we prove that under certain assumptions, Nash cavities are directly related to the value functions and cavity functions of Chapters 1 and 2.

Property 2. *For any graphical game $\mathcal{G}$, and node $v \in V$, the function Z_v defined by $Z_v(s_v) = J_{\mathcal{H},v}(s_v)$ is a Nash cavity function for $\mathcal{G}$. Moreover, if $\mathcal{H}$ admits a unique maximum, then the set $(Z_v)_{v \in V}$ forms a set of local Nash cavity functions for $\mathcal{G}$ (i.e., $(\text{argmax } Z_v(s_v))_{v \in V}$ is a Nash Equilibrium). Finally, for any tree graphical game $\mathcal{G}$, the Nash cavity functions $Z_{\mathcal{G}[u:s_u],v}(s_v) = 1_{J_{\mathcal{H}[u:s_u],v}(s_v) \geq 0}$ are exactly equal to the messages of the TreeProp and NashProp algorithms.*

The proof follows directly from Theorem 16.

Choosing the parameters

Finally, by assigning different values for the feasible assignments to each factor, we obtain different properties for the optimal solution of $\mathcal{H}$. We propose different models as follows:

Theorem 17.

(a) Suppose that $K = +\infty$, and that for all $v \in V$, complementary strategies $\mathbf{s}_{-v}$, and $s_v \in BR_v^\epsilon(s_{-v})$, we set $\phi_{a_v}(s_v, \mathbf{s}_{-v}) = 0$. Then, the set of all (ϵ, δ)-approximate Nash equilibria is exactly the set $\{\mathbf{s} \mid F_{\mathcal{H}}(\mathbf{s}) = 0\}$

(b) Suppose that for all $v \in V$, complementary strategies $\mathbf{s}_{-v}$, and $s_v \in BR_v^\epsilon(\mathbf{s}_{-v})$, we set $\phi_{a_v}(s_v, \mathbf{s}_{-v}) = \phi_v(s_v, \mathbf{s}_{-v})$. Then, the optimal solution of $\mathcal{H}$ is a (ϵ, δ)-approximate Nash equilibrium which maximizes the total utility.

(c) Suppose that there exists $u \in V$ such that for all $v \neq u$, complementary strategies $\mathbf{s}_{-v}$, and $s_v \in BR_v^\epsilon(\mathbf{s}_{-v})$, we set $\phi_{a_v}(s_v, \mathbf{s}_{-v}) = 0$. Furthermore, suppose that for all $\mathbf{s}_{-u}$ and $s_u \in BR_u^\epsilon(\mathbf{s}_{-u})$, $\phi_{a_u}(s_u, \mathbf{s}_{-u}) = \phi_u(s_u, \mathbf{s}_{-u})$. Then, the optimal solution of $\mathcal{H}$ is a (ϵ, δ)-approximate Nash equilibrium which maximizes the utility of player u.

(d) Consider a collection of independent random variables $X_{v,x}$ uniformly distributed over $[0, 1]$ and indexed by player v and decision $x \in \chi$. Suppose that for all $v \in V$, complementary strategies $\mathbf{s}_{-v}$, and $s_v \in BR_v^\epsilon(\mathbf{s}_{-v})$, we set $\phi_{a_v}(s_v, \mathbf{s}_{-v}) = \mathbb{E}_{\mathbf{s}}[\sum_{x_v} X_{v,x_v}]$. Then, the optimal solution of $\mathcal{H}$ is unique with probability 1.

Proof. For the first statement, simply note that $\mathcal{H}$ is designed so that $F_{\mathcal{H}}(\mathbf{s})$ is equal to $-\infty$ if $\mathbf{s}$ is not a Nash equilibrium, and 0 otherwise. For the second (resp. third), note that $\mathcal{H}$ is designed so that $F_{\mathcal{H}}(\mathbf{s})$ is negative if $\mathbf{s}$ is not a Nash equilibrium, and equal to the total utility (resp. utility of player u) induced by $\mathbf{s}$ otherwise. Finally, for the last statement, note that the joint distribution of the utility of two distinct discretized solution has a density, and therefore, two distinct discretized solutions have almost surely distinct payoffs. Since there is only a finite number of discretized solutions, the probability that two have the same payoff is zero. Therefore, the solution which maximizes $F_{\mathcal{H}}$ is unique. □

Note that the uniform distribution assumption of the last point can be replaced by any nonnegative distribution with a density. In most cases, the approximate Nash equilibrium which maximizes the total utility (or a given player utility) is unique. This is for instance the case with probability 1 if the utility functions are random variables and if their distribution has a density. In this case, from Property 2, computing said Nash equilibrium and computing the local Nash cavity functions is equivalent.

5.5.3 Search algorithms

In the previous section, we showed that computing a Nash equilibrium for $\mathcal{G}$ is equivalent to computing the optimal solution of a suitably defined factor graph. It is therefore natural to wonder if the message-passing algorithms exposed in Chapters 1 and 2 can be converted into message-passing algorithms for graphical games. We show that this is partially the case, as the cavity expansion algorithm, applied to the factor graph of a tree game, is equivalent to the TreeProp algorithm. Using the cavity expansion algorithm, we also generalize the TreeProp algorithm into the *Nash Cavity Expansion* (NCE) algorithm, a message-passing family of heuristics which aims to compute Nash cavity functions for graphical games. Furthermore, we show that NCE algorithms, when they are designed to compute binary Nash cavities, always converge, and always decrease the size of the search space for NE. We show in particular that the NashProp algorithm of Ortiz and Kearns belongs to the NCE family. Finally, we show how to modify the TreeProp algorithm so that it computes local Nash cavities, thus removing the need for the second "upstream" phase of the TreeProp algorithm.

TreeProp as a special case of the Cavity Expansion

Take a graphical game $\mathcal{G}$ (we do not suppose for now that $\mathcal{G}$ is a tree), and consider the reduction from $\mathcal{G}$ to $\mathcal{H}$ where K is set to $+\infty$, and assignments $(s_v, \mathbf{s}_{-v})$ in best-response for factor v have utility $\phi_{a_v}(s_v, \mathbf{s}_{-v}) = 0$. Note that in the graphical model $\mathcal{H}$, any value function $J_{\mathcal{H},v}(s_v)$ can in fact be considered as the negative of a cavity function (or more precisely, of a censored cavity function, in the meaning of Chapter 4). Indeed, we have $J_{\mathcal{H},v}(s_v) = J_{\mathcal{H},v}(s_v) - J_{\mathcal{H}}$, since $J_{\mathcal{H}}$ is always zero (as there always exists a Nash equilibrium for $\mathcal{G}$, we know that the value function of $\mathcal{H}$ is 0). It is sometimes more intuitive to think of a censored cavity $J_{\mathcal{H}} - J_{\mathcal{H},v}(s_v)$ as a difference $J_{\mathcal{H},v}(\mathrm{f}) - J_{\mathcal{H},v}(s_v)$, where f represents the "free" action, meaning that v can optimize over it.

Now, consider any node $v \in V$ with neighbors $\{w_1, \ldots, w_d\}$, a strategy s_v, and suppose we wish to compute $J_{\mathcal{H},v}(s_v)$. In $\mathcal{H}$, v has neighbors $\{a_v, a_{w_1}, \ldots, a_{w_d}\}$. Let us apply the cavity recursion for factor graphs (2.10) in order to compute $J_{\mathcal{H},v}(s_v)$. We obtain:

$$\begin{aligned} J_{\mathcal{H},v}(s_v) &= J_{\mathcal{H},v}(s_v) - J_{\mathcal{H}} \\ &= \sum_{1 \leq i \leq d} \mu_{v \leftarrow a_{v_i}}(s_v, M_{\mathcal{H}(v,i,s_v)}) + \mu_{v \leftarrow a_v}(s_v, M_{\mathcal{H}(v,d+1,s_v)}) \end{aligned}$$

We obtain that $J_{\mathcal{H},v}(s_v)$ is 0 if and only if each term of the sum is 0 as well. We now make the following observation: since we chose the factor a_v to be last in our expansion, in each of the $\mathcal{H}(v, 1, s_v)$, the decision of v in factor a_v is actually fixed to the value f, the "free" decision variable. Therefore, the corresponding μ function is 0 (since there always exists a Nash equilibrium for any game, even if the decision of v is constrained to be s_v, as long as we do not check if v is in best-response to its neighbors). The first d terms of the sum are therefore zero (this can be verified through careful checking of the definitions). We are thus left with only one term, $\mu_{v \leftarrow a_v}(s_v, M_{\mathcal{H}(v,d+1,s_v)})$. In $\mathcal{H}(v, d+1, s_v)$, both the node v and its factor a_v has been removed, and in every other factor v was involved in, the decision of v is fixed to s_v. Thus, $\mathcal{H}(v, d+1, s_v)$ is simply $\mathcal{H}[v : s_v]$, and checking definitions, we find that $M_{\mathcal{H}[v:s_v]}$ is equal to $J_{\mathcal{H}[v:s_v],\mathbf{w}}(\mathbf{s_w})$. Finally, it is easy to check that $\mu_{v \leftarrow a_v}(s_v, J_{\mathcal{H}[v:s_v],\mathbf{w}}(\mathbf{s_w}))$ is equal to 0 if and only if there exists $\mathbf{s_w}$ such that:

- $s_v \in \mathrm{BR}_v(s_{w_1}, \ldots, s_{w_d})$
- $Z_{\mathcal{G}[v:s_v],\mathbf{w}}(\mathbf{s_w}) = 1$. In other words, there exists a NE in $\mathcal{G}[v : s_v]$ in which each player w_i plays s_{w_i}

Note that the above statement is in fact obvious, and that for completely general graphical games, the Cavity Expansion does not provide useful information. However, by applying Theorem 3 (Chapter 2) to $\mathcal{H}$, we immediately recover that $J_{\mathcal{H}[v:s_v],\mathbf{w}}(\mathbf{s_w}) = \sum_i J_{\mathcal{H}[v:s_v],w_i}(s_{w_i})$, or in other words that $Z_{\mathcal{G}[v:s_v],\mathbf{w}}(\mathbf{s_w}) = 1$ if and only if $Z_{\mathcal{G}[v:s_v],w_i}(s_{w_i}) = 1$ for all i. This is the mathematical way to say that there exists a NE in $\mathcal{G}[v:s_v]$ in which each player w_i plays s_{w_i} if and only if for each i, there exists a NE in $\mathcal{G}[v:s_v]$ in which w_i plays s_{w_i} (since all players w_i are decoupled after conditioning on v). We thus recovered the TreeProp equations as a special case of the CE algorithm.

Proposition 18. *Given a tree graphical game $\mathcal{G}$ and the corresponding graphical model $\mathcal{H}$, the Cavity Expansion algorithm applied to $\mathcal{H}$ has the same output as the output of TreeProp for $\mathcal{G}$, and thus computes exactly the Nash cavity functions.*

Nash cavity expansion

Let us try to go further. The following lemmas will help design useful (although non-exact) recursions for computing the Nash cavity functions:

Lemma 19. *Consider an arbitrary set $T \subset \{1,\dots,n\}$ and 0-1 function G, which takes as input an object $(\mathbf{s}_T, F_1, \dots, F_k)$ where:*

- *$\mathbf{s}_T$ is a set of strategies for the elements of T: $\mathbf{s}_T = (s_v)_{v\in T}$*
- *For each i, F_i is a function from $\mathcal{S}_\delta(\chi)$ to $\{0,1\}$*

Suppose that for any $\mathbf{s}_T$ and $(F_1,\dots,F_k)$, $G(\mathbf{s}_T, F_1, \dots, F_k)$ is equal to one if and only if there exists a set of strategies $\mathbf{s}_{-T} = (s_v)_{v\notin T}$ such that, after forming a solution $\mathbf{s} = (\mathbf{s}_T, \mathbf{s}_{-T})$, we have:

- *For each $v \in T$, $s_v \in BR_v(s_v, s_{-v})$*
- *For each i, $F_i(\mathbf{s}) = 1$*

Then, G is an increasing function of $F_1,\dots,F_k$.

Proof. Consider two sequences $(F_1,\dots,F_k)$ and $(F'_1,\dots,F'_k)$ with $F_i \le F'_i$ for all i, and a set of strategies $\mathbf{s}_T$. If $G(\mathbf{s}_T, F_1, \dots, F_k) = 0$, then clearly $G(\mathbf{s}_T, F'_1, \dots, F'_k) \ge 0 = G(\mathbf{s}_T, F_1, \dots, F_k)$. If $G(\mathbf{s}_T, F_1, \dots, F_k) = 1$, then there exists $\mathbf{s}_{-T}$ such that the conditions above hold for the sequence F_i. But then, since $F_i \le F'_i$, the conditions above also hold for the sequence F'_i, and $G(\mathbf{s}_T, F'_1, \dots, F'_k)$ is also equal to 1. □

Lemma 19 can be used to show that a wide family of message-passing algorithms for computing binary Nash cavity functions converges. It suffices to think of the F_i as the set of all messages, and G as the update rule for one of the messages. Since the update rule is monotonic and belong to a finite space, using a generalized version of the technique used by Ortiz and Kearns [OK03], we can show that the messages have to converge.

Nash Cavity Expansion
INPUT: A graphical game $\mathcal{G}$, two sequences $(S_1, \ldots, S_k)$ and $(T_1, \ldots, T_k)$ of subsets of V such that $T_i \cap S_i = \emptyset$ for all i, and a sequence $(K_1, \ldots, K_k)$ of subsets of $\{1, \ldots, k\}$
BEGIN
For each i, initialize Y_i^0 as a function of strategy sets $\mathbf{s}_{S_i}$ and $\mathbf{s}_{T_i}$, equal to 1 for all values of the input.
While $Y_i^r \neq Y_i^{r+1}$ for all i DO:
For all i DO:
Update $Y_i^r(\mathbf{s}_{S_i}, \mathbf{s}_{T_i})$ as follows:
IF there exists $\mathbf{s}_{-(T_i \cup S_i)} = (s_v)_{v \notin S_i \cup T_i}$ such that:

- For all $v \in T_i$, $s_v \in \mathrm{BR}_v(\mathbf{s})$
- For all $j \in K_i$, we have $Y_j^{r-1}(\mathbf{s}_{S_j}, \mathbf{s}_{T_j}) = 1$

THEN set $Y_i^r(\mathbf{s}_{S_i}, \mathbf{s}_{T_i}) = 1$
ELSE
set $Y_i^r(\mathbf{s}_{S_i}, \mathbf{s}_{T_i}) = 0$ OUTPUT The set of converged functions Y_i.

Lemma 20. *Nash Cavity Expansion always terminates, and at termination, for any NE* $\mathbf{s}^*$ *of* $\mathcal{G}$, *we have*

$$Y_i(\mathbf{s}^*_{S_i}, \mathbf{s}^*_{T_i}) = 1$$

Proof. Since Y_i^0 is identically one, we have $Y_i^1 \leq Y_i^0$ for all i. From Lemma 19, we obtain $Y_i^r \leq Y_i^{r-1}$ for all i and $r \geq 1$. Finally, since Y_i^r decreases and has a finite number of configurations, it converges. For the equality, it suffices to check by induction that all NE $\mathbf{s}^*$ are stable through the iterations. □

The second part of Lemma 20 ensures that we have effectively reduced the search space for Nash equilibria, and improves the chances of the NashSearch algorithm to obtain a valid Nash equilibrium.

The final "step" of designing a good NCE algorithm is to design the clusters and take advantage of the graphical structure to ensure that, when updating Y_i, only a few of the Y_j are checked, as well as checking over all strategies $\mathbf{s}_{-T_i \cup S_i}$ can be made on a smaller set.

Consider for instance any node v with neighbors $\{u, w_1, \ldots, w_d\}$, consider the oriented edge $u \leftarrow v$ and let $S(u \leftarrow v) = \{u\}$, $T(u \leftarrow v) = \{v\}$ and $K(u \leftarrow v) = \{v \leftarrow w_i, 1 \leq i \leq d\}$. Then it is easy to check that by the set of all $S(u \leftarrow v), T(u \leftarrow v)$, the corresponding NCE is NashProp.

By slightly augmenting the size of the clusters, one can easily obtain an algorithm which is strictly stronger than NashProp, yet whose complexity can be manageable. For instance, consider clusters of size 3: for any node v with neighbors $\{w_1, \ldots, w_d\}$, let $S(v, w_i, w_j) = \{w_i, w_j\}$, $T(v, w_i, w_j) = 1$, and $K(v, w_i, w_j)$ be the set of permutations of (v, w_i, w_j). The number of iterations of NashProp is upper bounded by $|E|1/\delta^2$, and each iteration takes $O(1/\delta^\Delta)$ steps, so that the overall compexity of NashProp is $O(|E|1/\delta^{\Delta+2})$. In constrast, the 3-cluster algorithm takes at most $|E|\Delta\delta^3$ iterations, and each iteration is $O(1/\delta^{2\Delta})$, so that the overall complexity is $O(|E|\Delta 1/\delta^{2\Delta+3})$.

5.6 Correlation decay and local Nash equilibrium

In this section, we develop a correlation decay analysis for graphical games, in the restrictive framework of directed tree games. Nevertheless, there are many reasons to believe the technique we develop for directed tree games can be extended to general tree games, and perhaps, to a lesser extent, to general games on arbitrary networks. We will assume for simplicity that $|\chi| = 2$. All results can be extended to the case $|\chi| \geq 2$, with more complex computations of the coupling and correlation constants.

Consider an arbitrary decomposable, directed tree game $\mathcal{G} = (V, E, o, \chi, \Phi)$. Recall that

for any $u \in V$ with neighbors $\{v_1, \ldots, v_d\}$, the utility function of u can be decomposed as:

$$\phi_u(x_u, x_{v_1}, \ldots, x_{v_d}) = \psi_u(x_u) + \sum_{v \in \mathcal{N}(u)} \phi_{u \leftarrow v}(x_u, x_v)$$

Recall that for a directed tree game with random costs with a jointly continuous distribution, there exists a unique pure Nash equilibrium. For any directed tree game $\mathcal{H}$ and node v, we will denote $Z_{\mathcal{H},v}$ the action of v in the PNE of $\mathcal{H}$. (We use the same notation as a Nash cavity function since the Nash cavity function $Z^p_{\mathcal{H},v}(x)$ will be nonnegative for a unique x. As such, there is a clear bijection between a Nash cavity function and an optimal decision in the PNE). Our probabilistic model is as follows:

Assumption 3. *There exist two nonnegative real numbers I_1 and I_2, and distributions F_ψ and F_ϕ such that the following two assumptions hold:*

- *For all u, $\psi_u(0) = I_1 \psi'_u(0)$ and $\psi_u(1) = I_1\, \psi'_u(1)$, where the set $(\psi'_u(0), \psi'_u(1))_{u \in V}$ is a set of i.i.d. random variables with common distribution F_ψ. We also suppose that F_ψ has a bounded density, and denote α_ψ the bound on the density.*

- *For all $(u, v) \in E$, and $x_u, x_v \in \chi$, $\phi_{u \leftarrow v}(x_u, x_v) = I_2\, \phi'_{u \leftarrow v}(x_u, x_v)$, where the set $(\phi'_{u \leftarrow v}(x_u, x_v))_{(u,v) \in E, (x_u, x_v) \in \chi^2}$ is a set of i.i.d. random variables with common distribution F_ϕ.*

We denote $\alpha_\phi = \mathbb{E}[\max(\phi'_{u \leftarrow v_i}(0,0) - \phi'_{u \leftarrow v_i}(1,0), \phi'_{u \leftarrow v_i}(0,1) - \phi'_{u \leftarrow v_i}(1,1)) - \min(\phi'_{u \leftarrow v_i}(0,0) - \phi'_{u \leftarrow v_i}(1,0), \phi'_{u \leftarrow v_i}(0,1) - \phi'_{u \leftarrow v_i}(1,1))] \geq 0$.

For any $u \in V$, let $\mathcal{G}_u$ (resp. $\mathcal{G}^r_u$) denote the game induced by the subtree rooted at node u (resp. the game induced by the subtree rooted at u with depth at most r), both $\mathcal{G}_u$ and $\mathcal{G}^r_u$ are decomposable directed tree games. Let Z^r_u be the optimal decision of u in $\mathcal{G}^r_u$, and Z_u be the optimal decision of u in $\mathcal{G}$ (which is also the optimal decision of u in $\mathcal{G}_u$, by Property 1). Our objective is to identify sufficient conditions pertaining to F_ψ, F_ϕ, and the maximum degree Δ, to guarantee the following correlation decay condition:

Definition 3. *For a nonnegative function $\rho(r)$ which decreases and converges to 0, we say that a directed tree game $\mathcal{G}$ exhibits the correlation decay property with rate ρ if*

$$\forall u, r \geq 0, \ \mathbb{P}(Z^r_u \neq Z_u) \leq \rho(r) \tag{5.9}$$

Correlation decay is said to be exponential if $\rho(r)$ is of the form $K\alpha^r$, with $K > 0$ and $\alpha < 1$, where K and α do not depend on the network topology or the number of nodes, but solely on the distribution of the costs.

The condition is of interest because of the following property, which follows from a trivial application of the union bound.

Property 3. *Suppose that the correlation decay condition* (5.9) *holds. Let $x^r = (Z_u^r)_{u \in V}$. Then,*

$$\mathbb{P}(x^r \text{ is a NE}) \geq 1 - |V|\rho(r)$$

Note that Property 3 is not a statement about computation times, since it is very easy to compute Nash equilibria for directed tree games. Instead, it should be seen as a locality property of a Nash equilibrium in a random game. In other words, in a directed tree game which satisfies a correlation decay condition, the decision of an agent in the tree depends only on a local neighborhood around it — see Section 3.5 in Chapter 3 for a discussion about decentralization.

5.6.1 Results

We now give our main results.

Theorem 18. *Suppose that Assumption 3 holds, and that F_ϕ and F_ψ both are the distribution of uniform random variables over $[0, 1]$. If*

$$\Delta \frac{I_2}{I_1} < 1$$

then the exponential correlation decay property holds, and there exists a local NE for the directed tree game $\mathcal{G}$.

Theorem 19. *Suppose that Assumption 3 holds, and that F_ϕ and F_ψ both are the distribution of standard Gaussian random variables. If*

$$\frac{\Delta I_2}{\sqrt{2(I_1^2 + (\Delta - 1)I_2^2)}} < 1$$

then the exponential correlation decay property holds, and there exists a local NE for the directed tree game $\mathcal{G}$.

5.6.2 Branching argument

In this section, we consider the simplest argument to prove the existence of local Nash equilibria for decomposable graphical games on trees (this particular argument actually trivially extends to general graphical games). The conditions we obtain will serve as a benchmark for the more powerful bounds we will obtain by using a more refined correlation decay analysis. Consider a decomposable game $\mathcal{G}$ defined on a tree $\mathcal{T}$, and let u be an arbitrary node with d neighbors $v_1, \ldots, v_d$. Let β be the probability that the best reponse reponse function BR_u of node u is always a fixed action $x \in \chi$, no matter what the input strategies $(s_{v_1}, s_{v_2}, \ldots, s_{v_d})$ are.

$$\beta = \mathbb{P}\left(\exists x \in \chi,\ \forall (s_{v_1}, \ldots, s_{v_d}) \in \mathcal{S}(\chi)^d, \mathrm{BR}_u(s_{v_1}, \ldots, s_{v_d}) = x\right) \tag{5.10}$$

β is called the branching parameter of the system, and provides a simple way of proving the existence of a local Nash equilibria:

Property 4. *Assume $\Delta(1-\beta) < 1$. Then $\mathcal{G}$ exhibits correlation decay with rate $\rho(r) = (\Delta(1-\beta))^r$*

Proof. We prove the result by induction. The result is clearly true for $r = 0$, since for any agent u, $\mathbb{P}(Z_u^0 \neq Z_u) \leq 1 \leq (\Delta(1-\beta))^0$. Suppose now the result is true for a given r, and let us compute $\mathbb{P}(Z_u^{r+1} \neq Z_u)$. Let $(v_1, \ldots, v_d)$ be the children of u in Z_u.

$$\mathbb{P}(Z_u^{r+1} \neq Z_u) = \mathbb{P}(\exists i \text{ s.t. } Z_{v_i}^r \neq Z_{v_i}) \mathbb{P}(Z_u^{r+1} \neq Z_u \mid \exists i \text{ s.t. } Z_{v_i}^r \neq Z_{v_i})$$

By induction hypothesis and the union bound, $\mathbb{P}(\exists i \text{ s.t. } Z_{v_i}^r \neq Z_{v_i}) \leq \Delta \cdot ((1-\beta)\Delta)^r$. Furthermore, $\mathbb{P}(Z_u^{r+1} \neq Z_u \mid \exists i \text{ s.t. } Z_{v_i}^r \neq Z_{v_i}) = 1 - \mathbb{P}(Z_u^{r+1} = Z_u \mid \exists i \text{ s.t. } Z_{v_i}^r \neq Z_{v_i}) \leq (1-\beta)$. Combining both bounds, we obtained the desired result. □

Proposition 19. *Suppose for all u, $\psi_u(0)$ and $\psi_u(1)$ are independent and uniformly distributed over $[0, I_1]$, and for all (u, v), $\phi_{u \leftarrow v}(0,0), \phi_{u \leftarrow v}(0,1), \phi_{u \leftarrow v}(1,0)$ and $\phi_{u \leftarrow v}(1,1)$ are independent and uniformly distributed over $[0, I_2]$. Then,*

$$(1-\beta) \leq \Delta \frac{I_2}{I_1}$$

More generally, under Assumption 3, we have

$$(1-\beta) \leq \Delta \frac{I_2}{I_1} \alpha_\psi \alpha_\phi$$

Proof. First, note that we have

$$\beta = \mathbb{P}(A \cup B)$$

with

$$A = \left\{ \forall (s_{v_1}, \ldots, s_{v_d}) \in \mathcal{S}(\chi)^d, \mathrm{BR}_u(s_{v_1}, \ldots, s_{v_d}) = 1 \right\}$$
$$B = \left\{ \forall (s_{v_1}, \ldots, s_{v_d}) \in \mathcal{S}(\chi)^d, \mathrm{BR}_u(s_{v_1}, \ldots, s_{v_d}) = 0 \right\}$$

It is easy to see that

$$\forall (y_1, \ldots, y_d) \in \chi^d, \Phi_u(1, \mathbf{y}) \geq \Phi_u(0, \mathbf{y}) \tag{5.11}$$

is a necessary and sufficient condition for

$$\forall (s_{v_1}, \ldots, s_{v_d}) \in \mathcal{S}(\chi)^d, \Phi_u(1, \mathbf{s}) \geq \Phi_u(0, \mathbf{s}) \tag{5.12}$$

to hold. It is clearly necessary, and is sufficient since the payoff of a mixed strategy is a convex combination of payoffs of pure strategies. The payoffs are decomposable, and therefore

$$\Phi_u(1, \mathbf{y}) - \Phi_u(0, \mathbf{y}) = (\psi_u(1) - \psi_u(0)) - \sum_{1 \leq i \leq d} (\phi_{u \leftarrow v_i}(0, y_i) - \phi_{u \leftarrow v_i}(1, y_i)) \tag{5.13}$$

For any i, let $C_i = \max(\phi_{u \leftarrow v_i}(0,0) - \phi_{u \leftarrow v_i}(1,0), \phi_{u \leftarrow v_i}(0,1) - \phi_{u \leftarrow v_i}(1,1))$. From Equation (5.13), we obtain that (5.11) holds if and only if

$$(\psi_u(1) - \psi_u(0)) \geq \sum_{1 \leq i \leq d} C_i \tag{5.14}$$

Similarly, let $D_i = \min(\phi_{u \leftarrow v_i}(0,0) - \phi_{u \leftarrow v_i}(1,0), \phi_{u \leftarrow v_i}(0,1) - \phi_{u \leftarrow v_i}(1,1))$. Then, event

B holds if and only if

$$(\psi_u(1) - \psi_u(0)) \leq \sum_{1 \leq i \leq d} D_i \tag{5.15}$$

We finally obtain

$$1 - \beta = \mathbb{P}(A^c \cap B^c) = \mathbb{P}(\sum_{1 \leq i \leq d} D_i < \psi_u(1) - \psi_u(0) < \sum_{1 \leq i \leq d} C_i) \tag{5.16}$$

If for any x, $\psi_u(x) = I_1 \psi'_u(x)$, and the density of $\psi'_u(x)$ is upper bounded by α_ψ, then the density of $\psi_u(x)$ is upper bounded by $\frac{\alpha_\psi}{I_1}$. It follows that the density of $\psi_u(1) - \psi_u(0)$ is also upper bounded by $\frac{\alpha_\psi}{I_1}$, and we obtain

$$\begin{aligned}
1 - \beta =& \mathbb{E}\left[1_{\sum_{1 \leq i \leq d} D_i < \psi_u(1) - \psi_u(0) < \sum_{1 \leq i \leq d} C_i}\right] \\
=& \mathbb{E}\left[\mathbb{E}\left[1_{\sum_{1 \leq i \leq d} D_i < \psi_u(1) - \psi_u(0) < \sum_{1 \leq i \leq d} C_i} \mid \sum_i C_i, \sum_i D_i\right]\right] \\
=& \mathbb{E}\left[\mathbb{E}\left[\mathbb{P}\left(\sum_{1 \leq i \leq d} D_i < \psi_u(1) - \psi_u(0) < \sum_{1 \leq i \leq d} C_i\right) \middle| \sum_i C_i, \sum_i D_i\right]\right] \\
\leq& \frac{\alpha_\psi}{I_1} \mathbb{E}\left[\sum_{1 \leq i \leq d} C_i - \sum_{1 \leq i \leq d} D_i\right] \\
\leq& \frac{\alpha_\psi \alpha_\phi \Delta I_2}{I_1}
\end{aligned}$$

Note that the first inequality comes from the fact that $C_i \geq D_i$ almost surely. Finally, for uniform random variables, $\alpha_\phi \leq 1$, and simple algebra (very similar to the calculations of Section 3.4.2 in Chapter 3) shows that $\alpha_\psi = \frac{14}{15} \leq 1$, leading to the desired upper bounds.

Remark: The branching argument, through Property 19, shows that $\mathcal{G}$ exhibits correlation decay as soon as $\Delta^2 \frac{I_2}{I_1}$ is less than some constant K. This is a weaker result than the conditions of Theorems 18 and 19, showing that the branching argument alone does not suffice to show our correlation decay result. Through lengthy computations, one could in fact show that in the uniform case (and most likely for a vast number of distributions), $(1 - \beta)$ is at most $K' \sqrt{\Delta} \frac{I_2}{I_1}$ for some constant K', which shows that the best that the

branching argument could ever do is a condition of the form $\Delta^{3/2}\frac{I_2}{I_1} < 1$, still weaker than Theorems 18 and 19. This shows that the correlation decay phenomenon truly is more than a simple branching argument in a tree.

5.6.3 Dobrushin trick

In this section, we introduce a stronger technique to prove that the correlation decay property holds. It is based on a simple interpolation trick Dobrushin used in order to prove his condition for uniqueness of Gibbs distribution (c.f. [Dob68a]), and the resulting condition we obtain for correlation decay in graphical games is quite similar to a uniqueness condition for Gibbs fields.

Consider an arbitrary directed tree game $\mathcal{H}$. Recall that for any node $u \in \mathcal{H}$ with children $(v_1, v_2, \ldots, v_d)$, we have

$$Z_u = \mathrm{BR}_u(Z_{v_1}, Z_{v_2}, \ldots, Z_{v_d})$$

Also, for any $r > 0$,

$$Z_u^r = \mathrm{BR}_u(Z_{v_1}^{r-1}, Z_{v_2}^{r-1}, \ldots, Z_{v_d}^{r-1})$$

Finally, for all v, Z_v^0 is an arbitrary decision in $\{0,1\}$. Introduce $e_r = \sup_v P(Z_v \neq Z_v^r)$. Also, given distributions F_ψ and F_ϕ, for any $d \geq 0$, let $\alpha_v(d)$ denote the maximum probability that two vectors of d decisions that are identical in all but one component have a different best response.

$$\alpha_v(d) \triangleq \sup_{i, y_{-i} \in \chi^{d-1}} \mathbb{P}(\mathrm{BR}(0, y_{-i}) \neq \mathrm{BR}(1, y_{-i})) \tag{5.17}$$

Note that implicit in the definition of α is the fact that the best response function BR is sampled according to $(I_1 F_\psi, I_2 F_\phi)$ and corresponds to a node of degree d. Our main result is the following Lemma:

Lemma 21. $e_r \leq \sup_{v \in V, d \leq \Delta}(d\alpha_v(d))\, e_{r-1}$

Proof. Consider any node $v \in V$. Since $Z_v \in \{0,1\}$, note that $\mathbb{P}(Z_u \neq Z_u^r) = \mathbb{E}\,|Z_u - Z_u^r|$. For any $0 \leq i \leq d$, define the vector $\mathbf{Z_v}[i] = (Z_{v_1}[i], \ldots, Z_{v_d}[i])$, where $Z_{v_j}[i] = Z_{v_j}$ if $j \leq i$ and is equal to $Z_{v_j}^{r-1}$ otherwise. Note that $\mathbf{Z_v}[0] = (Z_{v_1}, \ldots, Z_{v_d})$ and $\mathbf{Z_v}[d] =$

$(Z^r_{v_1}, Z^r_{v_2}, \ldots, Z^r_{v_d})$. Using a telescoping sum,

$$\begin{aligned}\mathbb{P}(Z_u \neq Z^r_u) =& \mathbb{E}\,|Z_u - Z^r_u| = \mathbb{E}\,|\mathrm{BR}_u(\mathbf{Z_v}[d]) - \mathrm{BR}_u(\mathbf{Z_v}[0])| \\ =& \mathbb{E}\left|\sum_{1\leq i\leq d} (\mathrm{BR}_u(\mathbf{Z_v}[i]) - \mathrm{BR}_u(\mathbf{Z_v}[i-1]))\right| \\ \leq& \sum_{1\leq i\leq d} E\,|\mathrm{BR}_u(\mathbf{Z_v}[i]) - \mathrm{BR}_u(\mathbf{Z_v}[i-1])| \end{aligned} \tag{5.18}$$

Now, notice that $\mathbf{Z_v}[i]$ and $\mathbf{Z_v}[i-1]$ differ only on Z_{v_i}: $Z_{v_i}[i] = Z_{v_i}$, while $Z_{v_i}[i-1] = Z^{r-1}_{v_i}$. Conditioning on the event $\{Z_{v_i} = Z^{r-1}_{v_i}\}$, we obtain

$$\begin{aligned} E\,|\mathrm{BR}_u(\mathbf{Z_v}[i]) - \mathrm{BR}_u(\mathbf{Z_v}[i-1])| =& \mathbb{P}(Z_{v_i} = Z^{r-1}_{v_i})\; E\left[|\mathrm{BR}_u(\mathbf{Z_v}[i]) - \mathrm{BR}_u(\mathbf{Z_v}[i-1])| \;\middle|\; Z_{v_i} = Z^{r-1}_{v_i}\right] \\ &+\mathbb{P}(Z_{v_i} \neq Z^{r-1}_{v_i})\; E\left[|\mathrm{BR}_u(\mathbf{Z_v}[i]) - \mathrm{BR}_u(\mathbf{Z_v}[i-1])| \;\middle|\; Z_{v_i} \neq Z^{r-1}_{v_i}\right] \\ \leq& \mathbb{P}(Z_{v_i} = Z^{r-1}_{v_i})\; 0 + \mathbb{P}(Z_{v_i} \neq Z^{r-1}_{v_i})\; \alpha(d) \end{aligned}$$

by definition of α. Substituting the last inequality into (5.18), we obtain

$$\mathbb{P}(Z_u \neq Z^r_u) \leq \sum_{1\leq i\leq d} \alpha \mathbb{P}(Z_{v_i} \neq Z^{r-1}_{v_i}) \leq \alpha(d) d e_{r-1}$$

Taking the supremum over all nodes u, we obtain the desired result. □

Lemma 21 trivially implies the following

Proposition 20. *If for all $d \leq \Delta$, $\alpha(d)d < 1$, then $\mathcal{G}$ exhibits correlation decay with rate $\rho(r) = \left(\sup_{d\leq\Delta}(\alpha(d)d)\right)^r$*

All what remains to do in order to prove Theorems 18 and 19 is to compute desired upper bounds on $\alpha(d)$

Proposition 21. *Suppose that Assumption 3 holds, and that both F_ϕ and F_ψ are the distribution of uniform random variables over $[0,1]$. Then,*

$$\forall d > 0, \quad \alpha(d) \leq \frac{I_2}{I_1} \tag{5.19}$$

Suppose instead that F_ϕ and F_ψ are the distribution of standard Gaussian random variables.

Then,

$$\forall d > 0, \quad \alpha(d) \leq \frac{I_2}{\sqrt{2(I_1^2 + (d-1)I_2^2)}} \tag{5.20}$$

Proof. Since $\psi_u(1)$ and $\psi_u(0)$ are i.i.d. random variables, and so is the collection of $(\phi_{u\leftarrow v}(x_u, x_v))_{(x_u,x_v)\in\chi^2}$. Therefore, the quantity $\mathbb{P}(\mathrm{BR}(0, y_{-i}) \neq \mathrm{BR}(1, y_{-i}))$ does not depend on i or y_{-i}, and we take i to be 1 and y_{-i} to be identically zero. Letting $\mathbf{0}_{d-1}$ denote the vector composed of $d-1$ zeroes, we therefore have

$$\alpha(d) = \mathbb{P}\left(\{\mathrm{BR}(0, \mathbf{0}_{d-1}) = 0, \mathrm{BR}(1, \mathbf{0}_{d-1}) = 1\} \cup \{\mathrm{BR}(0, \mathbf{0}_{d-1}) = 1, \mathrm{BR}(1, \mathbf{0}_{d-1}) = 0\}\right)$$

The event $\{\mathrm{BR}(0, \mathbf{0}_{d-1}) = 0\}$ is equivalent to

$$\psi_u(0) + \phi_{u\leftarrow v_1}(0,0) + \sum_{2\leq i\leq d} \phi_{u\leftarrow v_i}(0,0) \geq \psi_u(1) + \phi_{u\leftarrow v_1}(1,0) + \sum_{2\leq i\leq d} \phi_{u\leftarrow v_i}(1,0) \tag{5.21}$$

On the other hand, $\{\mathrm{BR}(0, \mathbf{1}_{d-1}) = 1\}$ is equivalent to

$$\psi_u(0) + \phi_{u\leftarrow v_1}(0,1) + \sum_{2\leq i\leq d} \phi_{u\leftarrow v_i}(0,0) \leq \psi_u(1) + \phi_{u\leftarrow v_1}(1,1) + \sum_{2\leq i\leq d} \phi_{u\leftarrow v_i}(1,0) \tag{5.22}$$

Let $X = \psi_u(1) + \sum_{2\leq i\leq d} \phi_{u\leftarrow v_i}(1,0) - \psi_u(0) - \sum_{2\leq i\leq d} \phi_{u\leftarrow v_i}(0,0)$. Together, Equations (5.21) and (5.22) imply that the event $\{\mathrm{BR}(0, \mathbf{0}_{d-1}) = 0, \mathrm{BR}(1, \mathbf{0}_{d-1}) = 1\}$ is equivalent to:

$$\phi_{u\leftarrow v_1}(0,1) - \phi_{u\leftarrow v_1}(1,1) \leq X \leq \phi_{u\leftarrow v_1}(0,0) - \psi_{u\leftarrow v_1}(1,0) \tag{5.23}$$

Conversely, event $\{\mathrm{BR}(0, \mathbf{0}_{d-1}) = 1, \mathrm{BR}(1, \mathbf{0}_{d-1}) = 0\}$ is equivalent to

$$\phi_{u\leftarrow v_1}(0,0) - \phi_{u\leftarrow v_1}(1,0) \leq X \leq \phi_{u\leftarrow v_1}(0,1) - \psi_{u\leftarrow v_1}(1,1) \tag{5.24}$$

Letting $Y = \phi_{u\leftarrow v_1}(0,0) - \phi_{u\leftarrow v_1}(1,0)$ and $Z = \phi_{u\leftarrow v_1}(0,1) - \psi_{u\leftarrow v_1}(1,1)$, by combining Equations (5.23) and (5.24), we finally obtain

$$\alpha(d) = \mathbb{P}(\min(Y, Z) \leq X \leq \max(Y, Z)) \tag{5.25}$$

Let us now upper bound this probability for the case of uniform random variables. The density of X can be very crudely upper bounded by the density of $\psi_u(1) - \psi_u(0)$ (for

any independent random variables X and Y, the density of $X+Y$ is upper bounded by the minimum of the maximum density of X and the maximum density of Y), itself upper bounded by $\frac{1}{I_1}$. Therefore,

$$\begin{aligned}\alpha(d) =& \int\int d\mathbb{P}_Y d\mathbb{P}_Z \mathbb{P}(\min(y,z) \leq X \leq \max(y,z)) \\ \leq& \frac{1}{I_1} \int\int d\mathbb{P}_Y d\mathbb{P}_Z \max(y,z) - \min(y,z) \\ \leq& \frac{1}{I_1} \mathbb{E}[\max(Y,Z) - \min(Y,Z)] = \frac{I_2}{I_1}\alpha_\phi \leq \frac{I_2}{I_1}\end{aligned}$$

For normally distributed random variables, X is a zero-mean Gaussian random variable with variance $2(I_1^2 + (d-1)I_2^2)$. The density of X is therefore upper bounded by $\frac{1}{\sqrt{2(I_1^2+(d-1)I_2^2)}}$. Using the same method as above, we obtain:

$$\alpha(d) \leq \alpha_\phi \frac{I_2}{\sqrt{2(I_1^2 + (d-1)I_2^2)}}$$

and it is easy to show that for Gaussian variables (see similar computations in Chapter 3, we also have $\alpha_\phi \leq 1$, giving us the desired result. □

5.7 Conclusions

In this chapter, we switched focus from a classical optimization setting where agents of the networks are cooperating towards a common goal, to a game-theoretic setting in which our network is composed of selfish agents locally interacting with each other. Our approach is twofold. First, following the lead of Daskalakis *et al.*, we established a connection between optimization in graphical models, and computation of Nash equilibrium in graphical games. In particular, we found that TreeProp, a tree-optimal message-passing algorithm for computing Nash in graphical games, was a special case of a family of heuristics derived from the cavity expansion. We suggested a simple modification of TreeProp which ensures that a Nash equilibrium can be computed locally and without coordination between agents, and developed a new family of search heuristics. Next, we developed a notion of correlation decay for graphical games, although in the restricted setting of directed tree games, and found sufficient conditions for the correlation decay property to hold. In particular, we developed a Dobrushin-like trick to prove these conditions, which we demonstrate to be

stronger than simple coupling techniques. These findings point to several areas for future research. An important open problem is to generalize the correlation decay proof technique to general graphs, or at least non-directed tree graphs. While the Dobrushin method we introduced can be trivially applied to tree graphs, it is hard to compute the correlation coefficients $\alpha(d)$ in a non-tree setting. Furthermore, we need to extend our development and analysis of message-passing algorithms. We see at least two directions in which this is important. The first direction consists of developing algorithmic methods which would allow us to link a tree-based correlation decay methodology to general graphs. The second direction consists in identifying specific game-theoretic models for which simple algorithms such as NashProp would converge to the exact Nash cavity functions.

Chapter 6

Application of graphical models and message-passing techniques to the early diagnosis of Alzheimer's disease

6.1 Introduction

In this last chapter, we shift our focus from theoretical questions to an applied problem, and investigate applications of graphical models and message-passing algorithms to the early diagnosis of *Alzheimer's disease* (AD). By doing so, we aim to demonstrate the practical relevance of the mathematical frameworks we considered throughout this thesis. In particular, the statistical model developed in this chapter is a graphical model, and the key algorithm used to perform inference will use the Belief Propagation algorithm.

Alzheimer's disease is a neuro-degenerative disease, the most common form of dementia, the third most expensive disease and the sixth leading cause of death in the United States. It affects more than 10% of Americans over age 65, nearly 50% of people older than 85, and it is estimated that the prevalence of the disease will triple within the next 50 years [MMS, Mat04]. While no known cure exists for Alzheimer's disease, a number of medications are believed to delay the symptoms (and perhaps causes) of the disease.

The progression of the disease can be categorized in four different stages. The first

stage is known as *Mild Cognitive Impairment* (MCI), and corresponds to a variety of symptoms — most commonly amnesia — which do not significantly alter daily life. Between 6% and 25% of people affected with MCI progress to AD every year. The next stages of Alzheimer's disease (*mild and moderate Alzheimer's disease*) are characterized by increasing cognitive deficits, decreasing independence, culminating in the patient's complete dependence on caregivers and a complete deterioration of personality (*severe Alzheimer's disease*) [SYA+01].

Early diagnosis of Alzheimer's disease, and in particular diagnosis of MCI, is important for several reasons [CFC+02a, CFC+02b, SHY05, BJZGA07, BLT+08]:

- A negative diagnostic may ease anxiety over memory loss associated with aging. It also allows for early treatments of reversible conditions with similar symptoms (such as thyroidal problems, depression, and nutrition or medication problems).
- Early diagnosis of AD also allows prompt treatment of psychiatric symptoms such as depression or psychosis, and as such reduces the personal and societal costs of the disease.
- Current symptoms-delaying medications have a given time frame during which they are effective. Early diagnosis of MCI helps ensure prescription of these medications when they are most useful. As research progresses, preventive therapies may be developed. Early diagnosis raises the chance of treating the disease at a nascent stage, before the patient suffers permanent brain damage.
- Finally, positive diagnoses give the patient and his family time to inform themselves about the disease, to make life and financial decisions related to the disease, and to plan for the future needs and care of the patients. Furthermore, as institutionalization accounts for a large part of health care costs incurred because of AD, by preserving patients' independence longer and preparing families for the needs of AD patients, timely diagnosis further decreases the societal cost of the disease.

Medical diagnosis of Alzheimer's disease is hard, and symptoms are often dismissed as normal consequences of aging. Diagnosis is usually performed through a combination of extensive testing and eliminations of other possible causes. Psychological tests such as mini mental state examinations (MMSE), blood tests, neurological examination, and increasingly, imaging techniques are used to help diagnose the disease [AA03, SGR07, PBM+07].

Our approach is based on medical studies which show that many neurophysiological diseases (such as Alzheimer's disease) are often associated with abnormalities in neural synchronicity. It is indeed well known the neural activity of different parts of a healthy brain is, to some extent, synchronized. In contrast, it has frequently been reported that these diseases cause brain signals from different brain regions [Mat01, Jeo04] to become less coherent. Therefore, developing methods to reliably detect degradations in brain-signal coherence may help to diagnose such diseases.

A common type of brain activity are so-called *electroencephalograms* (EEGs); these are measurements of electrical activity produced by the brain as recorded from electrodes placed on the scalp [NS06]. In particular, we will focus on the problem of quantifying the coherence of EEG signals (*EEG synchronicity*).

In general, quantifying the statistical interdependence between time series is an important but challenging problem. Although it is relatively easy to quantify linear dependencies (through the measure of statistical correlation, for instance), the extension to non-linear dependencies is far from trivial. This is especially true in the case of EEG anomalies, since it is important to detect brain diseases as early as possible, and fluctuations in brain signal coherence are usually very weak at this stage.

Following this last hypothesis, we developed a novel similarity measure for the purpose of detecting perturbations in EEG synchronicity. We will refer to this measure as "Stochastic Event Synchrony (SES)", since it tries to capture stochastic interactions between certain events in the time series.

6.2 Basic principle

In this section, we will briefly describe the problem of quantifying similarity of synchronicity between time series or point processes, and give a high-level, qualitative description of our algorithm.

6.2.1 Measures of synchronicity

Finding good measures of similarity between data sets is a problem of tremendous practical importance. Classical mathematical notions of distance between time series often do not correspond to the qualitative separation the practitioner desires to achieve, and it is therefore often necessary to carefully design a new metric of separation which takes into

account knowledge of the problem at hand.

More specifically, being able to measure alignment or *synchronicity* (which we will loosely define as the alignment of oscillatory processes) of different time series or point processes has been found to be the mathematical problem at the center of many various practical applications, including oceanography (e.g., oceanic "normal modes" caused by convection [KC00]), seismography (e.g., free earth oscillations and earth oscillations induced by earthquakes, hurricanes, and human activity [AFR72]), biochemistry (e.g., oscillatory events in calcium imaging data are due to oscillations of intracellular calcium [VLM+07]), proteomics [LNRE05, LKBJ08], speech recognition and stereo vision [LP98, LNRE05], and lastly, our application of interest, neuroscience.

Finding quantitative measures of synchronicity in brain activity is indeed an important topic in neuroscience. For instance, it is hotly debated whether the synchronous firing of neurons plays a role in cognition [VLRM01]. The synchronous firing paradigm has also attracted substantial attention in both the experimental [ABMV93] and the theoretical neuroscience literature [ANWS03].

6.2.2 Stochastic Event Synchrony

Stochastic event synchrony is a new measure of the interdependence of generic point processes, and as such can be used to measure alignment of point processes coming from various fields. We will, however, solely consider time series that occur in the context of neuroscience, in particular, electroencephalograms (EEG).

Point process representation of EEG data

A potential problem for using SES with EEG signals is that, as we will see, SES is developed to be a measure of synchronicity of point processes, while EEG signals are continuous time series. More generally, we would like to be able to use SES to compare any collection of time series. In order to do so, we will assume that the data obtained is, potentially after transformation, sparse in some domain, and therefore well approximated by a point process representing "bursts" of activity. Sparse representation of data has attracted a lot of interest recently through the development of compressed sensing.

Returning to the field of computational neuroscience, it is in fact known that EEG signals are, after appropriate pre-processing, well approximated by point processes: the time-frequency maps (*spectrograms*) of EEG signals are indeed sparse, as shown in Fig. 6-1

(top). They contain discrete regions of strong activity, commonly referred to as *oscillatory events*, which are believed to contain much of the information encoded in brain signals. The brain, which is a network of over a hundred billion neurons, can be considered as a huge network of coupled oscillators; as a consequence, oscillations, and oscillatory events in particular, are a key concept in the analysis in brain signal measurements such as EEG.

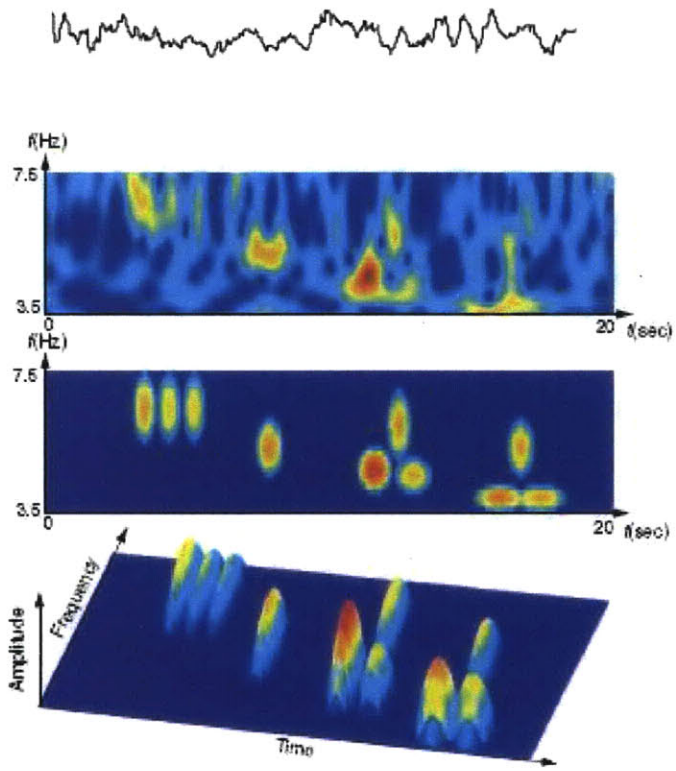


Figure 6-1: Bump modeling: the original EEG signal (top) is transformed in time-frequency domain (second from top). Then, a bump model (bottom two figures) is extracted from the resulting time-frequency map.

Following this intuitive reasoning, we pre-processed our EEG data and approximated it by a sparse representation, which we describe at a high-level as follows (see appendix C for full details on the pre-processing used):

1. *wavelet transform*

2. *normalization* of the wavelet coefficients

3. *bump modeling* of the normalized wavelet representation,

4. *aggregation* of the resulting bump models in several regions.

We used Morlet wavelets (well-known to be useful in the extraction of oscillatory patters in EEG data, see [TBBDP96]). The output of the wavelet transform is a time-frequency signal (cf. Fig. 6-1 (top)), in which clear bursts of activity can be identified.

Therefore one may consider approximating each wavelet transform by a sequence of (half-ellipsoid) basis functions ("bumps") [VMD+07]. The resulting bump models represent the most prominent oscillatory activity, and can be represented as points in a multi-dimensional space, two dimensions for the center of the bump, two for the width and height, and one for the intensity.

The basic principle behind SES

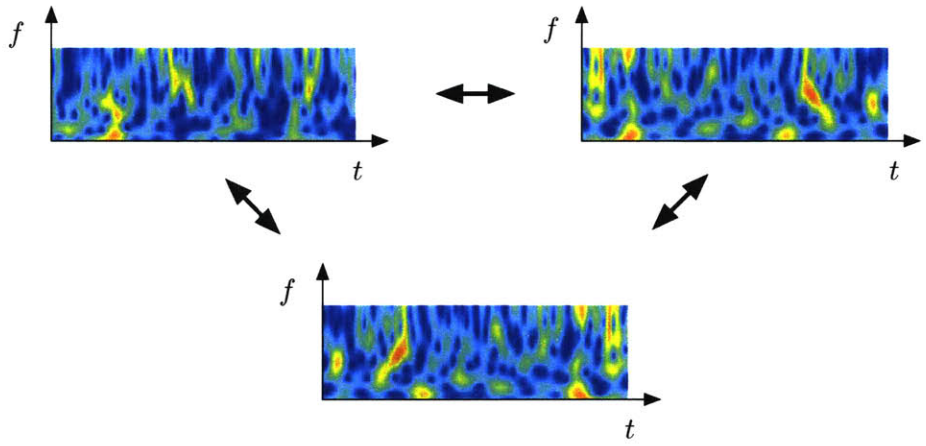
The idea underlying SES itself is very simple, and consists in matching events from one point process to events from other processes, as illustrated in Fig. 6-2. The better the matching, the more similar the original signals are. Let us reiterate that this approach differs from the classical approaches mentioned earlier in one important point: classical measures are usually directly computed from the original signals, either in time or time-frequency domain. In contrast, we determine the similarity based on point processes extracted from those signals, e.g., oscillatory events in EEG.

Suppose for now that we are only comparing two time series ($N = 2$; see Fig. 6-3). Bumps in one time-frequency map may not be present in the other map (*non-coincident* or *orphan* bumps); other bumps are present in both maps (*coincident* or *matched* bumps), but appear at slightly different positions on the maps.

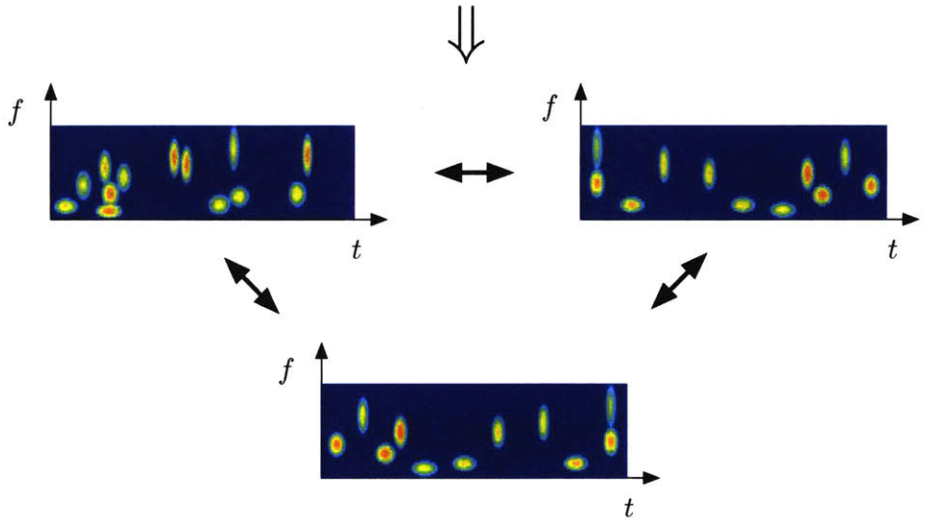
The black lines in Fig. Fig. 6-3 connect the centers of coincident bumps; hence, they show the offset in position between pairs of matched bumps. Stochastic event synchrony consists in this case of five parameters that quantify the alignment of two bump models:

- ρ: fraction of orphan bumps,

- δ_t and δ_f: average time and frequency offset between matched bumps,



(a) Time-frequency maps.



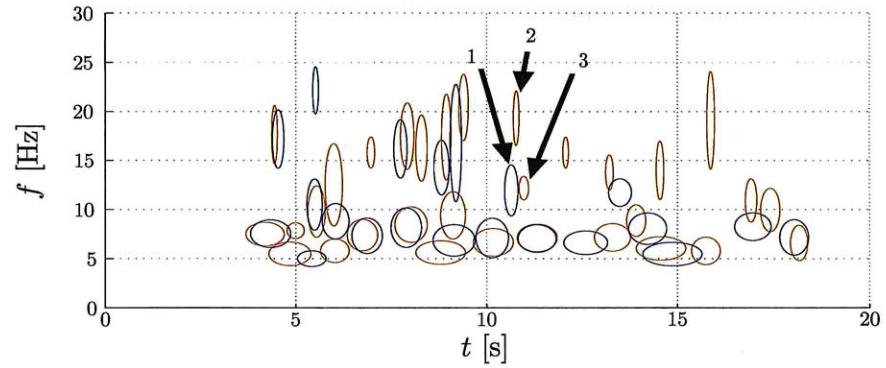
(b) Bump models.

Figure 6-2: Stochastic event synchrony of three EEG signals ($N = 3$); from their time-frequency transforms (top), one extracts two-dimensional point processes ("bump models"; bottom), which are then aligned by the proposed algorithm.

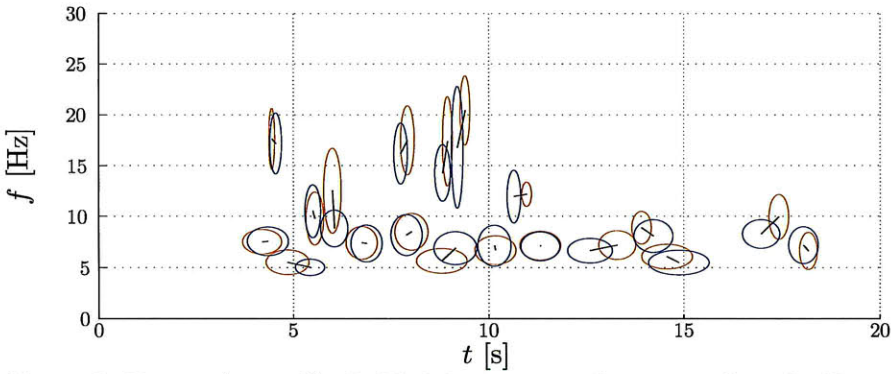
- s_t and s_f: variance of the time and frequency offset between matched bumps.

It is noteworthy that SES tolerates shifts in time and frequency between both bump models, and this is also the case for the multivariate formulation of SES, as we will explain later.

We align the two bump models and determine the above parameters by coordinate de-



(a) Bump models of two EEG channels; the arrows are described in Section 6.3.



(b) matched bumps ($\rho = 27\%$); the black lines connect the centers of matched bumps.

Figure 6-3: Bump models of two EEG signals ($N = 2$), one model is depicted in red, the other in blue; some bumps are matched (bottom), others are orphans.

scent, iterating between the following two steps (in a fashion similar to the EM algorithm):

1. For given estimates of δ_t, δ_f, s_t, and s_f, we align the two bump models (cf. Fig. 6-3 (bottom)). In section 6.3, we show that the alignment of two bump models can be recast as a maximum weighted matchings problem.

2. Given this alignment, the SES parameters are updated by maximum a posteriori (MAP) estimation.

The five SES parameters are determined from the resulting alignment by maximum a posteriori (MAP) estimation. The parameters ρ and s_t are the most relevant for the present study, since they quantify the synchronicity between bump models (and hence, the original time-frequency maps); low ρ and s_t implies that the two time-frequency maps at hand are well synchronized.

So far, we have described SES for *pairs* of signals (as in Fig. 6-3). In practice, however, one often needs to analyze multiple signals simultaneously. For example, EEG is usually recorded by an array of 21, 64, or 256 electrodes [NS06]. In principle, one may apply SES to each pair of signals, and average the SES parameters over all those pairs, resulting in a global measure for synchronicity. This approach, however, may become unwieldy as the number of pairs grows quadratically with the number of electrodes. This is one reason why we wish to consider all signals simultaneously. Secondly, multivariate SES also allows us to investigate interactions between more than two signals; for example, it enables us to distinguish events that occur in all signals from those that only occur in a subset of signals.

At a high level, multivariate SES is built upon the same idea as bivariate SES (see Fig. 6-4): events from the different signals are matched with each other. If the point processes are similar, the matched events form clearly distinguishable clusters as in Fig. 6-4, each containing at most one event from each point process. Events within each cluster are then similar and nearly simultaneous (apart from a potential shift in time and/or frequency). On the other hand, if the point processes are less similar, there may be clusters with fewer events, and the events within each cluster may be less similar and may occur at substantially different times. In summary, the similarity of point processes can be characterized by the average number of events per cluster, and the timing dispersion and similarity of the events within each cluster.

We will give a full mathematical development of bivariate and multivariate SES in the next sections.

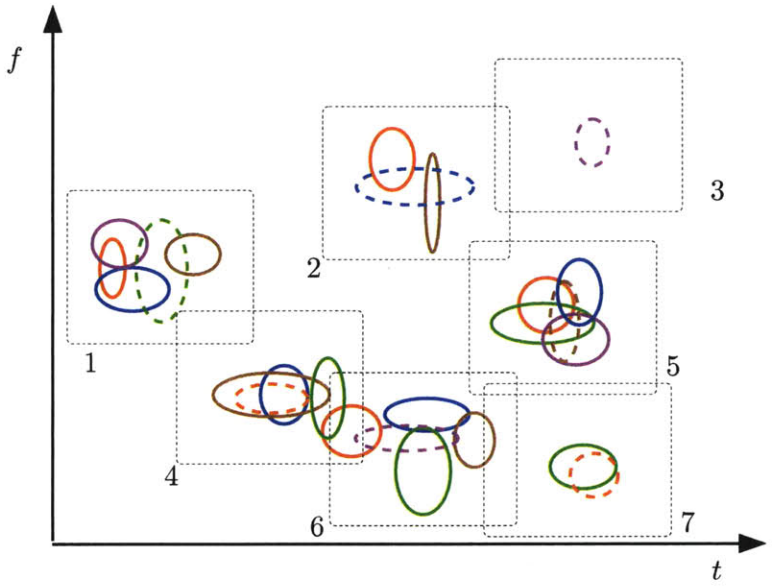


Figure 6-4: Five bump models on top of each other ($N = 5$), each color corresponds to one model; the dashed boxes indicate clusters, the dashed ellipses correspond to exemplars; cluster 1, 5 and 6 contain bumps from all 5 models, cluster 2, 4 and 7 contains bumps from 3, 4, and 2 models respectively, and cluster 3 consists of a single bump.

6.3 A class of statistical model measuring similarity between two point processes

6.3.1 Bivariate SES

In this section, we will focus on the interdependence of two multi-dimensional point processes (the special case of two one-dimensional point processes in $\mathbb{R}$ benefits from a great deal of additional structure, and is thoroughly studied in [DVWC09]). As a concrete example, we will consider multi-dimensional point processes in the time-frequency domain, in particular bump models; the proposed algorithm, however, is not restricted to that particular situation, and will be clearly generalizable to any pair of k-dimensional point processes.

Suppose that we are given a pair of continuous signals, e.g., EEG signals recorded from two different channels, each converted into a bump model. Each bump is described by five parameters: time X, frequency F, width ΔX, height ΔF, and amplitude W. The resulting bump models $Y = ((X_1, F_1, \Delta X_1, \Delta F_1, W_1), \ldots, (X_n, F_n, \Delta X_n, \Delta F_n, W_n))$ and $Y' = ((X'_1, F'_1, \Delta X'_1, \Delta F'_1, W'_1), \ldots, (X'_{n'}, F'_{n'}, \Delta X'_{n'}, \Delta F'_{n'}, W'_{n'}))$ represent the most prominent oscillatory activity in the signals at hand. In the statistical model we expose in this section, the heights, widths, and amplitude actually will actually play no role, and Y and Y' could be considered two-dimensional point process (bumps positions). However, for the application of early diagnosis of AD, we will use a slight variation of this model which does involve these variables; for the sake of consistency, we will keep widths, heights and amplitudes as data points of our processes.

The development of SES was derived from the following observation (see Fig. 6-3): bumps in one time-frequency map may not be present in the other map ("non-coincident" or orphan bumps); other bumps are present in both maps ("coincident or matched"), but appear at slightly different positions on the maps. The black lines in Fig. 6-3 connect the centers of matched bumps, and hence, visualize the offsets between pairs of matched bumps.

Statistical model

SES is intrinsically a measure of statistical similarity. We assume that the data at hand was generated from a statistical model whose parameters need to be inferred from MAP

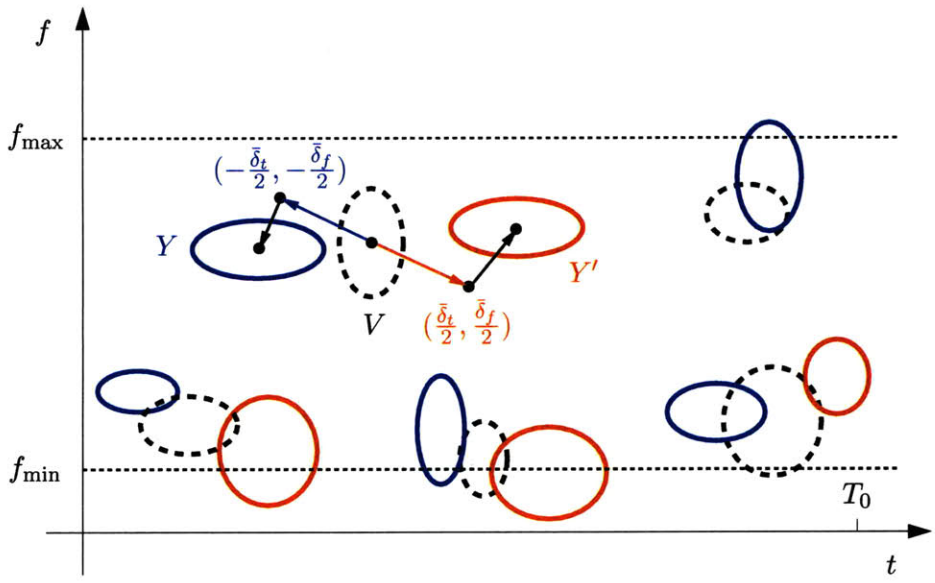


Figure 6-5: Generative model

estimation (given appropriate priors). These parameters form the set of SES metrics. Let us now describe the generative model of the observed data. We assume that there exists a hidden, unobserved point process V — the *mother* process — from which we obtain by perturbation the two observed processes Y and Y'. We make the following statistical assumptions about V, Y and Y':

The mother process V has ℓ events, where ℓ is a geometric random variable with intensity λ:

$$\mathbb{P}(\ell) = (1-\lambda)\lambda^{\ell}. \tag{6.1}$$

Let $V = ((\tilde{X}_1, \tilde{F}_1, \Delta\tilde{X}_1.\Delta\tilde{F}_1, \tilde{W}_1), \ldots, (\tilde{X}_l, \tilde{F}_l, \Delta\tilde{X}_l.\Delta\tilde{F}_l, \tilde{W}_l))$ and let us first suppose that the experiment goes over a period of time of length T_0, i.e., for any bump, $X \in [0, T_0]$. The centers $(\tilde{x}_k, \tilde{f}_k)$ are uniformly distributed over a period of time $[0, T_0]$ and a frequency band $[f_{\min}, f_{\max}]$; and as a consequence:

$$\mathbb{P}(\tilde{x}, \tilde{f}|\ell) = \frac{1}{T_0^{\ell}(f_{\max} - f_{\min})^{\ell}}. \tag{6.2}$$

The amplitudes and widths (in time and frequency) of the bumps V_k are independently and identically distributed according to distributions p_w, $p_{\Delta x}$ and $p_{\Delta f}$ respectively (once

again, let us mention that because of the perturbation model used, these distributions do not affect the resulting SES metrics).

From the mother bump process V, the bump processes Y and Y' are then generated as follows:

1. Two identical copies $\tilde{Z}$ and $\tilde{Z}'$ of bump model V are made.

2. For every $i = 1, \ldots, \ell$, the bumps $\tilde{Z}_i$ and $\tilde{Z}'_i$ are randomly removed. More precisely, each bump is deleted with probability p_d, independently of the other bumps. The results of the deletions are the two point processes Z and Z'.

3. The heights Δf_k, widths Δx_k, and amplitudes w_k of all remaining bumps in Z and Z' are randomly perturbed; more precisely, they are redrawn independently from the priors p_w, $p_{\Delta x}$ and $p_{\Delta f}$ respectively.

4. The bump sequence Y (resp. Y') is obtained by shifting the position of the bumps Z_k and Z'_k by $(-\frac{\delta_t}{2}, -\frac{\delta_f}{2})$ and $(\frac{\delta_t}{2}, \frac{\delta_f}{2})$, and by adding small random perturbations to the position of the bumps Y_k and Y'_k (cf. Fig. 6-5), modeled as zero-mean Gaussian random vectors with diagonal covariance matrix (s_t, s_f):

Joint distribution of the hidden process and observed samples

Let us detail some notations. Let $n_{\text{del}}^{\text{double}}$ be the number of double deletions, i.e., the number of bumps of V for which the copies in $\tilde{Z}$ and $\tilde{Z}'$ both were deleted. Let n_{del} (resp. n'_{del}) be the number of deleted bumps of Z (resp. Z') for which the other bump was not deleted (single deletions). By definition,

$$n_{\text{del}} + n + n_{\text{del}}^{\text{double}} = \ell, \tag{6.3}$$

and likewise:

$$n'_{\text{del}} + n' + n_{\text{del}}^{\text{double}} = \ell, \tag{6.4}$$

and the total number of deleted bumps is therefore given by:

$$n_{\text{del}}^{\text{tot}} = n_{\text{del}} + n'_{\text{del}} + 2n_{\text{del}}^{\text{double}} = 2\ell - n - n', \tag{6.5}$$

Note that the bump parameters w, Δx, Δf are generated independently for each bump, and therefore, they do not provide any information about bump matchings. As a result,

the SES inference algorithm (see Section 6.3.2) does not depend on the specific choice of the priors p_w, $p_{\Delta x}$ and $p_{\Delta f}$. Without loss of generality, we will adopt improper priors $p_w = p_{\Delta x} = p_{\Delta f} = 1$.

Since there is a total of 2ℓ bumps and $2\ell - n - n'$ deleted bumps, and since each bump is deleted with i.i.d. probability p_d, the joint probability of V, Z, Z' and ℓ is:

$$\begin{aligned} \mathbb{P}(v, z, z', l) &= \mathbb{P}(\ell)\mathbb{P}(v \mid \ell)\mathbb{P}(z, z' \mid v, \ell) && (6.6) \\ &= (1-\lambda)\lambda^{\ell} \frac{1}{T_0^{\ell}(f_{\max} - f_{\min})^{\ell}} (1-p_d)^{n+n'} p_d^{2\ell-n-n'}. && (6.7) \end{aligned}$$

By introducing parameters β and γ:

$$\beta = p_d \sqrt{\frac{\lambda}{T_0(f_{\max} - f_{\min})}} \tag{6.8}$$

$$\gamma = (1-\lambda)\Big(\frac{1-p_d}{p_d}\Big)^{n+n'}, \tag{6.9}$$

we can rewrite (6.7) as:

$$\mathbb{P}(v, z, z', l) = \gamma\, \beta^{2\ell}. \tag{6.10}$$

Finally, we use the relation $2\ell = n + n' + n_{\text{del}} + n'_{\text{del}} + 2n_{\text{del}}^{\text{double}}$ to obtain:

$$\mathbb{P}(v, z, z', l) = \gamma'\, \beta^{2n_{\text{del}}^{\text{double}}} \beta^{n_{\text{del}} + n'_{\text{del}}}, \tag{6.11}$$

with $\gamma' = \gamma\, \beta^{n+n'}$.

Before we can write down the complete model (including Y and Y'), we need to introduce some more notation. We attach to each bump Y_i a binary variable B_i, which indicates whether Y_i has a matching bump in Y'. In particular, Y_i is equal to one iff the corresponding bump in Y' was deleted. Along the same lines, we associate variables B'_i to each bump Y'_i. Furthermore, we introduce binary variables $C_{kk'}$ for any k and k': the variables $C_{kk'}$ are equal to one if bumps Y_k and $Y'_{k'}$ are matched (i.e., copies of the same bump in V), and 0 otherwise. Since each bump Y_k in Y is either unmatched (i.e., $B_k = 1$)

or corresponds to exactly one bump in Y', the following matching constraints hold:

$$\forall k, \quad \sum_{k'=1}^{n'} c_{kk'} + b_k = 1 \tag{6.12}$$

$$\forall k', \quad \sum_{k=1}^{n} c_{kk'} + b'_k = 1 \tag{6.13}$$

Note also that $\sum_k b_k = n'_{\text{del}}$ and $\sum_{k'} b'_{k'} = n_{\text{del}}$. Therefore, the exponent $n'_{\text{del}} + n_{\text{del}}$ of β in Equation (6.11) can be written in terms of B and B':

$$n'_{\text{del}} + n_{\text{del}} = \sum_k b_k + \sum_{k'} b'_{k'}. \tag{6.14}$$

Finally, let i_k be the index of the bump in V that generated Y_k, and similarly, $i'_{k'}$ stands for the index of the bump in V that generated $Y'_{k'}$. Note that if $C_{kk'} = 1$, we have $i_k = i'_{k'}$. Finally, let $\theta = (\delta_t, s_t, \delta_f, s_f)$, and let $P(\theta)$ denote the prior on θ, on which we will later elaborate. In this representation, the joint probability of the entire set of random variables $(\ell, V, Y, Y', C, C', B, B')$ of the global statistical model is given by:

$$\begin{aligned}
\mathbb{P}(\ell, v, y, y', c, c', b, b', \theta) \propto\ & P(\theta)\, \beta^{\sum_k b_k + \sum_{k'} b'_{k'}} \\
& \cdot \prod_{k=1}^{n} \prod_{k'=1}^{n'} \left(p\Big(x_k - \tilde{x}_{i_k}; -\delta_t/2, s_t/2\Big)\, n\Big(x'_{k'} - \tilde{x}_{i_k}; \delta_t/2, s_t/2\Big) \right)^{c_{kk'}} \\
& \cdot \prod_{k=1}^{n} \prod_{k'=1}^{n'} \left(p\Big(f_k - \tilde{f}_{i_k}; -\delta_f/2, s_f/2\Big)\, p\Big(f'_{k'} - \tilde{f}_{i_k}; \delta_f/2, s_f/2\Big) \right)^{c_{kk'}} \\
& \cdot \prod_{k} \left(p\Big(x_k - \tilde{x}_{i_k}; -\delta_t/2, s_t/2\Big)\, p\Big(f_k - \tilde{f}_{i_k}; -\delta_f/2, s_f/2\Big) \right)^{b_k} \\
& \cdot \prod_{k'} \left(p\Big(x'_{k'} - \tilde{x}'_{i'_{k'}}; -\delta_t/2, s_t/2\Big)\, p\Big(f'_{k'} - \tilde{f}'_{i_k}; -\delta_f/2, s_f/2\Big) \right)^{b'_{k'}} \\
& \cdot \prod_{k'=1}^{n'} \left(\delta[b'_{k'} + \sum_{k=1}^{n} c_{kk'} - 1]\right) \prod_{k=1}^{n} \left(\delta[b_k + \sum_{k'=1}^{n'} c_{kk'} - 1]\right),
\end{aligned} \tag{6.15}$$

where $p(x, m, s)$ represents the density of a normal variable with mean m and standard deviation s at point x, and δ the dirac function equal to 1 if its argument is zero, and 0

otherwise. The variables $C_{kk'}$, B_k, and $B_{k'}$ are binary. The first four factors in (6.15) correspond to bump pairs $(Y_k, Y'_{k'})$ (with $C_{kk'} = 1$); the next four factors correspond to orphan bumps ($B_k = 1$ and $B'_{k'} = 1$). The last two factors in (6.15) encode the constraints (6.12).

Our objective is to estimate the parameters θ and alignment variables C and C', since those quantities contain information about the similarity of Y and Y'. We integrate over the structural variables V, B and B', and after some straightforward algebraic manipulations, we eventually obtain:

$$\mathbb{P}p(y, y', c, \theta) \propto \prod_{k=1}^{n} \prod_{k'=1}^{n'} \left(p\Big(x'_{k'} - x_k; \delta_t, s_t\Big)\, p\Big(f'_{k'} - f_k; \delta_f, s_f\Big) \beta^{-2} \right)^{c_{kk'}} \cdot P(\theta) I(c), \tag{6.16}$$

where $I(c)$ is equal to 1 if and only if for all k, $\sum_{k'} c_{kk'} \in \{0, 1\}$ and for all k', $\sum_k c_{kk'} \in \{0, 1\}$. The factor $I(c)$ encodes the partial matching constraints (6.12).

We now comment on the priors of the parameters $\theta = (\delta_t, s_t, \delta_f, s_f)$. Since we usually we do not need to encode prior information about δ_t and δ_f, we may choose improper priors $p(\delta_t) = 1 = p(\delta_f)$. On the other hand, one may have prior knowledge about s_t and s_f. For example, in the case of spontaneous EEG (see Section 6.5), we a priori expect s_t to be larger than s_f: we expect bumps to appear at about the same frequency in both time-frequency maps, but there might be a delay of up to about 500ms between them. Indeed, frequency shifts can only be caused by non-linear transformations, which are hard to justify from a physiological perspective; on the other hand, signals may propagate over large distances in the brain, and therefore, time shifts arises quite naturally. For example, bump nr. 1 in Fig. 6-3(a) ($t = 10.7$s) should then be paired with bump nr. 3 ($t = 10.9$s) and not with nr. 2 ($t = 10.8$s), since the former is much closer in frequency than the latter. One may encode such prior information by means of conjugate priors for s_t and s_f, i.e., scaled inverse chi-square distributions:

$$p(s_t) = \frac{(s_{0,t}\nu_t/2)^{\nu_t/2}}{\Gamma(\nu_t/2)} \frac{e^{-\nu_t s_{0,t}/2s_t}}{s_t^{1+\nu_t/2}} \tag{6.17}$$

$$p(s_f) = \frac{(s_{0,f}\nu_f/2)^{\nu_f/2}}{\Gamma(\nu_f/2)} \frac{e^{-\nu_f s_{0,f}/2s_f}}{s_f^{1+\nu_f/2}}, \tag{6.18}$$

where ν_t and ν_f are the degrees of freedom and $\Gamma(x)$ is the Gamma function. In the

example of spontaneous EEG, the widths $s_{0,t}$ and $s_{0,f}$ are chosen such that $s_{0,t} > s_{0,f}$, since s_f is expected to be smaller than s_t.

6.3.2 Statistical inference for bivariate SES

As previously described, our two point processes Y and Y' will intuitively be considered as synchronous when they are identical with a few exceptions: (i) Small time and frequency shifts δ_t and δ_f; (ii) small deviations in the event occurence times ("event timing jitter") and in the frequencies; (iii) a few event insertions and/or deletions. More precisely, the event timing jitter should be significantly smaller than the average inter-event time, and the number of deletions and insertions should only comprise a small fraction of the total number of events.

Armed with the stochastic model of section 6.3.1, we are now in a position to rigorously define Stochastic Event Synchrony (SES): given two point processes y, and y', SES is defined as the triplet $(\delta_t,\ s_t,\ \rho)$, where:

$$\rho \triangleq \frac{n_{\text{del}} + n'_{\text{del}}}{n + n'} = \frac{\sum_{k=1}^{n} \hat{b}_k + \sum_{k=1}^{n'} \hat{b}'_{k'}}{n + n'}. \tag{6.19}$$

The estimates $(\hat{c}, \hat{\theta})$ are obtained by maximum a posteriori (MAP) estimation:

$$(\hat{c}, \hat{\theta}) = \text{argmax}_{c,\theta}\, \mathbb{P}(c, \theta | y, y'). \tag{6.20}$$

Since for given y and y', the factor $p(c, \theta|y, y')$ is proportional to $p(y, y', c, \theta)$ (cf. (6.16)), we can rewrite (6.20) as:

$$(\hat{c}, \hat{\theta}) = \text{argmax}_{c,\theta}\, \mathbb{P}(y, y', c, \theta). \tag{6.21}$$

The MAP estimate (6.21) is hard to compute, and we obtain it by coordinate descent: first, the parameters θ are initialized (e.g., $\hat{\delta}_t^{(0)} = 0 = \delta_f^{(0)}$, $\hat{s}_t^{(0)} = s_{0,t}$, and $\hat{s}_f^{(0)} = s_{0,f}$), then one alternates the following two update rules until convergence (or until the available time has elapsed):

$$\hat{c}^{(i+1)} = \text{argmax}_c\, \mathbb{P}(y, y', c, \hat{\theta}^{(i)}) \tag{6.22}$$

$$\hat{\theta}^{(i+1)} = \text{argmax}_\theta\, \mathbb{P}(y, y', \hat{c}^{(i+1)}, \theta). \tag{6.23}$$

Update of the continuous parameters

The estimate $\hat{\theta}^{(i+1)}$ (6.23) is available in closed-form; indeed, it is easily verified that the point estimates $\hat{\delta}_t^{(i+1)}$ and $\hat{\delta}_f^{(i+1)}$ are the (sample) mean of the timing and frequency offset respectively, computed over all pairs of matched events:

$$\hat{\delta}_t^{(i+1)} \triangleq \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} (\hat{x}_k'^{(i+1)} - \hat{x}_k^{(i+1)}) \tag{6.24}$$

$$\hat{\delta}_f^{(i+1)} \triangleq \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} (\hat{f}_k'^{(i+1)} - \hat{f}_k^{(i+1)}) \tag{6.25}$$

where $n^{(i+1)}$ is the number of coincident bump pairs in iteration $i+1$, and where we used the shorthand notation $\hat{x}_k^{(i+1)}$ and $\hat{x'}_k^{(i+1)}$ to represent the times of k^{th} matched pair during the i^{th} iteration of the algorithm, and likewise for $\hat{f}_k^{(i+1)}$ and $\hat{f'}_k^{(i+1)}$. Using the conjugate priors, the estimates $\hat{s}_t^{(i+1)}$ and $\hat{s}_f^{(i+1)}$ are obtained as:

$$\hat{s}_t^{(i+1)} = \frac{\nu_t s_{0,t} + n^{(i+1)} \hat{s}_{t,\text{sample}}^{(i+1)}}{\nu_t + n^{(i+1)} + 2} \tag{6.26}$$

$$\hat{s}_f^{(i+1)} = \frac{\nu_f s_{0,f} + n^{(i+1)} \hat{s}_{f,\text{sample}}^{(i+1)}}{\nu_f + n^{(i+1)} + 2}, \tag{6.27}$$

where ν_t, ν_f, $s_{0,t}$ and $s_{0,f}$ are the parameters of the conjugate priors (6.17) and (6.18), and $s_{t,\text{sample}}$ and $s_{f,\text{sample}}$ are the (sample) variance of the timing and frequency offset respectively, computed over all pairs of coincident events.

Update of the discrete parameters

We now address the update (6.22), i.e., finding the optimal bivariate alignment C for *given* values $\hat{\theta}^{(i)}$ of the parameters θ. In the following, we will show that it is equivalent to a standard problem in combinatorial optimization, i.e., max-weight bipartite matching (see, e.g., [Ger95, Pul, BSS08, BBCZ08, HJ07, San07, San07]).

First, note that the maximization (6.22) is equivalent to:

$$\hat{c}^{(i+1)} = \text{argmax}_c \log p(y, y', c, \hat{\theta}^{(i)}). \tag{6.28}$$

Using (6.16), we can rewrite (6.28) as:

$$\hat{c}^{(i+1)} = \text{argmax}_c \sum_{kk'} w_{kk'}\, c_{kk'} + \log I(c), \tag{6.29}$$

with

$$\begin{aligned} w_{kk'} = &- \frac{\left(x'_{k'} - x_k - \hat{\delta}_t^{(i)}\right)^2}{2s_t} - \frac{\left(f'_{k'} - f_k - \hat{\delta}_f^{(i)}\right)^2}{2s_f} + 2\log\beta \\ &- 1/2\log 2\pi s_t - 1/2\log 2\pi s_f, \end{aligned} \tag{6.30}$$

where the weights $w_{kk'}$ can be positive or negative. Bump pairs $(Y_k, Y'_{k'})$ with large weights $w_{kk'}$ are likely to be matched to each other. The closer the bumps $(Y_k, Y'_{k'})$ on the time-frequency plan, the larger their weight $w_{kk'}$. From the definition of β (6.8), we can also see that the weights increase as the deletion probability p_d decreases. Indeed, if p_d is large, a significant number of bumps from Y cannot be matched with bumps from Y' and vice versa. In addition, the weights $w_{kk'}$ are large if the concentration of bumps on the time-frequency plane, i.e., the ratio λ/S with $S = T_0(f_{\max} - f_{\min})$, is small. Indeed, if there are few bumps in each model (per square unit) and a bump Y_k of Y happens to be close to a bump $Y'_{k'}$ of Y', they are most likely a matched bump pair.

The term $\log(I(c))$, on the other hand, is equal to 0 if and only if for all k and k', both $\sum_{k'} C_{kk'}$ and $\sum_k C_{kk'}$ are either zero or one (binary), and is equal to $-\infty$ otherwise (infeasible solution). Therefore, the maximization problem is equivalent to finding:

$$\begin{aligned} &\max \sum_{k,k'} w_{kk'} C_{kk'} \\ &\text{s.t.} \\ &\forall k, \sum_{k'} C_{kk'} \in \{0,1\} \\ &\forall k', \sum_{k} C_{kk'} \in \{0,1\}. \end{aligned} \tag{6.31}$$

This is exactly the formulation of the imperfect, bipartite, maximum-weight matching (IBMWM) problem. Note that if $w_{kk'} < 0$, since it is an imperfect matching problem, we know that $c_{kk'} = 0$ in the optimal solution, and the corresponding variable can be removed. In practice, this will make the matching problem very sparse, since for a given

bump, only neighboring bumps (in the time-frequency plane) will have positive weight and be considered for matching. As a result, this observation naturally transforms our general optimization problem into an optimization problem in a graphical model, where two bumps k and k' are neighbors if and only if they are from a different process and $w_{kk'} > 0$.

The IBMWM problem can be solved (in polynomial time) by at least three different methods:

- by the Edmond-Karp [Edm69] or auction algorithm [TB89],
- by using the tight LP relaxation to the integer programming formulation of bipartite max-weight matching [Ger95, Pul],
- by applying the max-product algorithm [BSS08, BBCZ08, HJ07, San07].

The Edmond-Karp [Edm69] and auction algorithm [TB89] both result in the optimum solution of (6.29). The same holds for the linear programming relaxation approach and the max-product algorithm as long as the optimum solution is unique. If the latter is not unique, the linear programming relaxation method may result in non-integer solutions and the max-product algorithm may not converge, as shown in [San07]. Note that in many practical problems, the optimum matching (6.29) is unique with probability one. This is in particular the case for the bump models described above. Each method was tested, and the message-passing paradigm was found to be very efficient for several reasons:

- It does not require a complex, potentially commercial LP solver.
- It is a very simple iterative algorithm, and takes natural advantage of the sparsity of the underlying bipartite graph. For example, note that if $w_{kk'} < 0$, the edge between bumps Y_k and Y'_k can be removed.
- It is very modular: simple modifications of the model often translate into simple modifications of the iterative equations. Convergence and optimality may not be guaranteed anymore, but in practice the algorithm has very good performance, whereas specialized combinatorial algorithms may fail.

Finally, we note that the SES inference algorithm is guaranteed to converge. Indeed, this algorithm is an instance of coordinate descent, which is guaranteed to converge if the iterated conditional maximizations have unique solutions. This holds for the SES inference algorithm: the conditional maximization (6.23) has unique solutions (cf. (6.24)), and the

same also holds for (6.22) in most practical applications, in particular, the application considered in Section 6.5.

We close this section by mentioning that not all statistical assumptions made were necessary. As a matter of fact, the only assumptions necessary for tractability are the geometric number of mother bumps, and the uniform (or at least piecewise uniform) location of these bumps. All other assumptions can be generalized; in particular, the assumption of Gaussian deviations, while convenient and plausible, is not required.

6.4 Comparing multiple point processes at the same time

We now consider the extension of SES from pairs of point processes to collections of point processes. We will do so in a fully general framework. Consider $N > 2$ signals $S_1, \ldots, S_N$ from which we extract point processes $Y_1, \ldots, Y_N$ by some method. Each point process Y_i is a list of n_i points (or events) in a given multi-dimensional set $\mathcal{S} \subseteq \mathbb{R}^M$, i.e., $Y_i = \{Y_{i,1}, Y_{i,2}, \ldots, Y_{i,n_i}\}$ with $Y_{i,k} \in \mathcal{S}$ for $k = 1, \ldots, n_i$ and $i = 1 \ldots N$. By analogy with the previous section, we will call events "bumps". Intuitively speaking, N bump models Y_i may be considered well-synchronized if bumps appear in all models (or almost all) simultaneously, potentially with some small offset in time and frequency. In other words, if one overlays N bump models (cf. Fig. 6-4 with $N = 5$), bumps naturally appear in clusters that contain precisely one bump from all (or almost all) bump models. In the example of Fig. 6-4, clusters 1, 5 and 6 contain bumps from all five models Y_i, clusters 2, 4 and 7 contains bumps from 3, 4, and 2 models respectively, and cluster 3 consists of a single bump.

6.4.1 Principle of multivariate SES

This intuitive concept of similarity may be translated into a generative stochastic model in the same spirit as section 6.3. In that model, the N point processes Y_i are treated as independent noisy observations of a hidden "mother" process $\tilde{X}$. An observed sequence Y_i is obtained from $\tilde{X}$ by the following three-step procedure:

1. COPY: generate a copy of the mother bump model $\tilde{X}$,

2. DELETION: delete some of the copied mother bumps,

3. PERTURBATION: slightly alter the position and shape of the remaining mother bump copies, resulting in the bump model Y_i.

As a result, each sequence Y_i consists of "noisy" copies of a non-empty subset of mother bumps. The point processes Y_i may be considered well-synchronized if there are only a few deletions (cf. Step 2) and if the bumps of Y_i are "close" to the corresponding mother bumps (cf. Step 3). One way to determine the synchronicity of given point processes Y_i is to first reconstruct the hidden mother process $\tilde{X}$, and to next determine the number of deletions and the average distance between the point processes Y_i and the mother process $\tilde{X}$. When comparing more than two point processes, inferring the mother process is a high-dimensional estimation problem, as the underlying probability distribution typically has a large number of local extrema. Therefore, we will use an alternative procedure: we will assume that each cluster contains one *identical* copy of a mother bump; the other bumps in that cluster are *noisy* copies of that mother bump. The identical copy, referred to as *exemplar*, plays the role of "center" or "representative" of each cluster (see Fig. 6-4). We will assume, without loss of generality, that there is one exemplar for each mother bump. Note that under this assumption, the mother process $\tilde{X}$ is equal to the list of all exemplars.

The exemplar-based formulation amounts to the following inference problem: given the point processes Y_i (with $i = 1, 2, \ldots, N$), we need to identify the bumps that are exemplars and those that are noisy copies of some exemplar, with the constraint that an exemplar and its noisy copies all stem from different point processes. Obviously, this inference problem also has potentially many locally optimal solutions. However, in contrast to the original (continuous) inference problem, we can in practice find the global optimum by message-passing and integer programming (see sections 6.4.3). This model choice is related to exemplar-based approaches for clustering such as affinity propagation [FD07, FD06] and the convex clustering algorithm of [LG07]. As we will see, the exemplar-based formulation will allows us to extend the bivariate similarity to multivariate similarity measures.

6.4.2 Stochastic model for multivariate SES

We now describe the underlying stochastic model in more detail. The mother process $\tilde{X} = \{\tilde{X}_1, \ldots, \tilde{X}_M\}$, which is the source of all points in $Y_1, Y_2, \ldots Y_N$, is modeled as follows:

- The number M of points in $\tilde{X}$ is geometrically distributed with parameter $\lambda \operatorname{vol}(S)$:

$$\mathbb{P}(M) = (1 - \lambda \operatorname{vol}(S))\big(\lambda \operatorname{vol}(S)\big)^M, \tag{6.32}$$

where $\mathrm{vol}(S)$ is the multi-dimensional volume of set S.

- Each point $\tilde{x}_m$ for $m = 1, \ldots, M$ is uniformly distributed in S:

$$\mathbb{P}(\tilde{x}|M) = \mathrm{vol}(S)^{-M}. \tag{6.33}$$

With those two choices, the prior of the mother process $\tilde{X}$ equals:

$$\mathbb{P}(\tilde{x}, M) = \mathbb{P}(M)\mathbb{P}(\tilde{x}|M) = (1 - \lambda\,\mathrm{vol}(S))\lambda^M. \tag{6.34}$$

For convenience we will in the following use the short-hand notation $p(\tilde{x})$ for $p(\tilde{x}, M)$, i.e., we will not explicitly mention the dependency on M.

From the mother process $\tilde{X}$, the point processes Y_i for $i = 1, \ldots, N$ are generated according to the following steps:

- For each event $\tilde{X}_m$ in the mother process $\tilde{X}$, one of the point processes Y_i with $i \in \{1, \ldots, N\}$ is chosen at random, denoted by $Y_{i(m)}$, and a copy of mother event $\tilde{X}_m$ is created in $Y_{i(m)}$; this identical copy is referred to as "exemplar". For convenience, we will adopt a uniform prior $\mathbb{P}\big(i(m) = i\big) = 1/N$ for $i = 1, \ldots, N$. Next, for each event $\tilde{X}_m$ in the mother process $\tilde{X}$ (with $m = 1, \ldots, M$), a "noisy" copy may be created in the point processes Y_j with $j \neq i(m)$, at most *one* copy per point process Y_j; the latter restriction ensures that all events in a cluster come from different point processes (cf. Fig. 6-4). The noisy copies are modeled as follows.

- The number C_m of copies is modeled by a prior $\mathbb{P}(c_m|\theta_c)$, parameterized by θ_c, which in turn has a prior $p(\theta_c)$. As a priori for C_m, we take a binomial distribution with $N - 1$ trials and probability of success p_s, and adopt the beta distribution $\mathrm{B}(\kappa, \lambda)$ as a conjugate prior for p_s. Note that a binomial prior $\mathrm{Bi}(p_s)$ for C_m is equivalent to deleting copies of the mother events *independently* with probability $1 - p_s$ (cf. DELETION step). In appendix D, we show how to extend the binomial distribution to a multinomial distribution $\mathrm{Mult}(\gamma)$ with parameter γ and conjugate Dirichlet $\mathrm{Di}(\zeta)$ respectively.

- Conditional on the number C_m of copies, the copies are attributed uniformly at random to other signals Y_j, with the constraints of at most one copy per signal and $j \neq i(m)$; since there are $\binom{N-1}{c_m}$ possible attributions $\mathcal{A}_m \subseteq \{1, \ldots, i(m) -$

$1, i(m)+1, \ldots, N\}$ with $|\mathcal{A}_m| = c_m$, the probability mass of an attribution $\mathcal{A}_m$ is $p(\mathcal{A}_m|c_m) = \binom{N-1}{c_m}^{-1}$.

- The process of generating a noisy copy $Y_{i,r}$ from a mother bump $\tilde{X}_m$ is described by a conditional distribution $p_x(x_{i,r}|\tilde{x}_m; \theta_x^i)$, parameterized by some vector θ_x^i that may differ for each point process Y_i.

  In the case of bump models (cf. Fig. 6-4), a simple mechanism to generate copies is to slightly shift the mother bump center while the other mother bump parameters (width, height, and amplitude) are drawn from some prior distribution, independently for each copy. The latter four bump parameters could be taken into account in a less trivial way, but we omit such extensions here, as they are not required for the application at hand. The center offset may be modeled as a bivariate Gaussian random variable with mean vector $(\delta_{t,i}, \delta_{f,i})$ and diagonal non-isotropic covariance matrix $V_i = \mathrm{diag}(s_{t,i}, s_{f,i})$, and hence, $\theta_x^i = (\delta_{t,i}, \delta_{f,i}, s_{t,i}, s_{f,i})$. For simplicity, we will assume that $s_{t,i} = s_t$ and $s_{f,i} = s_f$ for all i. We adopt the improper priors $p(\delta_{t,i}) = 1 = p(\delta_{f,i})$ for $\delta_{t,i}$ and $\delta_{f,i}$ respectively, and conjugate priors for s_t and s_f, i.e., scaled inverse chi-square distributions:

$$p(s_t) = \frac{(s_{0,t}\nu_t/2)^{\nu_t/2}}{\Gamma(\nu_t/2)} \frac{e^{-\nu_t s_{0,t}/2s_t}}{s_t^{1+\nu_t/2}} \tag{6.35}$$

$$p(s_f) = \frac{(s_{0,f}\nu_f/2)^{\nu_f/2}}{\Gamma(\nu_f/2)} \frac{e^{-\nu_f s_{0,f}/2s_f}}{s_f^{1+\nu_f/2}}, \tag{6.36}$$

  where ν_t and ν_f are the degrees of freedom, and $s_{0,t}$ and $s_{0,f}$ are the width of the scaled inverse chi-square distributions, and $\Gamma(x)$ is the Gamma function.

Joint distribution

For later convenience, we will introduce some more notation. The exemplar associated to the mother event $\tilde{X}_m$ is denoted by $Y_{i(m),k(m)}$, it is the event $k(m)$ in point process $Y_{i(m)}$. We denote the set of pairs $(i(m), k(m))$ by $\mathcal{I}^{\mathrm{ex}}$. A noisy copy of $\tilde{X}_m$ is denoted by $Y_{j(m),\ell(m)}$, it is the event $\ell(m)$ in point process $Y_{j(m)}$ with $j(m) \in \mathcal{A}_m$. We denote the set of all pairs $(j(m), \ell(m))$ associated to $\tilde{X}_m$ by $\mathcal{I}_m^{\mathrm{copy}}$, and furthermore define $\mathcal{I}^{\mathrm{copy}} \triangleq \mathcal{I}_1^{\mathrm{copy}} \cup \cdots \cup \mathcal{I}_M^{\mathrm{copy}}$

and $\mathcal{I} = \mathcal{I}^{\text{ex}} \cup \mathcal{I}^{\text{copy}}$. In this notation, the overall probabilistic model may be written as:

$$\mathbb{P}(\tilde{X}, X, \mathcal{I}, \theta) = p(\theta_c)p(\theta_x)(1 - \lambda \operatorname{vol}(S))\lambda^M N^{-M} \prod_{m=1}^{M} \delta\big(x_{i(m),k(m)} - \tilde{x}_m\big)$$
$$\cdot\, p(c_m|\theta_c)\binom{N-1}{c_m}^{-1} \prod_{(i,j)\in\mathcal{I}_m^{\text{copy}}} p_x(x_{i,j}|\tilde{x}_m, \theta_x). \tag{6.37}$$

Given point processes $X = (Y_1, \ldots, Y_N)$, we wish to infer $\mathcal{I}$ and θ, since those variables contain information about similarity. In particular, we are interested in the timing jitter s_t and the number of events per cluster; the latter is given by $c_m + 1$, i.e., there is one exemplar in each cluster and c_m noisy copies. The smaller the timing jitter s_t and the more events contained in each cluster, the more similar the point processes are considered to be. The parameter s_t is part of θ, and the variables c_m can be directly extracted from $\mathcal{I}$. (Note: in section 6.5, we will denote the average number of events per cluster by n_c.)

6.4.3 Statistical inference for multivariate SES

By considering the minus logarithm of the above stochastic model:

$$-\log p(\tilde{X}, X, \mathcal{I}, \theta) = -\log p(\theta_c) - \log p(\theta_x) - \log(1 - \lambda \operatorname{vol}(S)) - M \log \frac{\lambda}{N}$$
$$- \sum_{m=1}^{M} \log \delta\big(x_{i(m),k(m)} - \tilde{x}_m\big) - \log\left(p(c_m|\theta_c)\binom{N-1}{c_m}^{-1}\right)$$
$$- \sum_{(i,j)\in\log\mathcal{I}_m^{\text{copy}}} \log p_x(x_{i,j}|\tilde{x}_m, \theta_x). \tag{6.38}$$

The term $-\log p_x(x_{i,j}|\tilde{x}_m, \theta_x)$ may be interpreted as a measure of the distance between $x_{i,j}$ and $\tilde{x}_m$; note that this measure is not necessarily symmetric or non-negative. If p_x is a Gaussian distribution (as in the case of bump models), this measure is nothing but the Euclidean distance. In other applications, non-Euclidean distances may be more appropriate. The proposed algorithm can straightforwardly handle arbitrary distance measures. Let us now consider specific choices for $p(c_m|\theta_c)$; if the latter is a binomial distribution with $N - 1$ trials and probability of success p_s, and the prior for p_s is a beta distribution

$\mathrm{B}(\kappa, \lambda)$, we have:

$$\begin{aligned} -\log p(\tilde{X}, X, \mathcal{I}, \theta) = &-\log \mathrm{B}(p_s; \kappa, \lambda) - \log p(\theta_x) - \log(1 - \lambda \operatorname{vol}(S)) - M \log \frac{\lambda}{N} \\ &- \sum_{m=1}^{M} \log \delta\big(x_{i(m),k(m)} - \tilde{x}_m\big) - M(N-1) \log \delta \\ &- \sum_{m=1}^{M} (N - 1 - c_m) \log \frac{1 - p_s}{p_s} - \sum_{(i,j) \in \mathcal{I}_m^{\mathrm{copy}}} \log p_x(x_{i,j} | \tilde{x}_m, \theta_x), \end{aligned} \tag{6.39}$$

which we can rewrite as:

$$\begin{aligned} -\log p(\tilde{X}, X, \mathcal{I}, \theta) = &-\log \mathrm{B}(p_s; \kappa, \lambda) - \log p(\theta_x) - \log(1 - \lambda \operatorname{vol}(S)) + \alpha M \\ &- \sum_{m=1}^{M} \log \delta\big(x_{i(m),k(m)} - \tilde{x}_m\big) + \beta \sum_{m=1}^{M} (N - 1 - c_m) \\ &- \sum_{(i,j) \in \mathcal{I}_m^{\mathrm{copy}}} \log p_x(x_{i,j} | \tilde{x}_m, \theta_x), \end{aligned} \tag{6.40}$$

where

$$\alpha = -\log \frac{\lambda}{N} - (N-1) \log p_s \quad \text{and} \quad \beta = \log\Big(\frac{p_s}{1 - p_s}\Big). \tag{6.41}$$

A reasonable approach to infer $(\mathcal{I}, \theta)$ is maximum a posteriori (MAP) estimation:

$$(\hat{\mathcal{I}}, \hat{\theta}) = \operatorname{argmax}_{(\mathcal{I}, \theta)} \log p(\tilde{X}, X, \mathcal{I}, \theta). \tag{6.42}$$

As there exists no closed form expression for (6.42), we need to resort to numerical methods. A simple technique to try to find (6.42) is coordinate descent: We first choose initial values $\hat{\theta}^{(0)}$, and then perform the following updates for $r \geq 1$ until convergence:

$$\hat{\mathcal{I}}^{(r)} = \operatorname{argmax}_{\mathcal{I}} \log p(\tilde{X}, X, \mathcal{I}, \hat{\theta}^{(r-1)}) \tag{6.43}$$

$$\hat{\theta}^{(r)} = \operatorname{argmax}_{\theta} \log p(\tilde{X}, X, \hat{\mathcal{I}}^{(r)}, \theta). \tag{6.44}$$

First we consider the update (6.43), which we will carry out by integer programming. Next, we treat the update (6.44) of the parameters θ.

Integer Program

We write the update (6.43) as an integer program, i.e., a discrete optimization problem with linear objective function and linear (equality and inequality) constraints. To this end, we introduce the following variables:

- $S_{i,k}$ is a binary variable equal to one iff the k-th event of Y_i is an exemplar.
- $C_{i,k,i',k'}$ is a binary variable equal to one iff the k-th event of Y_i is copy of exemplar $Y_{i',k'}$.
- $M_{i,i',k'}$ is a binary variable equal to one iff no event of Y_i is a copy of exemplar $Y_{i',k'}$.

Note that $c_{i,k,i,k'} = 0$ for all k and k' and $m_{i,i,k'} = 1$ for all i and k', since Y_i must not contain a noisy copy of a mother event $\tilde{X}_m$ if it already contains the exemplar associated to $\tilde{X}_m$.

First assume that the parameters θ_x and p_s of the binomial prior are constant. By substituting (6.40) in (6.43), it can be easily shown that with the above choice of variables b, the conditional maximization (6.43) may be cast as the following integer program in b:

$$\min_{b} \quad \hat{\alpha}^{(r-1)} \sum_{i,1\le k\le n_i} s_{i,k} + \hat{\beta}^{(r-1)} \sum_{i,i'\neq i,1\le k'\le n_{i'}} m_{i,i',k'} - \sum_{i,i',1\le k\le n_i,1\le k'\le n_{i'}} c_{i,k,i',k'} \log p_x(x_{i,k}|x_{i',k'};\hat{\theta}^{(r-1)}) + C \tag{6.45}$$

subject to

$$\forall i,k, \qquad \sum_{i',k'} c_{i,k,i',k'} + s_{i,k} = 1 \tag{6.46}$$

$$\forall i, i' \neq i, k', \quad m_{i,i',k'} = s_{i',k'} - \sum_{1\le k\le n_i} c_{i,k,i',k'}, \tag{6.47}$$

where C is an irrelevant constant, and

$$\hat{\alpha}^{(r-1)} = -\log\frac{\lambda}{N} - (N-1)\log\hat{p}_s^{(r-1)} \tag{6.48}$$

$$\hat{\beta}^{(r-1)} = \log\left(\frac{\hat{p}_s^{(r-1)}}{1-\hat{p}_s^{(r-1)}}\right). \tag{6.49}$$

The sum $\sum_{i,k} b_{i,k}$ in (6.45) is equal to the number of exemplars M; therefore, the first term in (6.45) assigns a cost α to each exemplar. The second term in (6.45) associates a cost β to every deletion. Indeed, if (i', k') is not an exemplar, $\sum_i b_{i,i',k'}$ is equal to zero; if (i', k') is the exemplar associated to the m-th mother event, $\sum_i b_{i,i',k'} = (N - 1 - c_m)$, which is the number of deletions in the m-th cluster. The third term assigns a cost to each copy (i, k) of exemplar (i', k'), proportional to the "distance" $-\log p_x$ between both events.

The constraint (6.46) ensures that each event is either an exemplar or a copy of an exemplar. The constraint (6.47), combined with the fact that $b_{i,i',k'}$ is a binary variable, encodes the following:

- $c_{i,k,i',k'}$ can only be equal to one if $s_{i',k'}$ is equal to one, i.e., (i, k) can be a copy of (i', k') iff (i', k') is an exemplar,
- at most one event in Y_i can be a copy of (i', k'),
- $m_{i,i',k'}$ is one iff (i', k') is an exemplar but has no copy in Y_i.

The discrete optimization problem (6.45)-(6.47) is an integer program in b, since the objective function (6.45) and constraints (6.46) (6.47) are linear in the variables b.

This optimization problem may be solved by max-product message-passing on a sparse graph of $p(\tilde{X}, X, \mathcal{I}, \theta)$ (6.37), along the lines of the algorithm of [DVW+09]. We implemented this approach for the problem described in Section 6.5, and found that Belief Propagation converged in about 70% of instances. In those instances it converged, it found a solution which was always optimal or near-optimal. In the remaining 30% of instances, BP either diverged, or converged to an extremely poor solution (choosing every single bump to be an exemplar). Visual inspection of the instances for which BP converged and those where BP did not surprisingly did not reveal any qualitative differences between the bump profiles. As a result, we opted for the following scheme for optimization: we first run the BP algorithm, inspect the resulting solution, and, if poor, use an integer programming algorithm (CPLEX) to solve the instance. CPLEX was in fact found to solve most of these instances fairly fast.

6.5 Application to early diagnosis of Alzheimer's disease

In this last section, we investigate the applicability of both bivariate and multivariate SES to the problem of early prediction of Alzheimer's disease.

6.5.1 EEG Data

The EEG data used here has been analyzed in previous studies concerning early diagnosis of Alzheimer's disease (AD) [CNM+07, CSM+05, HSK+03, MAY+02, VCD+05]. They consist of rest, eyes-closed EEG data recorded from 21 sites on the scalp based on the 10–20 system (see Fig. C-2)). The sampling frequency f_s was 200Hz, which allowed for signals of up to 100Hz to be represented. As in [CNM+07, CSM+05, HSK+03, MAY+02, VCD+05], the signals were band-pass filtered between 4Hz and 30Hz using a third-order Butterworth filter.

The subjects comprised two study groups. The first consisted of a group of 25 patients who had complained of memory problems. These subjects were then diagnosed as suffering from mild cognitive impairment (MCI) and subsequently developed mild AD. The criteria for inclusion into the MCI group were a mini mental state exam (MMSE) score = 24, though the average score in the MCI group was 26 (SD of 1.8). The other group was a control set consisting of 56 age-matched, healthy subjects who had no memory or other cognitive impairments. The average MMSE of this control group was 28.5 (SD of 1.6). The ages of the two groups were 71.9 $\pm$ 10.2 and 71.7 $\pm$ 8.3, respectively. Finally, it should be noted that the MMSE scores of the MCI subjects studied here are quite high compared to a number of other studies. For example, in [HSK+03] the inclusion criterion was MMSE = 20, with a mean value of 23.7, while in [CNM+07], the criterion was MMSE = 22 (the mean value was not provided); thus, the disparity in cognitive ability between the MCI and control subjects was comparatively small, making the present classification task relatively difficult.

All recording sessions were conducted with the subjects in an awake but resting state with eyes closed.

After recording, the EEG data was carefully inspected. Indeed, EEG recordings are prone to a variety of artifacts, for example due to electronic smog, head movements, and muscular activity. The EEG data has been investigated independently by three EEG experts. EEG segments were considered as artifact-free if all three experts agreed. We retained in our analysis only those subjects whose EEG recordings contained at least 20s of artifact-free data. Based on this requirement, the number of subjects in the two groups described above was further reduced to 22 and 38, respectively.

We first applied bivariate SES to the 10 region pairs, and averaged the results over those pairs, resulting in one set of (average) bivariate-SES parameters per subject. Then,

we applied multivariate SES to the 5 regions *simultaneously.* Besides SES, we applied a large variety of classical approaches, as we will explain in the following section.

Methods and Statistics

We studied the following statistics, directly extracted from the bump model:

- $\overline{\Delta T}$: average width of bumps,
- $\overline{\Delta F}$: average height of bumps,
- $\bar{F}$: average frequency of bumps.

From bivariate SES, we obtained the following statistics:

- $s_t^{2\mathrm{D}}$: timing jitter variance,
- ρ: fraction of unmatched ("orphan") bumps.

From multidimensional SES, we obtained:

- s_t: variance in time domain ("time jitter"),
- p_c^i: the fraction of clusters with i bumps (for each $i = 1, \ldots, N$),
- n_c: average number of bumps per cluster.

We also consider the linear combination h_c of all parameters p_c^i that optimally separates both subject groups, obtained through leave-one-out cross-validation.

Besides SES, we applied a variety of classical synchronicity measures to the EEG data:

- Pearson cross-correlation coefficient [NS06],
- mean-square and phase coherence [NS06],
- Granger causality [KL05], in particular, Granger coherence, partial coherence, partial directed coherence (PDC), directed transfer function (DTF), full-frequency directed transfer function (ffDTF), and direct directed transfer function (dDTF),
- the recently proposed corr-entropy coefficient and wave-entropy coefficient [XBCP06],

- phase synchronicity indices derived from the Hilbert transform and time-frequency maps [LRMV99], global field synchronization (GFS) [KLS+01b], evolution map approach (EMA), and the instantaneous period approach (IPA) [RCB+02],
- mutual information, both in time domain (I) [KSG04] and time-frequency domain (I_W) [Avi05],
- information-theoretic divergence measures [Avi05] (in time-frequency domain), in particular, Kullback-Leibler, Rényi, Jensen-Shannon, and Jensen-Rényi divergence,
- state space based measures, in particular, the non-linear interdependence indices N^k, S^k, H^k [QQKKG02], and the S-estimator [CKIDF05].

For the sake of brevity, we will not expand on the technical details here (see [DVMC] for a study based on those measures), and instead only discuss the results.

6.5.2 Results and Discussion

The main results are summarized in Table 6.1, which shows the sensitivity of the synchronicity measures for diagnosing MCI. More precisely, it contains p-values obtained by the Mann-Whitney test. This test indicates whether the statistics at hand, in particular, the synchronicity measures, take different values for the two subject populations: low p-values indicate large difference in the medians of the two populations. Note that while a low p-value does not necessarily imply small classification error, it is a necessary condition for good classification, and an indicator of the classification strength. We therefore use p-values as a way to identify potentially good features for classification; we investigate classification in the following section.

Note that the p-values in Table 6.1 need to be statistically corrected. Indeed, since we consider many different measures simultaneously, it is likely that a few of those measures have small p-values due to stochastic fluctuations and *not* due to systematic difference between MCI patients and control subjects. Therefore, the p-values need to be corrected accordingly, for example, by means of Bonferroni [Bon36] or Sidak [Sid67] post-correction or step-down methods [Hoc88, Hol79]. In the most conservative Bonferroni post-correction, the p-values of Table 6.1 need to be multiplied by the number of synchronicity measures. As was shown in [DVMC], however, many synchronicity measures are strongly correlated: one can distinguish a small number of families of synchronicity measures. As a consequence,

Measure	Cross-correlation	Coherence	Phase Coherence	Corr-entropy	Wave-entropy	
p-value	**0.028***	0.060	0.72	0.27	**0.012*†**	
References	[NS06]			[XBCP06]		
Measure	Granger coherence	Partial Coherence	PDC	DTF	ffDTF	dDTF
p-value	0.15	0.16	0.60	0.34	**0.0012**†**	**0.030***
References	[KL05]					
Measure	Kullback-Leibler	Rényi	Jensen-Shannon	Jensen-Rényi	I_W	I
p-value	0.072	0.076	0.084	0.12	0.080	0.060
References	[Avi05]					[KSG04]
Measure	N^k	S^k	H^k	S-estimator		
p-value	**0.032***	0.29	0.090	0.33		
References	[QQKKG02]			[CKIDF05]		
Measure	Hilbert Phase	Wavelet Phase	Evolution Map	Instantaneous Period	GFS	
p-value	0.15	0.082	0.072	**0.020***	0.51	
References	[LRMV99]		[RCB+02]		[KLS+01b]	
Bump	$\overline{\Delta T}$	$\overline{\Delta F}$	$\bar{F}$			
p-value	**$2.3.10^{-4}$**†**	**0.023***	**2.10^{-3}****			
2D-SES	s_t^{2D}	ρ				
p-value	0.12	**0.00041**†**				
multi-SES	p_c^1	p_c^2	p_c^3	p_c^4	p_c^5	
p-value	**0.016***	**$2.9\ 10^{-4}$**†**	0.089	0.59	**0.0054***	
multi-SES	n_c	h_c	s_t			
p-value	**1.10^{-3}**†**	**1.10^{-4}**†**	0.46			

Table 6.1: Sensitivity of average synchronicity for early prediction of AD (p-values for Mann-Whitney test; * and ** indicate $p < 0.05$ and $p < 0.005$ respectively; † indicates p-values that remain significant after post-correction).

a less conservative but arguably more reasonable approach is to multiply the p-values of Table 6.1 by the number of synchronicity measure families (four or five). The p-values in Table 6.1 that remain significant after post-correction are indicated by †, i.e., wave-entropy (p = 0.012), full-frequency DTF (0.0012), ρ (p = 0.00041), p_c^2 ($p = 2.9\ 10^{-4}$), n_c ($p = 110^{-3}$), and $\overline{\Delta T}$ ($p = 2.310^{-4}$). Note that even after the most conservative Bonferroni post-correction, the most discriminative SES measures (ρ, p_c^2, n_c) remain significant. In figures D-1 and D-2, we show boxplots for these synchronicity measures.

The strongest observed effect is a significantly higher degree of non-correlated activity in MCI patients, more specifically, a high number of non-coincident, non-synchronous oscillatory events, as quantified by the statistics (ρ, p_c^i, n_c); in MCI patients, there is an increase in the fraction ρ of orphan events (see Fig. D-1(d)), a decrease in the average number of bumps per cluster (see Fig. D-1(e)), an increase in the number of clusters with 1 and 2 bumps (see Fig. D-2(a) and D-2(b)), and a decrease in the number of clusters of size 4 and 5 (see Fig. D-2(d) and D-2(e)). Interestingly, we did not observe a significant effect on the timing jitter s_t of the coincident events (see Fig. D-1(c)). In other words, MCI seems to be associated with a significant increase of non-coincident background activity,

while the *coincident* activity remains well synchronized. Of course, those observations beg for a physiological explanation. However, this clearly goes beyond the scope of the present study.

We verified that the SES measures are not correlated with other synchronicity measures (Pearson $r, p > 0.10$). In contrast to the classical measures, SES quantifies the synchronicity of oscillatory events instead of more conventional amplitude or phase synchronicity, and therefore provides complementary information about EEG synchronicity.

6.5.3 Classification

Combining the SES parameters (ρ, p_c, h_c, n_c) with ffDTF or bump parameters (e.g., $\overline{\Delta T}$) yields good classification of MCI vs. control patients; some examples are shown in figures 6-6, 6-7, and 6-8. The classification error, determined by leave-one-out cross validation, was found to be between 10% and 15%. These results are promising, as there are currently very few methods to reliably predict the onset of MCI. The methods of [CNM+07, CSM+05, HSK+03, MAY+02, VCD+05], applied to the same EEG data, have significantly higher classification errors and use vastly more features and thus are more prone to over-fitting. In addition, the classical synchronicity measures lead to poorer classification performance. For instance, the classical measure ffDTF, which is the most discriminative classical synchronicity measure for the EEG data at hand, leads to classification errors of about 30% (obtained through leave-one-out cross validation). Combining ffDTF with SES parameters leads to vastly better results; this can be explained by the fact that SES provides complementary information about synchronicity.

However, the classification errors we obtained are admittedly still too large to allow us to predict AD reliably. To this end, we would need to combine those synchronicity measures with complementary features, perhaps from different modalities such as PET, MRI, or biochemical indicators. We wish to point out, however, that in the data set at hand, patients did not carry out any specific task. In addition, we considered recordings of 20s, which is very short. It is plausible that the sensitivity of EEG synchronicity could be further improved by increasing the length of the recordings and by recording the EEG before, during, and after patients carry out specific tasks, e.g., working memory tasks. As such, the separation shown in Fig. 6-6, 6-7, and 6-8 might be applied to screen a population for MCI, since it requires only an EEG recording system. The latter is a relatively simple and low-cost technology, at present available in most hospitals. Moreover, the iterative

algorithms developed to perform inference are simple and have very efficient running time, so that no additional equipment would be needed to analyze the EEG data.

6.6 Conclusions

In this chapter, we introduced an alternative method to quantify the similarity of time series, referred to as stochastic event synchrony (SES). As a first step, the algorithm extracts events from the time series, resulting in point processes. These events are then optimally aligned. The better the alignment, the more similar the original time series are considered to be.

Obviously, it is important to extract meaningful events from the given time series. In the case of spike trains, individual spikes can naturally be considered as events. Note that for certain neurons, however, it may actually be more appropriate to define a burst of spikes as a single event. As we have shown, for spontaneous EEG signals, it is natural to consider oscillatory events from the time-frequency representation. However, even in this case there might be interesting alternatives, depending on the nature of the EEG.

Since the proposed similarity measure does not take the entire time series into account but focuses exclusively on certain events, it provides complementary information about synchronicity. Therefore, we believe that it may prove to be useful to blend our similarity measure with classical measures such as the Pearson correlation coefficient, Granger causality, or phase synchronicity indices.

We have applied the proposed approach to the problem of diagnosing Alzheimer's disease (AD) at an early stage based on electroencephalograms (EEG). We demonstrated that the SES measures are sensitive to AD-induced perturbations in EEG synchronicity; they allow an improvement of early diagnosis of AD, by combining those novel measures with classical measures.

The SES model is very modular and can be extended in several different ways. For instance, in the present study, the SES parameters are assumed to be constant in time. By considering time-varying parameters, we are able to study neural response to timed stimuli.

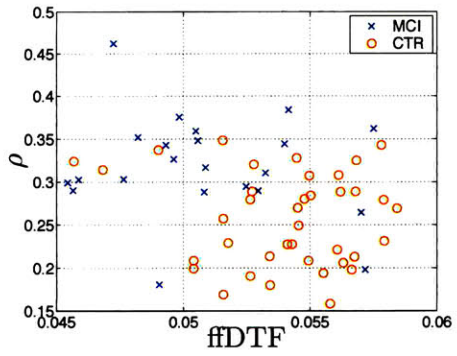
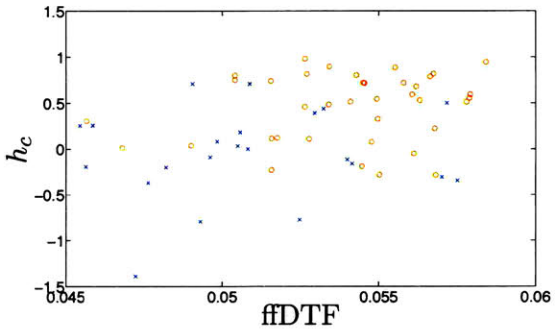
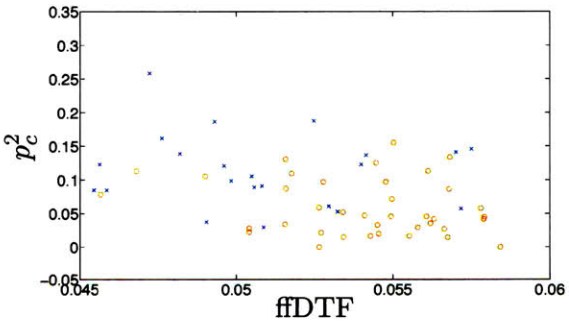


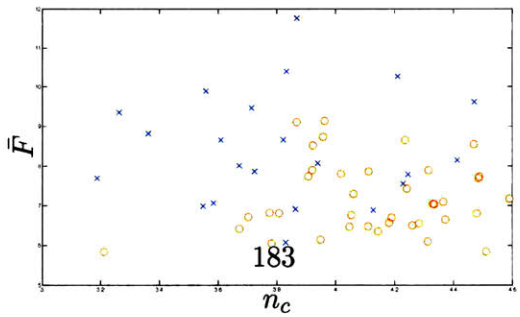
Figure 6-6: Combining bivariate-SES parameter ρ with ffDTF; red circles: CTR, blue crosses: MCI.



(a) h_c vs. ffDTF

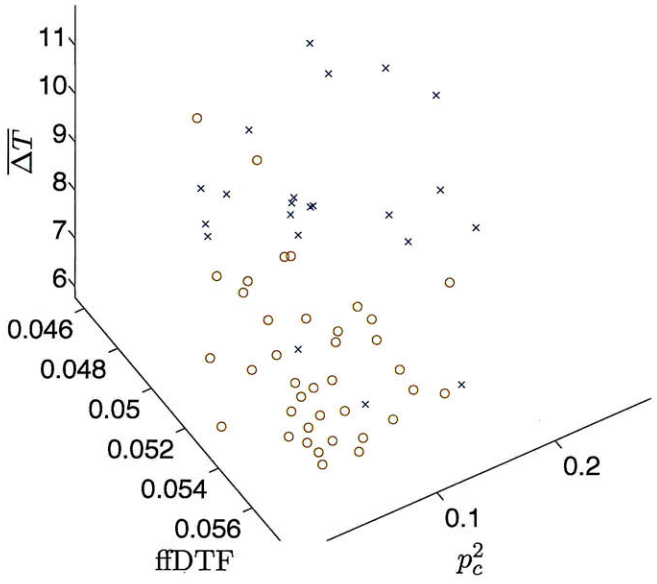


(b) p_c^2 vs. ffDTF

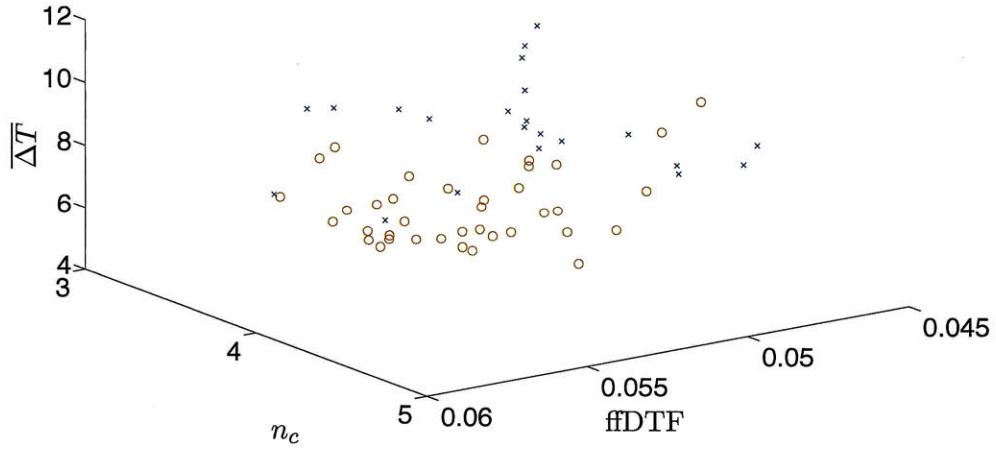


(c) $\bar{F}$ vs. n_c

Figure 6-7:]
Combination of two features; (top, middle) multivariate-SES parameters combined with ffDTF, and (bottom) multivariate-SES parameter n_c combined with bump parameter $\bar{F}$ (average frequency); red circles: CTR, blue crosses: MCI.



(a) $\overline{\Delta T}$ vs. ffDTF vs. p_c^2



(b) $\overline{\Delta T}$ vs. n_c vs. ffDTF

Figure 6-8: Combination of three features; multivariate-SES parameters combined with ffDTF and bump parameter $\overline{\Delta T}$; red circles: CTR, blue crosses: MCI.

Appendix A

Glossary

Chapter 1

$\mathcal{G} = (V, E, \Phi, \chi)$	Decision network
Φ	Set of interaction and potential functions
χ	Decision set: $\{0, 1, \ldots, T-1\}$
T	Number of actions
$J_{\mathcal{G}}$	Value function of network $\mathcal{G}$
$J_{\mathcal{G},\mathbf{v}}(x)$	Value function when nodes in $\mathbf{v}$ are constrained to take decisions $\mathbf{x}$
$B_{\mathcal{G},v}(x)$	Bonus of node v when taking action x
$B_{\mathcal{G}.v}$	Bonus of node v when taking action 1, when $\chi = \{0, 1\}$ ($B_{\mathcal{G},v} \in \mathbb{R}$ in this case)
$\mu_{u \leftarrow v}(x, B)$	partial cavity function sent v to u : $\max_y(\Phi_{u,v}(x, y) + B(y)) - \max_y(\Phi_{u,v}(0, y) + B(y))$

Chapter 2

$\mathcal{G} \setminus \{u\}$	Subnetwork induced by removal of node u
$\mathcal{G}(u, j, x)$	j^{th} modified subnetwork of $\mathcal{G}$ with action x, see section 2.2.2, Equation (2.1)
$\hat{B}(\mathcal{G}, u, r)$	Approximation of the cavity function $B_{\mathcal{G},u}$ with depth r, see section 2.3
$\mathcal{C}$	Boundary condition for the Cavity Expansion algorithm
CE$[\mathcal{G}, u, r, x, \mathcal{C}]$	Cavity Expansion for network $\mathcal{G}$, node u, depth r, action x, see section 2.3.1

$\rho(r)$	Correlation decay rate
K_c, α_c	Exponential correlation decay parameters
g	Bound on the density of the cavities
x_v^r	$\text{argmax}_x \text{CE}[\mathcal{G}, v, r, x]$
u	node of interest
$(v_1, \ldots, v_d)$	neighbors of u
d	number of neighbors of u
$\mathcal{G}_j$	$\mathcal{G}(u, j, 1)$
$(v_{j1}, \ldots, v_{jn_j})$	neighbors of v_j in $\mathcal{G}_j$ (i.e., neighbors of v_j in $\mathcal{G}$, except for u)
n_j	number of neighbors of v_j in $\mathcal{G}$
$\mathcal{C}, \mathcal{C}'$	Two different boundary conditions
$B(r)$	$\text{CE}[\mathcal{G}, u, r, 1, \mathcal{C}]$
$B'(r)$	$\text{CE}[\mathcal{G}, u, r, 1, \mathcal{C}']$
$B_j(r-1)$	$\text{CE}[\mathcal{G}_j, v_j, r-1, 1, \mathcal{C}]$
$B'_j(r-1)$	$\text{CE}[\mathcal{G}_j, v_j, r-1, 1, \mathcal{C}']$
$\mathbf{B}(r-1)$	$(B_j(r-1))_{1 \le j \le d}$
$\mathbf{B}'(r-1)$	$(B'_j(r-1))_{1 \le j \le d}$
$B_{jk}(r-2)$	$\text{CE}[\mathcal{G}_j(v_j, k, 1), v_j, r-2, 1, \mathcal{C}]$
$B'_{jk}(r-2)$	$\text{CE}[\mathcal{G}_j(v_j, k, 1), v_j, r-2, 1, \mathcal{C}']$
$\mathbf{B_j}(r-2)$	$(B_{jk}(r-2))_{1 \le k \le n_j}$
$\mathbf{B'_j}(r-2)$	$(B'_{jk}(r-2))_{1 \le k \le n_j}$
$\mu_{u \leftarrow v_j}(z)$	$\mu_{u \leftarrow v_j}(1, z)$
$\mu_{v_j \leftarrow v_{jk}}(z)$	$\mu_{v_j \leftarrow v_{jk}}(1, z)$
$\Phi^1_{u,v}$	$\Phi_{u,v}(1,0) - \Phi_{u,v}(1,1)$, for any (u, v)
$\Phi^2_{u,v}$	$\Phi_{u,v}(0,0) - \Phi_{u,v}(0,1)$, for any (u, v)
$\Phi^3_{u,v}$	$\Phi_{u,v}(1,1) - \Phi_{u,v}(0,1)$, for any (u, v)
$X_{u,v}$	$\Phi^1_{u,v} + \Phi^2_{u,v}$
$Y_{u,v}$	$\Phi^2_{u,v} - \Phi^1_{u,v} = \Phi_{u,v}(1,1) - \Phi_{u,v}(1,0) - \Phi_{u,v}(0,1) + \Phi_{u,v}(0,0)$
$\partial\Phi_v$	$\Phi_v(0) - \Phi_v(1)$
$E^+_{u \leftarrow v}(x, x')$	$\{x \ge \partial\Phi_v + \max(\Phi^1_{u \leftarrow v}, \Phi^2_{u \leftarrow v})\} \cap \{x' \ge \partial\Phi_v + \max(\Phi^1_{u \leftarrow v}, \Phi^2_{u \leftarrow v})\}$
$E^-_{u \leftarrow v}(x, x')$	$\{x \le \partial\Phi_v + \min(\Phi^1_{u \leftarrow v}, \Phi^2_{u \leftarrow v})\} \cap \{x' \le \partial\Phi_v + \min(\Phi^1_{u \leftarrow v}, \Phi^2_{u \leftarrow v})\}$
$E_{u \leftarrow v}(x, x')$	$E^+_{u,v}(x, x') \cup E^-_{u,v}(x, x')$

$W(U)$	Weight of set U: $\sum_{u\in U} W_u$
$M(\mathcal{G})$	Size $\lvert\mathcal{G}\rvert$ of graph $\mathcal{G}$
$\alpha(\mathcal{G})$	Independence number of $\mathcal{G}$
I^s	Maximum Independent set of $\mathcal{G}$
I^*	Maximum Weighted Independent Set of $\mathcal{G}$
$\mathcal{I}$	Independent set output by the PTAS
$C_{\mathcal{G}}(i)$	Censored cavity of node i in graph $\mathcal{G} \triangleq \max(B_{\mathcal{G}(i)}, 0)$
$C^-_{\mathcal{G}}(i,r)$	Lower bound and approximation of $C_{\mathcal{G}}(i)$ resulting from the CE of depth r
$C^+_{\mathcal{G}}(i,r)$	Upper bound and approximation of $C_{\mathcal{G}}(i)$ resulting from the CE of depth r
ϵ'	$\epsilon^2/2$
$\mathcal{I}(r,\epsilon)$	$\{i \mid C^-_{\mathcal{G}(\epsilon')}(i,r) > 0\}$
$\mathcal{G}(\epsilon)$	Graph obtained from the original graph $\mathcal{G}$ after removing each node with probability ϵ'
$\mathcal{H}$	Subgraph of $\mathcal{G}(\epsilon)$
I^*_0	MWIS of $\mathcal{G}(\epsilon)$
$M_{\mathcal{G}}(i)$	$\mathbb{E}[\exp(-C_{\mathcal{G}}(i))]$
$M^-_{\mathcal{G}}(i,t)$	$\mathbb{E}[\exp(-C^-_{\mathcal{G}}(i,t))]$
$M^+_{\mathcal{G}}(i,t)$	$\mathbb{E}[\exp(-C^+_{\mathcal{G}}(i,t))]$
$M^j_{\mathcal{H}}(i)$	$\mathbb{E}[\exp(-\alpha_j\, C_{\mathcal{H}}(i))]$
$M^{-,j}_{\mathcal{H}}(i,t)$	$\mathbb{E}[\exp(-\alpha_j\, C^-_{\mathcal{H}}(i,t))]$
$M^{+,j}_{\mathcal{H}}(i,t)$	$\mathbb{E}[\exp(-\alpha_j\, C^+_{\mathcal{H}}(i,t))]$

Chapter 5

ϕ_u	Payoff function of player u
$\mathbf{s}_u$	Strategy of player u
$\mathbf{s}$	Strategy profile
$\mathcal{S}(\chi)$	strategies over the set χ
BR	Best-response function
$\phi_{u \leftarrow v}$	contribution of agent v to the payoff of agent u, in a decomposable game
$Z_{\mathcal{G},w}$	Nash cavity function
$\mathcal{H}$	Graphical model derived from graphical game $\mathcal{G}$
Z_v	In a directed tree game, optimal decision of agent v
β	branching rate
$\alpha_v(d)$	Dobrushin coefficient for correlation decay in graphical games of degree d

Chapter 6

ρ	Fraction of orphan bumps
δ_t	Average time offset
δ_f	Average frequency offset
σ_t	Variance of the time offset
σ_f	Variance of the frequency offset
l	Number of bumps in the mother process
λ	Exponential parameter for l
T_0	Measurement time
f_min	Minimum frequency sampled
f_max	Maximum frequency sampled
Y,Y'	Observed bump processes
Z, Z'	Processes obtained after bumps deletion
$n_{\text{del}}, n_{\text{del}}$	Number of deletions in Z, Z'
$n_{\text{del}}^{\text{double}}$	Number of double deletions
p_d	Probability of deletion
β	$(1-\lambda)(\frac{1-p_d}{p_d})^{n+n'}$
$c_{kk'}$	Indicator variable of assignment of bump k to bump k'
b_k	Indicator variable of orphan bump
$\overline{\Delta T}$	average width of bumps
$\overline{\Delta F}$	average height of bumps
$\bar{F}$	average frequency of bumps
s_t	Variance in time domain
p_c^i	Fraction of clusters with i bumps
n_c	average number of bumps per cluster
h_c	Best linear combination of n_c's

Appendix B

Notions in complexity theory and approximation algorithms

In this appendix, we will give a very brief primer on the main notions of complexity theory used in the rest of the thesis. The first section deals with the P and NP classes in optimization and the notion of approximation algorithm, and the second deals with the PPAD class used in game theory.

We refer the reader to Sipser's book [Sip96] on complexity theory and computation for more details on these topics.

B.1 The P and NP classes, approximation algorithms

A *decision problem* is defined as a set $\mathcal{P} \triangleq (I, v)$, where I is a set of instances, and v a function from I to $\{0, 1\}$. The objective is to compute the function $v(i)$ for any $i \in I$. An algorithm $\mathcal{A}$ for the decision problem $\mathcal{P}$ is a sequence of non-ambiguous instructions which can be simulated on a *Turing machine*, takes as input an instance i and outputs the value $v(i)$. The running time of the algorithm is the number of steps the Turing machine takes in order to produce an answer.

Optimization problems are similary defined. An optimization problem is a set $\mathcal{P} \triangleq (I, f, v)$, where I is a set of instances, f is a feasibility function, and v a value function. For any i, $f(i)$ is the set of feasible solution, and $v(i, .)$ is a function from $f(i)$ to $\mathbb{R}$. The objective is to find for any i a solution $x \in f(i)$ which maximizes the function $v(i, .)$, or in other words, finding $x \in f(i) \cap \text{argmax}_y(v(i, y))$. Such a solution x is called *optimal*.

An algorithm $\mathcal{A}$ for problem $\mathcal{P}$ is a sequence of non-ambiguous operations which can be simulated on a Turing machine, which takes as input an instance i and outputs a candidate solution $x \in f(i)$. $\mathcal{A}$ is said to be optimal if it always outputs an optimal solution.

In many optimization problem, we can define a natural *size* of the instance $M(i)$. We will often be interested in upper bounding the running time of an algorithm $\mathcal{A}$ as a function of the size of the instance it is running on.

P and NP

An algorithm $\mathcal{A}$ is said to be running in polynomial time if its computation takes a number of steps at most polynomial in the size of the input.

The set of all decision problems for which there exists a polynomial time algorithm is called P. Similarly, we will also call P the set of all optimization problems for which there exists a polynomial time running algorithm. P is, in essence, the set of all problems for which there exists "fast" or "efficient" algorithms.

Because it proved hard to rigorously show that problems do not belong to P, researchers introduced new complexity classes of problems which are believed to be strict supersets of P.

Perhaps the most famous such class is NP. A decision problem is said to be in NP if for all problems which have answer 'yes', the answer can be verified in polynomial time. In other words, a decision problem is in NP if for all its positive instances, there exists a polynomial checkable certificate of positivity (i.e., a "short proof" that the answer is, in fact, 'yes'). Formally, a problem is in NP if it can be simulated on a non-deterministic Turing machine).

For any optimization problem $\mathcal{P} = (I, f, v)$, define a corresponding decision problem $\mathcal{P}' = (I', v')$, where $I' = (I, \mathbb{R})$, and where for any $i \in I$ and $\alpha \in \mathbb{R}$, v' is the answer to the question "does there exist a solution x such that $f(x) \geq \alpha$?". Then, we say that $\mathcal{P}$ is in NP if and only if $\mathcal{P}'$ is.

A reduction from a problem $\mathcal{P} = (I, v)$ to $\mathcal{P}' = (I', v')$ is a function g from I to I' such that for any $i \in I$, $g(i) \in I'$, and $v(i) = v'(g(i))$. A reduction is said to be polynomial if the size of $g(i)$ is always polynomial in the size of i, and $g \in \mathcal{P}$. If there exists a polynomial reduction from $\mathcal{P}$ to $\mathcal{P}'$, this means that problem $\mathcal{P}'$ is in some sense harder than $\mathcal{P}$, since we can always solve the decision problem $\mathcal{P}$ through its reduction g.

A reduction from an optimization problem $\mathcal{P} = (I, f, v)$ to $\mathcal{P}' = (I', f', v')$ is a pair of

functions (g, h), where g is a function from $\mathcal{P}$ to $\mathcal{P}'$ and a h a function from $\{i', f(i')\}$ to $\{i, f(i)\}$, such that for any $i \in I$, an optimal solution x of $g(i)$ can be converted through h into an optimal solution $h(x)$ of i.

Finally, a decision or optimization problem $\mathcal{P}$ is said to be NP-hard if for any problem $\mathcal{P}'$ in NP, there exists a polynomial reduction from $\mathcal{P}'$ to $\mathcal{P}$. In other words, NP-hard problems are harder than all problems in NP, since if we can solve an NP-hard problem, we can solve any NP problem. A problem is NP-complete if it is both NP-hard and in NP. Most computer scientists believe that P≠NP, implying that there exists no polynomial time algorithms for NP-hard problems.

Randomized algorithms

Intuitively, a randomized algorithm $\mathcal{A}$ is a sequence of non-ambiguous instructions which can be simulated on a Turing machine, with the additional assumption that the Turing machine can read from an infinitely long tape which contains a sequence of random bernoulli numbers drawn independently with probability 1/2.

Approximation algorithm

An approximation algorithm $\mathcal{A}$ for a problem $\mathcal{P}$ with additive error ϵ is an algorithm which for any $i \in I$, outputs a solution $\mathcal{A}(i)$ such that for all i, $|\max_y v(i, y) - v(i, \mathcal{A}(i))| \leq \epsilon$.

For a problem $\mathcal{P}$ such that for all i, $v(i)$ is a positive function, an approximation algorithm $\mathcal{A}$ for a problem $\mathcal{P}$ with approximation ratio ρ is an algorithm which for any $i \in I$, outputs a solution $\mathcal{A}(i)$ such that for all i, $\frac{\max_y v(i,y)}{v(i,\mathcal{A}(i))} \leq \rho$. We also say that $\mathcal{A}$ is a ρ-factor approximation algorithm for $\mathcal{P}$.

A polynomial-time approximation scheme (PTAS) is a parametrized algorithm $\mathcal{A}(\epsilon)$, such that for for every $\epsilon > 0$, $\mathcal{A}(\epsilon)$ is an $(1 + \epsilon)$-factor approximation algorithm which runs in polynomial time. A similar notion can be defined for the additive error, where an algorithm $\mathcal{A}(\epsilon)$ is called a PTAS if for any $\epsilon > 0$ it is an AS with additive error ϵ.

An efficient polynomial-time approximation scheme (EPTAS) is a PTAS whose running time is upper bounded by some $h(\epsilon)\, n^{O(1)}$, where n is the size of the input.

Finally, a fully-polynomial-time approximation scheme (FPTAS) is a PTAS whose running time is a polynomial in n and ϵ, where n is the size of the input.

Approximation schemes which use randomized algorithms can also be defined, in which case they are respectively called randomized approximation scheme, polynomial-time, ran-

domized approximation scheme (PRAS), efficient polynomial-time, randomized approximation scheme (EPRAS), and fully polynomial-time, randomized approximation scheme (FPRAS).

Specific notions for approximation algorithms in decision networks

We give here give a specific definition of approximation algorithm in the context used by the main body of this thesis. Note that since our problem is non-standard (random input), our definition slightly differ from classical ones.

For any network $\mathcal{G}$, we call $M(\mathcal{G}) = \max(|V|, |E|, |\chi|)$ the size of the network. Since we will exclusively consider graphs with degree bounded by a constant, for all practical purposes we can think of $|V|$ as the size of the instance. We will say that an algorithm is an additive (resp. multiplicative) PTAS with high probability if for all $\epsilon > 0$ it outputs in time polynomial in $|V|$ a solution $\hat{x}$ such that $\mathbb{P}(|J_{\mathcal{G}} - F(\hat{x})| > \epsilon) < \epsilon$ (resp. $\mathbb{P}(\frac{J_{\mathcal{G}}}{F(\hat{x})} > 1+\epsilon) < \epsilon$). EPTAS and FPTAS with high probability are similarly defined, as are their randomized counterparts.

B.2 The PPAD class

We closely follow the exposition of Daskalakis et al. *[DGP09])*

A search problem is defined as $\mathcal{P} = (I, S)$, where I is a set of instances, and S is a function from I to a set of candidate solutions (for simplicity, the set of all finite binary strings). The objective is, for all $i \in I$, to find an element $x \in S(i)$. A total search problem is a search problem for which $S(i)$ is nonempty for all i.

As an example, finding a pure Nash equilibrium is a search problem, where the set of solutions is the set of all decisions vectors which are in best response to the complement decision vectors (see Chapter 5, Equation 5.2); it is not necessarily total.

In contrast, finding an approximate Nash equilibrium is, by Nash's Theorem, a total search problem.

PPAD can intuitively be defined as the set of all total search problems whose totality can be established by the following argument: in any directed graph, if there exists an imbalanced node (a node whose outdegree is different from its indegree), then there exists another imbalanced node.

Formally, a problem in PPAD is defined by two functions P, S from $\{0,1\}^n$ to $\{0,1\}^n$ (n is the size of the input), such that $P(0_n) = 0_n \neq S(0_n)$; the search problem consists in finding $x \in \{0,1\}^n$ such that $P(S(x)) \neq x$ or $S(P(x)) \neq x \neq 0_n$.

Appendix C

Preprocessing of brain data: Wavelet Transform and Bump Modeling of EEG data

This appendix is based on work by F. Vialatte

We successively apply the following transformations to the EEG signals:

1. wavelet transform,
2. normalization of the wavelet coefficients,
3. bump modeling of the normalized wavelet representation,
4. aggregation of the resulting bump models in several regions.

As previously explained, as soon as the EEG is transformed through an appropriate wavelet, discrete patterns of activity become apparent. However, before extracting bump models from the time-frequency maps, it is necessary to normalize the latter. EEG signals have a very non-flat spectrum with an overall 1/f shape, disrupted by state-dependent peaks at specific frequencies [NS06]. Therefore, most energy of the time-frequency maps is located at low frequencies f. If we directly apply bump modeling to the (unnormalized) time-frequency maps, most bumps will be located in the low-frequency range; in other words, the high-frequency range will be under-represented. Since relevant information might be contained at high frequency, we normalize the map $S(t, f)$ before extracting the

bump models. As a result, the bumps are more or less uniformly distributed on the time-frequency maps. (In the statistical model underlying SES, we use a uniform prior for the bump position (see Appendix 6.3)).

After normalization, we extract a bump model from each EEG channel; since there are typically many EEG channels (usually at least 20), we aggregate the models in several regions (see, e.g., Fig. C-2), resulting in one bump model per region. In principle, one may apply SES to the original bump models, but it is computationally attractive to first reduce the number of models. This aggregation procedure is the last pre-processing step. Next, SES is applied to the bump models of each region.

Eventually, we compute the SES parameters for each pair of aggregated bump models. In the following, we detail each of those five operations.

Wavelet Transform

In order to extract the oscillatory patterns in the EEG, we apply a wavelet transform. More specifically, we use the complex Morlet wavelets [GGM84, DEG+92]:

$$\psi(t) = A \exp\left(-t^2/2\sigma_0^2\right) \exp(2i\pi f_0 t), \tag{C.1}$$

where t is time, f_0 is frequency, σ_0 is a (positive) real parameter, and A is a (positive) normalization factor. The Morlet wavelet (C.1) has proven to be well suited for the time-frequency analysis of EEG (see [TBBDP96]). The product $w_0 = 2\pi f_0 \cdot \sigma_0$ determines the number of periods in the wavelet ("wavenumber"). This number should be sufficiently large (≥ 5), otherwise the wavelet $\psi(t)$ will not fulfill the admissibility condition:

$$\int \frac{|\psi(t)|^2}{t} dt < \infty, \tag{C.2}$$

and as a result, the temporal localization of the wavelet becomes unsatisfactory [GGM84, DEG+92]. In the present study, we choose a wavenumber $w_0 = 7$, as in the earlier studies [TBBDP96, VMD+07]; this choice yields good temporal resolution in the frequency range we consider in this study.

The wavelet transform $X(t,s)$ of an EEG signal $X(t)$ is obtained as:

$$X(t,s) \triangleq \sum_{t'=1}^{K} X(t')\, \psi^*\left(\frac{t'-t}{s}\right), \tag{C.3}$$

where $\psi(t)$ is the Morlet "mother" wavelet (C.1), s is a scaling factor, and $K = f_s T$, with f_s the sampling frequency and T the length of the signal. For the EEG data at hand, we have $T = 20s$ and $f_s = 200\text{Hz}$ and hence $K = 4000$. The scaled and shifted "daughter" wavelet in (C.3) has center frequency $f \triangleq f_0/s$. In the following, we will use the notation $X(t, f)$ instead of $X(t, s)$.

Next we compute the squared magnitude $S(t, f)$ of the coefficients $X(t, f)$:

$$S(t, f) \triangleq |X(t, f)|^2. \tag{C.4}$$

Intuitively speaking, the time-frequency coefficients $S(t, f)$ represents the energy of oscillatory components with frequency f at time instances t. It is noteworthy that $S(t, f)$ contains no information about the phase of that component.

It is well known that EEG signals have very non-flat spectrum with an overall 1/f shape (besides state-dependent peaks at specific frequencies). Therefore, the map $S(t, f)$ contains most energy at low frequencies f. If we directly apply bump modeling to the map $S(t, f)$, most bumps will be located in the low-frequency range, in other words, the high-frequency range would be under-represented. Since relevant information might be contained at high frequency, we normalize the map $S(t, f)$ before extracting the bump models.

Normalization

The coefficients $S(t, f)$ are centered and normalized, resulting in the coefficients $\tilde{Z}(t, f)$:

$$\tilde{Z}(t, f) \triangleq \frac{S(t, f) - m_S(f)}{\sigma_S(f)}, \tag{C.5}$$

where $m_S(f)$ is obtained by averaging $S(t, f)$ over the whole length of the EEG signal:

$$m_S(f) = \frac{1}{K} \sum_{t=1}^{K} S(t, f). \tag{C.6}$$

Likewise, $\sigma_S^2(f)$ is the variance of $S(t, f)$:

$$\sigma_S^2(f) = \frac{1}{K} \sum_{t=1}^{K} \big(S(t, f) - m_S(f)\big)^2. \tag{C.7}$$

In other words: the coefficients $Z(t,f)$ encode fluctuations from the baseline EEG power at time t and frequency f. The normalization (C.5) is known as z-score [BC02]. The coefficients $\tilde{Z}(t,f)$ are positive when the activity at t and f is stronger than the baseline $m_S(f)$ and negative otherwise. In the application of diagnosing AD (see Section 6.5), we concentrate on regions in $\tilde{Z}(t,f)$ with large activity, so-called oscillatory events. For convenience, we shift the coefficients (C.5) in the positive direction before bump modeling by adding a constant α, the remaining negative coefficients are set to zero:

$$Z(t,f) \triangleq \left\lceil \tilde{Z}(t,f) + \alpha \right\rceil^{+} = \left\lceil \frac{S(t,f) - m_S(f)}{\sigma_S(f)} + \alpha \right\rceil^{+}, \tag{C.8}$$

where $\lceil x \rceil^{+} = x$ if $x \geq 0$ and $\lceil x \rceil^{+} = 0$ otherwise. In our experiments (see Section 6.5), we set $\alpha = 3.5$, and as a consequence, virtually all values of $\tilde{Z}(t,f) + \alpha$ are then positive. The top row of Fig. 6-2 shows the normalized wavelet map Z (C.8) of two EEG signals.

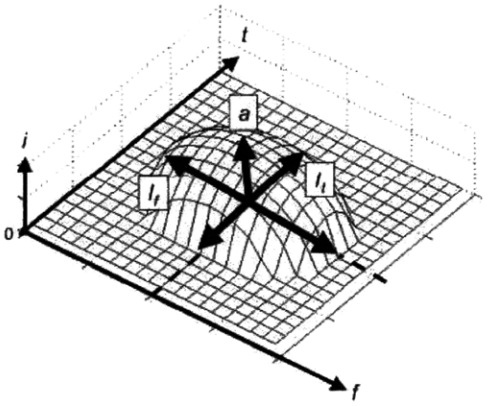
Bump Modeling

Next, bump models are extracted from the coefficient maps Z (see Fig. 6-2 and [VMD+07]). We approximate the map $Z(t,f)$ as a sum $Z_{\text{bump}}(t,f,\theta)$ of a "small" number of smooth basis functions or "bumps" (denoted by f_{bump}):

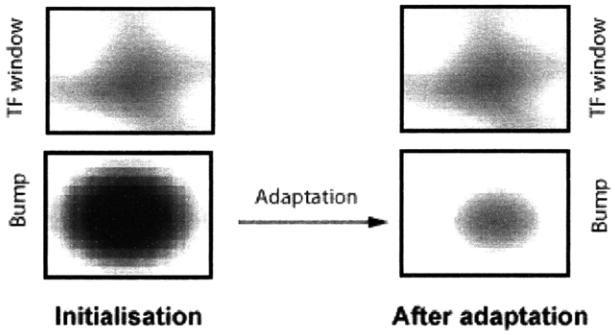
$$Z(t,f) \approx Z_{\text{bump}}(t,f,\theta) \triangleq \sum_{k=1}^{N_b} f_{\text{bump}}(t,f,\theta_k), \tag{C.9}$$

where θ_k are vectors of bump parameters and $\theta \triangleq (\theta_1, \theta_2, \ldots, \theta_{N_b})$. The sparse bump approximation $Z_{\text{bump}}(t,f,\theta)$ represents regions in the time-frequency plane where the EEG contains more power than the baseline; in other words, it captures the most significant oscillatory activities in the EEG signal.

We choose half-ellipsoid bumps since they are well-suited for our purposes [Via05, VMD+07] (see Fig. C-1). Since we wish to keep the number of bump parameters as low as possible, the principal axes of the half ellipsoid bumps are constrained to be parallel to the time-frequency axes. As a result, each bump is described by five parameters (see Fig. C-1(a)): the coordinates of its center (i.e., time x_k and frequency f_k), its amplitude $w_k > 0$, and the extension Δx_k and Δf_k in time and frequency respectively, in other words,



(a) Parameters



(b) Learning the parameters

Figure C-1: Half ellipsoid bump.

$\theta_k = (x_k, f_k, w_k, \Delta x_k, \Delta f_k)$. More precisely, the ellipsoid bump function $f_{\text{bump}}(t, f, \theta_k)$ is defined as:

$$f_{\text{bump}}(t, f, \theta_k) = \begin{cases} w_k \sqrt{1 - \kappa(t, f, \theta_k)} & \text{for } 0 \leq \kappa \leq 1 \\ 0 & \text{for } \kappa > 1, \end{cases} \tag{C.10}$$

where

$$\kappa(t, f, \theta_k) = \frac{(t - x_k)^2}{\Delta^2 x_k} + \frac{(f - f_k)^2}{\Delta^2 f_k}. \tag{C.11}$$

For the EEG data described in Section 6.5.1, the number of bumps N_b (cf. (C.9)) is typically between 50 and 100, and therefore, $Z_{\text{bump}}(t, f, \theta)$ is fully specified by a few hundred parameters. On the other hand, the time-frequency map $Z(t, f)$ consists of between 10^4 and 10^5 coefficients; the bump model $Z_{\text{bump}}(t, f, \theta)$ is thus a sparse (but approximate) representation of $Z(t, f)$.

The bump model $Z_{\text{bump}}(t, f, \theta)$ is extracted from $Z(t, f)$ by the following algorithm [Via05, VMD+07]:

1. Define appropriate boundaries for the map $Z(t, f)$ in order to avoid finite-size effects.

2. Partition the map $Z(t, f)$ into small zones. The size of these zones depends on the time-frequency ratio of the wavelets, and is optimized to model oscillatory activities lasting 4 to 5 oscillation periods. Larger oscillatory patterns are modeled by multiple bumps.

3. Find the zone $\mathcal{Z}$ that contains the most energy.

4. Adapt a bump to that zone; the bump parameters are determined by minimizing the quadratic cost function (see Fig. C-1(b)):

$$\mathcal{E}(\theta_k) \triangleq \sum_{t,f \in \mathcal{Z}} \big(Z(t, f) - f_{\text{bump}}(t, f, \theta_k)\big)^2. \tag{C.12}$$

   Next withdraw the bump from the original map.

5. The fraction of total intensity contained in that bump is computed:

$$F = \frac{\sum_{t,f \in \mathcal{Z}} f_{\text{bump}}(t, f, \theta_k)}{\sum_{t,f \in \mathcal{Z}} Z(t, f)}. \tag{C.13}$$

   If F is smaller than a threshold $G \in \mathbb{R}^+$ for three consecutive bumps, and hence those bumps contain only a small fraction of the energy of map $Z(t, f)$, stop modeling and proceed to (6), otherwise iterate (3).

6. After all signals have been modeled, define a threshold $T \geq G$, and remove the bumps with least energy until $F < T$. This allows us to trade off the information loss and modeling of background noise.

In the present application, we used a threshold $G = 0.05$. With this threshold, each bump model contains many bumps. Some of those bumps may actually model background noise. Therefore, we further pruned the bump models (cf. Step 6). We tested various values of the threshold $T \in [0.05, 0.4]$; the results presented did not depend much on the specific choice of T. We refer to [Via05, VMD+07] for more information on bump modeling. In particular, we used the same choice of boundaries (Step 1) and partitions (Step 2) as in those references.

Eventually, we obtain 21 bump models, i.e., one per EEG channel. In the following, we describe how those models are further processed.

Aggregation

As a next step, we group the 21 electrodes into 5 regions, as illustrated in Fig. C-2. From the 21 bump models obtained by sparsification (cf. Section C), we extract a single bump model for each of the 5 zones by means of the aggregation algorithm described in [VMD+07]. This vastly reduces the computational complexity: instead of computing the bivariate-SES parameters between all possible pairs of 21 electrodes (210 in total), we compute those parameters for all pairs of 5 regions (10 in total). Next, we average the SES parameters over those 10 pairs, resulting in a triplet (ρ, δ_t, s_t) for each subject. On the other hand, we apply multivariate SES to all 5 bump models *simultaneously*.

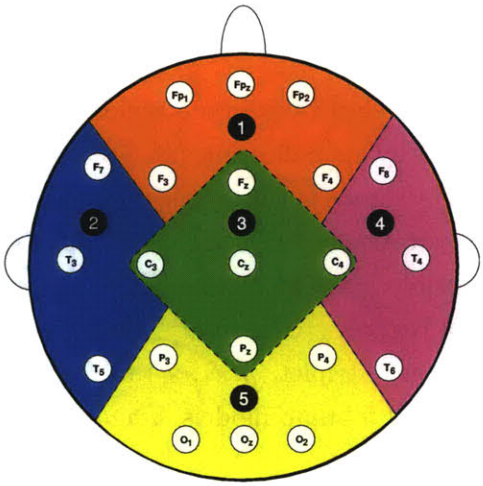


Figure C-2: The 21 electrodes used for EEG recording, distributed according to the 10–20 international placement system [NS06]. The clustering into 5 zones is indicated by the colors and dashed lines (1 = frontal, 2 = left temporal, 3 = central, 4 = right temporal and 5 = occipital).

Appendix D

Complements on multivariate SES and additional graphs

D.1 Computational hardness of multivariate SES

The combinatorial problem (6.43) is very similar to solving a *maximum weighted N-dimensional matching*. For the purpose of understanding the combinatorial hardness of the problem, we show that for $N \geq 5$, the *maximum 3-dimensional matching* problem can be reduced to (6.43) when forgoing the euclidean costs assumptions. Since *maximum 3-dimensional matching* is NP-hard, it results that (6.43) (with general costs) is also NP-hard. Therefore, the extension of SES from 2 time series to more than 2 is far from trivial.

Proposition 22. *The combinatorial problem (15) is NP-hard if* $N \geq 5$.

We include a sketch of the proof; it is based on a reduction from *maximum weighted 3-dimensional matching* optimization, which is known to be NP-hard and APX-hard [Kan91] [GJB$^+$].

Proof. Let $T \subset X \times Y \times Z$, where X, Y, Z are disjoint sets. Let us construct an instance of problem (6.43) which can be used to reconstruct the optimal 3-dimensional matching of T. Since finding maximum cardinality 3-dimensional matching is NP-hard, so is problem (XXX). As a first set, let us create empty sets X', Y', Z', T', U', which will serve as containers for carefully chosen "bumps" (points). In the rest of the proof denote $s_{a,b}$ the cost of assigning bump a to exemplar bump b (i.e., the cost of the variable $b_{a,b}$).

For every $x \in X$ (resp. $y \in Y$, $z \in Z$), create two corresponding bumps $x \in X'$ and $\tilde{x} \in U'$ (resp. $y \in Y'$, $z \in Z'$, $\tilde{y} \in U$, $\tilde{z} \in U'$) and for every $t = (x, y, z) \in T$, create two bumps $t \in T'$ and $\tilde{t} \in U'$.
Then, let us choose the cost function as follows:

- $p_s = 1 - \epsilon$, where ϵ is an extremely small positive constant (practically equal to 0)
- $\lambda = N \exp(1)$
- For any $t = (x, y, z) \in T$, let $s_{x,t} = s_{y,t} = s_{z,t} = 0$. For any bump $b \in X' \cup Y' \cup Z' \cup T'$, let $s_{b,\tilde{b}} = \beta$. For any other two bumps b_1, b_2, let s_{b_1,b_2} be equal to M, where M is a very large positive constant (practically, $+\infty$).

The first two assumptions effectively set α to -1 and β to a very large constant. Since for any bump $u \in U'$, and for any other bump b, $s_{u,b}$ is infinite, bumps in U' can never be assigned to any other bump, and all have to be exemplars.The total cost of bumps in U' being exemplars is therefore an additive constant which does not change the solution. Moreover, for any bump $u \in U'$, there exists a unique bump $b \in X' \cup Y' \cup Z' \cup T'$ that can be assigned to it (i.e., it is the unique bump b such that $s_{b,u} < +\infty$). Denote $f(u)$ the unique bump b of $X' \cup Y' \cup Z' \cup T'$ such that $s_{b,u} < +\infty$. Then, if $f(u)$ is assigned to u, the assignment cost $s_{b,u}$ is equal to β, and since it is the only bump which can be assigned to u, three bumps will be missing, for an extra cost of 3β. The total cost of the cluster is therefore 4β. On the other hand, if $f(u)$ is not assigned to u, the cluster of exemplar u does not contain any other bumps — 4 bumps are missing, for a total cluster cost again equal to 4β. Therefore, the cluster costs of bumps in U' are additive constants, and bumps in $X' \cup Y' \cup Z' \cup T'$ can be assigned to their corresponding exemplar in U' for no extra cost. For this reason, exemplars in U' can be considered as "fake exemplars" (they serve as bins for unmatched bumps in X', Y', Z' and T').
The next step consists in observing all other exemplars have to be in T'. Indeed, for any bump $b_1 \in X' \cup Y' \cup Z'$, and any other bump b_2, s_{b_2,b_1} is infinite. It results that bumps in $X' \cup Y' \cup Z'$ can never be exemplars.
Since $\alpha = -1$, the optimization effectively aims at maximizing the number of exemplars in T'. Finally, because the cost of missing bumps β is very large, all clusters with exemplars in T' have to contain a bump from each time series X', Y', Z'. Let $t = (x, y, z) \in T$. Then the only possible cluster for exemplar $t \in T'$ consists of the corresponding bumps x, y, z in X', Y', Z' (all other assignments bring the cost up to infinity). Finally, since each bump in

X', Y', Z' can only be assigned to one exemplar in T', the clusters differ in each coordinate. It finally follows that the set of real clusters is the maximum 3-dimensional matching of $T \subset X \times Y \times Z$. □

D.2 Extension of multivariate SES to the multinomial prior

First we assume that the parameters θ_x and γ of the multinomial prior are constant. If $p(c_m|\theta_c)$ is a multinomial distribution Mult(γ) with parameter γ, and the prior for γ is a Dirichlet distribution Di(ζ), the expression (6.38) becomes:

$$\begin{aligned} -\log p(\tilde{X}, X, \mathcal{I}, \theta) = &-\log \mathrm{Di}(\gamma;\zeta) - \log p(\theta_x) - \log(1-\lambda\,\mathrm{vol}(S)) + \phi M \\ &- \sum_{m=1}^{M} \log \delta\big(x_{i(m),k(m)} - \tilde{x}_m\big) + g(c_m) \\ &- \sum_{(i,j)\in \log \mathcal{I}_m^{\mathrm{copy}}} \log p_x(x_{i,j}|\tilde{x}_m, \theta_x). \end{aligned} \tag{D.1}$$

where $\phi = -\log \frac{\lambda}{N}$, and the non-linear function g is defined as:

$$g(c_m) = -\log \gamma_m + \log \binom{N-1}{c_m}. \tag{D.2}$$

Next we consider a multinomial prior for C_m, which results in a non-linear objective function. By introducing auxiliary variables, this objective function can be written as a linear function in the resulting augmented parameter space, and the associated combinatorial optimization problem can be formulated as an integer program (see Section D.2).

By substituting (D.1) in (6.43), the conditional maximization (6.43) results in the following combinatorial optimization problem:

$$\begin{aligned} \min_{b} \quad & \phi \sum_{i,1\le k\le n_i} b_{i,k} + \sum_{i',1\le k'\le n_{i'}} b_{i',k'}\ \hat{g}^{(r-1)}\big(N-1-\sum_{i\neq i'} b_{i,i',k'}\big) \\ & - \sum_{i,i',1\le k\le n_i, 1\le k'\le n_{i'}} b_{i,k,i',k'}\ \log p_x(x_{i,k}|x_{i',k'};\hat{\theta}^{(r-1)}) + \tilde{C}, \end{aligned} \tag{D.3}$$

subject to the constraints (6.46) (6.47), where $\tilde{C}$ is an arbitrary constant and and the

non-linear function $g^{(r-1)}$ is defined as:

$$\hat{g}^{(r-1)}(c) = -\log \hat{\gamma}_c^{(r-1)} + \log \binom{N-1}{c}, \tag{D.4}$$

for $c = 0, 1, \ldots, N-1$. Note that the objective function (D.3) is non-linear in b since it involves the non-linear function g. We will now introduce auxiliary variables such that the objective function (D.3) is linear in those variables; we will then reformulate (D.3) as an integer program in the augmented space of variables.

Let us first point out that for an arbitrary function f we can always write:

$$f(x) = \sum_{x' \in \mathcal{X}} f(x')\delta[x - x'], \tag{D.5}$$

with discrete (finite or infinite) set $\mathcal{X}$. By introducing variables $D_{x'}$, we can rewrite (D.5) as:

$$f(x) = \sum_{x' \in \mathcal{X}} f(x') d_{x'}, \tag{D.6}$$

with the constraint $d_{x'} = \delta[x - x']$. The key observation here is that (D.6) is linear in $D_{x'}$.

In this vein, we introduce the binary variables $d_{v,i',k'}$ and rewrite the objective function (D.3) as:

$$\begin{aligned} \min_b \quad & \phi \sum_{i, 1 \le k \le n_i} b_{i,k} + \sum_{v, i', 1 \le k' \le n_{i'}} g_v^{(r-1)}\, d_{v,i',k'} \\ & - \sum_{i, i', 1 \le k \le n_i, 1 \le k' \le n_{i'}} b_{i,k,i',k'} \log p_x(x_{i,k} | x_{i',k'}; \theta) + \tilde{C}, \end{aligned} \tag{D.7}$$

where $g_v^{(r-1)} = g^{(r-1)}\big(N - 1 - v\big)$. This alternative formulation is equivalent to the original expression (D.3) iff $d_{v,i',k'}$ equals one if both $v = \sum_{i \neq i'} b_{i,i',k'}$ and $b_{i',k'} = 1$, and is zero

otherwise. We express those constraints on $d_{v,i',k'}$ as follows:

$$v - \sum_{i \neq i'} b_{i,i',k'} \leq a_{v,i',k'}, \tag{D.8}$$

$$\sum_{i \neq i'} b_{i,i',k'} - v \leq a_{v,i',k'}, \tag{D.9}$$

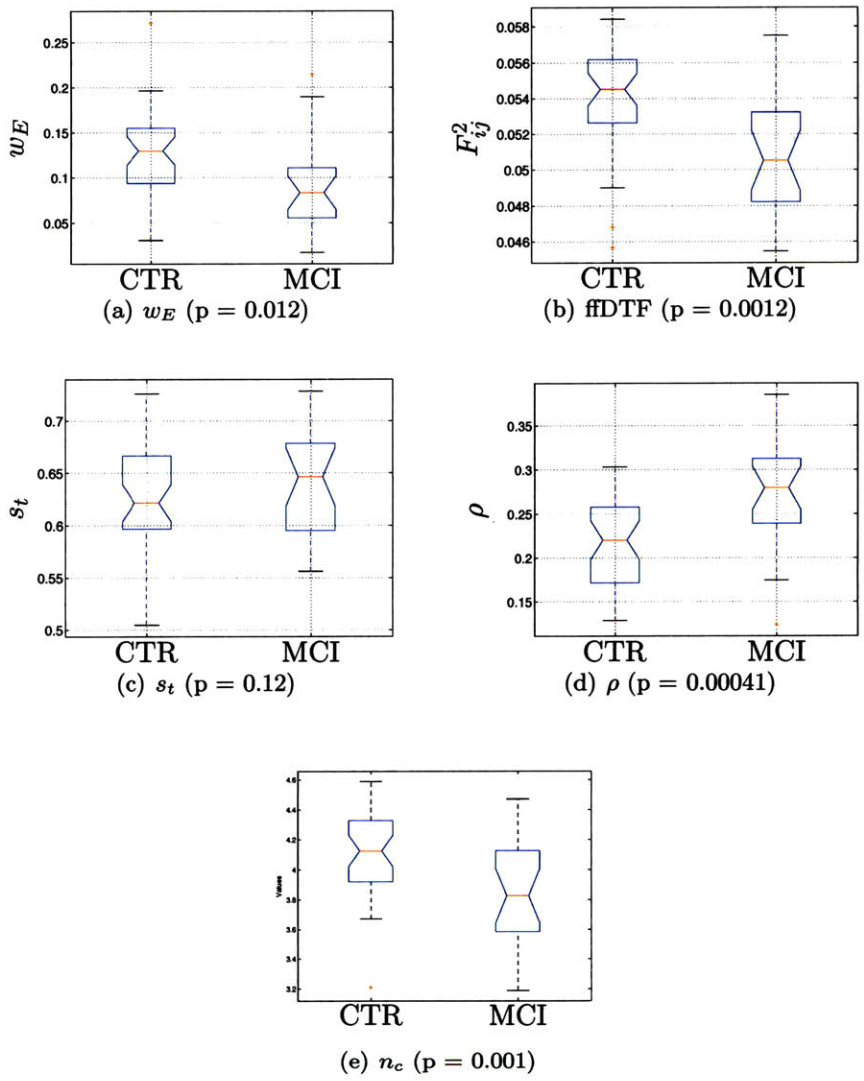
$$a_{v,i',k'} \leq N\,(1 - d_{v,i',k'}), \tag{D.10}$$

$$\sum_{v} d_{v,i',k'} = b_{i',k'}, \tag{D.11}$$

where $a_{v,i',k'}$ are additional auxiliary binary variables. The first two constraints encode that $a_{v,i',k'} \geq |v - \sum_{i \neq i'} b_{i,i',k'}|$; note that as a consequence, $a_{v,i',k'}$ is non-negative. If $v \neq \sum_{i \neq i'} b_{i,i',k'}$, the variable $a_{v,i',k'}$ is strictly positive, and from the third inequality it follows that $d_{v,i',k'}$ equals zero. On the other hand, if $v = \sum_{i \neq i'} b_{i,i',k'}$, the first two constraints no longer force $a_{v,i',k'}$ to be non-zero, and they do not impose any constraint on $d_{v,i',k'}$. However, from the fourth constraint it follows that if $b_{i',k'} = 1$ and hence if (i', k') is an exemplar, one of the $d_{v,i',k'}$ (with fixed i' and k') is equal to one. By setting $d_{v,i',k'}$ equal to one if $v = \sum_{i \neq i'} b_{i,i',k'}$ and zero otherwise, one fulfills then all four constraints. If $b_{i',k'} = 0$ and hence if (i', k') is not an exemplar, all $d_{v,i',k'}$ (with fixed i' and k') are equal to zero. By setting all $d_{v,i',k'}$ equal to zero, all four constraints are then fulfilled.

In summary: the non-linear combinatorial optimization problem with objective (D.3) and constraints (6.46) (6.47) is equivalent to the integer program with objective (D.7) and constraints (6.46) (6.47) combined with (D.8)–(D.11).

D.3 Extra figures



(a) w_E (p = 0.012)

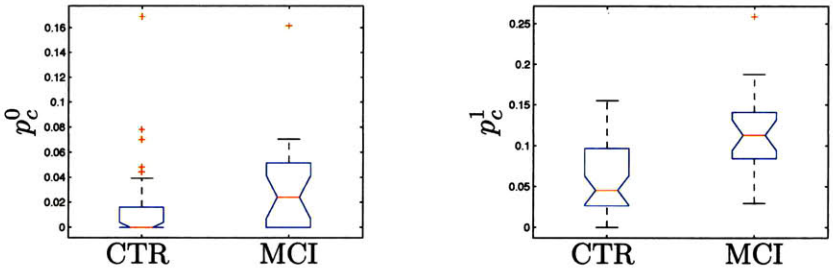
(b) ffDTF (p = 0.0012)

(c) s_t (p = 0.12)

(d) ρ (p = 0.00041)

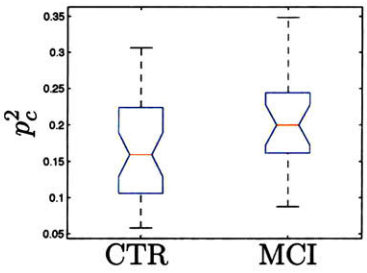
(e) n_c (p = 0.001)

Figure D-1: Box plots for the most discriminant classical measures (wave-entropy w_E and full-frequency directed transfer function (ffDTF)) and the bivariate-SES parameters s_t and ρ.

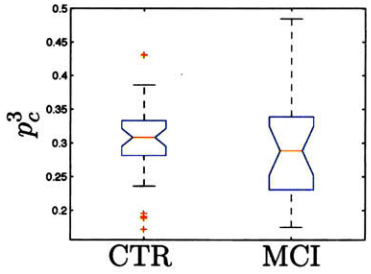


(a) p_1^c (p = 0.016)

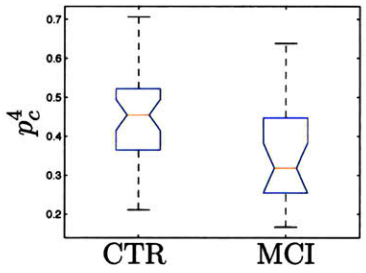
(b) p_2^c (p = 0.0.00029)



(c) p_3^c (p = 0.089)



(d) p_4^c (p = 0.59)



(e) p_c^5 (p = 0.0054)

Figure D-2: Box plots for the multivariate-SES parameters p_c^i, the fraction of clusters with i bumps.

Bibliography

[AA03] E. Arnaiz and O. Almkvist, *Neuropsychological features of mild cognitive impairment and preclinical Alzheimer's disease*, Acta Neurologica Scandinavica **107** (2003), no. s179, 34–41.

[ABMV93] M. Abeles, H. Bergman, E. Margalit, and E. Vaadia, *Spatiotemporal firing patterns in the frontal cortex of behaving monkeys*, Journal of Neurophysiology **70** (1993), no. 4, 1629–1638.

[AFR72] B. Alder, S. Fernbach, and M. Rotenberg, *Seismology: Surface Waves and Earth Oscillations*, Methods in Computational Physics **11** (1972).

[Ald92] D. Aldous, *Asymptotics in the random assignment problem*, Probability Theory and Related Fields **93** (1992), no. 4, 507–534.

[Ald01] ———, *The $\zeta(2)$ limit in the random assignment problem*, Random Structures and Algorithms **18** (2001), 381–418.

[ANWS03] S. Amari, H. Nakahara, S. Wu, and Y. Sakai, *Synchronous firing and higher-order interactions in neuron pool*, Neural computation **15** (2003), no. 1, 127–142.

[AS03] D. Aldous and J. M. Steele, *The objective method: Probabilistic combinatorial optimization and local weak convergence*, Discrete Combinatorial Probability, H. Kesten Ed., Springer-Verlag, 2003.

[AS04] D. Aldous and J. Steele, *The objective method: probabilistic combinatorial optimization and local weak convergence*, Probability on Discrete Structures **110** (2004), 1–72.

[Avi05] S. Aviyente, *A measure of mutual information on the time-frequency plane*, IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05), vol. 4, 2005.

[Bat89] G. Battail, *Coding for the Gaussian channel- The promise of weighted-output decoding*, International journal of satellite communications **7** (1989), 183–192.

[BB72] U. Bertele and F. Brioschi, *Nonserial dynamic programming*, Academic Pr, 1972.

[BBCZ07] M. Bayati, C. Borgs, J. Chayes, and R. Zecchina, *Belief-Propagation for Weighted b-Matchings on Arbitrary Graphs and its Relation to Linear Programs with Integer Solutions*, Arxiv preprint arXiv:0709.1190 (2007).

[BBCZ08] ———, *On the exactness of the cavity method for weighted b-matchings on arbitrary graphs and its relation to linear programs*, Journal of Statistical Mechanics: Theory and Experiment **6001** (2008), 1–10.

[BC02] M. Browne and TRH Cutmore, *Low-probability event-detection and separation via statistical wavelet thresholding: an application to psychophysiological denoising*, Clinical Neurophysiology **113** (2002), no. 9, 1403–1411.

[BFH09] F. Brandt, F. Fischer, and M. Holzer, *Symmetries and the complexity of pure Nash equilibrium*, Journal of Computer and System Sciences **75** (2009), no. 3, 163–177.

[BG06] A. Bandyopadhyay and D. Gamarnik, *Counting without sampling: new algorithms for enumeration problems using statistical physics*, Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm, 2006, pp. 890–899.

[BGK+07] M. Bayati, D. Gamarnik, D. Katz, C. Nair, and P. Tetali, *Simple deterministic approximation algorithms for counting matchings*, Proc. of the 39th annual ACM Symposium on Theory of computing, 2007, pp. 122–127.

[BGT93] C. Berrou, A. Glavieux, and P. Thitimajshima, *Near Shannon limit error-correcting coding and decoding: Turbo-codes.*, IEEE International Conference on Communications, 1993. ICC 93. Geneva. Technical Program, Conference Record, vol. 2, 1993.

[BJZGA07] R. Brookmeyer, E. Johnson, K. Ziegler-Graham, and H.M. Arrighi, *Forecasting the global burden of Alzheimer's disease*, Alzheimer's & Dementia: The Journal of the Alzheimer's Association **3** (2007), no. 3, 186–191.

[BK98] P. Berman and M. Karpinski, *On some tighter inapproximability results*, Tech. report, Electronic Colloquium on Computational Complexity, 1998.

[BLT+08] C. Ballard, M.M. Lana, M. Theodoulou, S. Douglas, R. McShane, R. Jacoby, K. Kossakowski, L.M. Yu, E. Juszczak, et al., *A randomised, blinded, placebo-controlled trial in dementia patients continuing or stopping neuroleptics (the DART-AD trial)*, PLoS Medicine **5** (2008), no. 4.

[BMZ05] A. Braunstein, M. Mezard, and R. Zecchina, *Survey propagation: An algorithm for satisfiability*, Random Structures and Algorithms **27** (2005), no. 2, 201–226.

[Bod06] H.L. Bodlaender, *Treewidth: Characterizations, applications, and computations*, Lecture Notes in Computer Science **4271** (2006), 1–14.

[Bon36] C.E. Bonferroni, *Teoria statistica delle classi e calcolo delle probabilita*, Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze **8** (1936), no. 3.

[BSK06] B. Blum, C.R. Shelton, and D. Koller, *A continuation method for Nash equilibria in structured games*, Journal of Artificial Intelligence Research **25** (2006), 457–502.

[BSS05] M. Bayati, D. Shah, and M. Sharma, *Maximum weight matching via max-product belief propagation*, Proceedings of the 2005 International Symposium on Information Theory, 2005, pp. 1763–1767.

[BSS08] ———, *Max-Product for Maximum Weight Matching: Convergence, Correctness, and LP Duality*, IEEE Transactions on Information Theory **54** (2008), no. 3, 1241–1251.

[BT00] V.D. Blondel and J.N. Tsitsiklis, *A survey of computational complexity results in systems and control*, Automatica-Kidlington **36** (2000), no. 9, 1249–1274.

[BVV07] I. Barany, S. Vempala, and A. Vetta, *Nash equilibria in random games*, Random Structures and Algorithms **31** (2007), no. 4, 391–405.

[CC06a] M. Chertkov and V.Y. Chernyak, *Loop calculus in statistical physics and information science*, Phys Rev E **73** (2006), no. 6, 65102.

[CC06b] ———, *Loop series for discrete statistical models on graphs*, Journal of Statistical Mechanics: Theory and Experiment **6** (2006), P06009.

[CDM08] A. Dimakis C. Daskalakis and E. Mossel, *Connectivity and Equilibrium in Random Games*, Arxiv preprint math/0703902 (2008).

[CFC+02a] J.L. Cummings, J.C. Frank, D. Cherry, N.D. Kohatsu, B. Kemp, L. Hewett, and B. Mittman, *Guidelines for managing Alzheimer's disease: part I. Assessment*, American Family Physician **65** (2002), no. 11, 2263–2276.

[CFC+02b] ———, *Guidelines for managing Alzheimer's disease: Part II. Treatment*, American family physician(1970) **65** (2002), no. 12, 2525–2534.

[Che08] M. Chertkov, *Exactness of belief propagation for some graphical models with loops*, Journal of Statistical Mechanics: Theory and Experiment **2008** (2008), P10016.

[CJW08] V. Chandrasekaran, J. Johnson, and A. Willsky, *Estimation in Gaussian Graphical Models Using Tractable Subgraphs: A Walk-Sum Analysis*, IEEE Transactions on Signal Processing **56** (2008), no. 5, 1916–1930.

[CKIDF05] C. Carmeli, M.G. Knyazeva, G.M. Innocenti, and O. De Feo, *Assessment of EEG synchronization based on state-space analysis*, Neuroimage **25** (2005), no. 2, 339–354.

[CNM+07] R.M. Chapman, G.H. Nowlis, J.W. McCrary, J.A. Chapman, T.C. Sandoval, M.D. Guillily, M.N. Gardner, and L.A. Reilly, *Brain event-related potentials: Diagnosing early-stage Alzheimer's disease*, Neurobiology of aging **28** (2007), no. 2, 194–201.

[Coo90] G.F. Cooper, *Research Note The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks*, Artificial intelligence **42** (1990), 393–405.

[CRVRL06] R. Cogill, M. Rotkowitz, B. Van Roy, and S. Lall, *An approximate dynamic programming approach to decentralized control of stochastic systems*, Lecture notes in control and information sciences **329** (2006), 243.

[CS02] V. Conitzer and T. Sandholm, *Complexity of mechanism design*, Arxiv preprint cs/0205075 (2002).

[CS03] ———, *Complexity results about Nash equilibria*, Proceedings of the 18th International Joint Conference on Artificial intelligence, 2003, pp. 765–771.

[CS08] ———, *New complexity results about Nash equilibria*, Games and Economic Behavior **63** (2008), no. 2, 621–641.

[CS09] S. Chien and A. Sinclair, *Convergence to approximate nash equilibria in congestion games*, Games and Economic Behavior (2009), in press.

[CSH08] V. Chandrasekaran, N. Srebro, and P. Harsha, *Complexity of inference in graphical models*, UAI, vol. 8, Citeseer, 2008, pp. 70–78.

[CSM+05] A. Cichocki, S.L. Shishkin, T. Musha, Z. Leonowicz, T. Asada, and T. Kurachi, *EEG filtering based on blind source separation (BSS) for early detection of Alzheimer's disease*, Clinical Neurophysiology **116** (2005), no. 3, 729–737.

[Daw92] AP Dawid, *Applications of a general propagation algorithm for probabilistic expert systems*, Statistics and Computing **2** (1992), no. 1, 25–36.

[DEG+92] N. Delprat, B. Escudie, P. Guillemain, R. Kronland-Martinet, P. Tchamitchian, B. Torresani, and M. LMA-IM, *Asymptotic wavelet and Gabor analysis: extraction of instantaneousfrequencies*, IEEE Transactions on Information Theory **38** (1992), no. 2 Part 2, 644–664.

[DFP06] C. Daskalakis, A. Fabrikant, and C. Papadimitriou, *The Game World Is Flat: The Complexity of Nash Equilibria in Succinct Games*, Lecture notes in Computer Science **4051** (2006), 513.

[DGP09] C. Daskalakis, P.W. Goldberg, and C.H. Papadimitriou, *The complexity of computing a Nash equilibrium*, SIAM Journal of Computing (2009).

[DGS07] B. Dilkina, C.P. Gomes, and A. Sabharwal, *The impact of network topology on pure Nash equilibria in graphical games*, Proceedings of the 22nd AAAI Conference On Artificial Intelligence, vol. 22, 2007, p. 42.

[DMP07] C. Daskalakis, A. Mehta, and C. Papadimitriou, *Progress in approximate nash equilibria*, Proceedings of the 8th ACM conference on Electronic commerce, 2007, pp. 355–358.

[DMP09] C. Daskalakis, A. Mehta, and C. Papadimitriou, *A note on approximate Nash equilibria*, Theoretical Computer Science **410** (2009), no. 17, 1581–1588.

[DMU04] C. Di, A. Montanari, and R. Urbanke, *Weight distributions of LDPC code ensembles: combinatorics meets statistical physics*, Proceedings of the 2004 International Symposium on Information Theory, 2004.

[Dob68a] R.L. Dobrushin, *The description of the random field by its conditional distributions and its regularity conditions*, Teoriya Veroyatnostei i ee Primeneniya **13** (1968), no. 2, 201–229.

[Dob68b] RL Dobrushin, *The problem of uniqueness of a Gibbsian random field and the problem of phase transitions*, Functional Analysis and its Applications **2** (1968), no. 4, 302–312.

[DP06] C. Daskalakis and C. Papadimitriou, *Computing pure nash equilibria in graphical games via markov random fields*, Proceedings of the 7th ACM conference on Electronic commerce, 2006, pp. 91–99.

[Dud99] R.M. Dudley, *Uniform central limit theorems*, Cambridge university press, 1999.

[DVMC] J. Dauwels, F. Vialatte, T. Musha, and A. Cichocki, *A comparative study of synchrony measures for the early diagnosis of Alzheimer's disease based on EEG*, Neuroimage **49**, no. 1, 668–693.

[DVW+09] J. Dauwels, F. Vialatte, T. Weber, T. Musha, and A. Cichocki, *Quantifying Statistical Interdependence by Message Passing on Graphs—Part II: Multidimensional Point Processes*, Neural computation **21** (2009), no. 8, 2203–2268.

[DVWC09] J. Dauwels, F. Vialatte, T. Weber, and A. Cichocki, *Quantifying Statistical Interdependence by Message Passing on Graphs—Part I: One-Dimensional Point Processes*, Neural Computation **21** (2009), no. 8, 2152–2202.

[Edm69] J. Edmonds, *Theoretical improvements in algorithmic efficiency for network flow problems*, International Conference on Combinatorial Structures and Their Applications, 1969.

[EGG06] E. Elkind, L.A. Goldberg, and P. Goldberg, *Nash equilibria in graphical games on trees revisited*, Proceedings of the 7th ACM conference on Electronic commerce, ACM, 2006, p. 109.

[EGG07] E. Elkind, L.A. Golberg, and P.W. Goldberg, *Computing good Nash equilibria in graphical games*, Proceedings of the 8th ACM conference on Electronic commerce, ACM, 2007, p. 171.

[FD06] B. Frey and D. Dueck, *Mixture modeling by affinity propagation*, Advances in Neural Information Processing Systems **18** (2006), 379.

[FD07] B.J. Frey and D. Dueck, *Clustering by passing messages between data points*, Science **315** (2007), no. 5814, 972.

[FJ70] G. Forney Jr, *Convolutional codes I: Algebraic structure*, IEEE Transactions on Information Theory **16** (1970), no. 6, 720–738.

[FM98] B.J. Frey and D.J.C. MacKay, *A revolution: Belief propagation in graphs with cycles*, Advances in Neural Information Processing Systems, 1998.

[FT91] D. Fudenberg and J. Tirole, *Game theory*, MIT Press Books **1** (1991).

[Gal60] R.G. Gallager, *Low density parity-check codes*, Ph.D. thesis, MIT, 1960.

[Gal63] RG Gallager, *Low density parity check codes. Number 21 in Research monograph series*, 1963.

[GDKV03] C. Guestrin, R. Parr D. Koller, and S. Venkataraman, *Efficient solution algorithms for factored MDPs*, Journal of Artificial Intelligence Research **19** (2003), no. 10, 399–468.

[Geo88] H. O. Georgii, *Gibbs measures and phase transitions*, de Gruyter Studies in Mathematics 9, Walter de Gruyter & Co., Berlin, 1988.

[Ger95] AMH Gerards, *Matching. Volume 7 of¡ i¿ Handbooks in Operations Research and Management Science¡/i¿, Chapter 3*, 1995.

[GG09] D. Gamarnik and D.A. Goldberg, *Randomized greedy algorithms for independent sets and matchings in regular graphs: Exact results and finite girth corrections*, Combinatorics, Probability and Computing (2009), 1–25.

[GGM84] P. Goupillaud, A. Grossmann, and J. Morlet, *Cycle-octave and related transforms in seismic signal analysis*, Geoexploration **23** (1984), no. 1, 85–102.

[GGS05] G. Gottlob, G. Greco, and F. Scarcello, *Pure Nash equilibria: Hard and easy games*, Journal of Artificial Intelligence Research **24** (2005), no. 195-220, 26–37.

[GJ07] A. Globerson and T. Jaakkola, *Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations*, Advances in Neural Information Processing Systems **21** (2007).

[GJB+] MR Garey, D.S. Johnson, R.C. Backhouse, G. von Bochmann, D. Harel, CJ van Rijsbergen, J.E. Hopcroft, J.D. Ullman, A.W. Marshall, I. Olkin, et al., *Computers and Intractability: A Guide to the Theory of*, Springer.

[GK07a] D. Gamarnik and D. Katz, *A Deterministic Approximation Algorithm for Computing a Permanent of a 0, 1 matrix*, Arxiv preprint math.CO/0702039 (2007).

[GK07b] ———, *Correlation decay and deterministic FPTAS for counting list-colorings of a graph*, Proc. of the 18th annual ACM-SIAM Symposium On Discrete Algorithms, 2007, pp. 1245–1254.

[GNS06] D. Gamarnik, T. Nowicki, and G. Swirscsz, *Maximum weight independent sets and matchings in sparse random graphs. Exact results using the local weak convergence method*, Random Structures and Algorithms **28** (2006), no. 1, 76–106.

[GS04] G. Greco and F. Scarcello, *Constrained pure Nash equilibria in graphical games*, Proceedings of the 16th Eureopean Conference on Artificial Intelligence, 2004, p. 181.

[GSW09] D. Gamarnik, D. Shah, and Y. Wei, *Belief propagation for min-cost network flow: Convergence and correctness*, To appear in proceedings of the 2010 ACM-SIAM symposium on Discrete algorithm, 2009.

[Has96] J. Hastad, *Clique is hard to approximate within n*, Proceedings of the 37th annual Symposium on Foundations of Computer Science, 1996, pp. 627–636.

[HH89] J. Hagenauer and P. Hoeher, *A Viterbi algorithm with soft-decision outputs and its applications*, Proc. IEEE Globecom, vol. 89, 1989, pp. 1680–1686.

[HJ07] B. Huang and T. Jebara, *Loopy belief propagation for bipartite maximum weight b-matching*, Artificial Intelligence and Statistics (AISTATS) (2007).

[Hoc88] Y. Hochberg, *A sharper Bonferroni procedure for multiple tests of significance*, 1988, pp. 800–802.

[Hoc97] D. Hochbaum, *Approximation algorithms for NP-hard problems*, WS Publishing Company, Boston, MA, 1997.

[Hol79] S. Holm, *A simple sequentially rejective multiple test procedure*, Scandinavian Journal of Statistics (1979), 65–70.

[HSK+03] M.J. Hogan, G.R.J. Swanwick, J. Kaiser, M. Rowan, and B. Lawlor, *Memory-related EEG power and coherence reductions in mild Alzheimer's disease*, International Journal of Psychophysiology **49** (2003), no. 2, 147–163.

[HW05] A. Hartmann and M. Weigt, *Phase transitions in combinatorial optimization problems: basics, algorithms and statistical mechanics*, Vch Verlagsgesellschaft Mbh, 2005.

[IFW06] A.T. Ihler, JW Fisher, and A.S. Willsky, *Loopy belief propagation: Convergence and effects of message errors*, Journal of Machine Learning Research **6** (2006), no. 1, 905.

[Isi24] E. Ising, *Beitrag zur Theorie des Ferro-und Paramagnetismus (Thesis, Hamburg, 1924)*, 1924.

[Jeo04] J. Jeong, *EEG dynamics in patients with Alzheimer's disease*, Clinical Neurophysiology **115** (2004), no. 7, 1490–1505.

[Jer03] M. Jerrum, *Counting, sampling and integrating: algorithms and complexity.*

[JLB07] A.X. Jiang and K. Leyton-Brown, *Computing pure nash equilibria in symmetric action graph games*, Proceedings of the 22nd AAAI national conference on Artificial intelligence, vol. 22, 2007, p. 79.

[JMW06] J. Johnson, D. Malioutov, and A. Willsky, *Walk-Sum Interpretation and Analysis of Gaussian Belief Propagation*, Advances in Neural Information Processing Systems **18** (2006), 579–586.

[Jor98] M. Jordan, *Learning in Graphical Models*, Kluwer Academic Publishers, 1998.

[Jor04] ———, *Graphical models*, Statistical Science (Special Issue on Bayesian Statistics) **19** (2004), 140–155.

[JS97] M. Jerrum and A. Sinclair, *The Markov chain Monte Carlo method: an approach to approximate counting and integration*, Approximation algorithms for NP-hard problems (D. Hochbaum, ed.), PWS Publishing Company, Boston, MA, 1997.

[JS07] K. Jung and D. Shah, *Inference in Binary Pair-wise Markov Random Fields through Self-Avoiding Walks*, Proceedings of Allerton Conference on Computation, Communication and Control, 2007, p. 8.

[JVV86] M. Jerrum, L. Valiant, and V. Vazirani, *Random generation of combinatorial structures from a uniform distribution*, Theoret. Comput. Sci. **43** (1986), no. 2-3, 169ñ188.

[Kan91] V. Kann, *Maximum bounded 3-dimensional matching in max snp-complete*, Information Processing Letters **37** (1991), no. 1, 27–35.

[KC00] LH Kantha and C.A. Clayson, *Numerical models of oceans and oceanic processes*, Academic Press, 2000.

[KKLO03] S. Kakade, M. Kearns, J. Langford, and L. Ortiz, *Correlated equilibria in graphical games*, Proceedings of the 4th ACM conference on Electronic commerce, 2003, pp. 42–47.

[KL05] M. Kaminski and H. Liang, *Causal influence: advances in neurosignal analysis.*, Critical reviews in biomedical engineering **33** (2005), no. 4, 347.

[KLS01a] M. Kearns, M. Littman, and S. Singh, *Graphical models for game theory*, Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence, 2001, pp. 253–260.

[KLS+01b] T. Koenig, D. Lehmann, N. Saito, T. Kuginuki, T. Kinoshita, and M. Koukkou, *Decreased functional connectivity of EEG theta-frequency activity in first-episode, neuroleptic-naive patients with schizophrenia: preliminary results*, Schizophrenia research **50** (2001), no. 1-2, 55–60.

[KM03] D. Koller and B. Milch, *Multi-agent influence diagrams for representing and solving games*, Games and Economic Behavior **45** (2003), no. 1, 181–221.

[Kol06] V. Kolmogorov, *Convergent tree-reweighted message passing for energy minimization*, IEEE Transactions on Pattern Analysis and Machine Intelligence **28** (2006), no. 10, 1568.

[KP00] D. Koller and R. Parr, *Policy Iteration for Factored MDPs*, Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence table of contents, 2000, pp. 326–334.

[KSG04] A. Kraskov, H. Stoegbauer, and P. Grassberger, *Estimating mutual information*, Physical Review E **69** (2004), no. 6, 66138.

[KW05] V. Kolmogorov and M. Wainwright, *On the optimality of tree-reweighted max-product message passing*, Uncertainty in Artificial Intelligence, Citeseer, 2005.

[Lau96] S.L. Lauritzen, *Graphical models*, Oxford University Press, USA, 1996.

[LDH+07] H.A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, FR Kschischang, and Z. ETH, *The factor graph approach to model-based signal processing*, Proceedings of the IEEE **95** (2007), no. 6, 1295–1322.

[LETB04] H.A. Loeliger, AG Endora Tech, and S. Basel, *An introduction to factor graphs*, IEEE Signal Processing Magazine **21** (2004), no. 1, 28–41.

[LG07] D. Lashkari and P. Golland, *Convex clustering with exemplar-based models*, Advances in Neural Information Processing Systems (2007).

[LKBJ08] T. Lin, N. Kaminski, and Z. Bar-Joseph, *Alignment and classification of time series gene expression in clinical studies*, Bioinformatics **24** (2008), no. 13, i147.

[LKS02] M.L. Littman, M. Kearns, and S. Singh, *An efficient exact algorithm for singly connected graphical games*, Advances in Neural Information Processing Systems, 2002.

[LM00] P. La Mura, *Game networks*, Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, vol. 342, 2000.

[LMM07] Y. Lu, C. Méasson, and A. Montanari, *TP decoding*, Arxiv preprint arXiv:0710.0564 (2007).

[LNRE05] J. Listgarten, R.M. Neal, S.T. Roweis, and A. Emili, *Multiple alignment of continuous time series*, Advances in Neural Information Processing Systems **17** (2005), 817–824.

[LP98] F. Liu and R.W. Picard, *Finding periodicity in space and time*, International Conference on Computer Vision, vol. 1, Citeseer, 1998, pp. 376–383.

[LRMV99] J.P. Lachaux, E. Rodriguez, J. Martinerie, and F.J. Varela, *Measuring phase synchrony in brain signals*, Human Brain Mapping **8** (1999), no. 4, 194–208.

[LV97] M. Luby and E. Vigoda, *Approximately counting up to four*, Proceedings of the 29d Annual ACM Symposium on the Theory of Computing (1997), 682–687.

[Mar55] J. Marschak, *Elements for a theory of teams*, Management Science **1** (1955), no. 2, 127–137.

[Mat01] H. Matsuda, *Cerebral blood flow and metabolic abnormalities in Alzheimer's disease*, Annals of Nuclear Medicine **15** (2001), no. 2, 85–92.

[Mat04] M.P. Mattson, *Pathways towards and away from Alzheimer's disease*, Nature **430** (2004), no. 7000, 631–639.

[MAY+02] T. Musha, T. Asada, F. Yamashita, T. Kinoshita, Z. Chen, H. Matsuda, M. Uno, and W.R. Shankle, *A new EEG method for estimating cortical neuronal impairment that is sensitive to early stage Alzheimer's disease*, Clinical Neurophysiology **113** (2002), no. 7, 1052–1058.

[MJW06] D. Malioutov, J.Johnson, and A. Willsky, *Walk-sums and belief propagation in gaussian graphical models*, Journal of Machine Learning Research **7** (2006), 2031–2064.

[MK05] J.M. Mooij and H.J. Kappen, *Sufficient conditions for convergence of loopy belief propagation*, Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence, Citeseer, 2005, pp. 396–403.

[MM96] R.D. McKelvey and A. McLennan, *Computation of equilibria in finite games*, Handbook of computational economics **1** (1996), 87–142.

[MM08] M. Mezard and A. Montanari, *Information, physics and computation*, Oxford: Oxford University Press, 2008.

[MMS] PD Meek, K. McKeithan, and GT Schumock, *Economic considerations in Alzheimer's disease.*, Pharmacotherapy **18**, no. 2 Pt 2, 68.

[MMW07] E. Maneva, E. Mossel, and M.J. Wainwright, *A new look at survey propagation and its generalizations*, Journal of the ACM **54** (2007), no. 4, 1–41.

[Moo08] J.M. Mooij, *Understanding and improving belief propagation.*

[MPV87] M. Mezard, G. Parisi, and M. A. Virasoro, *Spin-glass theory and beyond,* vol 9 of *Lecture Notes in Physics*, World Scientific, Singapore, 1987.

[MR72] J. Marschak and R. Radner, *Economic theory of teams*, Yale Univ. Press, 1972.

[MR07] C. Moallemi and B. Van Roy, *Convergence of the min-sum algorithm for convex optimization*, Arxiv preprint arXiv:0705.4253 (2007).

[MR09] ———, *Convergence of min-sum message passing for quadratic optimization*, IEEE Transactions on Information Theory **55** (2009), no. 5, 2413–2423.

[MS09] A. Montanari and A. Saberi, *Convergence to equilibrium in local interaction games*, ACM SIGecom Exchanges **8** (2009), no. 1, 11.

[MU07] A. Montanari and R. Urbanke, *Modern coding theory: The statistical mechanics and computer science point of view*, preprint (2007).

[MWKR07] J. Mooij, B. Wemmenhove, H. Kappen, and T. Rizzo, *Loop corrected belief propagation*, Proc. of the 11th International Conference on Artificial Intelligence and Statistics, vol. 11, 2007.

[Nas50] J. Nash, *Nash. Equilibrium points in n-person games*, Proceedings of the National Academy of Sciences **36** (1950), 48–49.

[Nas51] ———, *Non-cooperative games*, The Annals of Mathematics **54** (1951), no. 2, 286–295.

[NS06] P.L. Nunez and R. Srinivasan, *Electric fields of the brain: the neurophysics of EEG*, Oxford University Press, USA, 2006.

[OK03] L. Ortiz and M. Kearns, *Nash Propagation for Loopy Graphical Games*, Advances in Neural Information Processing Systems (2003), 817–824.

[Ons39] L. Onsager, *Electrostatic Interaction of Molecules.*, Journal of Physical Chemistry **43** (1939), no. 2, 189–196.

[Ons44] ———, *Crystal statistics. I. A two-dimensional model with an order-disorder transition*, Physical Review **65** (1944), no. 3-4, 117–149.

[PBM+07] K. Palmer, AK Berger, R. Monastero, B. Winblad, L. Backman, and L. Fratiglioni, *Predictors of progression from mild cognitive impairment to Alzheimer disease*, Neurology **68** (2007), no. 19, 1596.

[Pea82] J. Pearl, *Reverend Bayes on inference engines: A distributed hierarchical approach*, Proceedings of the National Conference on Artificial Intelligence, 1982, pp. 133–136.

[Pea00] ———, *Causality: models, reasoning, and inference*, Cambridge Univ Pr, 2000.

[PS88] J. Pearl and G. Shafer, *Probabilistic reasoning in intelligent systems: networks of plausible inference*, JSTOR, 1988.

[PT87] C.H. Papadimitriou and J.N. Tsitsiklis, *The complexity of Markov decision processes*, Mathematics of operations research **12** (1987), no. 3, 441–450.

[Pul] WR Pulleyblank, *Matchings and Extension. 179–232*, Handbook of Combinatorics ed, 0–444.

[QQKKG02] R. Quian Quiroga, A. Kraskov, T. Kreuz, and P. Grassberger, *Performance of different synchronization measures in real data: A case study on electroencephalographic signals*, Physical Review E **65** (2002), no. 4, 41903.

[Rad62] R. Radner, *Team decision problems*, The Annals of Mathematical Statistics **33** (1962), no. 3, 857–881.

[RCB+02] M.G. Rosenblum, L. Cimponeriu, A. Bezerianos, A. Patzak, and R. Mrowka, *Identification of coupling direction: Application to cardiorespiratory interaction*, Physical Review E **65** (2002), no. 4, 41909.

[Rot96] D. Roth, *On the hardness of approximate reasoning*, Artificial Intelligence **82** (1996), no. 1-2, 273–302.

[RR01] P. Rusmevichientong and B. Van Roy, *An analysis of belief propagation on the turbo decoding graph withGaussian densities*, IEEE Transactions on Information Theory **47** (2001), no. 2, 745–765.

[RR03] , *Decentralized decision-making in a large team with local information*, Games and Economic Behavior **43** (2003), no. 2, 266–295.

[RS00] Y. Rinott and M. Scarsini, *On the number of pure strategy Nash equilibria in random games*, Games and Economic Behavior **33** (2000), no. 2, 274–293.

[RU08] T. Richardson and R. Urbanke, *Modern coding theory*, Cambridge University Press, 2008.

[San07] S. Sanghavi, *Equivalence of LP Relaxation and Max-Product for Weighted Matching in General Graphs*, Information Theory Workshop, 2007, 2007, pp. 242–247.

[SGJ08] D. Sontag, A. Globerson, and T. Jaakkola, *Clusters and Coarse Partitions in LP Relaxations*, Advances in Neural Information Processing Systems, 2008.

[SGR07] B.J. Small, E. Gagnon, and B. Robinson, *Early identification of cognitive deficits: Preclinical Alzheimer's disease and mild cognitive impairment.*

[SHY05] K.M. Sink, K.F. Holden, and K. Yaffe, *Pharmacological treatment of neuropsychiatric symptoms of dementia A review of the evidence*, 2005, pp. 596–608.

[Sid67] Z. Sidak, *Rectangular confidence regions for the means of multivariate normal distributions*, Journal of the American Statistical Association (1967), 626–633.

[Sip96] M. Sipser, *Introduction to the Theory of Computation*, International Thomson Publishing, 1996.

[SJ09] D. Sontag and T. Jaakkola, *Tree Block Coordinate Descent for MAP in Graphical Models*, Proceedings of the 12th International Conference on Artificial Intelligence and Statistics, vol. 12, 2009.

[SK75] D. Sherrington and S. Kirkpatrick, *Solvable model of a spin-glass*, Physical review letters **35** (1975), no. 26, 1792–1796.

[SMG+08a] D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss, *Tightening LP relaxations for MAP using message passing*, Conf. Uncertainty in Artificial Intelligence (UAI), Citeseer, 2008.

[SMG+08b] D. Sontag, T. Meltzer, A. Globerson, Y. Weiss, and T. Jaakkola, *Tightening LP relaxations for MAP using message-passing*, Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence table of contents, 2008.

[Spi71] F. Spitzer, *Markov random fields and Gibbs ensembles*, American Mathematical Monthly (1971), 142–154.

[SSW04] S. Singh, V. Soni, and M. Wellman, *Computing approximate Bayes-Nash equilibria in tree-games of incomplete information*, Proceedings of the 5th ACM conference on Electronic commerce, 2004, pp. 81–90.

[SSW07] V. Soni, S. Singh, and M.P. Wellman, *Constraint satisfaction algorithms for graphical games*, Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems, ACM, 2007, p. 67.

[SSW08] S. Sanghavi, D. Shah, and A. Willsky, *Message-passing for Maximum Weight Independent Set*, Arxiv preprint arXiv:0807.5091 (2008).

[SYA+01] A. Shimokawa, N. Yatomi, S. Anamizu, S. Torii, H. Isono, Y. Sugai, and M. Kohno, *Influence of deteriorating ability of emotional comprehension on interpersonal behavior in Alzheimer-type dementia*, Brain and Cognition **47** (2001), no. 3, 423–433.

[Tal03] M. Talagrand, *Spin glasses: a challenge for mathematicians: cavity and mean field models*, Springer Verlag, 2003.

[Tan81] R. Tanner, *A recursive approach to low complexity codes*, IEEE Transactions on Information Theory **27** (1981), no. 5, 533–547.

[TB89] J.N. Tsitsiklis and DP Bertsekas, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall Englewood Cliffs, NJ, 1989.

[TBBDP96] C. Tallon-Baudry, O. Bertrand, C. Delpuech, and J. Pernier, *Stimulus specificity of phase-locked and non-phase-locked 40 Hz visual responses in human*, Journal of Neuroscience **16** (1996), no. 13, 4240–4249.

[TJ02] S. Tatikonda and M. Jordan, *Loopy belief propagation and Gibbs measures*, Proceedings of the 2002 Annual Conference on Uncertainty in Artificial Intelligence, vol. 18, 2002, pp. 493–500.

[Tre01] L. Trevisan, *Non-approximability results for optimization problems on bounded degree instances*, Proceedings of the thirty-third annual ACM symposium on Theory of computing, 2001, pp. 453–461.

[TS07] H. Tsaknakis and P.G. Spirakis, *An optimization approach for approximate Nash equilibria*, Lecture Notes in Computer Science **4858** (2007), 42.

[VCD+05] F. Vialatte, A. Cichocki, G. Dreyfus, T. Musha, T.M. Rutkowski, and R. Gervais, *Blind source separation and sparse bump modelling of time frequency representation of EEG signals: New tools for early detection of Alzheimer's disease*, IEEE Workshop on Machine Learning for Signal Processing, 2005, pp. 27–32.

[Via05] F. Vialatte, *Modélisation en bosses pour l'analyse des motifs oscillatoires reproductibles dans l'activité de populations neuronales : applications à l'apprentissage olfactif chez l'animal et à la détection précoce de la maladie d'alzheimer*, Ph.D. thesis, Paris VI University, 2005.

[VK02] D. Vickrey and D. Koller, *Multi-Agent Algorithms for Solving Graphical Games*, Proceedings of the national conference on Artificial Intelligence, 2002, pp. 345–351.

[VK05] P.O. Vontobel and R. Koetter, *Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes*, Arxiv preprint cs/0512078 (2005).

[VLM+07] M. Volkers, C.M. Loughrey, N. MacQuaide, A. Remppis, B.R. DeGeorge, F. Wegner, O. Friedrich, R.H.A. Fink, W.J. Koch, G.L. Smith, et al., *S100a1*

decreases calcium spark frequency and alters their spatial characteristics in permeabilized adult ventricular cardiomyocytes, Cell Calcium **41** (2007), no. 2, 135–143.

[VLRM01] F. Varela, J.P. Lachaux, E. Rodriguez, and J. Martinerie, *The brainweb: phase synchronization and large-scale integration*, Nature Reviews Neuroscience **2** (2001), no. 4, 229–239.

[VMD+07] F.B. Vialatte, C. Martin, R. Dubois, J. Haddad, B. Quenet, R. Gervais, and G. Dreyfus, *A machine learning approach to the analysis of time–frequency maps, and its application to neural dynamics*, Neural networks **20** (2007), no. 2, 194–209.

[Wei00] Y. Weiss, *Correctness of local probability propagation in graphical models with loops*, Neural computation **12** (2000), no. 1, 1–41.

[Wei06] D. Weitz, *Counting independent sets up to the tree threshold*, Proc. 38th Ann. Symposium on the Theory of Computing, 2006.

[WJ08] M. Wainwright and M. Jordan, *Graphical models, exponential families, and variational inference.*, Foundations and Trends in Machine Learning **1** (2008), 1–305.

[WJW03a] M. Wainwright, T. Jaakkola, and A. Willsky, *Exact MAP estimates by (hyper) tree agreement*, Advances in neural information processing systems (2003), 833–840.

[WJW03b] M. Wainwright, T. Jaakkola, and A. Willsky, *Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudomoment matching*, Workshop on Artificial Intelligence and Statistics, 2003.

[WJW05a] , *MAP estimation via agreement on trees: message-passing and linear programming*, IEEE Transactions on Information Theory **51** (2005), no. 11, 3697–3717.

[WJW05b] M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky, *A new class of upper bounds on the log partition function*, IEEE Transactions on Information Theory **51** (2005), no. 7, 2313–2335.

[WYM07] Y. Weiss, C. Yanover, and T. Meltzer, *Map estimation, linear programming and belief propagation with convex free energies*, Uncertainty in Artificial Intelligence, Citeseer, 2007.

[XBCP06] J.W. Xu, H. Bakardjian, A. Cichocki, and JC Principe, *EEG Synchronization Measure: a Reproducing Kernel Hilbert Space Approach*, submitted to IEEE Transactions on Biomedical Engineering Letters (2006).

[YFW00] J. Yedidia, W. Freeman, and Y. Weiss, *Understanding Belief Propagation and its generalizations*, Tech. Report TR-2001-22, Mitsubishi Electric Research Laboratories, 2000.

[YMW06] C. Yanover, T. Meltzer, and Y. Weiss, *Linear Programming Relaxations and Belief Propagation–An Empirical Study*, The Journal of Machine Learning Research **7** (2006), 1907.

Index

algorithm, 191
    additive error, 193
    approximation algorithm, 193
    multiplicative error, 193
    polynomial time, 192
approximation scheme
    EPTAS, 193
    FPTAS, 193
    PTAS, 193

Bayesian networks, 21
belief Propagation, 24
belief propagation, 29
    loopy belief propagation, 38
    max-product, 29
    min-sum, 29
    MPLP, 42
    sum-product, 29
    tree reweighted belief propagation, 41
best response, 120

canonical parameters, 20
cavity, 34
    censored cavity, 98
    function, 34
    partial cavity function, 35
cavity expansion algorithm, 24, 45, 53
    boundary conditions, 66
cavity method, 30
cavity propagation, 58
cavity recursion
    general, 46
    tree, 36
    trees, 35
censored cavity function, 134
coincidence, 154
constraint satisfaction problem, 117
correlation decay, 24, 26
    definition, 66
    exponential decay, 66
    rate, 66
counting problems, 23

decision, 20
decision network, 20
    pairwise costs, 32
decision problem, 191

energy function, 20
exponential family distributions, 20
exponential parameters, 20

free energy, 20

game
    normal form game, 118
Gibbs distribution, 20

graph, 15
    hypergraph, 15
graphical game
    decomposable game, 121
    directed tree game, 121
graphical model, 20
    pairwise costs, 32
graphs
    factor graphs, 15

instance size, 192
interaction functions, 20
inverse temperature, 20
Ising model, 19

junction tree, 40
junction tree algorithm, 40

log partition function, 20
long-range independence, 62
low density parity check code, 30

Markov random fields, 21
maximum weight independent set, 26
message-passing algorithms, 23

Nash Cavity Expansion, 133
Nash cavity function, 122
    binary, 122
    local, 122
Nash cavity functions, 27
Nash equilibrium, 119
    pure Nash equilibrium, 119
NP(complexity class), 192

optimization problem, 191
optimization problems, 23

P(complexity class), 192
partition function, 20
phase-type distribution, 112
potential functions, 20

recursive distributional equations, 30

sampling problems, 23
search problem, 194
    total search problem, 194
self avoiding walk, 45, 46
SES
    mother process, 160
solution, 118
spin glass, 30
strategy, 118
    discretized strategy, 118
    pure strategy, 118
    strategy profile, 118
synchronicity, 152

utility function, 118