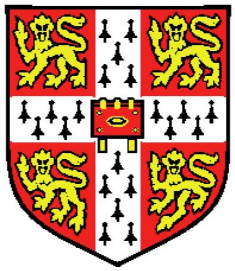
Combining Approximations for Inference



Frederik Eaton
Department of Engineering
University of Cambridge

A thesis submitted for the degree of
Doctor of Philosophy

August 23, 2011

Title: *Combining Approximations for Inference*

Declaration:

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text.

Statement of length for Engineering Department:

This dissertation does not exceed 65,000 words in length, and 150 figures

Supervisor: Professor Zoubin Ghahramani

Advisor: Doctor Bill Byrne

Abstract

Combining Approximations for Inference

by Frederik Eaton

The problem of Bayesian statistical inference, or "approximate inference", is fundamental to Bayesian Machine Learning and statistics. Much effort has been spent in studying the application of approximate inference techniques to specific models, or in analysing the behaviour of specific algorithms. Here we address the broader goal of creating more powerful and general approximate inference algorithms. We propose to embark upon this long and difficult journey by first investigating the ways in which it can be effective to combine the outputs of multiple approximate inference algorithms.

In this thesis, we describe four such ways: partitioning a statistical model between multiple approximations for cooperative solution (chapter 3), competitively comparing the accuracy of two different approximations (chapter 4), exchanging information between two approximations, a "teacher" and a "student" (chapter 5), and harnessing cooperation and competition in simulated evolution, in order to optimise over the space of approximations to a model (chapter 6).

Contents

This thesis is dedicated to my parents

Frederik Eaton
July, 2011

Chapter 1

Introduction

In this chapter, we motivate our choice of "combining approximations" as a research topic, after developing the reasoning which underlies our own view of approximate inference as an approach to the study of artificial intelligence.

Chapter 2 reviews some of the technical background for approximate inference and complexity theory, which is needed in the rest of the thesis.

1.1 The dream of Artificial Intelligence

We would like for computers to be more intelligent. We are tired of supervising them. Computers can be very good at producing quick responses to simple questions, but when we use them to solve difficult problems we find ourselves having to break our questions down, to oversee the solution of the various parts, and this puts us in the position of responding to a tool which we had wanted to be responding to us. We would like to be able to say: "optimise this circuit" or "prove this theorem" and receive the results by email, without having to worry about whether we specified the right parameters (search depth, annealing schedule) and without too much concern that a competitor might be solving the same problem faster and with fewer resources.

We will refer to this end goal as Artificial Intelligence (AI).

We don't think of an AI as being defined interactively, as a computer program which is able to carry on "polite conversation" masquerading undetected as a human.[1] Our picture of AI is more like a computer program which can emulate an idealised mathematician. It should be able to take

[1] A.M. Turing. "Computing machinery and intelligence". In: *Mind* 59.236 (1950), pp. 433–460.

an abstract but well-defined problem and go away and work on it, considering all possible approaches, partitioning its time between them in a rational manner, and returning a solution when it is done. The two definitions may well be seen as equivalent, but we think that the second one leaves less room for misinterpretation.

More precisely, *interaction* with an external environment may play a role in the intellectual functioning of a real mathematician, but to the extent that it does, he will usually be communicating with other mathematicians regarding topics of shared interest. However, in such cases we could view the totality of mathematicians as a single mathematician - we could imagine putting them in a box, submitting a problem through a slot in this box, and receiving a solution through the same slot. Thus, although successive cycles of interaction, involving distinct notions of agent and environment, may be helpful in describing the internal workings of this box, we do not see it as playing a part in the outward definition of intelligence.

How should we go about trying to advance the goals of AI? In the rest of this chapter, we attempt to address this question. We will sometimes use technical terms or concepts, whose definitions we review in chapter 2. Readers unfamiliar with these terms may wish to skip ahead and read chapter 2 before continuing.

1.2 A probabilistic approach to intelligence

Many past approaches to AI have proceeded by breaking down difficult problems using assorted application-specific rules and heuristics, in the hope that such activities could build insight into the more general aspects of intelligence. For instance, chess was once considered the domain of AI, and the standard approach to solving it - traversing a game tree using heuristics to control depth and to establish priority between branches - was typical of the AI plan of the 1950-1980 period. Some of these methods were successful in tackling specific, circumscribed problems, but none succeeded in elucidating the general principles underlying intelligence. This failure has often been attributed to the fact that such methods provided no general answer to the difficult, which is to say presumably exponential time complexity of their input problems. Heuristics which have been developed to work well against one class of tasks, such as analysing chess positions, tend to break down when that class is generalised (say, to include Go).

A fresh approach to AI might start by identifying some elementary facility common to all intelligence which allows for the concise representation

of difficult problems, for instance of the NP-complete variety, while abbreviating the more sophisticated behaviours of our idealised mathematician - for example, his capacities for emotion, communication, or visualisation. We could, for instance, theorise that since intelligence is able to work towards goals, it must have the ability to choose between actions which would have different utility in advancing it towards these goals. By quantising such advancement into a unit of currency and offering such a system choices between p units now or 1 unit if and only if some proposition (event) x turns out to be true, we can discover a value $p(x)$ at which the alternatives have equal weight. A well-known theorem called the Dutch Book Theorem[2] states that if the system is rational then the $p(x)$ which has been elicited in this way behaves like a probability in all salient respects. We call such probabilities "beliefs" and conclude that intelligent systems can be made to express beliefs about any potential set of events; these beliefs are encoded as probabilities. We can describe many useful probability distributions in terms of the relationships between small groups of variables. The problem of calculating the marginals of a distribution specified in this manner is called statistical inference. Performing statistical inference is known to be NP-hard, which means that difficult problems such as boolean satisfiability can be reduced to it, and so it fulfils our criterion for the sought-after facility: it is fundamental, common to all intelligent systems; it consists of a simple framework, based on probability theory; and at the same time it is difficult, and capable of representing other difficult problems of interest.

A distinction can be drawn between two kinds of statistical inference: exact inference, which produces marginals that are accurate within the limits of machine precision, and approximate inference, which produces marginals which are only accurate to within some possibly larger error bound.[3] Both problems are NP-hard, but we prefer to focus on approximate inference as it is more general. Often, it is possible to cope with approximate marginals, and considering approximate algorithms allows us to make trade-offs between time and accuracy. In such circumstances, the problem of approximate inference must no longer be considered to have a certain time complexity, because arbitrarily weak approximations can be produced within any given time constraint. The term "statistical inference" will be used to refer to both types of inference in this dissertation.

[2] B. De Finetti. "Probabilism: A critical essay on the theory of probability and on the value of science". In: *Erkenntnis* 31.2 (1989), pp. 169–223; E.T. Jaynes and G.L. Bretthorst. *Probability theory: the logic of science*. Cambridge University Press, 2003.

[3] None of the algorithms which we consider are actually able to provide tight bounds on the error of the marginals they produce, but such bounds exist in theory.

Crossing the border into a land where everything is a probability might seem like a strange way to start a journey whose ultimate destination is to emulate an idealised mathematician. Below we raise and respond to two potential objections.

Symbolic representations A first objection to this plan could be that we rarely hear people, even mathematicians, communicating directly about probabilities. They may be uncertain or disagree, but they use words like "maybe", "definitely", "possibly" rather than "0.5", "0.95", "0.1". Perhaps this shows that the use of symbols is more fundamental and should be examined first; or perhaps our intention to represent all the various correlations between variables in a large model is misguided - that task is too difficult, we should be using per-variable, "fuzzy" measures of uncertainty, and combine them using local, approximate rules. In reply, we reiterate that it is actually possible to elicit probabilities from intelligent beings as described above; whether probabilities are stored internally using a completely different - perhaps symbolic or qualitative - form, for instance assigning greater certainty to propositions which we have heard expressed in rhyme, an intelligent system ought to be able to produce probabilities as output (e.g. when asked to value a bet). Furthermore, beliefs elicited in this manner ought to behave like probabilities when we query joints or conditionals, and frameworks such as fuzzy logic[4] which ignore this desideratum can result in probability estimates which are arbitrarily bad.

Deterministic models A second, more serious objection, would be that the purpose of probabilities is to represent uncertainty about truly random systems in our environment. A mathematician, on the other hand, deals with deterministic statements, i.e. statements which are either true or false. The same is true for a circuit optimiser, or a chess player. Surely the key to modelling such systems is to have a proper internal representation of what is known to be true, together with rules for extending this representation with additional true statements, and some intelligent way to apply these rules. Even if uncertain beliefs can be elicited, they are of little use since over time any beliefs about propositions in the model should converge to 1 or 0. To this objection we reply with the practical observation that mathematicians can and do employ uncertainty at a very fundamental level. Any conjecture

[4]L.A. Zadeh. "Fuzzy sets". In: *Information and control* 8.3 (1965), pp. 338–353; D. Klaua. "Über einen Ansatz zur mehrwertigen Mengenlehre". In: *Monatsberichte der Deutschen Akademie der Wissenschaften Berlin* 7 (1965), pp. 859–867.

is uncertain until it is proven, and sometimes such uncertainty persists for hundreds of years, even around important conjectures which are the focus of much research. Deciding whether to try to prove any theorem requires us to form beliefs about whether that theorem is true or not. And although rules for using logic and applying axioms are deterministic, often one applies heuristics or makes analogies which are non-deterministic in an attempt to "reason backwards" from a theorem whose proof is desired. One might see that $A \implies B$ and form the hypothesis that $B \implies A$; in forward reasoning or "deduction" this would be an erroneous step, termed a "fallacy"[5], but in backwards reasoning, sometimes called "induction", it could be used to construct useful approximate beliefs about the statement $B \implies A$. Combined with other heuristics, such approximate beliefs would help us decide whether to further investigate $B \implies A$. Such inductive reasoning, though approximate in nature, is described by many mathematicians as central to their work.[6] Finally, as an example we note that the Survey Propagation algorithm for k-SAT is based on a fundamentally probabilistic analysis of an inherently deterministic problem, yet has managed to advance the state of the art for a certain class of deterministic problems - randomly generated k-SAT instances. The success of such probabilistic approaches might be seen as additional evidence in favour of a probabilistic approach to AI.

1.3 The complexity of inference

The foregoing section has argued that we should approach AI using frameworks which are able to represent uncertainty, such as statistical inference. Once we have decided upon statistical inference as the avenue through which to investigate AI, we are confronted with another problem: the time complexity of statistical inference. One of our reasons for selecting inference in the first place was its NP-hardness, which means that it is able to represent[7] NP problems, some of which are considered difficult yet amenable to intelligent solution.

We should say a few words about why we are interested in members of NP, rather than some other complexity class, as being representative of the sort of difficult problems on the solution of which AI should be able to distinguish itself. We could consider, for example the following problem:

[5] This particular fallacy is called "affirming the consequent".

[6] G. Pólya. *Mathematics and Plausible Reasoning: Patterns of plausible inference.* Princeton University Press, 1954.

[7] I.e. NP problems can be reduced to an NP-hard problem in polynomial time; see section 2.5.

given an input integer n (represented using $\log n$ bits), compute the result of applying a standard cryptographic hash function n times in succession to some initial string like "hello". This is certainly a difficult problem, requiring time exponential in the size of its input. But it doesn't have the appearance of being a useful problem to solve - even if it were possible to improve on the time-complexity of the naive algorithm for solving it. This is in contrast to problems in NP such as boolean satisfiability, or SAT, a canonical NP-complete problem which is useful in many domains, such as scheduling and optimisation of circuit layout. The difference may be due to the fact that NP comprises that class of problems whose solutions can be verified easily (in polynomial time) - for example, by checking that each SAT clause is satisfied by the suggested variable assignment, or that the schedule has no conflicts or the circuit layout is valid and fits within the allotted area. By contrast, the preceding hash function example involves solutions which can apparently only be verified by recomputing them all over again. On reflection, easy verification of solutions is a very natural property to demand of problems which are targets for AI. Putting solutions of such problems to practical use, as one can see with the scheduling or optimisation examples, is tantamount to verifying them, and so easy verifiability is related to being able to use a solution without the overhead associated with computing it. If we want to use AI as a *tool* for building results which can stand on their own, so that this tool can be taken away and subsequently applied to new tasks, then the NP complexity class presents itself as a natural target domain.

Yet we are convinced that the average-case time complexity of the NP-hard class (and its subset, NP-complete) is exponential in the problem size,[8] although this has not yet been proven, and the No Free Lunch Theorems (NFLT)[9] tell us that even if we manage to do better than this for certain problems, such improvement can only be realized at the expense of performance on other problems. One might argue that since we cannot improve the average-case performance of our inference algorithms (only being able to speed them up by a constant factor or change the base of the exponential), it follows that programs which attempt to solve such general problems are bound to be slow, and we should devote our energy to attacking more specialised problems for which economical solutions may be available.

[8] R. Impagliazzo and R. Paturi. "Complexity of k-SAT". In: *Computational Complexity, 1999. Proceedings. Fourteenth Annual IEEE Conference on.* IEEE. 2002, pp. 237–240.

[9] D.H. Wolpert and W.G. Macready. *No free lunch theorems for search.* Tech. rep. SFI-TR-95-02-010. Santa Fe Institute, 1995; D.H. Wolpert and W.G. Macready. "No free lunch theorems for optimization". In: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), pp. 67–82.

Since most inference algorithms apply to arbitrary statistical inference problems, this view has led to a good deal of research which considers problems and inference algorithms in pairs, intending to show that specific inference techniques (such as BP or Gibbs sampling) apply well to specific practical problems (such as document classification or change-point detection).

Perhaps it is only apparent to an external observer that the research process thus described - in which humans analyse the complexity of different inference tasks, perform experiments to identify the best algorithms for solving them, and then hand the problems over to a computer to be solved using the appropriate algorithm - could, assuming AI is possible, theoretically be performed entirely by computer from start to finish.

But this also describes the mechanism by which we might hope an "intelligent" inference algorithm would circumvent the NFLT: by dividing the space of statistical inference problems into subclasses of varying complexity, and solving the simpler subclasses as quickly as their structure allows. Since there are many more complex problems than simple ones, we can realize such a speed-up by slowing down the solution of complex problems by a small constant factor (the cost of exploring simpler solutions simultaneously). The difference is then credited to the account of the simple problems, so that the average time-complexity remains unchanged. Such an approach is taken implicitly by Marcus Hutter's 2001 paper, "The Fastest and Shortest Algorithm for All Well-Defined Problems", which matches the speed of any algorithm (for which there exists a proof of its correctness and a time bound) on any problem, to within a factor of $4 + \epsilon$ plus a (huge) constant needed to budget for searching through the space of possible programs and correctness proofs. Hutter's result, although only relevant to theorists, embodies what we hope to be able to achieve with AI - a procedure that tackles each problem optimally, but with some overhead (ideally smaller than in Hutter's proof of concept) which can be thought of as being devoted to the analysis of such problems.

In this section and the previous one we have argued that AI should be approached through a probabilistic framework, and that it should be able to adapt to problem complexity much as humans are able to do. So far, the vision of AI that we have advocated does not differ substantially from an automated version of a typical researcher in Machine Learning, which is the field that uses statistical inference techniques to draw conclusions about data. Our point of departure from Machine Learning can be located in the methods which machine learning advocates for evaluating the appropriateness of an approximate inference algorithm on a given problem. Since ma-

chine learning is the primary consumer of approximate inference techniques, we will first review the methodology behind this body of applications for contrast, before going on to describe our own philosophy.

1.4 Methodology of machine learning

Machine Learning (ML) is considered here to be a branch of data analysis. It arose together with the relatively recent development of computer technology and the internet, as a body of techniques for extracting useful information from the increasingly large and complex structured data-sets which had become available as a result of such technology. It is used in fields such as linguistics, computer vision, finance, astronomy, and biology, as well as in business applications such as search engines and product recommender systems, where the techniques of early 20th-century statistics, having been developed for simple or low-dimensional datasets, were not appropriate. The methods of Machine Learning can be classified according to their philosophy of data modelling:

- **Deterministic** methods attempt to model data with deterministic, functional relationships between variables. Parameters defining such relationships are optimised to reduce a measure of the error of the approximation, and overfitting is avoided using cross-validation to select the best model. Approximate inference is used rarely if at all.
- **Probabilistic** methods view the data as having been produced by a "generative model", essentially a stochastic program. Parameters may be selected to maximise the likelihood of the data, and overfitting can be avoided by averaging over models of different complexity. Both of these tasks, as well as that of using the fitted model to characterise the data or make predictions, employ the methods of (and constitute the principal applications for) approximate inference.

These are only rough, high-level descriptions of the two main approaches to ML. In practice, there is much overlap between them. Deterministic methods may make use of probabilistic techniques like model averaging, and probabilistic methods may use deterministic techniques like cross-validation. Since our interest is approximate inference, we will be concerned with the probabilistic side of the spectrum. In this thesis, we use the term "Machine Learning" to mean "Probabilistic Machine Learning".

The central role played by data in ML strongly influences the way that the field employs statistical inference. In ML, data is used not only to infer

the parameters and occasionally to learn the structure of a model, but also as a source of samples from "ground truth". These samples can be used in conjunction with inference in several ways:

Use of data to validate the inference algorithm Similarly to cross-validation, a particular choice of approximation can be evaluated against another by testing their predictions against the data being modelled, to see which approximation assigns a higher probability to the data. Alternatively, if the purpose of inference is to enable us to make decisions based on some input, we can evaluate the accuracy of inference by measuring the quality of the decisions.

Use of data to signal convergence In the application of sampling algorithms such as MCMC, we can guess when a sampling run has sufficiently converged by measuring the probability of the data, for instance by sampling over latent variables and calculating the probability of observed variables in a data set with each setting, and waiting for the average of this quantity to stop increasing.

Use of data to replace inference In complex models, we may use data from an external system which is assumed to do inference well on our model as a kind of manual replacement for some of the inference we find too difficult to do ourselves. This is done in some game-playing algorithms, for instance, where data from experts is used to augment the difficult process of inference on the game tree. Another example might include physical simulations, where the measured values of some quantities such as melting point or molecular bond length might be incorporated directly into a simulation rather than inferring these "from scratch".

1.5 Our approximate inference philosophy

We have enumerated several ways in which traditional ML methodology relies on data when applying inference. We can see that these constitute a kind of "interaction with an external environment", which was originally found to be superfluous to our definition of AI. This is not to say that such interaction is not important - using data in this way forms the basis of science. Nor do we wish to suggest that AI could not be applied to the task of reasoning about data which comes from an external source. On the

contrary, we expect that such applications will dominate. The problem we have with this use of data is that from the perspective of statistical inference, it forms a kind of crutch, and complicates the task of judging the quality of our inference algorithms in isolation. A chess player does not have access to samples of optimal moves, and our "mathematician in a box" has no recourse to a source of samples of true theorems - other than what has formed his education, which is assumed to have already taken place. We anticipate that by avoiding this reliance upon data, we can focus our efforts to build a better approximate inference algorithm. (Such an algorithm could then be used, among other things, to facilitate an advancement of the state of the art in data modelling techniques.)

When we apply approximate inference to problem domains where the statistical model comes with "real data", we make it difficult to know how much of our success to credit to the inference algorithm and how much to credit to the way in which the data was collected or used, or the way the model was learned from it. Furthermore, there may be a tendency to avoid tackling certain difficult problems in approximate inference, such as the question of how to evaluate the quality of an approximation to a large model, when our focus is on applications in which "real" data can often be employed to give case-by-case workarounds.

We now wish to highlight the existence of another continuum of approximate inference methodologies. On the one extreme, we have situations where a model is learned from data consisting of i.i.d. samples in which each of the variables are observed, so that statistical inference in the model can be validated or augmented using these samples. The application of approximate inference at this extreme we will call "applied" approximate inference. Traditional ML, as well as any other application area in which data plays an important role, falls near this extreme. On the other extreme is our idealised mathematician, or box of mathematicians, who must perform probabilistic reasoning in a model which has been fully specified using rules and axioms, and for which no source of data samples exists. This extreme would also include applications such as the optimisation of software or electric circuits, and playing games like chess or backgammon. We refer to applications at this extreme as "pure" approximate inference. Between these two extremes of "pure" and "applied" approximate inference, we find applications in which we only get to observe some subset of a model's variables, or in which we are only provided with sufficient statistics such as pairwise correlations. There may also be applications in which a large model has been learned with the help of data, but new data points (consisting of the outcomes of financial or medical experiments, perhaps) are so expensive that we don't wish to

consume them in validating inference.

Which approximate inference algorithms are best suited to the "pure" methodology, and which to the "applied"? This is not the most appropriate question to ask - since it is the applications, and not the algorithms themselves, which are described by these categories. Although we grant that it is correct to associate pure approximate inference with algorithms which are more autonomous, and applied approximate inference with algorithms that require problem-specific tuning or analysis, most algorithms have extensions which can be used in both ways. For example, Belief Propagation is a fairly general-purpose algorithm, which could therefore be seen as satisfying one of the goals of autonomy, thus making it "pure"; but it has extensions like Generalised Belief Propagation and Expectation Propagation whose operation should be parametrised to suit specific "applied" applications. Similarly, Gibbs sampling is not only general-purpose but produces arbitrarily accurate approximations over time, in a sense making it even more autonomous and "pure" than Belief Propagation; but it also has numerous extensions and optimisations (such as *collapsed* or *blocked* samplers) which invite "applied" problem-specific adaptation. It is hard to draw a clear line between both sets of algorithms. We can observe, however, that the work of extending algorithms shows a trend of development towards the more "applied" settings - which is to be expected.

In summary, although almost all of the contemporary consumers of approximate inference technology fall into the category of applications which we have called "applied", and although we would expect to see such applications continue to occupy an important place as artificial intelligence technology advances, we don't see the "applied" methodology as posing the most stimulating challenges when it comes to motivating such advancement. Instead, we focus on "pure" applications of approximate inference - in which a model is specified exactly, and not learned from data. We have devoted some space to this discussion because we expect most of our readers to come from a machine learning background, and to be most accustomed to applied approximate inference. Such readers may have a tendency to question or misunderstand our direction and priorities in the pages that follow. By highlighting the contrast between our own hypothetical applications and those of traditional ML, we hope to avoid some of these misunderstandings.

1.6 Our contribution

In this introductory chapter we have advocated studying approximate inference - of the "pure", data-free variety - as a bite-sized first step on the way to the bigger goal of AI. This leaves us with the question of how one should best approach approximate inference. For the most part, the various contributions of this thesis were developed independently of any one plan of attack. It was not until afterwards that we found that the various research directions we had pursued could all be unified under the umbrella of "methods for combining multiple different approximations". This concept was then selected as the subject of the dissertation. Although conceived in hindsight, we should see the study of ways to "combine approximations" as a simple and natural strategy for making progress on the problem at hand. Below, we establish this strategy with some simple principles and analogies.

We start by recalling the end goal: some kind of artificial intelligence.

We know that real intelligence can be used for communication and in fact we already made the argument, based upon our view of AI as an idealised mathematician, that the combination of multiple of these intelligences (e.g., by putting them together in a hypothetical "box") should be externally equivalent to a single one, aside from differences in speed or power. Furthermore, the mind of an animal is often said to function through the cooperative action of many different subcomponents; this parallelism is apparent whether we perform physical or electrical measurements on the brain, or psychological or behavioural experiments on the unconscious. The observation that intelligence can be composed or decomposed in this way leads us to hypothesise that perhaps by studying the various forms and functions to be realised in such compositions, we can understand how new intelligence should be created.

This is, of course, not such a radical idea. Indeed, after trying to distance ourselves from the applied methodology of Machine Learning, which combines approximations with data, we find that we are left with only approximations and no other type of object - the only thing left to do would seem to be to search for ways of combining these approximations with each other. And in fact many or all of the popular approximate inference algorithms in use today can be seen as doing this in one way or another. The algorithm of Belief Propagation, performed on a tree, combines the (exact) approximate marginals for a node at each of its isolated subtrees by multiplying them together as "messages". Sampling methods combine together a number of samples, each of which can be seen as a very poor set of approximate marginals, by averaging. Of course, it is important that

the samples be generated in an appropriate way (for instance by iterating a stochastic transition operator which satisfies detailed balance); similarly, in Belief Propagation it is necessary to multiply together each of the incoming messages exactly once (though some variations are possible[10]).

We can perhaps motivate the concept of "combining approximations" more rigorously by using arguments from complexity theory. The parallelisable nature of programs for solving problems in the complexity class NP, which class we have tried to characterise as being connected to intelligence, often seems to result in algorithms which work by decomposing a problem into a number of subproblems and then combining the results. If this is also the proper form for an effective inference algorithm, and if the subproblems consist of inference as well, then we expect to find a more or less direct relationship between approximate inference algorithms and ways of combining approximations.

For these reasons we chose to devote the subject of this thesis to combining approximations. In the chapters that follow, we investigate four questions related to the topic:

- How should we go about partitioning a statistical inference problem for cooperative solution between multiple approximate inference algorithms? (chapter 3)
- Suppose we are given multiple approximate inference algorithms which result in divergent beliefs about the same model. Are there methods that can identify which one is more accurate (without knowing the true marginals)? (chapter 4)
- Is there a good way for one algorithm to "teach" another about the interesting aspects of a model, without simply conveying his own (possibly imperfect) beliefs? (chapter 5)
- Is there a way to "evolve" approximate inference algorithms which produce more accurate results (again, without knowing the true marginals)? (chapter 6)

We give a positive answer to each question, together with experimental results supporting our claims.

[10] W. Wiegerinck and T. Heskes. "Fractional belief propagation". In: *Advances in Neural Information Processing Systems 15*. MIT Press. 2003, p. 455.

Chapter 2

Background

In this background chapter we give a more formal definition of approximate inference in discrete graphical models. We give a broad overview of different types of inference algorithms, and describe a few specific algorithms in detail. We discuss the semantics and generality of discrete factor graphs, roughly demarcating the broad class of problems which can be converted into the factor graph form required by our algorithms. We examine some common restrictions which can be placed on the structure of factor graphs, and present original results classifying the conditions under which general factor graphs can be converted into these restrictive forms. Finally, we give a brief overview of complexity theory and discuss the intractability of exact and approximate inference.

2.1 Definitions

Bayesian statistical inference, also called probabilistic inference, statistical inference or just inference, is defined here as the task of computing the probability distribution of some variable or set of variables in a statistical model which is defined over a possibly larger set of variables:

$$P(x_s) = \int P(x) \mathrm{d}x_{\setminus s} = \sum_{x_{\setminus s}} P(x) \tag{2.1}$$

where s is a set of variable indices of interest, x_s denotes the vector $(x_i : i \in s)$, and $x_{\setminus s}$ denotes $(x_i : i \notin s)$, i.e. the set of variables in the model but not in s. When this value is computed exactly (to within the limits of numerical precision determined by the computer architecture) we say that "exact inference" is being done; otherwise it is "approximate inference". We are

more concerned with approximate inference because it is more general and allows us to trade time for accuracy. Both exact inference, and approximate inference with accuracy bounds, are NP-hard, as discussed in section 2.5, although approximate inference with no accuracy guarantees can of course have arbitrary time complexity.

Other problems which are defined as belonging to inference include the problem of computing expectations of functions under probability distributions, which is also called numerical integration; and MAP (maximum a posteriori) estimation, or the problem of calculating $\operatorname{argmax}_x P(x)$.[1] Numerical integration can often be expressed in terms of marginals. MAP is an occasionally more tractable problem than computation of marginals. We do not address either of these problems further.

We consider probability distributions defined in the following way:

$$P(x) = \frac{1}{Z} \prod_{\alpha} \psi_\alpha(x_\alpha) \tag{2.2}$$

$$Z = \sum_{x} \prod_{\alpha} \psi_\alpha(x_\alpha) \tag{2.3}$$

where the values $x \equiv (x_1 \dots x_n)$ are members of some finite set $\mathcal{X} \equiv \mathcal{X}_1 \times \dots \times \mathcal{X}_n$, α indexes a finite set of subsets of variables, to each of which is associated a "factor" (also called local function,[2] potential or interaction[3]) ψ_α, a non-negative real-valued function

$$\psi_\alpha : \mathcal{X}_\alpha \to \mathbb{R}^+ \tag{2.4}$$

Here x is called a "state" of the model, or a "configuration" or an "assignment" of the variables. For $i \in \{1 : n\}$ we refer to x_i as a "random variable" or "variable" in the model, but we may also refer to the same random variable as "variable i" or the value of x_i as the "state" or "value" of variable i; likewise, a factor ψ_α may be called "the potential function of factor α".

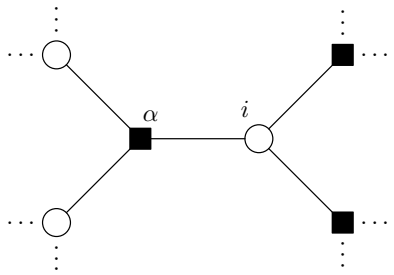
We sometimes write $\mathcal{F}$ for the set of factors and $\mathcal{V}$ $(= \{1, \dots, n\})$ for the set of variables.

[1] Y. Weiss and W.T. Freeman. "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs". In: *Information Theory, IEEE Transactions on* 47.2 (2002), pp. 736–744.

[2] F.R. Kschischang, B.J. Frey, and H.A. Loeliger. "Factor graphs and the sum-product algorithm". In: *IEEE Transactions on information theory* 47.2 (2001), pp. 498–519.

[3] JM Mooij. "Understanding and improving belief propagation". PhD thesis. Radboud Universiteit Nijmegen, 2008.

A problem formed in this way, i.e. as a set of variables $\mathcal{V}$, factors $\mathcal{F} \subseteq 2^{\mathcal{V}}$, and functions $\{\psi_\alpha : \alpha \in \mathcal{F}\}$, is loosely termed a "factor graph"[4] which may also refer to the connectivity graph induced by the factors (potentials). The latter is a bipartite graph with edges (i, α) for each $i \in \mathcal{V}$ and $\alpha \in \mathcal{F}$ such that $i \in \alpha$. We also write the adjacency relation in this graph as $\sim$ and regard a variable as the set of factors containing it, so that for instance $\prod_{\beta \sim i \setminus \alpha} f_\beta$ represents a product over $\{\beta \in \mathcal{F} : i \in \beta \wedge \beta \neq \alpha\}$[5]. Factor graphs are often drawn with variables represented as hollow circles and factors represented as black squares (factors containing only two variables may also be drawn as just an edge between the variables). Here is a depiction of part of a factor graph, with a variable i and factor α which neighbour each other, and which each have two other neighbours:



(2.5)

The probability of an element x of $\mathcal{X}$, i.e. $P(x)$, is called the "joint" probability; the probability of some subset $x_r \in \prod_{k \in r} \mathcal{X}_k$, written $P(x_r)$, and equal to $\sum_{x_{\setminus r}} P(x)$, is called a "marginal" probability. The conditional probability $P(x|y) = \frac{P(xy)}{P(y)}$ is called the "probability of x given y" or the "likelihood of y given x". When we update our beliefs about x after seeing some data y, we call $P(x)$ the prior, $P(y|x)$ the likelihood, and $P(x|y) = \frac{P(x)P(y|x)}{P(y)}$ the posterior. The last equation is called Bayes' Rule. We sometimes write $1 : k$ for the vector $(1, 2, \ldots, k)$ and $x_{1:k}$ for the vector $(x_1, \ldots, x_k)$.

When the arguments of a function of a model's variables can be understood from context, these are sometimes omitted. In such cases, variables of summation are indicated with capital letters, so in equation 2.10 below, instead of

$$Z_i(x_i) \equiv \sum_{x_{\setminus i}} \prod_\alpha \psi_\alpha(x_\alpha) \tag{2.6}$$

[4] Kschischang, Frey, and Loeliger, "Factor graphs and the sum-product algorithm", op. cit.

[5] Here and in the rest of this thesis, $\wedge$ means "and" and $\vee$ means "or".

we could have written

$$Z_i \equiv \sum_{X_{\setminus i}} \prod_{\alpha} \psi_\alpha \tag{2.7}$$

2.2 Some classes of inference algorithms

Statistical inference algorithms can be broadly divided into two classes: *deterministic*, and *stochastic*. Deterministic algorithms include algorithms for exact inference such as junction tree,[6] cutset conditioning[7] and join-tree propagation,[8] as well as those for approximate inference such as Belief Propagation (BP),[9] Generalised Belief Propagation (GBP),[10] Mean Field (MF),[11] Expectation Propagation (EP)[12] and the Expectation Consistent

[6] F.V. Jensen, K.G. Olesen, and S.K. Andersen. "An algebra of Bayesian belief universes for knowledge-based systems". In: *Networks* 20.5 (1990).

[7] J. Pearl. "Fusion, propagation, and structuring in belief networks". In: *Artificial intelligence* 29.3 (1986), pp. 241–288.

[8] S.L. Lauritzen and D.J. Spiegelhalter. "Local computations with probabilities on graphical structures and their application to expert systems". In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1988), pp. 157–224.

[9] J. Pearl. "Reverend Bayes on inference engines: A distributed hierarchical approach". In: *Proceedings of the AAAI National Conference on AI*. 1982, pp. 133–136; RG Gallager. *Low Density Parity Check Codes. Number 21 in Research monograph series*. 1963; K. Nakanishi. "Two- and three-spin cluster theory of spin-glasses". In: *Physical Review B* 23.7 (1981), pp. 3514–3522; H.A. Bethe. "Statistical Theory of Superlattices". In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 150.871 (1935), pp. 552–575; R. Peierls. "On Ising's model of ferromagnetism". In: *Mathematical Proceedings of the Cambridge Philosophical Society*. Vol. 32. 03. Cambridge University Press. 1936, pp. 477–481.

[10] J.S. Yedidia, W.T. Freeman, and Y. Weiss. "Generalized belief propagation". In: *Advances in Neural Information Processing Systems 13* (2001), pp. 689–695.

[11] M. Mézard and G. Parisi. "Mean-field equations for the matching and the travelling salesman problems". In: *EPL (Europhysics Letters)* 2 (1986), p. 913; M. Mézard and G. Parisi. "Mean-field theory of randomly frustrated systems with finite connectivity". In: *EPL (Europhysics Letters)* 3 (1987), p. 1067; C. Peterson and J.R. Anderson. "A mean field theory learning algorithm for neural networks". In: *Complex systems* 1.5 (1987), pp. 995–1019; M.I. Jordan et al. "An introduction to variational methods for graphical models". In: *Machine learning* 37.2 (1999), pp. 183–233; J. Winn and C.M. Bishop. "Variational message passing". In: *Journal of Machine Learning Research* 6.1 (2006), p. 661.

[12] T.P. Minka. "Expectation propagation for approximate Bayesian inference". In: *Uncertainty in Artificial Intelligence*. Vol. 17. 2001, pp. 362–369; M. Welling, T. Minka, and Y.W. Teh. "Structured region graphs: Morphing EP into GBP". In: *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence*. 2005, pp. 609–616.

(EC) approximation,[13] Loop Corrected Belief Propagation (LCBP),[14] Tree-EP,[15] Tree Reweighted Belief Propagation (TRW-BP),[16] Fractional BP (FBP),[17] Conditioned BP,[18] and convergent double-loop algorithms such as the Convex Concave Procedure (CCCP)[19] and the algorithm of Heskes, Albers, and Kappen (HAK).[20] Many deterministic algorithms are based on "message passing", in which quantities called messages are assigned to nodes or edges of a graph and updated as a function of neighbouring messages. Stochastic (or "sampling") algorithms are always approximate and operate by endeavouring to draw samples from the distribution of interest, and using these samples to answer queries, for instance to compute marginal estimates (by counting the number of samples in which a variable takes each of its values).

Stochastic algorithms may be further divided into two classes: exact sampling algorithms, which attempt to draw uncorrelated (exact) samples directly from the desired distribution, e.g. "coupling from the past"[21] or "systematic stochastic search"[22] or sampling from a causal network;[23] and Markov Chain Monte Carlo algorithms,[24] which draw correlated samples

[13] M. Opper and O. Winther. "Expectation consistent approximate inference". In: *The Journal of Machine Learning Research* 6 (2005), pp. 2177–2204; T. Heskes et al. "Approximate inference techniques with expectation constraints". In: *Journal of Statistical Mechanics: Theory and Experiment* 2005 (2005), P11015.

[14] J.M. Mooij et al. "Loop corrected belief propagation". In: *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*. 2007.

[15] T. Minka and Y. Qi. "Tree-structured approximations by expectation propagation". In: *Advances in Neural Information Processing Systems 16*. 2004, p. 193.

[16] Martin Wainwright, Tommi Jaakkola, and Alan Willsky. "A New Class of Upper Bounds on the Log Partition Function". In: *Proceedings of the Eighteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-02)*. 2002, pp. 536–54.

[17] Wiegerinck and Heskes, "Fractional belief propagation", op. cit.

[18] F. Eaton and Z. Ghahramani. "Choosing a variable to clamp: approximate inference using conditioned belief propagation". In: *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*. Vol. 5. 2009, pp. 145–152.

[19] A.L. Yuille and A. Rangarajan. "The concave-convex procedure". In: *Neural Computation* 15.4 (2003), pp. 915–936.

[20] Tom Heskes. "Convexity arguments for efficient minimization of the Bethe and Kikuchi free energies". In: *Journal of Artificial Intelligence Research* 26 (2006), pp. 153–190.

[21] J.G. Propp and D.B. Wilson. "Exact sampling with coupled Markov chains and applications to statistical mechanics". In: *Random structures and Algorithms* 9.1-2 (1996), pp. 223–252.

[22] V. Mansinghka et al. "Exact and Approximate Sampling by Systematic Stochastic Search". In: 5 (2009).

[23] J. Pearl. "Evidential reasoning using stochastic simulation of causal models". In: *Artificial Intelligence* 32.2 (1987), pp. 245–257.

[24] N. Meteopolis and S. Ulam. "The monte carlo method". In: *Journal of the American Statistical Association* 44.247 (1949), pp. 335–341; A.E. Gelfand and A.F.M. Smith.

that only converge to the desired distribution in the infinite limit. When using MCMC algorithms, one must concern oneself with the question of estimating when a particular algorithm has converged "sufficiently" to be producing useful samples. However, MCMC algorithms can be much faster than exact sampling algorithms and are among the most practical approximate inference algorithms. They include Gibbs sampling,[25] importance sampling (e.g.[26]), rejection sampling,[27] Metropolis,[28] Hybrid Monte-Carlo,[29] Tempered MCMC,[30] combinations of these, and so on. Stochastic algorithms from these two main classes all have the property that error decreases proportionally to $1/\sqrt{n}$ where n is the number of samples (because if the variance of one sample is σ^2, then the variance of a sum of n uncorrelated samples is $n\sigma^2$, the standard deviation is $\sqrt{n}\sigma$, and the standard deviation of the average is $\frac{\sigma}{\sqrt{n}}$; this argument also generalises to the case where samples are correlated).

2.3 Some inference algorithms

To provide a more concrete image of modern inference algorithms we describe some elementary examples from the two main classes: deterministic (below), and stochastic (section 2.3.2).

2.3.1 BP and GBP

We give a brief definition of Belief Propagation (BP), arguably the simplest message-passing algorithm for approximate inference, as well as Generalised

"Sampling-Based Approaches to Calculating Marginal Densities". In: *Journal of the American Statistical Association* 85.410 (1990), pp. 398–409.

[25] S. Geman and D. Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images". In: *IEEE transactions on pattern analysis and machine intelligence* 6.6 (1984), pp. 721–741.

[26] J. Geweke. "Bayesian Inference in Econometric Models Using Monte Carlo Integration". In: *Econometrica* 57.6 (1989), p. 1317.

[27] J. Von Neumann. "Various techniques used in connection with random digits". In: *Applied Math Series* 12.36-38 (1951), p. 1.

[28] N. Metropolis et al. "Equation of state calculations by fast computing machines". In: *The journal of chemical physics* 21.6 (1953), p. 1087; W.K. Hastings. "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika* 57.1 (1970), p. 97.

[29] AD Duane et al. "Hybrid monte carlo". In: *Physics letters B* 195.2 (1987), pp. 216–222.

[30] K. Kimura, K. Taki, and S.S.K.G.K. Kikō. *Time-homogeneous parallel annealing algorithm*. Institute for New Generation Computer Technology, 1991.

Belief Propagation (GBP) which is used in chapters 4 and 6.

Belief propagation, also known as the sum-product algorithm, is related to the Bethe approximation[31] which was developed in statistical physics. BP was first used as an inference algorithm by Gallager in an application to error correcting codes.[32] Its application to exact statistical inference on trees[33] was first recognised by Pearl[34] who also advocated the use of BP in "loopy" graphs for approximate inference.[35] The latter application of BP is sometimes called "Loopy BP", but we also refer to it as just "BP".

BP is the basis for a number of other algorithms, both exact and approximate. Approximate extensions of BP include Generalised Belief Propagation[36] (which can be used for the Cluster Variation Method[37]), Expectation Propagation,[38] and Loop Corrected Belief Propagation.[39] A simple exact extension of BP is called Cutset Conditioning,[40] which is based on the idea of turning a complex graphical model into a simpler one by conditioning on a set of variables. This "conditioning" approach to inference is further

[31] Bethe, "Statistical Theory of Superlattices", op. cit.; J.S. Yedidia, W.T. Freeman, and Y. Weiss. "Bethe free energy, Kikuchi approximations and belief propagation algorithms". In: *Advances in Neural Information Processing Systems 12* 13 (2000).

[32] Gallager, *Low Density Parity Check Codes. Number 21 in Research monograph series*, op. cit.

[33] Pearl's original algorithm applied to singly-connected (tree-structured) Bayesian networks in which each variable has only a single parent, sometimes called simply "trees". BP is more easily derived from a simple generalisation of this algorithm due to Jin Kim, which applies to *polytrees*, which are singly-connected Bayesian networks in which some variables may have multiple parents (J.H. Kim and J. Pearl. "A computational model for causal and diagnostic reasoning in inference systems". In: *Proceedings of the 8th International Joint Conference on Artificial Intelligence*. 1983, pp. 190–193; J. Pearl. "Fusion, propagation, and structuring in belief networks". In: *Artificial intelligence* 29.3 (1986), pp. 241–288; A. Darwiche. "Inference in Bayesian Networks: A Historical Perspective". In: *Heuristics, Probability and Causality. A Tribute to Judea Pearl*. College Publications,).

[34] Pearl, "Reverend Bayes on inference engines: A distributed hierarchical approach", op. cit.

[35] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988, p. 195.

[36] Yedidia, Freeman, and Weiss, "Generalized belief propagation", op. cit.

[37] R. Kikuchi. "A Theory of Cooperative Phenomena". In: *Physical Review* 81.6 (1951), pp. 988–1003; A. Pelizzola. "Cluster variation method in statistical physics and probabilistic graphical models". In: *J. Phys. A: Math. Gen* 38 (2005), R309–R339.

[38] T.P. Minka. "Expectation propagation for approximate Bayesian inference". In: *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence*. Vol. 17. 2001, pp. 362–369.

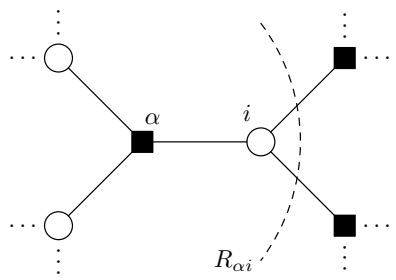
[39] Mooij et al., "Loop corrected belief propagation", op. cit.

[40] Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, op. cit.

explored in chapter 3.

We can derive the BP message updates by considering the problem of statistical inference in tree-structured graphical models. By writing inference in a subtree in terms of sub-subtrees, we can express the marginal of a variable in terms of recursively-defined quantities which we call "messages", which leads to a dynamic programming style algorithm. These updates can then be applied to loopy graphs without modification, although the resulting marginals will no longer be guaranteed to be exact.

Imagine that the following factor graph is a tree. Define $R_{\alpha i}$ to be the set of variables contained in the unique subtree which is rooted at i and contains the factor α (but no other factors neighbouring i).



(2.8)

We would like to know the entries of the marginal distribution of x_i:

$$P(x_i) = \frac{Z_i(x_i)}{Z = \sum_{x_i} Z_i(x_i)} \tag{2.9}$$

where

$$Z_i(x_i) \equiv \sum_{x_{\setminus i}} \prod_{\alpha} \psi_\alpha(x_\alpha) \tag{2.10}$$

Because of the tree structure, $Z_i(x_i)$ is actually a product of summations over independent sets of variables. We can make this explicit using the subtree region definitions $R_{\alpha i}$:

$$Z_i(x_i) = \prod_{\gamma \sim i} \left(m_{\gamma i}(x_i) \equiv \sum_{x_{R_{\gamma i} \setminus i}} \prod_{\delta \in R_{\gamma i}} \psi_\delta(x_\delta) \right) \tag{2.11}$$

where we have introduced the new "message" quantities $m_{\alpha i}(x_i)$ defined for every variable i and neighbouring factor $\alpha \sim i$. But we can rewrite these

quantities using messages from the neighbours of the neighbours of i:

$$m_{\alpha i}(x_i) = \sum_{x_{R_{\alpha i} \setminus i}} \prod_{\beta \in R_{\alpha i}} \psi_\beta(x_\beta) \tag{2.12}$$

$$= \sum_{x_{\alpha \setminus i}} \psi_\alpha(x_\alpha) \prod_{j \sim \alpha} \prod_{\beta \sim j \setminus \alpha} \left(\sum_{x_{R_{\beta j} \setminus j}} \prod_{\gamma \in R_{\beta j}} \psi_\gamma(x_\gamma) = m_{\beta i}(x_i) \right) \tag{2.13}$$

Thus the messages can be computed, for all α and $i \sim \alpha$, as:

$$m_{\alpha i} = \sum_{X_{\alpha \setminus i}} \psi_\alpha \prod_{j \sim \alpha} \prod_{\beta \sim j \setminus \alpha} m_{\beta i} \tag{2.14}$$

(here we have used the more concise notation from the end of section 2.1, which omits arguments of functions of the state x)

These are the BP message updates. When the underlying graph is a tree, no recursive dependencies are implied in these equations, and the "messages" can be computed with two passes over all the variables. Equations 2.15 and 2.16 can be used to calculate the true marginals. On a loopy graph, the updates can be made "destructive" - messages are initialised (say, to uniform values), and each update computes a new value with which to replace the old, repeating until convergence. Also, on a loopy graph, the normalisation of the messages no longer has a meaning in terms of Z_i, and the messages must be renormalised after each update in order to preserve stability. The marginals calculated on a loopy graph will be approximate, but are often close to the true marginals.

Note that the marginal distribution or "beliefs" b_i or b_α over one or more variables can be computed (exactly in the case of a tree) by multiplying the factors in that region by all the "incoming" messages. For single node beliefs $b_i(x_i)$ and factor beliefs $b_\alpha(x_\alpha)$ this gives

$$b_i(x_i) \propto \prod_{\alpha \sim i} m_{\alpha i}(x_i) \tag{2.15}$$

$$b_\alpha(x_\alpha) \propto \psi_\alpha(x_\alpha) \prod_{i \sim \alpha} \prod_{\beta \sim i \setminus \alpha} m_{\beta i}(x_i) \tag{2.16}$$

There are other ways of writing the messages. Sometimes a "dual form", which emphasises a kind of duality between variables and factors, is given as in section 3.3.2.

From equations 2.15 and 2.16 we can see that the message updates are equivalent to

$$m_{\alpha i}(x_i) \leftarrow \frac{\sum_{x_{\alpha \setminus i}} b_\alpha(x_\alpha)}{b_i(x_i)} m_{\alpha i}(x_i) \tag{2.17}$$

Note that the $m_{\alpha i}$ appearing explicitly in the right-hand side cancels with the factor appearing implicitly in the calculation of the denominator b_i, so that the right-hand side has no functional dependence on $m_{\alpha i}$, and the update takes the form of a projection. (A view of message passing in terms of projections is explored by Minka (2005),[41] which unifies EP, FBP, TRW-BP, and MF.)

This form gives an easy derivation of the GBP parent-child updates (other message definitions and updates are possible, see Yedidia et al 2005[42]) which we now describe. A GBP algorithm is parametrised by a set of user-defined regions $\mathcal{R}$. Each region comprises a set of variables. $\mathcal{R}$ is closed under intersection. For $r, s \in \mathcal{R}$, introduce the relationship $s \lesssim r$ which means "s is a direct subregion of r", i.e. $s \subset r$ and there is no $t \in \mathcal{R}$ such that $s \subset t \subset r$. For each $s \lesssim r$, a message $m_{rs}(x_s)$ is defined. The parent-child message updates can be written in analogy to 2.17:

$$m_{rs} \leftarrow \frac{\sum_{X_{r \setminus s}} b_r}{b_s} m_{rs} \tag{2.18}$$

where the b's should be expanded as:

$$b_r \equiv \left(\prod_{\alpha \subseteq r} \psi_\alpha \right) \left(\prod_{r' \cap r = s} m_{r's} \right) \tag{2.19}$$

As above, the right-hand side of the update does not depend on m_{rs} and so the update is a projection. By expanding and then cancelling like terms, it can also be written

$$m_{rs} \leftarrow \frac{\sum_{X_{r \setminus s}} \left(\prod_{\substack{\alpha \subseteq r \\ \alpha \not\subseteq s}} \psi_\alpha \right) \prod_{\substack{r' \cap r = s' \\ s' \not\subseteq s}} m_{r's'}}{\prod_{\substack{r' \subsetneq r \\ r' \cap s = s'}} m_{r's'}} \tag{2.20}$$

[41] T. Minka. "Divergence measures and message passing". In: *Microsoft Research, Cambridge, UK, Tech. Rep. MSR-TR-2005-173* (2005).

[42] JS Yedidia, WT Freeman, and Y. Weiss. "Constructing free-energy approximations and generalized belief propagation algorithms". In: *IEEE Transactions on Information Theory* 51.7 (2005), pp. 2282–2312.

which corresponds to equation 7 in Yedidia et al 2001.[43]

The stable fixed points of loopy belief propagation updates correspond to local minima of the "Bethe Free Energy":[44]

$$F_{\text{Bethe}}(b;\psi) \equiv \sum_{\alpha}\sum_{x_\alpha} b_\alpha(x_\alpha)\log\frac{b_\alpha(x_\alpha)}{\psi_\alpha(x_\alpha)} + \sum_i (1-|i|)\sum_{x_i} b_i(x_i)\log b_i(x_i) \tag{2.21}$$

Similarly, GBP finds local minima of the "Kikuchi free energy":

$$F_{\text{Kikuchi}}(b;\psi) \equiv \sum_r c_r \sum_{X_r} b_r \log\frac{b_r}{\varphi_r} \tag{2.22}$$

where $\varphi_r \equiv \prod_{\alpha\in r}\psi_\alpha$ and the "overcounting numbers" c_r are defined as

$$c_r = 1 - \sum_{t\supset r} c_t \tag{2.23}$$

Sometimes, Loopy BP or GBP updates fail to converge. This problem is worsened in models with strong factors.[45] It can be ameliorated with damping; however, it is not the case that every local minimum of F_{Bethe} or F_{Kikuchi} corresponds to a *stable* fixed point of the BP or GBP message updates, and in cases where it does not, damping will not help.[46] However, there are algorithms which are stable at all local minima of these free energies, which are based on a kind of coordinate ascent between two convex functions[47] and can also be seen as bounding the free energy and optimising this bound.[48] These are called "double-loop" algorithms because they consist of an inner loop and an outer loop, both of which are run to convergence. They tend to

[43] Yedidia, Freeman, and Weiss, "Generalized belief propagation", op. cit.

[44] T. Heskes. "Stable fixed points of loopy belief propagation are minima of the Bethe free energy". In: *Advances in Neural Information Processing Systems 15*. MIT Press. 2003, p. 359.

[45] JM Mooij and HJ Kappen. "On the properties of the Bethe approximation and loopy belief propagation on binary networks". In: *Journal of Statistical Mechanics: Theory and Experiment* 2005 (2005), P11012.

[46] Heskes, "Stable fixed points of loopy belief propagation are minima of the Bethe free energy", op. cit.

[47] A. L. Yuille and A. Rangarajan. "The Concave-Convex Procedure (CCCP)". In: *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press, 2002, pp. 1033–1040.

[48] T. Heskes, K. Albers, and B. Kappen. "Approximate inference and constrained optimization". In: *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence*. Vol. 13. 2003, pp. 313–320; Heskes, "Stable fixed points of loopy belief propagation are minima of the Bethe free energy", op. cit.

be much slower than BP or GBP, but we use the HAK version[49] of this idea in our experiments with GBP because its convergence properties are more important to us than speed.

2.3.2 Gibbs sampling

The simplest MCMC method is Gibbs sampling. Each sample consists of a state, or complete assignment of all of a model's variables, $x = x^*$. Each Gibbs update consists of resampling one variable from its distribution conditional on its neighbours,

$$x_i \leftarrow P(x_i|x^*_{\setminus i}) \tag{2.24}$$

which distribution can easily be calculated from the model's local factorisation:

$$P(x_i|x^*_{\setminus i}) \propto \prod_{\alpha \sim i} \psi_\alpha(x_i, x^*_{\alpha \setminus i}) \tag{2.25}$$

A new sample is typically formed by updating all the variables in this way, following some predefined order of the variables (although variants are possible). It is easy to check that each update satisfies *detailed balance*, and that the sequence of updates is *positive recurrent* (see Neal[50] for definitions of these terms) and this implies that the samples will converge to draws from the true distribution P in the infinite limit.

2.4 Generality of factor graphs

We motivate our decision to express our statistical models using discrete factor graphs rather than an alternative formalism. Below, we consider other graphical representations with richer independence structure. In section 2.4.2, we explain why we prefer models with discrete rather than continuous variables. In section 2.4.3, we discuss several restrictions that are often placed on factor graph structure and dimensionality, and we prove some original results characterising the universality of these restricted classes.

[49] Heskes, Albers, and Kappen, "Approximate inference and constrained optimization", op. cit.

[50] R.M. Neal and University of Toronto. Department of Computer Science. *Probabilistic inference using Markov chain Monte Carlo methods*. 1993.

2.4.1 Independence structure

Statistical models are often specified by decomposing or *factoring* the (possibly unnormalised) joint distribution $P(x_{1:n})$ into a product of functions of subsets of variables, as in equation 2.2. Such a decomposition is called a (probabilistic) graphical model. A factor graph, defined above, is a very general kind of graphical model, but in many cases a model will be given in terms of a factorisation which contains more structure than a simple factor graph. For instance, we define a Bayesian causal network, also called a causal network, or influence diagram, or belief network, as specifying a statistical model in terms of a procedure for sampling from the model's distribution:

$$P(x) = \prod_i P(x_i | x_{\gtrsim i}) \tag{2.26}$$

where $x_{\gtrsim i}$ indicates the set of parents of x_i (not ancestors) in a directed acyclic graph with all the variables as vertices.[51] The graph is acyclic, so we can find a total ordering consistent with it. Sampling variables one at a time according to such a total order is straightforward because the conditional distribution for x_i is given in terms of the parents $x_{\gtrsim i}$ whose values will already have been fixed when it is time to sample x_i. The name "causal network" comes from the existence of this natural sampling procedure, which may be a representation of a similar "causal" procedure in the "real-world" system that is being modelled.

But a causal network is less general than a factor graph because, aside from the fact that it yields a distribution which is already normalised, each factor also obeys its own normalisation rule:

$$\sum_{x_i} P(x_i | x_{\gtrsim i}) = 1 \qquad (\forall\ x_{\gtrsim i} \in \mathcal{X}_{\gtrsim i}) \tag{2.27}$$

Thus, the factors in a causal network enjoy additional structure. Such structure is commonly characterised in terms of its effect on the sets of independencies satisfied by variables in the model. Given three sets of variable indices, U, V, and W, we say that x_U is independent from x_V given x_W, or

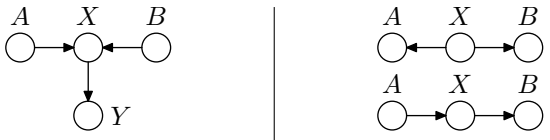
$$x_U \perp\!\!\!\perp x_V | x_W \tag{2.28}$$

when

$$P(x_U, x_V | x_W) = P(x_U | x_W) P(x_V | x_W) \qquad \forall x_W \tag{2.29}$$

[51] J. Pearl. *Causality: models, reasoning, and inference*. Cambridge University Press, 2000.

In a factor graph, such a relation holds (for arbitrary factors) when all paths between a variable in U and a variable in V in the connectivity graph contain a variable in W. In a causal network, the conditional independence rules are more complicated. For instance, here are three network structures and some of the conditional independencies satisfied by them:



$$A \perp\!\!\!\perp B \qquad A \not\perp\!\!\!\perp B$$
$$A \not\perp\!\!\!\perp B \mid X \qquad A \perp\!\!\!\perp B \mid X$$
$$A \not\perp\!\!\!\perp B \mid Y \tag{2.30}$$

The general rule is given by the "d-separation criterion",[52] whose cases are summarised by the above examples.

Causal networks are very general, and some important inference algorithms, such as BP,[53] were first defined on causal networks. It could be argued that since almost all real-world probabilistic systems are characterised by a procedure for drawing samples, and since a causal network can be used to define such a procedure, so all statistical inference algorithms should be defined in terms of causal networks.

We outline two reasons for not adopting such a convention. The first reason is that specifying certain models depends on the greater generality of factor graphs. For example, factor graphs are closed under the operation of "marginalising out" (summing over) a variable, but causal networks are not. Marginalising out A in the network $B \leftarrow A \rightarrow C \leftarrow D$ results in a graph with three variables whose independence relations are not consistent with any causal network. Instead, it is an example of an acyclically directed mixed graph, a generalisation of a causal network, with its own independence rules.[54]

[52] Ibid.

[53] Idem, "Reverend Bayes on inference engines: A distributed hierarchical approach", op. cit.

[54] T. Richardson. "Markov properties for acyclic directed mixed graphs". In: *Scandinavian Journal of Statistics* 30.1 (2003), pp. 145–157; T.S. Richardson. "A factorization criterion for acyclic directed mixed graphs". In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press. 2009, pp. 462–470.

Thus, when not every variable in the model can be easily characterised or enumerated, but rather some unspecified variables should be considered as having been marginalised out of the model, the causal network framework is not sufficient. Another example is the Ising model which models the distribution over spins in a magnetic solid:

$$P(s) \propto \exp\ \frac{1}{T}\left(J\sum_{i\sim j} s_i s_j + h\sum_i s_i\right) \tag{2.31}$$

where $s_i \in \pm 1$. Here, J defines the strength of interactions between spins, h is the strength of an external field, T the temperature, and $\sim$ defines an adjacency relation (usually representing adjacency on a two-dimensional grid). We could argue that the distribution indeed possesses a sampling procedure, like a causal network: pick a random spin, set it to a draw from its distribution conditional on its neighbours, and repeat forever (as in Gibbs sampling). But the resulting causal network has a node corresponding to each spin at every instant of time, and the final distribution of interest appears only at infinite depth in the network, in the limit where the spins have reached equilibrium. On the other hand, it is straightforward to represent the Ising model in a finite form as a product of factors (by distributing the "exp" across the sum).

The second reason for using factor graphs is that many algorithms which have been specially defined for causal networks are straightforward to generalise to factor graphs. There are some exceptions, two of which are described in section 3.2, but we do not find these inspiring. It is relatively more common for recent work on approximate inference algorithms to be expressed in terms of factor graphs. Perhaps it is a shortcoming of our algorithms that they are not able to leverage the additional information encoded in the causal network structure, but for the ideas considered in this dissertation, factor graphs were sufficient.

2.4.2 Discreteness

We have chosen to consider only discrete factor graphs, i.e. those whose variables take values in a finite domain (this is a slight abuse of the term "discrete", which traditionally might also include countably infinite domains such as the integers). This is a real restriction, since many useful models contain continuous-valued variables. As in the previous section, there are two main reasons for adopting such a restriction. One reason is that it is sometimes feasible to discretise continuous models by partitioning the state

space of each variable. A second reason is that it is often straightforward to generalise algorithms which have been defined on discrete variables to the continuous setting. For instance BP and MF can be generalised to EP, which applies to models with continuous variables. There are natural ways to generalise the algorithms presented in this dissertation to apply to continuous models, but working through such a generalisation does not seem instructive or useful at this point.

Working with continuous-valued variables results in additional complexity which comes from deciding on a parametrisation for continuous distributions (Gaussian, mixture of Gaussian), calculating projections to this parametric family, deciding how to generate example models, how to report error in marginals, and how to optimise and schedule simulations on a potentially much larger number of dimensions. Interesting instances where subclasses of continuous models are simple and tractable, such as Gaussian Belief Propagation,[55] exist, but the properties of such special cases do not seem to generalise to broader classes of models. For the general case there is an overhead to using continuous variables and at least at first glance it seems potentially more productive to concentrate on inference tasks in models with many variables, each taking values in a small domain, than tasks on models with a few variables taking values in a large or continuous domain.

The case of continuous-valued variables is only one of many generalisations which we could consider here but choose not to. Non-parametric Bayesian models deal with processes on various unbounded domains, which could correspond to factor graphs with a countably infinite (Beta process) or uncountably infinite (Gaussian process) number of continuous-valued variables. Non-parametric models will no doubt yield interesting connections to approximate inference, but at the present it seems better to concentrate on the discrete finite case, since we aren't specifically concerned with inference in such models here.

2.4.3 Converting between classes of discrete factor graphs

In this section we will consider ways of reducing the problem of inference in general discrete factor graphs to that of inference in restricted classes of discrete factor graphs. For us, this means converting the general input problem into the simpler framework, in such a way that the marginals of the converted graph bear a tractable correspondence to those of the original graph. This is similar to the idea of reductions in complexity theory,

[55] Y. Weiss and W.T. Freeman. "Correctness of belief propagation in Gaussian graphical models of arbitrary topology". In: *Neural computation* 13.10 (2001), pp. 2173–2200.

discussed in section 2.5. Such reductions are of interest because there are many inference algorithms, decompositions, and other results which apply only to restricted classes of factor graphs. In the subsections that follow, we discuss conversions of general discrete factor graphs to pairwise, binary, binary pairwise, and planar binary pairwise form. With the exception of the first conversion, the results we present are original and we are not aware of prior publication. We mention the results here in this background chapter because they answer important basic questions about the generality of the models we consider. We describe them in detail because, although not directly relevant to the rest of this dissertation, most of them are not described elsewhere and so cannot be abstracted.

We start by defining a "partial assignment" (PA) as an assignment of values to a subset of a model's variables. We use a notion of "representation" of a model G by a model H which can be defined as an injective correspondence f between PAs in G and PAs in H, which preserves probabilities and containments: for $r, s \subseteq \mathcal{V}$ and given $f(x_r = x_r^*) = (y_s = y_s^*)$ we have

$$P_G(x_r = x_r^*) = P_H(y_s = y_s^*) \tag{2.32}$$

and if additionally $r', s' \subseteq \mathcal{V}$ with $r \subseteq r'$ and $f(x_{r'} = x_{r'}^*) = (y_{s'} = y_{s'}^*)$ and $(x_{r'}^*)_r = x_r^*$ then we have

$$s \subseteq s' \tag{2.33}$$

and

$$(y_{s'}^*)_s = y_s^* \tag{2.34}$$

This relates to a kind of event algebra, and perhaps it is possible to generalise or weaken the above definition or say more about its formal properties, but the present form suffices for the material below. It should be clear that such a correspondence satisfies the primary desiderata, namely that it is easy to compute marginals in G from those in H, which implies that inference in G can be accomplished via inference in H. Normally we will only be interested in cases where a correspondence can be constructed in polynomial time in the size of the model, and this should be understood implicitly when we claim that one type of factor graph can be *converted* into another type of factor graph in the following sections.

As a special case, the notion of representation of G by H includes situations where the variables of G are included in H, but H has extra "latent" variables which need to be "marginalised out". The pairwise conversion

defined in Yedidia 2003, for example, conforms to our definition of representation because the variables in the new graph are a superset of the variable of the original graph, and the lifted containment relation suffices for the injective correspondence f.[56]

The requirement that the PA map f must be containment-preserving implies that for any $r \subseteq \mathcal{V}$, we have $f(x_r = x_r^*) = \bigcap_{i \in r} f(x_i = x_i^*)$, in other words the value of f at a particular PA can be found by intersecting the values of f at each variable in the PA.

If we had allowed our PA maps to be multi-valued, so that a PA in G could map under f to a union of PAs in H, then the expression $f(x_r = x_r^*) = \bigcap_{i \in r} f(x_i = x_i^*)$ would expand to a union of a number of PAs which is exponential in the size of r. As a result, calculating the probability of $x_r = x_r^*$ in G would correspond to adding up the probabilities of an exponential number of PAs in H. This would be at odds with our interest in polynomial-time conversions between graphs.[57]

2.4.3.1 Pairwise factor graphs

A common restriction imposed on factor graphs is to require all factors to have size 2; or size 1 or 2. Since singleton (size 1) factors can be incorporated into pairwise (size 2) factors (provided that there are no isolated variables), we consider both cases as roughly equivalent and refer to graphs which satisfy either restriction as "pairwise" factor graphs. Arbitrary (n-wise) factor graphs can be converted into pairwise form. The following theorem was outlined in Yedidia 2003 in a slightly different form:

Theorem 1. *Any factor graph can be converted to the form of a pairwise factor graph*

Proof. One way to effect this conversion is to create a variable (i or (α)) for each variable (i) and factor (α) in the old graph, introduce singleton factors $\{(\alpha)\}$ for each α and pairwise factors $\{i, (\alpha)\}$ for each $i \sim \alpha$, and assign to

[56] J.S. Yedidia, W.T. Freeman, and Y. Weiss. "Understanding belief propagation and its generalizations". In: *Exploring Artificial Intelligence in the New Millennium* (2003), pp. 239–236, p. 13.

[57] The drawback of not allowing multi-valued PA maps is that our conversions may not be invertible. Consider the conversion of Theorem 5: one assignment of a binary variable in the range can correspond to multiple values in the domain.

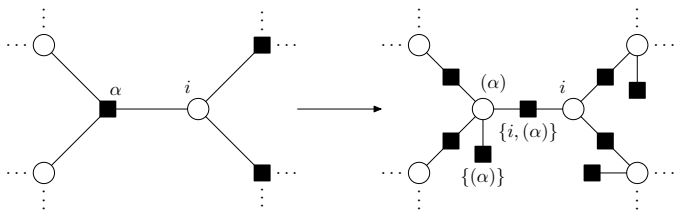
these factors the following potentials:

$$\hat{\psi}_{\{i,(\alpha)\}}(\hat{x}_i, \hat{x}_{(\alpha)}) = \begin{cases} 1 & \text{if } \hat{x}_i = [\hat{x}_{(\alpha)}]_i \\ 0 & \text{otherwise} \end{cases} \tag{2.35}$$

$$\hat{\psi}_{\{(\alpha)\}}(\hat{x}_{(\alpha)}) = \psi_\alpha([\hat{x}_{(\alpha)}]) \tag{2.36}$$

where the domains are $\hat{\mathcal{X}}_i = \mathcal{X}_i$ and $\hat{\mathcal{X}}_{(\alpha)} = (\mathcal{X}_\alpha)$. Here [ ] is used as a kind of inverse of ( ), so $x_\alpha = [\hat{x}_{(\alpha)}]$ indicates the set of variable assignments x_α (in the old graph) corresponding to the single variable assignment $\hat{x}_{(\alpha)} = (x_\alpha)$ (in the new graph).

The new pairwise potentials are constructed to enforce consistency between the representatives of the old variables and copies of them appearing in representatives of the old factors by assigning zero weight to illegal states, and the new singleton potentials incorporate the values of the old factors to ensure that the legal states have the same weight as in the original graph. The transformation is illustrated in the following diagram:



(2.37)

□

It is straightforward to check that, for the above construction, Belief Propagation on the converted pairwise graph is equivalent to BP on the original graph. MF, interestingly, does not carry over. (None of the other transformations we describe share this property of preserving BP fixpoints.)

2.4.3.2 Binary pairwise factor graphs

The next restriction we consider is to require all variable domains to have size two. Factor graphs satisfying this restriction are called "binary". This restriction is usually combined with the first, resulting in "binary pairwise" factor graphs. Several algorithms and decompositions have been proposed which only apply to binary pairwise graphs, such as Belief Optimization;[58]

[58] M. Welling and Y.W. Teh. "Belief optimization for binary networks: A stable alternative to loopy belief propagation". In: *Uncertainty in Artificial Intelligence* (2001).

the self-avoiding-walk (SAW) tree expansion[59] and loop decompositions,[60] so it is interesting to ask if it is possible to convert more general factor graphs to the binary pairwise form. Typically, such a conversion would operate by first converting the input to binary form, by choosing a binary representation for the input variables, and then adding auxiliary (latent) variables to implement the correct distribution over the new binary graph. We show that the second step is not possible in general:

Theorem 2. *There exist factor graphs which cannot be converted to binary pairwise form*

Proof. This is because the positive states of a binary pairwise factor graph correspond to solutions of 2-SAT, which obey a special *median graph* structure.

2-SAT is a special case of k-SAT, the problem of finding satisfying assignments to boolean formulae of the form

$$\bigwedge_c \left((\bigvee_{i \in c^+} v_i) \vee (\bigvee_{i \in c^-} \neg v_i) \right) \tag{2.38}$$

where c runs over a set of "clauses". Each clause $c \equiv c^+ \cup c^-$ is a set of variables, some of which are negated (those in c^-). Each clause c has size less than or equal to k: $|c| \leq k$. Here $\wedge$ signifies conjunction (binary "and") and $\vee$ signifies disjunction (binary "or"). Thus, the formula is a conjunction of disjunctive clauses - for the formula to be true, every clause must be satisfied, which means that at least one of the clause's positive variables must be true, or at least one of its negative variables must be false. For $k \geq 3$, we can find a k-SAT instance where a given set of assignments to some variables, and no other, satisfies the formula (possibly by introducing extra variables). This is not possible with 2-SAT, however, whose satisfying assignments form a structure called a "median graph" and have the property that given a

[59] K. Jung and D. Shah. "Inference in binary pair-wise markov random field through self-avoiding walk". In: (2006); D. Weitz. "Counting independent sets up to the tree threshold". In: *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*. ACM. 2006, pp. 140–149.

[60] Erik Sudderth, Martin Wainwright, and Alan Willsky. "Loop Series and Bethe Variational Bounds in Attractive Graphical Models". In: *Advances in Neural Information Processing Systems 20*. Cambridge, MA: MIT Press, 2008, pp. 1425–1432; Y. Watanabe and K. Fukumizu. "Loop series expansion with propagation diagrams". In: *Journal of Physics A: Mathematical and Theoretical* 42 (2009), p. 045001; Y. Watanabe and K. Fukumizu. "Graph zeta function in the Bethe free energy and loopy belief propagation". In: *Advances in Neural Information Processing Systems 22* (2009), pp. 2017–2025.

set of three satisfying assignments, if we construct a new assignment (the "median" of the three) in which each of the variables take the values they took in the majority of the other assignments, then the new assignment is also satisfying.[61]

Assume that the input graph is binary. Call this graph P and let it as usual be given by $P(x) = \frac{1}{Z}\prod_\alpha \psi_\alpha(x_\alpha)$.

We see that a state x^* has positive probability if and only if the following boolean expression is true:

$$\bigwedge_{\alpha\in\mathcal{F}} \bigwedge_{\substack{x_\alpha\in\mathcal{X}_\alpha \\ \psi_\alpha(x_\alpha)=0}} \bigvee_{i\in\alpha} x_i \neq x_i^* \tag{2.39}$$

Introduce a boolean variable v_i which is true if $x_i^* = 1$ and false otherwise; the expression becomes:

$$\bigwedge_{\alpha\in\mathcal{F}} \bigwedge_{\substack{x_\alpha\in\mathcal{X}_\alpha \\ \psi_\alpha(x_\alpha)=0}} \left(\left(\bigvee_{\substack{i\in\alpha \\ x_i^*=1}} v_i \right) \vee \left(\bigvee_{\substack{i\in\alpha \\ x_i^*=0}} \neg v_i \right) \right) \tag{2.40}$$

The positive states of P are thus instances of k-SAT, where k is the size of the largest factor in P. Any set of states can be realised as a solution set of k-SAT (perhaps by introducing auxiliary variables) when $k \geq 3$. But when $k = 2$, such sets must obey the median rule defined above. If we can show that our definition of reduction preserves lack of median structure, then we are done: an arbitrary model P (without median structure) cannot be reduced to a binary pairwise model Q (with median structure).

Let f be the PA-map of a representation of $P(x)$ by $Q(y)$, where Q has median structure. Consider a triple of states $x^{(1)}, x^{(2)}, x^{(3)}$ in P (i.e. these are full, not partial, assignments), each with positive probability, and let x^* be their median. These map under f to a triple of PAs $y_{r_1}^{(1)}$, $y_{r_2}^{(2)}$, $y_{r_3}^{(3)}$ in Q. Since each PA $y_{r_i}^{(i)}$ has positive probability $Q(y_{r_i}^{(i)}) = P(x^{(i)})$, it can be extended to a state $y^{(i)}$ with positive probability. The median of these three states let us call y^*. Since we assumed the median property for Q, we have $Q(y^*) > 0$. Now we would like to show that y^* is an extension of $f(x^*)$. This follows from the variable intersection rule for PA maps: $f(x) = \bigcap_i f(x_i)$. More specifically, let $i \in \mathcal{V}_P$. Since x_i^* is a median of $(x_i^{(1)}, x_i^{(2)}, x_i^{(3)})$, it must have the same value of two of these - say, WLOG, $x_i^{(1)}$ and $x_i^{(2)}$. But $y_{r_1}^{(1)}$

[61] T. Feder. "Network flow and 2-satisfiability". In: *Algorithmica* 11.3 (1994), pp. 291–319. ISSN: 0178-4617.

and $y^{(2)}_{r_2}$ will then both be consistent with $f(x^*_i) = f(x^{(1)}_i) = f(x^{(2)}_i)$. As a consequence, y^* will share this consistency: any variable which is fixed in $f(x^*_i)$ will appear in both $y^{(1)}$ and $y^{(2)}$ and hence y^*. Since we have shown y^* is consistent with $f(x^*_i)$ for all i, it follows that y^* must be an extension of $f(x^*)$. Now, $Q(y^*) > 0$ since we assumed Q to have median structure. But $y^* \in f(x^*)$ so $P(x^*) = Q(f(x^*)) \geq Q(y^*) > 0$. Thus x^* has positive probability in P. Hence, P has median structure.

We have proven that our reductions preserve lack of median structure, from which it follows that inference in a model whose positive states lack median structure cannot be reduced to inference in a binary pairwise factor graph. We have indicated that general factor graphs do not have median structure: see equation 2.42 for a concrete counterexample, the "XOR distribution".

□

Below we demonstrate the manner in which median structure is preserved by our notion of representation, for a toy example. The PA map of the example representation is given by $f(s_1 = 1) = \{t_1 = 1\}$, and $f(s_1 = -1) = \{t_1 = -1, t_2 = -1\}$, and $f(s_3 = -1) = \{t_6 = 1\}$, and so on. One can check that any assignment of values to the "wildcard" variables (marked "?") in the PAs above the line will be consistent with some refinement of the median PA below the line:

	s_1	s_2	s_3	t_1	t_2	t_3	t_4	t_5	t_6
	-1	1	1	-1	-1	?	1	-1	?
	1	-1	1	1	?	-1	?	-1	?
	1	1	-1	1	?	?	1	?	1
median:	1	1	1	1	?	?	1	-1	?

(2.41)

A concrete example of a distribution which is not representable with a binary pairwise graph is the "XOR distribution":

$$P(s_1, s_2, s_3) = \begin{cases} \frac{1}{4} & \prod_i s_i = -1 \\ 0 & \text{otherwise} \end{cases} \tag{2.42}$$

where $s_i \in \pm 1$. The median structure demands that "111" has a positive probability, since the following three positive configurations each have a

majority of 1 for each variable:

	s_1	s_2	s_3
	-1	1	1
	1	-1	1
	1	1	-1
median:	1	1	1

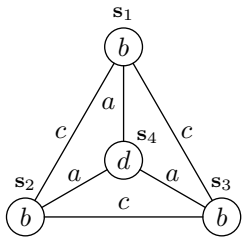
(2.43)

But the distribution assigns it a zero probability.

However, it is possible to construct a limit of binary pairwise graphs which approaches the XOR distribution with arbitrary precision. This is because it is possible to implement the following distribution as a binary-pairwise factor graph, for finite k:

$$P(s_1, s_2, s_3) = \exp\left(k \prod_{i=1}^{3} s_i\right) \tag{2.44}$$

The following explicit construction is due to Martijn Leisink.[62] Introduce an auxiliary variable s_4, and create a network:



(2.45)

with weights shown (corresponding to factors $\exp(as_1s_4)$, $\exp(bs_1)$, etc.), having values:

$$b = \frac{k}{4\,|k|}\,\mathrm{acosh}\left(e^{4|k|}\right) \tag{2.46}$$

$$c = -\,|b| \tag{2.47}$$

$$a = \frac{-k}{4\,|k|}\,\mathrm{acosh}\left(e^{8|b|}\right) \tag{2.48}$$

$$d = |a| \tag{2.49}$$

[62] Martijn Leisink. Personal communication. Jan. 2010.

This set of weights is not unique, since although there are 4 unknown weights and 4 unique (up to permutation) values for the state $s_{1:3}$, the partition function of the new model is an extra degree of freedom which can be constrained by the simplifying choice, $d = |a|$, from which follows $c = -|b|$ and the other two equations.

It is straightforward to check that the network induces the distribution P on $s_{1:3}$ when marginalising out s_4. This allows us to prove the following theorem:

Theorem 3. *Every factor graph can be represented arbitrarily closely by a binary pairwise graph*

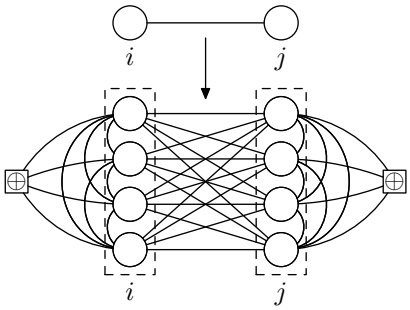
More precisely, the size of the output binary pairwise graph is a fixed function of the input graph, and only the parameters must change in order obtain a closer approximation to the input graph.

We note that since it has been known that inference in binary pairwise factor graphs is NP-hard (see for instance Barahona[63]), it follows that these models are expressive enough to represent NP-complete problems such as SAT, even though working out the details of such a representation might be cumbersome. However, it is not necessarily clear how to represent other NP-hard problems, such as inference in general factor graphs, using the binary pairwise form. As we saw in the previous theorem, it is not the case that there is a simple correspondence.

Proof. Assume, WLOG, that the original graph is in pairwise form. Now create a new graph with a variable $k = (i, x_i)$ for each of the (variable, value) pairs in the old graph, which will be by construction $\hat{x}_k = 1$ if the variable i takes state x_i, and $\hat{x}_k = 0$ otherwise. Introduce an edge $((i, x_i), (j, x_j))$ for each edge (i, j) in the old graph and each pair of states (x_i, x_j), with factor potentials equal to 1 if either $\hat{x}_{(i,x_i)} = 0$ or $\hat{x}_{(j,x_j)} = 0$ and equal to $\psi_{ij}(x_i, x_j)$ otherwise. One can see that this graph has an unnormalised joint which coincides with that of the original graph for each allowed state. We still need to exclude states where a variable takes "multiple values", i.e. $\hat{x}_{(i,x_i)} = \hat{x}_{(i,x'_i)} = 1$ for $x_i \neq x'_i$, and we need to ensure that at least one $\hat{x}_{(i,x_i)}$ is 1. For each variable i, introduce an edge $((i, x_i), (i, x'_i))$ for each pair of values $x_i \neq x'_i$ with factor potential equal to zero if both x_i and x'_i are 1, and equal to 1 otherwise. This ensures that no more than one $\hat{x}_{(i,x_i)}$ is 1 for each i, but the remaining case where $\hat{x}_{(i,x_i)} = 0$ for all x_i is not yet excluded by the new graph. We can exclude it by introducing a new XOR factor of

[63] F. Barahona. "On the computational complexity of Ising spin glass models". In: *Journal of Physics A: Mathematical and General* 15 (1982), p. 3241.

size $|\mathcal{X}_i|$ which ensures that an odd (and therefore non-zero) number of the $\hat{x}_{(i,x_i)}$ are equal to 1. The following diagram describes the transformation for the case $|\mathcal{X}_i| = |\mathcal{X}_j| = 4$ (the XOR factors are marked $\oplus$):



(2.50)

It is easy to see that an XOR factor of size n can be constructed by combining $n-2$ XOR factors of size 3 (and with a single edge when $n = 2$). Since XOR factors of size 3 can be achieved as a limit of binary pairwise graphs (by letting $k \to \pm\infty$ in 2.44) this completes the proof. □

This also shows

Corollary 4. *Any discrete factor graph can be converted to binary 3-wise form*

Since the 3-wise to pairwise transformation of Leisink only breaks down in the presence of potential functions with zero entries, we ask whether it is possible to perform the binary pairwise conversion without resorting to a limit when graph potentials are strictly positive.

The answer is "yes". We prove this result in two parts. First, we show how to convert a general factor graph with positive entries to positive-entry binary n-wise form. Then we describe how to convert a binary n-wise graph with positive entries into binary-pairwise form.

Theorem 5. *Any discrete factor graph can be converted to binary form, in such a way that if the original graph had strictly positive potentials then the output graph also has strictly positive potentials*

Proof. Choose a binary encoding for the values in each variable's domain $\mathcal{X}_i$. The encoded values may contain different numbers of bits, since $|X_i|$ may not be a power of 2. The encoding will correspond to a binary tree with $|X_i|$ leaves. In the new graph, for each variable i introduce k_i binary variables, where k_i is the maximum depth of the tree. To these variables

attach factors whose entries at a given (binary) assignment correspond to the entries of factors in the original graph at the decoding of that assignment. Lastly, we need to compensate for the fact that a single variable assignment x_i in the original graph may correspond to multiple assignments of the k_i binary variables, due to the presence of extra unused variables when x_i is encoded with fewer than k_i bits. To this end, attach a factor to the k_i binary variables corresponding to the variable i, with values corresponding to $2^{f_i(x_i)-k_i}$, where $f_i(x_i)$ is the length of the encoding of x_i. This ensures that summing over the unused variables gives the correct probability of an assignment in the original graph. □

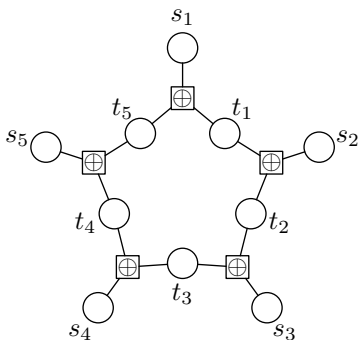
Proposition 6. *The n-wise soft-XOR factor*

$$P(s) \propto \exp\left(k \prod_{i=1}^{n} s_i\right) \tag{2.51}$$

(with k finite) can be represented in binary pairwise form.

Proof. A factor with $k \leq 0$ can be represented by a $k \geq 0$ factor by flipping the sign of one of the variables, so assume $k \geq 0$.

Consider connecting the n binary variables $s_{1:n}$ with 3-wise soft-XOR factors, each of strength k', in a loop as shown:



(2.52)

We will prove that for any k we can always find a k' such that the above graph implements the distribution of equation 2.51. The probability of a configuration of the s variables is

$$P(s) \propto \sum_{t} \exp(k'(t_n s_1 t_1 + t_1 s_2 t_2 + \ldots + t_{n-1} s_n t_n)) \tag{2.53}$$

This summation has 2^n terms. Observe that when two states s and s' have the same parity, then the terms in the summation for $P(s)$ are a permutation of those in the summation for $P(s')$. To see this, consider inverting a neighbouring pair of s variables, s_i and s_{i+1}. The effect is the same as inverting t_i, which permutes the terms of the sum over t, leaving the total value invariant. But a sequence of such inversions can be used to go between any s and s' if they have the same parity. In particular, inverting all t_i for which $\prod_{j=1}^{i} s_j = -1$ rearranges the terms to correspond either to $s = (1, 1, \ldots, 1)$ (if $\prod_{j=1}^{n} s_j = 1$) or $s = (-1, 1, \ldots, 1)$ (if $\prod_{j=1}^{n} s_j = -1$). This shows that

$$P(s) = \begin{cases} p_1 & : \prod_{j=1}^{n} s_j = 1 \\ p_2 & : \prod_{j=1}^{n} s_j = -1 \end{cases} \tag{2.54}$$

for some p_1 and p_2, which is the same as saying

$$P(s) \propto \exp(k \prod_i s_i) \tag{2.55}$$

where $k = \frac{1}{2} \log \frac{p_1}{p_2}$.

It remains to verify that the relationship $k' \mapsto k$ can be inverted. At least for small n this relationship appears to be strictly monotonic, but it is not necessary to prove that fact in general. All that is needed is to observe that $k' = 0$ gives $p_1 = p_2 \implies k = 0$, and $k' \to \infty \implies k \to \infty$. The second implication follows from the fact that the term with the largest exponent in p_1 is $\exp(k'n)$ (with coefficient 1), while that in p_2 is $\exp(k'(n-2))$ (with coefficient n) - or simply from observing the values of the network when the 3-wise soft-XORs become XORs. Finally, continuity and the intermediate value theorem imply that for any positive k, we can find a k' such that the above graph (2.52) is equivalent to a n-wise soft-XOR of strength k. □

Corollary 7. *Any n-wise binary factor with strictly positive entries can be implemented in binary pairwise form*

Proof. The 2^n functions $s_1^{e_1} s_2^{e_2} \ldots s_n^{e_n}$ with $e_i \in \{0, 1\}$ form an independent basis for the space of real-valued functions of s, so we can write the factor as $\exp(\sum_e a_e s_1^{e_1} s_2^{e_2} \ldots s_n^{e_n})$ for some coefficients a_e. But this can be implemented by constructing 2^n soft-XOR factors of strength a_e, each covering subsets of the variables defined by e. □

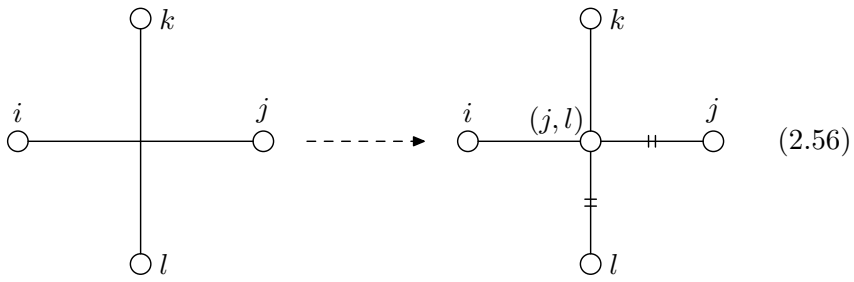
Together with Theorem 5, this proves:

Theorem 8. *Any factor graph with strictly positive factors can be implemented in binary pairwise form*

We note that many of the conversions in this section produce output graphs containing a variable for each member of the domain of each variable and factor in the original graph. Since these variables are usually fully connected by pairwise factors, the number of factors in the output graph is going to be proportional to the square of the size of the domains in the input graph. We do not give precise relationships, but note that even though the complexity of the conversion is polynomial, the resulting graphs may be quite large.

2.4.3.3 Planar binary pairwise graphs

Finally, we address the problem of converting an arbitrary factor graph to planar form. If n-ary variables are allowed in the planar graph, this task is easy: we simply draw the graph in two dimensions, and introduce a new variable wherever two edges cross. The new variable encodes the values at an endpoint of each of the two original edges.



(2.56)

Here, the factor between the new (j, l) variable and j enforces consistency between x_j and $x_{(j,l)}$; similarly for the factor between (j, l) and l (in both cases this is indicated with a double tic). The factors between i and (j, l) and between k and (j, l) copy the entries of ψ_{ij} and ψ_{kl}, respectively.

Finding a conversion for the binary pairwise planar case is more difficult since only two values can be used to propagate data across an intersection. Inference in binary pairwise planar graphs was shown to be NP-hard by Barahona (1982)[64] (although it is tractable in the special case of Ising "spin glasses", with soft-XOR pairwise factors and no singleton factors - also called "pure interaction potentials"[65]) so it is conceivable that there would be a way to convert from ordinary factor graphs to binary pairwise factor graphs.

[64] Ibid.

[65] M.E. Fisher. "On the dimer solution of planar Ising models". In: *Journal of Mathematical Physics* 7 (1966), p. 1776; A. Globerson and T.S. Jaakkola. "Approximate inference

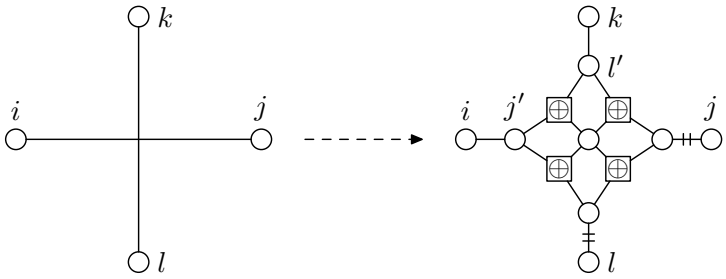
Such a conversion would be of interest because of the existence of a number of results which apply only to the planar binary pairwise case, in approximate inference[66] and statistical physics.[67]

We are not aware of a way to turn arbitrary factor graphs into binary pairwise planar graphs exactly, but it is straightforward to effect such a conversion in a limit.

Theorem 9. *Any discrete factor graph can be represented arbitrarily closely by a planar binary pairwise factor graph.*

As in Theorem 3, the structure of the output graph is fixed and only the parameters must vary to achieve an arbitrarily accurate representation.

Proof. Convert the graph to binary pairwise form as described above, and replace each pair of overlapping edges with the following subgraph, using soft-XOR 3-wise nodes of strength m.



(2.57)

As previously, edges with a double tic enforce the constraint that their endpoint variables match (i.e. in this case they have potentials $\left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$). The factor connecting i and j' in the new graph should be the same as ψ_{ij} in the old graph, and similarly for k and l'.

In the limit as $m \to \infty$, the soft-XOR factors become XOR factors; then, note that l' is forced to take the value of l, and j' to take the value of j. □

using planar graph decomposition". In: *Advances in Neural Information Processing Systems 19* 19 (2007), p. 473.

[66] Globerson and Jaakkola, "Approximate inference using planar graph decomposition", op. cit.; M. Chertkov, V. Gomez, and H. Kappen. *Approximate inference on planar graphs using loop calculus and belief progagation.* Tech. rep. Los Alamos National Laboratory (LANL), 2009.

[67] Fisher, "On the dimer solution of planar Ising models", op. cit.; P.W. Kasteleyn. "Dimer statistics and phase transitions". In: *Journal of Mathematical Physics* 4 (1963), p. 287.

2.4.3.4 Conclusion

We have demonstrated a number of formal conversions between different types of factor graphs, which prove that inference in one class of graphs can be implemented using inference in a more restrictive class. To the best of our knowledge, of the theorems appearing in this subsection only Theorem 1 has been published before.

We summarise the results. General discrete factor graphs can be converted to pairwise form and to binary form and in particular to binary 3-wise form. They can be converted to binary pairwise form if they have positive entries (and some discrete factor graphs with zeroes, such as the binary 3-wise XOR, cannot be implemented in binary pairwise form). If they have zero entries, they can still be represented arbitrarily closely by a binary pairwise graph. Additionally, general discrete factor graphs can be represented arbitrarily closely by a planar binary pairwise graph.

It is also interesting to consider the question of whether we can quantify the extent to which transformations such as the binary pairwise transformation make inference more difficult, either by distributing information across multiple variables, or by introducing frustrated factors (which is to say, factors whose effects tend to almost cancel each other out, as in the case of Leisink's 3-wise-to-pairwise transformation) or factors with very large or very small entries. As in the case of k-SAT vs. 2-SAT, it may be that inference is easier in binary pairwise graphs than more general graphs (which would partly justify the number of algorithms that only apply to binary pairwise graphs). Quantifying the extent to which the binary pairwise transformation amplifies "frustration" or some other measure of inference difficulty would help set bounds on the extent to which binary-pairwise-specific algorithms can be more powerful than more general algorithms. Numerical experiments, not described here[68], indicate that even sophisticated approximate inference algorithms perform poorly in models produced by reductions which make use of pairwise soft-XOR factors, for example the reduction of Theorem 3.

2.4.3.5 Acknowledgements

We are indebted to Martijn Leisink for his (unpublished) pairwise soft-XOR factor, and to Joris Mooij for pointing us to this construction.

[68] These results will be published separately.

2.5 Complexity of inference

This section discusses the computational complexity of inference. It assumes a basic familiarity with the theory of complexity (which is a measure of the difficulty of a computational task), including the complexity classes P and NP, NP-complete, NP-hard, and so forth. A brief glossary is provided for review:

- **Turing machine (TM)** - An idealised computer. The precise definition is not important, since all of the "reasonable" formal architectures which one might define, including that of the original TM, can be simulated on each other with at most a polynomial-time slow-down, leaving the basic complexity classes invariant.
- **Decision problem** - A problem with arbitrary input and boolean (accept/reject) output.
- **P** - The class of decision problems which are soluble on a TM in time bounded by a polynomial function of the input length.
- **Nondeterministic Turing Machine (NTM)** - A TM which is allowed to make "non-deterministic" choices - the input is accepted if some unspecified set of choices leads to an "accept state". Can be thought of as optionally forking a parallel TM thread at each step; if one of the threads accepts the input then the NTM accepts it.
- **NP** - The class of decision problems which are polynomial-time soluble on an NTM. Contains P. (Can also be thought of as problems whose "solutions" come with proofs that are polynomial-time verifiable on a TM)
- **P≠NP** - The conjecture that NP does not equal P, proposed by Cook 1971[69] and still unsolved.[70]
- **Polynomial-time reducible** - We can "reduce problem A to problem B in polynomial time" iff, given an oracle that solves problem B in constant time, we can solve problem A in polynomial time. (An oracle is like a magic subroutine)

[69] S.A. Cook. "The complexity of theorem-proving procedures". In: *Proceedings of the third annual ACM symposium on Theory of computing*. ACM. 1971, pp. 151–158.

[70] L. Fortnow. "The status of the P versus NP problem". In: *Communications of the ACM* 52.9 (2009), pp. 78–86.

- **NP-complete** - A problem is NP-complete if it is in NP and any other problem in NP can be reduced to it in polynomial time. Conjectured not to overlap with P (which would imply P=NP).
- **NP-hard** - A problem is NP-hard if problems in NP are polynomial time reducible to it (but it need not itself be in NP, or even be a decision problem).

The canonical NP-complete problem is boolean satisfiability or SAT, which can in turn be polynomial-time reduced to its subclasses k-SAT (defined in the previous section) for $k \geq 3$.

The NP-completeness of SAT is called the Cook-Levin theorem, which can be proven by showing that the problem of finding the correct non-deterministic choices to cause a specific NP program to be accepted by an NTM, can be polynomial-time reduced to boolean satisfiability. This is straightforward: introduce a variable corresponding to every aspect of the state of the NTM at every moment in time (only a polynomial-sized amount of memory needs to be modelled), form a boolean expression from these variables which is true iff their values correspond to a legal execution of the NTM, and show that the size of the resulting expression is bounded by a polynomial function of the input.

The problem of statistical inference is not a decision problem so it is not in P or NP. However, it is easy to reduce a k-SAT instance to the problem of computing marginals in a factor graph. Given the k-SAT instance

$$\bigwedge_c \left((\bigvee_{i \in c^+} v_i) \vee (\bigvee_{i \in c^-} \neg v_i) \right) \tag{2.58}$$

we construct a binary factor graph with factors c, and variables i, and potentials

$$\psi_c(v_c) = \begin{cases} 1 & \text{if } (\bigvee_{i \in c^+} v_i) \vee (\bigvee_{i \in c^-} \neg v_i) \\ 0 & \text{otherwise} \end{cases} \tag{2.59}$$

If the formula is satisfiable then the distribution

$$P(v) = \frac{1}{Z} \prod_c \psi_c(v_c) \tag{2.60}$$

is well-defined (has a non-zero normalising constant). If we can compute marginals for the distribution, i.e. if we can perform inference, then we

can find a satisfying assignment by selecting a variable i and value v_i^* with $P(v_i = v_i^*) > 0$, then e.g. updating the model to force $v_i = v_i^*$ (multiplying by a unary factor which is 1 when $v_i = v_i^*$ and 0 otherwise), and recalculating the marginals, and repeating. This procedure performs inference n times, where n is the number of variables, and so is a polynomial-time reduction of k-SAT to inference. This proves that exact inference is NP-hard.[71] A similar argument can be used to show that SAT can be reduced to the problem of finding marginals accurate to some constant ϵ, so even the problem of "do approximate inference to within a specified level of accuracy" shares the property of NP-hardness.[72] It is not clear if any classes of approximate inference tasks can be guaranteed as tractable for general models, although inference is easy in many restricted model classes such as trees (section 2.3.1).

Note that, as is common when applying complexity theory to numerical algorithms, these analyses sweep under the carpet the question of the influence of a need for greater or lesser numerical precision on time and memory usage; but we imagine the results should remain valid even under a more careful formalism.

The unproven but, some would say, intuitively obvious conjecture P≠NP would imply that SAT cannot be solved in polynomial time. This is to say that it is difficult to solve arbitrary SAT instances. It may be that the best possible time complexity is exponential in the number of variables (this is called the "exponential time hypothesis"[73]), which is the cost of a naive solver that examines each set of variable assignments one at a time. This would then be true for the cost of statistical inference as well. Yet the presumably exponential time-complexity of SAT does not deter various computer scientists from working on SAT solvers. Even if the worst-case performance of such solvers is exponential, it may be that there is considerable variation in speed, by constant factors, or by a different base for the exponent, between different solvers. Furthermore, one would like to be able to solve certain more tractable classes of SAT problems as quickly as their structure allows. Also, it is convenient to reduce other NP problems such as automated planning or electronic design automation to SAT, so a good SAT

[71] G.F. Cooper. "The computational complexity of probabilistic inference using Bayesian belief networks". In: *Artificial intelligence* 42.2-3 (1990), pp. 393–405; Barahona, "On the computational complexity of Ising spin glass models", op. cit.

[72] P. Dagum and M. Luby. "Approximate probabilistic reasoning in Bayesian belief networks is NP-hard". In: *Artificial Intelligence* 60 (1993), pp. 141–153; D. Roth. "On the hardness of approximate reasoning". In: *Artificial Intelligence* 82.1-2 (1996), pp. 273–302.

[73] Impagliazzo and Paturi, "Complexity of k-SAT", op. cit.

solver can have general applicability. Examples of popular algorithms for SAT include deterministic algorithms such as DPLL[74] (a variant of Davis Putnam[75]) and stochastic algorithms GSAT[76] and walk-SAT.[77]

The place of inference in machine learning is analogous to that of SAT in computer science. Inference can be done exactly, in time which is exponential in the number of variables in the model, and it is unlikely that we will be able to find inference algorithms which do any better in the worst case. But on the other hand, good statistical inference methods are useful in a variety of settings. There is a demand for exact inference solvers with low constant overhead, as there is for approximate inference solvers with good performance on certain tractable classes of models or for certain lenient types of prediction.

SAT-like problems, which is to say NP-complete ones, often arise in settings where we want computers to solve difficult, adversarial tasks (such as games), or to reason about their own behaviour (some problems in code optimisation and analysis and theorem proving).[78] These are all capabilities

[74]M. Davis, G. Logemann, and D. Loveland. "A machine program for theorem-proving". In: *Communications of the ACM* 5.7 (1962), pp. 394–397.

[75]M. Davis and H. Putnam. "A computing procedure for quantification theory". In: *Journal of the ACM (JACM)* 7.3 (1960), pp. 201–215.

[76]B. Selman, H. Levesque, and D. Mitchell. "A new method for solving hard satisfiability problems". In: *Proceedings of the tenth national conference on artificial intelligence*. 1992, pp. 440–446.

[77]B. Selman, H. Kautz, and B. Cohen. "Local search strategies for satisfiability testing". In: (1993).

[78]We should point out here that what is usually considered "theorem proving", which is theorem proving in first order or higher-order logic, is in fact undecidable, as are standard problems in program analysis such as the "halting problem," i.e. deciding whether a given program will terminate on a given input. "Undecidable" means that there is no algorithm which is guaranteed to find a solution to arbitrary such problems in finite time. This is closely related to Gödel's incompleteness theorem. Theorem proving in very simple logics, e.g. propositional logic, which is the same as SAT, is decidable. Undecidable problems can perhaps be thought of as NP-complete problems to which some form of abstraction has been added, so that there are no longer a finite number of solutions. Although we will try to restrict ourselves to proposing NP-complete problems as targets for approximate inference, we think that the distinction between NP-complete and undecidable is not so important here. Any intelligent system would have to be able to "call it quits" given some inputs, whether the problems it is being asked to solve are decidable or not. In practise, every computer has finite memory and every user has finite time, and it seems like it ought to make little immediate difference whether these resources are exhausted on a problem whose potential complexity is theoretically unbounded or theoretically very large. Additionally, most systems for theorem proving or for program analysis, despite having to tackle undecidable problems in general, make heavy use of subsystems which are solving problems in SAT.

which would be natural to demand of an "autonomous" or "intelligent" system. As we argued in the introduction, it is also natural to associate the NP complexity class with intelligence, because it comprises those problems which may be difficult but whose solutions can be associated with correctness proofs which are easy to verify.

Statistical inference should be seen as a continuous version of SAT, one which is capable of reasoning about uncertainty using the laws of probability. In the introduction, we raised the question: given that everything is ultimately either true or false, is reasoning about uncertainty really necessary? We argued that uncertainty should play a fundamental role in reasoning, even in situations such as SAT where the problem to be solved is well-defined and completely deterministic. We based this argument on analogies to problem-solving in mathematics, as well as recent work on the SAT problem itself which employs methods from statistics, namely *survey propagation*. Survey Propagation works by sending messages, composed of probabilities, between clauses in a SAT instance, and has been successful in solving previously difficult[79] random k-SAT instances generated close to the "hard SAT" region near the SAT-UNSAT phase transition. It is equivalent to running BP on a special graph.[80]

These arguments were intended to point us to the conclusion that probabilistic reasoning frameworks such as statistical inference are a more natural foundation for intelligent systems than outwardly deterministic frameworks such as SAT. In view of this, one might venture to argue that too much work is being devoted to complexity theory and the study of NP-complete problems, and not enough to statistical inference. But of course, it is difficult to know when one has identified the correct framework until one has done something useful with it, and neither framework can claim significant progress towards the goal of autonomy in digital computers. And although we are interested here in approximate inference in discrete factor graphs, some of the same arguments which we have put forward to advocate such a probabilistic framework could also be used to argue for the use of continuous factor graphs, which would be better able to reason about their own (continuous-valued) beliefs, or non-parametric models which could potentially represent uncertainty in meta-beliefs. It is difficult to envision a

[79] M. Mézard and R. Zecchina. "Random K-satisfiability problem: From an analytic solution to an efficient algorithm". In: *Physical Review E* 66.5 (2002), p. 56126; A. Braunstein, M. Mezard, and R. Zecchina. "Survey propagation: an algorithm for satisfiability". In: *Random Structures and Algorithms* 27.2 (2005), pp. 201–226.

[80] A. Braunstein and R. Zecchina. "Survey propagation as local equilibrium equations". In: *Journal of Statistical Mechanics: Theory and Experiment* (2004).

framework which is "complete" in this sense, so we have stopped at one which seems to make a reasonable compromise between simplicity and expressivity, *viz.*, discrete factor graphs. In any case, none of the techniques presented in this dissertation seem to have very natural analogs in the simpler SAT framework.

Chapter 3

Conditioned Belief Propagation

Abstract

Conditioning is a simple technique for partitioning the task of inference in a statistical model between two or more sub-models. We describe an approximate, divide-and-conquer application of variable conditioning to statistical inference. Our algorithm is based on recursive conditioning and Belief Propagation. We consider the problem of choosing which variable to condition at each level of recursion, and propose a fast heuristic using reverse-mode automatic differentiation (i.e. back-propagation) to obtain potential gradients at BP fixpoints.

3.1 Introduction

Cutset conditioning (CC) is an exact inference algorithm described by Pearl,[1] which functions by creating sub-models in which a model is conditioned on each possible value of some fixed subset of its variables. The results of running BP in each sub-model are combined to obtain marginals for the original model. When the condition variables form a "cutset", meaning that they intersect every loop in the model, then the conditioned sub-models are singly-connected and inference is exact; this is the CC algorithm proper. We explore an approximate, recursive variant of the same idea. In our algorithm, condition variables are selected one at a time and need not form a cutset.

[1] Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, op. cit.

We explore ways of choosing condition variables at each level of recursion. We develop a heuristic based on an application of reverse-mode automatic differentiation (RAD) (known as "back-propagation" in the neural network field) to BP, and present experiments characterising its effectiveness. We show that our algorithm is competitive with published inference algorithms on some standard models. We also identify some basic shortcomings and discuss directions for improvement.

The organisation of this chapter is as follows. First we describe existing work on cutset conditioning and on automatic differentiation of BP. Then we review our definitions, with brief descriptions of BP, conditioning, and back-propagation. We then present a description of the algorithm itself, followed by the results of experiments which compare the performance of our algorithm with other popular algorithms.

3.2 Background

The CC algorithm, mentioned in the introduction, is described in more detail in section 3.3.3.

An approximate application of CC in causal networks was described by Horvitz et al,[2] but their inference setting is slightly different from ours. As with CC, the time complexity of their algorithm is exponential in the cutset size. They only consider trading time for accuracy in updating the beliefs after modifying the model through the incorporation of "evidence" from observations. Their algorithm is only useful, in other words, when alternating inference with learning. But we are only concerned with the "pure" setting of approximate inference, in which a model has already been learned.

Darwiche also describes an approximate version of CC, which again applies only to causal networks.[3] It creates a Boolean formula describing statements which are approximately true of the model. Each entry in each of in the network's conditional probability tables which is smaller than some threshold value is used to generate an implication which is added to this formula. Darwiche's algorithm seems to be of limited usefulness, not only

[2] E.J. Horvitz, H.J. Suermondt, and G.F. Cooper. *Bounded conditioning: Flexible inference for decisions under scarce resources.* Tech. rep. Stanford University. Medical Computer Science. Knowledge Systems Laboratory, 1990.

[3] A. Darwiche. "Conditioning methods for exact and approximate inference in causal networks". In: *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence.* 1995.

because it doesn't apply to general factor graphs, but also because it is possible to obtain arbitrarily unlikely states in models whose probability tables are arbitrarily close to uniform, even though such models would not benefit from the algorithm.

Our description of a "recursive" form of cutset conditioning may risk conflation with another algorithm of Darwiche called "recursive conditioning"[4] (RC). RC is an exact algorithm which has the property of being able to smoothly trade-off consumption of time and space by caching intermediate results. His use of "recursion" is in a sense which is nearly orthogonal to our own: making use of topological information in the graphical structure, it recursively decomposes a model into disconnected submodels through the instantiation of each combination of variables in a set. We are not interested in exact inference; and we ignore graph topology, since we assume that the presence of dense but weak factors may make topology uninformative in most models.

Our application of RAD to BP, which we call BBP, is related to Welling and Teh's work on "linear response" (LR).[5] Their LR algorithm applies forward-mode automatic differentiation (FAD) techniques to BP to calculate a full set of partial derivatives $\frac{\partial b_i(x_i)}{\partial \psi_j(x_j)}$. Computing such a matrix requires time quadratic in the number of variables and states (proportional to n runs of BP), but their algorithm can be straightforwardly modified to output a vector of first derivatives of the beliefs b corresponding to an input vector of perturbations of ψ (which is the usual task of FAD). Our reverse-mode algorithm calculates the gradient of some objective function with respect to the factor potentials, given the gradient with respect to the beliefs. It has the same time complexity as this modified LR. After publication of the paper on which this chapter is based, we discovered that the two propagation algorithms can be used to calculate the same quantities and are in fact related by a somewhat involved transformation. Our formulation of BBP is more general because it avoids the use of division and so can be applied to models with zeroes, or very small values, but LR can easily be adapted to have the same property. It is interesting to consider why the two methods for propagating derivatives, one which follows messages forward in time and the other which follows them backwards, should be equivalent. This may be a fruitful topic for future research.

[4] A. Darwiche. "Recursive conditioning". In: *Artificial Intelligence* 126.1-2 (2001), pp. 5–41.

[5] M. Welling and Y.W. Teh. "Linear response algorithms for approximate inference in graphical models". In: *Neural computation* 16.1 (2004), pp. 197–221.

3.3 Prologue

3.3.1 Factor graphs

In this chapter we define graphical models using a variation on the conventions of section 2.1, explicitly including univariate factors ψ_i to emphasise a kind of variable-factor duality. We write the normalisation constant (partition function) using a subscript (Z_P) to distinguish it from others introduced later. Our model is then defined as

$$P(x_1, \ldots, x_N) = \frac{1}{Z_P} \prod_i \psi_i(x_i) \prod_\alpha \psi_\alpha(x_\alpha) \tag{3.1}$$

where

$$Z_P = \sum_x \prod_i \psi_i(x_i) \prod_\alpha \psi_\alpha(x_\alpha) \tag{3.2}$$

A probability distribution q will often be defined by normalising a given measure m:

$$q(x) = \frac{m(x)}{\sum_{x'} m(x')} \tag{3.3}$$

When there is no room for confusion, for each such q we will define the *normalisation constant* $Z_q \equiv \sum_x m(x)$ and *unnormalised measure* $\bar{q}(x) \equiv m(x) = Z_q q(x)$.

3.3.2 Belief Propagation

BP was defined in 2.3.1. We present a dual form for the BP message updates, which are passed between every factor α and each of its neighbouring variables $i \in \alpha$.

$$n_{i\alpha}(x_i) \leftarrow \frac{1}{Z_{n_{i\alpha}}} \psi_i(x_i) \prod_{\beta \sim i \backslash \alpha} m_{\beta i}(x_i) \tag{3.4}$$

$$m_{\alpha i}(x_i) \leftarrow \frac{1}{Z_{m_{\alpha i}}} \sum_{x_{\alpha \backslash i}} \psi_\alpha(x_\alpha) \prod_{j \sim \alpha \backslash i} n_{j\alpha}(x_j) \tag{3.5}$$

These messages should be propagated until convergence. The following equations yield estimates for variable and factor marginals:

$$P_i(x_i) \approx b_i(x_i) = \frac{1}{Z_{b_i}} \psi_i(x_i) \prod_{\alpha \sim i} m_{\alpha i}(x_i) \tag{3.6}$$

$$P_\alpha(x_\alpha) \approx b_\alpha(x_\alpha) = \frac{1}{Z_{b_\alpha}} \psi_\alpha(x_\alpha) \prod_{i \sim \alpha} n_{i\alpha}(x_i) \tag{3.7}$$

BP also provides an estimate of Z_P using the *Bethe free energy*:

$$\begin{aligned} -\log Z_P \approx & F_{\text{Bethe}} \equiv \sum_{\alpha} \sum_{x_\alpha} b_\alpha(x_\alpha) \log \left(\frac{b_\alpha(x_\alpha)}{\psi_\alpha(x_\alpha)} \right) \\ & + \sum_{i} \sum_{x_i} b_i(x_i) \log \left(\frac{b_i(x_i)}{\psi_i(x_i)} \right) \\ & - \sum_{i} \sum_{x_i} |i|\, b_i(x_i) \log \left(b_i(x_i) \right) \end{aligned} \tag{3.8}$$

3.3.3 Conditioning

One technique for improving the performance of BP, and indeed any inference algorithm providing an estimate of the partition function Z_P, is called "conditioning". The idea of conditioning is to write a model as a sum of simpler models, apply inference to each sub-model, and combine the resulting approximate marginals using the Z_P estimates from the sub-models. The decomposition into sub-models is expressed using a "condition" variable c:

$$\bar{P}(x) = \bar{P}(x)(I_c + I_{\neg c}) \equiv \bar{P}(x|c) + \bar{P}(x|\neg c) \tag{3.9}$$

$$Z_P = Z_{P(|c)} + Z_{P(|\neg c)} \tag{3.10}$$

(Recall that $\bar{P} \equiv Z_P P$ is the unnormalised distribution. I_c is an "indicator variable" for c; in other words it takes the value 1 when c is true and 0 otherwise.) Dividing equation 3.9 by Z_P yields the more familiar

$$P(x) = P(x|c)P(c) + P(x|\neg c)P(\neg c) \tag{3.11}$$

where $P(c) \equiv \frac{Z_{P(|c)}}{Z_P}$ and $P(\neg c) \equiv \frac{Z_{P(|\neg c)}}{Z_P}$.

Since equation 3.10 gives a rule for estimating the partition function of the original model, the conditioning idea can be applied recursively in a divide-and-conquer fashion. We will refer to any combination of BP and conditioning using divide-and-conquer as "conditioned BP" or CBP.

The CC algorithm can be seen as an instance of CBP, conditioning on all possible values of a set of variables (the *cutset*) whose removal makes G singly connected (tree-like). Since BP is exact on trees, the CC algorithm is also exact. The drawback of CC is that its run-time is exponential in the cutset size, so it is only applicable to small or tree-like graphs.

Here we will only consider conditions c of the form $\{x_i = \hat{x}_i\}$ for some variable i and state $\hat{x}_i$. Then $I_c(x) = \delta_{\hat{x}_i}(x_i)$ and $I_{\neg c}(x) = 1 - \delta_{\hat{x}_i}(x_i)$, so the two conditioned submodels can be expressed by replacing the original factor

$\psi_i(x_i)$ with $\delta_{\hat{x}_i}\psi_i(x_i)$ and $(1-\delta_{\hat{x}_i})\psi_i(x_i)$, respectively. In the first submodel, the variable x_i is "clamped"[6] to the state $\hat{x}_i$, and in the second it is required to take any state *but* $\hat{x}_i$ (there may be more than one). Each sub-model has fewer states than the original, without extending the original factor graph; so we might hope that combining the submodels would yield more accurate approximate marginals. The analogous technique of Rao-Blackwellization, which applies to sampled estimates, is guaranteed to produce estimates with lower variance. Empirically, the combined BP estimates are usually but not always more accurate (section 3.5).

After obtaining a pair of sub-models, CBP applies conditioning recursively to each one, so that eventually a binary tree of conditions is explored - we will call this the "condition tree" - with BP being run only at the leaves.

An implementation of CBP is defined by how it decides which variable to clamp to which value at each level of recursion. One simple method is to choose a uniformly random variable and value - we call this "CBP-rand". In the next section, we explore a more advanced heuristic for choosing condition variables.

3.4 Algorithm

Our proposal is based on the following idea: because BP uses local messages, it poorly represents correlations between distant variables. Since CBP must rely on conditioning to capture distant correlations, it should condition on variables whose potentials have the greatest "influence" on the rest of the graph. Conditioning on a set $\mathcal{X}_i' \subseteq \mathcal{X}_i$ has the same effect as setting to zero any value of $\psi_i(x_i)$ with $x_i \notin \mathcal{X}_i'$. We propose that the notion of the "influence" of a variable i and value x_i can be usefully approximated as the effect of *infinitesimal* variation of $\psi_i(x_i)$. The effect should be measured with respect to some function of the BP beliefs, call it $V(b)$. Although we can compute the change in V directly by clamping each variable i in the graph to each possible value x_i and running BP, such a procedure would have time complexity quadratic in the number of graph variables. If we can make do with querying infinitesimal changes in V, then we only need the derivatives $\frac{\mathrm{d}V}{\mathrm{d}\psi_i(x_i)}$. We can compute a full set of such derivatives in linear time complexity using a standard technique called *back-propagation*.

[6] In our terminology, to "clamp" a variable to a value means to condition the model so that the variable takes that value. Thus clamping is a special case of conditioning.

3.4.1 Back-Propagation and BP

In this section, we will apply RAD (back-propagation) to Belief Propagation, deriving an iterative algorithm for estimating the gradient of any differentiable objective function of the BP beliefs with respect to the factors of the model. We will refer to this algorithm as *back-belief-propagation* or BBP.[7]

The BP updates may not converge when initialised at a given point in the configuration space of factors and initial messages, but if they do, then typically there will be a smooth function between some open ball in that space, and the BP approximate marginals. The derivatives of this function are well-defined, and are what we seek to calculate.

RAD computes the derivatives of a common objective function V with respect to various quantities, call them y. We will abbreviate these derivatives (not following any existing convention) as

$$\not{d}y \equiv \frac{\mathrm{d}V}{\mathrm{d}y} \tag{3.12}$$

This is called the *adjoint* of y (with respect to V). Given $V(f(y), g(y))$, we can apply the chain rule to compute $\not{d}y = \frac{\partial f}{\partial y}\not{d}f + \frac{\partial g}{\partial y}\not{d}g$. This process can be extended to general function compositions and is called *back-propagation* or *reverse-mode automatic differentiation*.[8]

We can assume that the BP messages are updated in parallel, and index them with a variable $t \in [0, T]$. Then, given an objective function specified in terms of BP beliefs $V(b)$, we can compute the adjoints of the factors ψ_i and ψ_α by following the BP messages backwards through time. The functional dependencies are depicted in figure 3.1. Using the chain rule, we can derive equations for the back-propagation of BP message and factor adjoints.

Note that to use these equations, we must save the message normalisers from the BP run, since we need to calculate unnormalised adjoints from the normalised adjoints (e.g. to calculate $\not{d}\bar{n}_{i\alpha}$ from $\not{d}n_{i\alpha}$ requires $Z_{n_{i\alpha}}$). This is done using the following general formula. Given a normalised distribution $q(x) = \frac{1}{Z_q}\bar{q}(x)$ as in equation 3.3, and assuming that V doesn't depend

[7] There is a more recent method for computing the same quantities, which approximates derivatives with difference quotients. It makes use of the observation that the BP Jacobian is symmetric. It has the same time complexity and smaller constant overhead, but is less numerically stable than BBP (Justin Domke. "Implicit Differentiation by Perturbation". In: *Advances in Neural Information Processing Systems 23*. 2010, pp. 523–531).

[8] A. Griewank. "On automatic differentiation". In: *Mathematical Programming: Recent Developments and Applications* (1989), pp. 83–108.

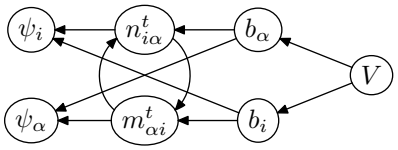


Figure 3.1: Functional dependencies of BP

explicitly on Z_q, we can write $\not{d}\bar{q}$ in terms of $\not{d}q$ and Z_q:

$$\not{d}\bar{q}(x) = \frac{1}{Z_q}\left(\not{d}q(x) - \sum_{x'} q(x')\not{d}q(x')\right) \tag{3.13}$$

We should be able to take $T \to \infty$ and get sensible answers. But the factor adjoints involve a sum of the message adjoints over time, which would diverge if the message adjoints did not converge to zero. In fact, for non-degenerate problems BP converges to attractive fix-points, which means that to a certain extent it is insensitive to the initial values of messages. This means that if we go far enough back in time, the messages have diminishing contribution to the final beliefs, and their adjoints should converge to zero as required. In this way, the BBP algorithm is both sensitive to initial conditions (which are used to specify the objective function), and convergent to a stable fixed point.

The equations can be simplified. Since BP will have converged, we can assume the messages are constant with respect to time, and drop the t superscripts from the BP messages (and their normalisers).

Furthermore, the factor adjoints $\not{d}\psi_i(x_i)$ and $\not{d}\psi_\alpha(x_\alpha)$ are expressed as initial values plus a sum involving message adjoints over time. We can compute such quantities incrementally, by making sure that each time we update a message adjoint, we also update the appropriate factor adjoint.

This yields the following algorithm:

Algorithm (BBP)

Input: The beliefs and messages (and normalisers thereof) of a BP run, also an objective function $V(b)$ defining initial adjoints $\not{d}b_i$ and $\not{d}b_\alpha$

Output: $\not{d}\psi_i(x_i)$, $\not{d}\psi_\alpha(x_\alpha)$

Precompute the following quantities:

$$T_{i\alpha}(x_i) = \prod_{\beta \sim i \setminus \alpha} m_{\beta i}(x_i) \tag{3.14}$$

$$U_{\alpha i}(x_\alpha) = \prod_{j \sim \alpha \setminus i} n_{j\alpha}(x_j) \tag{3.15}$$

$$S_{i\alpha j}(x_i, x_j) = \sum_{x_{\alpha \setminus i \setminus j}} \psi_\alpha(x_\alpha) \prod_{k \sim \alpha \setminus i \setminus j} n_{k\alpha}(x_k) \tag{3.16}$$

$$R_{\alpha i \beta}(x_i) = \psi_i(x_i) \prod_{\gamma \sim i \setminus \alpha \setminus \beta} m_{\gamma i}(x_i) \tag{3.17}$$

Initialise:

$$\not{d}\psi_i(x_i) \leftarrow \left(\prod_{\alpha \sim i} m_{\alpha i}(x_i) \right) \not{d}\bar{b}_i(x_i) \tag{3.18}$$

$$\not{d}\psi_\alpha(x_\alpha) \leftarrow \left(\prod_{i \sim \alpha} n_{i\alpha}(x_i) \right) \not{d}\bar{b}_\alpha(x_\alpha) \tag{3.19}$$

$$\not{d}n_{i\alpha}(x_i) \leftarrow \sum_{x_{\alpha \setminus i}} \psi_\alpha(x_\alpha) \prod_{j \sim \alpha \setminus i} n_{j\alpha}(x_j) \not{d}\bar{b}_\alpha(x_\alpha) \tag{3.20}$$

$$\not{d}m_{\alpha i}(x_i) \leftarrow \psi_i(x_i) \prod_{\beta \sim i \setminus \alpha} m_{\beta i}(x_i) \not{d}\bar{b}_i(x_i) \tag{3.21}$$

Then, apply the following updates in parallel (for every factor α and variable i) until the message adjoints converge to zero:

$$\not{d}\psi_i(x_i) \leftarrow \not{d}\psi_i(x_i) + T_{i\alpha}(x_i) \not{d}\bar{n}_{i\alpha}(x_i) \tag{3.22}$$

$$\not{d}n_{i\alpha}(x_i) \leftarrow \sum_{j \sim \alpha \setminus i} \sum_{x_j} S_{i\alpha j}(x_i, x_j) \not{d}\bar{m}_{\alpha j}(x_j) \tag{3.23}$$

$$\not{d}\psi_\alpha(x_\alpha) \leftarrow \not{d}\psi_\alpha(x_\alpha) + U_{\alpha i}(x_\alpha) \not{d}\bar{m}_{\alpha i}(x_i) \tag{3.24}$$

$$\not{d}m_{\alpha i}(x_i) \leftarrow \sum_{\beta \sim i \setminus \alpha} R_{\alpha i \beta}(x_i) \not{d}\bar{n}_{i\beta}(x_i) \tag{3.25}$$

The individual updates, which must be performed for each edge in the factor graph until convergence, are (assuming a bounded state-space for each variable) of complexity quadratic in the largest number of variables in a given factor, and in the largest number of factors containing a given variable.

3.4.1.1 Sequential updates

The above parallel algorithm occasionally suffers from numerical stability problems. It is straightforward (but slightly more involved) to derive a sequential algorithm which computes the same quantities, one message at a time, by ensuring that $m_{\alpha i}^{t+1} = m_{\alpha i}^{t}$ for all but one (α, i) (see section 3.8, figure 3.5). The sequential algorithm has the same time complexity as the above, but it allows us to fine-tune the order in which updates are performed. In particular, we can record the order in which BP messages are sent (which may be according to a dynamic schedule as in RBP[9] or its refinements[10]), and send BBP messages in the reverse of this order. Empirically, this method yields slightly better convergence than the parallel algorithm, and is therefore used throughout the experiments (section 3.5), with BP messages scheduled as in RBP (this means that message updates are sorted so that those which cause the greatest change in their messages are processed first). However, only the results for the "alarm" graph (figure 3.4) were noticeably improved by the use of sequential BBP updates.

3.4.2 CBP-BBP

We have not yet addressed the question of which objective function $V(b)$ to use with BBP. There are several possibilities, and the following proposal seems to perform well. The performance of some other objective functions is shown in the appendix (figure 3.6).

We use a brief run of Gibbs sampling to select a random state $x^\star$ of the model. Then we define

$$V^{\mathrm{G},x^\star}(\{b_\alpha\}_\alpha) = \sum_\alpha b_\alpha(x_\alpha^\star) \tag{3.26}$$

The intuition behind this choice is that we want to find a condition which "pushes" the model's beliefs in a certain direction. If the model's probability mass is concentrated in several modes, then we expect Gibbs sampling to produce a sample $x^\star$ from one of them; the objective function $V^{G,x^\star}$ then helps us identify a variable that pushes the beliefs in the direction of that mode.

The complete CBP-BBP algorithm becomes

[9] G. Elidan, I. McGraw, and D. Koller. "Residual belief propagation: Informed scheduling for asynchronous message passing". In: *Proceedings of the Twenty-second Conference on Uncertainty in AI (UAI), Boston, Massachussetts*. Vol. 6. 6.4. 2006, pp. 6–4.

[10] C. Sutton and A. McCallum. "Improved dynamic schedules for belief propagation". In: *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*. 2007.

Algorithm (CBP-BBP)

Input: A graphical model, and a maximum number of variables to clamp

Output: A set of approximate beliefs, and an approximate Z

1. Run BP. If the maximum number of variables have been clamped, return the BP beliefs and estimated Z. Otherwise,

2. Run Gibbs sampling to get a state $x^\star$

3. Run BBP with $V = V^{\mathrm{G},x^\star}$

4. Find the pair (i, x_i) with largest $\not{d}\psi_i(x_i)$

5. Clamp variable i to x_i and recurse

6. Clamp variable i to $\mathcal{X}_i \setminus x_i$ and recurse (i.e., condition on membership in $\mathcal{X}_i \setminus x_i$)

7. Combine the results from steps 5 and 6 as described in section 3.3.3

An implementation of this algorithm based on libDAI[11] can be downloaded from `http://mlg.eng.cam.ac.uk/frederik/aistats2009_choosing.php`.

3.5 Experiments

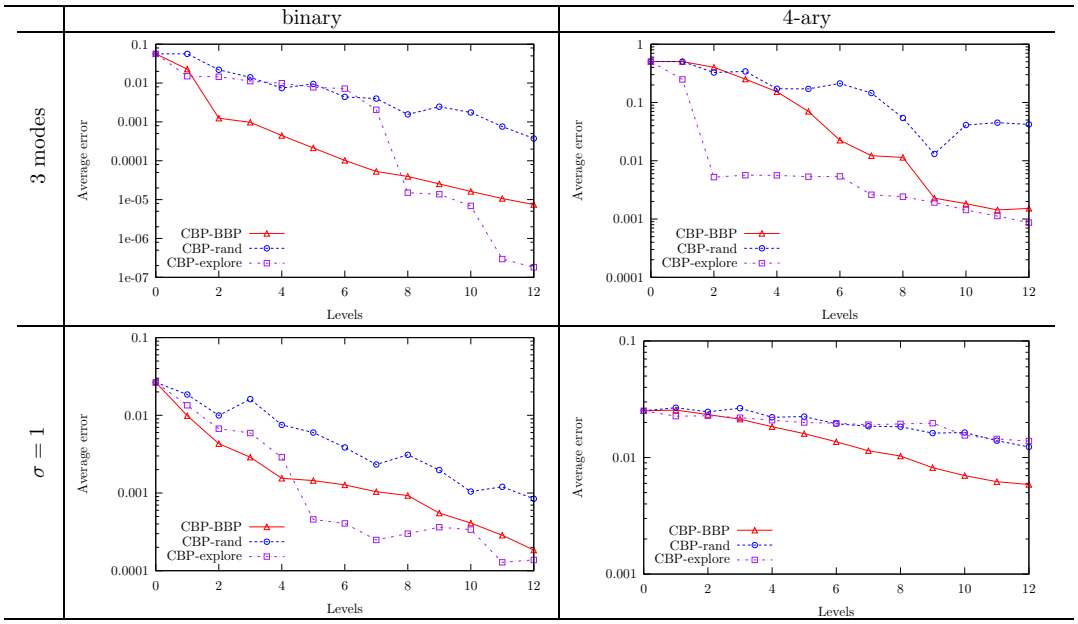
Our experiments use 8 graphical models, representing 2^3 combinations of topology (square grid, random regular), variable arity (2, 4), and potential initialisation (modes, random). The "random" potentials are created by setting each factor entry to $\exp(\sigma W)$ where W is a standard normal deviate and $\sigma = 1$. The "modes" potentials are created by choosing 3 random configurations $x^{\star,1\ldots3}$ for the graph variables, and setting each factor entry $\psi_\alpha(x_\alpha)$ to some constant c if it is consistent with one or more of them (i.e. $x_\alpha = x_\alpha^{\star,k}$ for some k) and to 1 otherwise. We arbitrarily choose c to be 4. This is a way of creating graphs with long-distance correlations. Most of the model's probability mass will be concentrated in the 3 selected "modes".

Our first task is to establish that the CBP-BBP algorithm chooses better conditioning variables than CBP-rand. We calculate the accuracy of the approximation generated from a condition tree of fixed uniform depth; this depth is called the "clamping level". Plots of accuracy vs. clamping level are shown in figure 3.2. In all the plots in this chapter, each CBP-BBP and CBP-rand data point shows the result of averaging error and timing data for 5 runs of the algorithm. The errors are computed as total variation distance[12]

[11] J.M. Mooij. *libDAI 0.2.2: A free/open source C++ library for Discrete Approximate Inference methods*. `http://mloss.org/software/view/77/`. 2008.

[12] The total variation distance is half of the L_1 distance: $\frac{1}{2}\sum_x |P(x) - Q(x)|$

Random regular graph, 25 variables and 30 factors size 3



8 by 8 square grid

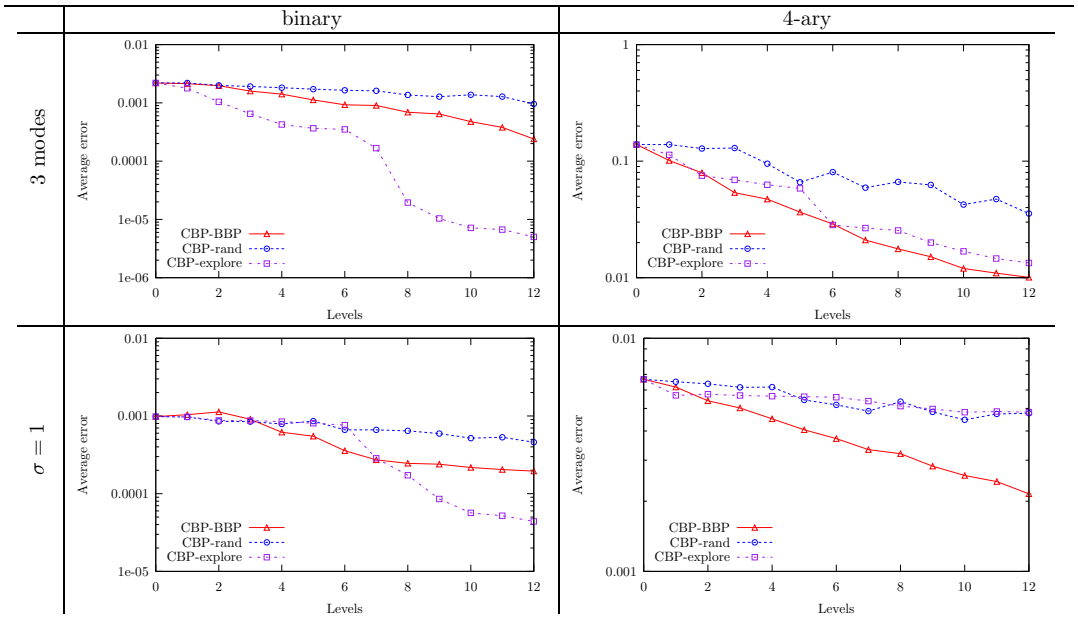
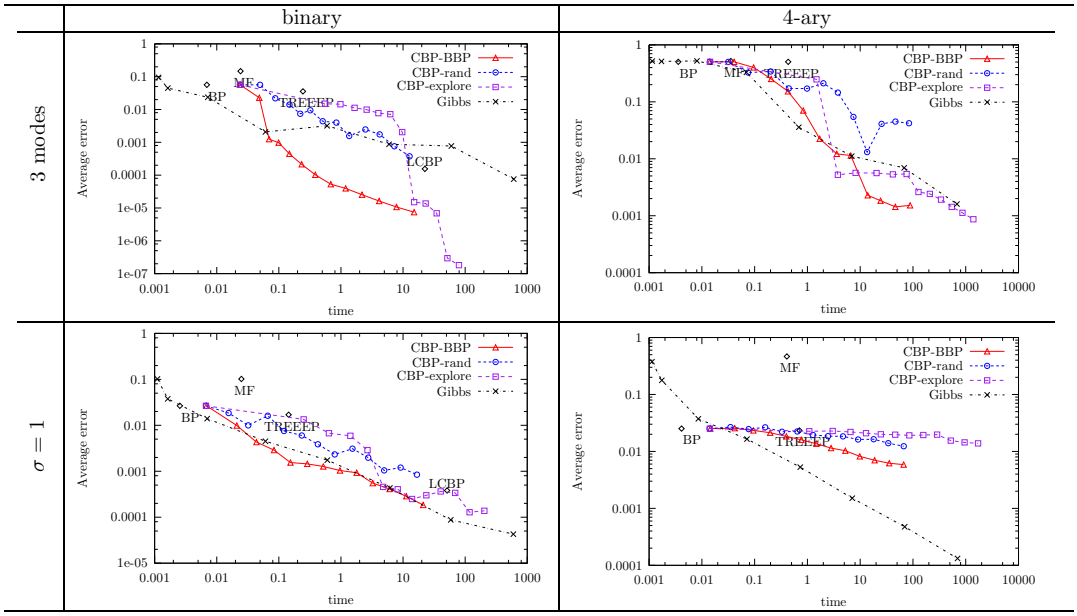


Figure 3.2: Comparisons of BBP clamping (CBP-BBP) to random clamping (CBP-rand) and "exploratory clamping" (CBP-explore), on eight example graphs.

Random regular graph, 25 variables and 30 factors size 3



8 by 8 square grid

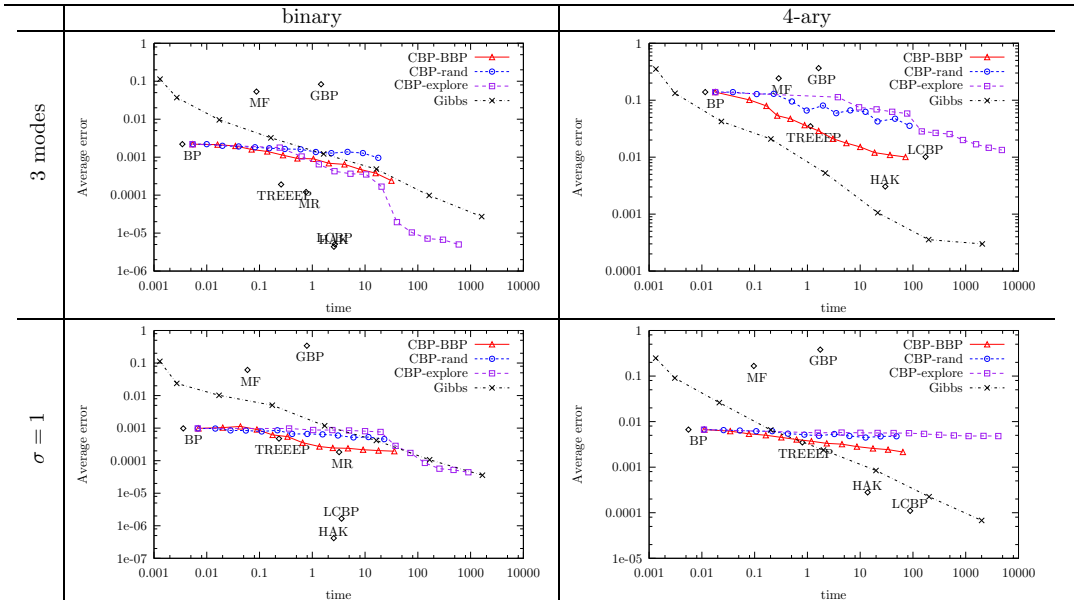


Figure 3.3: Performance for different versions of CBP and other standard approximate inference algorithms. Some algorithms, such as LCBP and MR, required too much time or memory to run on some graphs (such as those with 4-ary variables and random regular topology).

between our single-node marginals, and exact marginals (calculated by the junction tree algorithm with HUGIN updates[13]).

In each model, CBP-BBP is usually better than CBP-rand at a given level, but the difference is sometimes small. Also included for comparison is an algorithm "CBP-explore" which (as suggested at the beginning of section 3.4) prospectively clamps each variable to each value and runs BP, choosing the (variable, value) pair which produces marginals that are maximally different (L^1 distance) from the current marginals. This is slower than CBP-rand or CBP-BBP, being quadratic in the size of the graph, but gives an interesting comparison and often produces more accurate results. We are not sure why CBP-BBP is sometimes more accurate than CBP-explore.

Next, we compare the performance of CBP-BBP to other algorithms. We use the implementations found in libDAI[14] for these experiments.

Gibbs - Gibbs sampling, using runs with from 1000 to 10^7 samples

BP - Belief Propagation

MF - Mean Field

TreeEP - algorithm of Minka and Qi[15]

GBP - Generalised Belief Propagation,[16] using loops of size 4

HAK - algorithm of Heskes, Albers, and Kappen,[17] using loops of size 4

LCBP - Loop Corrected Belief Propagation (full cavities)[18]

MR - algorithm of Montanari and Rizzo[19]

The last two algorithms are based on propagating *cavity distributions* and have complexity exponential in cavity size. The random regular graphs of arity > 2 have cavities which are too large, so these algorithms can only be tested on the other graphs. Also, MR requires binary variables and could not be run on the 4-ary graphs.

Figure 3.3 shows the results of these experiments.[20] Notice that CBP-

[13] Jensen, Olesen, and Andersen, "An algebra of Bayesian belief universes for knowledge-based systems", op. cit.

[14] Mooij, *libDAI 0.2.2: A free/open source C++ library for Discrete Approximate Inference methods*, op. cit.

[15] Minka and Qi, "Tree-structured approximations by expectation propagation", op. cit.

[16] Yedidia, Freeman, and Weiss, "Generalized belief propagation", op. cit.

[17] Heskes, Albers, and Kappen, "Approximate inference and constrained optimization", op. cit.

[18] Mooij et al., "Loop corrected belief propagation", op. cit.

[19] A. Montanari and T. Rizzo. "How to compute loop corrections to the Bethe approximation". In: *Journal of Statistical Mechanics: Theory and Experiment* 10 (2005), P10011.

[20] The poor accuracy of GBP in these comparisons may be surprising. Note that GBP has the same fixpoints as HAK, a slower, double-loop alternative, with better convergence guarantees. Under ideal circumstances, the accuracy of the two algorithms should be the

BBP still typically dominates CBP-rand, not just when comparisons are binned by clamping level but also runtime, even though it is usually somewhat slower due to the overhead of BBP[21]. Gibbs sampling eventually performs better than our algorithm, for long runs. Figure 3.4 shows per-level comparisons and performance plots for the "alarm" graph, which is part of libDAI. There were convergence problems for BP and BBP on this graph, which may explain the poor results.

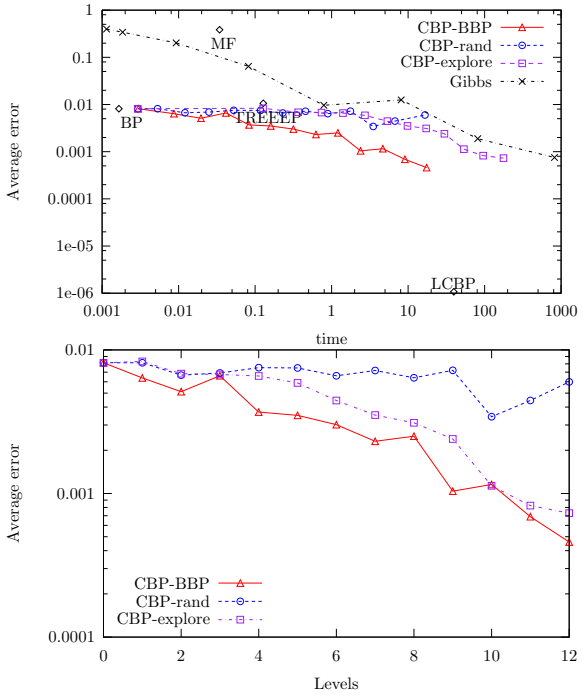


Figure 3.4: Performance on "alarm" graph; level comparison to CBP-rand on same graph

Our implementation of both CBP-BBP and CBP-rand include the op-

same. The poor accuracy of GBP may be due to its implementation in libDAI which uses message updates defined in the inner loop of HAK, rather than the parent-to-child updates endorsed by Yedidia, Freeman and Weiss (JS Yedidia, WT Freeman, and Y. Weiss. "Constructing free-energy approximations and generalized belief propagation algorithms". In: *IEEE Transactions on Information Theory* 51.7 (2005), pp. 2282–2312; Tom Minka. Personal communication. Feb. 2010).

[21] In our experiments, CBP-rand is up to 2.1 times faster than CBP-BBP for a fixed level of clamping. Sometimes, however, due to faster convergence of the BP runs in CBP-BBP, CBP-rand is slower (by up to about 1.5 times).

timisation of not clamping a variable whose BP marginal is already close to 0 or 1. The number of levels of recursion is otherwise fixed. We have experimented with heuristics for controlling recursion depth automatically, for instance recursing until the current Z estimate is smaller than a certain fraction of the top-level Z, but these did not have notably different performance than fixed-level recursion.

3.6 Discussion and future work

We have presented an approximate inference algorithm, based on a divide-and-conquer approach to inference, which combines BP and variable conditioning. The time complexity of the algorithm is proportional to the cost of a BP run, the square of the maximum variable or factor degree, and is exponential in the number of clamped variables. This "clamping level" can be specified by the user to achieve a desired speed/accuracy trade-off. One advantage of our algorithm is that it can be applied to models with large or densely connected underlying graphs. It seems particularly promising on models with long-range correlations between variables. It performs well on models where probability mass is divided between a few isolated "modes" which can be separated by appropriately chosen conditions.

The rest of this section describes some directions in which the CBP-BBP algorithm might be improved, and also poses some difficult problems which provide additional motivation for the research in later chapters. The descriptions are very rough and are aimed at readers interested in extending our research; the general reader may skip this section without detracting from the rest of the thesis.

In our opinion, the main drawback of CBP is its poor performance on models with weak coupling, or with multiple groups of variables that are independent or almost independent. For instance, consider how CBP behaves on a model G^n, consisting of n disconnected copies of a model G. Clamping a variable in the first copy will not change the marginals in any of the other copies; it is necessary to repeat this clamping n times, once for each copy of G, to have the same effect which clamping that variable had in the original G. But due to the hierarchical nature of CBP, these n variables cannot be explored separately, i.e. in parallel, taking advantage of the n-way factorisation of G^n, but rather each of the 2^n combinations of their values must be enumerated and associated to a run of BP. In general, to get the same accuracy in G^n which we had been able to obtain by exploring a condition tree with k levels in G, we need to explore a condition tree with nk levels,

which will be a power of n more expensive ($2^{nk} = (2^k)^n$). For example, for a fixed recursion depth, simply duplicating a model makes CBP quadratically slower. By contrast, in BP or Gibbs sampling or almost any other inference method, inference in a duplicated model is only about twice as hard, and G^n is n times as hard as G. The comparative difficulties that CBP experiences on G^n would presumably carry over to a perturbed version of G^n as well, in which factors are introduced to create a weak coupling between the various copies of G, so that they are no longer strictly independent but only approximately so.

Poor performance on models with weak coupling is not specific to CBP but characterises any naive application of divide-and-conquer which is based on conditioning. Understanding how to remedy this shortcoming, perhaps with some "parallel" version of CBP, should be considered a prerequisite to making the CBP concept useful for more general applications. We are not sure how to do this, but can say a few words about the ideas that we have considered. They are all based on the observation that, once one has selected a tree of successively more specific conditions, it is straightforward to run BP on all the conditioned models simultaneously. This is done by extending the definition of messages so that each message has one vector of probabilities for each leaf in the condition tree; the message updates must also be modified to take the conditions into account, by modifying the appropriate factors for each message component. But once the algorithm has been transformed in this way, it is straightforward to locally prune the tree of conditions, so that certain conditions are omitted in parts of the graph where they have negligible effect. (See also Tom Minka's "gates" approach to message passing in mixture models.[22]) When messages pass into a region with fewer conditions, the message components corresponding to any missing condition must be merged using a weighted average. The weights can be derived from a local form of the Bethe free energy, analogously to the way in which submodel weights were derived from the standard Bethe free energy in CBP. One can easily extend the same idea to the case where completely different condition trees are used in different overlapping regions, as long as the Cartesian product of the sets of conditions is tracked in the region of overlap. When some members of this Cartesian product are absent, so that the condition tree looks more like a forest, then it is not clear what one should do to recombine the messages. For example, such a situation would correspond to running BP with x_1 clamped to 0 and then 1; and then with x_2 clamped to 0 and then 1; but not exploring combinations such

[22] T. Minka and J. Winn. "Gates: A graphical notation for mixture models". In: (2008).

as $(x_1 = 0, x_2 = 1)$. One can think of creative ways to combine the output marginals resulting from such runs, perhaps by using a weighted combination of the x_1-clamped marginals to approximate marginals of variables which seem to be more strongly influenced by x_1 than x_2 (using some heuristic to assess "influence"), and correspondingly for those variables which are more strongly influenced by x_2.

This line of reasoning soon confronts us with a more serious problem, however, which seems difficult to overcome with heuristics. This is the problem of deciding which sets of conditions to explore, and at which locations in the graph to do so. The BBP heuristic presented in this chapter, whose performance was moderately encouraging, only had to choose a single variable and value at each step. Taking advantage of the small number of (variable, value) pairs in a graphical model we were able to compare the performance of our heuristics with that of other more expensive heuristics that explored all such choices exhaustively. But the space of choices to be made is much larger in the case of an algorithm which associates a different set of condition trees locally to each variable. Furthermore, in such cases we must also decide how messages between such variables could be combined, and so forth. We have investigated the implementation of a simple form of parallel CBP based on some naive heuristics, but we were not impressed with the results.

We note that the problem we are considering at this point is close to the problem of how to do a "sparse Generalised Belief Propagation". The main drawback of GBP is that its cost is exponential in the size of the regions, which limits its potential for simplifying inference in graphs with long-range correlations. Yet one can imagine that in many instances, only a few assignments of the variables in a region have significant probability mass. If we could modify GBP to track only these values, then we could use it with much larger regions. There are heuristics for choosing GBP regions,[23] and it is possible to construct a simple "sparse GBP" by giving a "default value" to all but some of a region's variable assignments. Combining these tasks has not to our knowledge been given an elegant solution, and moreover it seems unclear to us that such a simplistic approach to sparse GBP is proper - in the first place, it still requires us to examine all of the possible probabilities of variables in a region before deciding which ones are small enough be discarded. However, just as GBP is parametrised by a set of regions, we

[23] M. Welling. "On the choice of regions for generalized belief propagation". In: *Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence*. 2004, pp. 585–592.

could view a hypothetical sparse GBP algorithm as being parametrised by a set of "partial assignments" (PAs), which we define as assignments to some subset of the variables in the graph. The information in a PA is the same as that which is needed to characterise a condition on a set of variables, so perhaps the task of designing a good sparse GBP algorithm runs up against the same problems as that of a parallel CBP algorithm: even if we knew of a good way for PAs to interact, we would still be faced with the problem of choosing which ones to keep track of; and such a decision involves selecting from a very large space of possibilities.

At first glance, in order to solve the problem of selecting sets of PAs (or, equivalently, conditions) one would desire a way for these PAs to compete against each other in a kind of artificial evolution. The goal would be to obtain a near-optimal set of PAs using a selection process. Perhaps such a mechanism could also be used to answer the first question as well, *viz.*, how the PAs should interact (in the message-passing sense), since competition is itself a kind of interaction. But the way forward may on the other hand involve more complex considerations, since a set of approximate marginals is calculated using *cooperation* between different parts of the set of regions/messages/PAs, and assigning or partitioning credit for the algorithm's overall accuracy among these various parts may not be as simple as asking them to *compete* as individuals. We could illustrate this with an analogy to football (soccer): for instance, we wouldn't choose each of the members of a football team by the same metrics, or by having potential members play against each other one on one; but rather, each player should be chosen to have certain qualities - offence, defence, goal-keeping, etc. - so that the combination makes a good team. Accordingly, we might want to envision a system where a set of PAs is considered to make up the "genes" of an "agent", and different combinations of genes are explored by combining the genotypes of different agents. To return to the football analogy, this would not be a very efficient way of choosing a team, but it would at least evaluate a set of players using the proper criteria, namely by their ability to function as a team when playing against some other team. Such analogies are explored more fully, and compared to prior work in the field of Genetic Algorithms, in section 6.6.

We have by this point posed so many questions as to invite the criticism that this chapter contains more speculation than actual results. Yet we hope that we may have cast some useful light on the motivation behind this research direction, and on the ways in which it can proceed, which may provide illumination for other workers investigating the same topic.

Another objective of this discussion was to present a new connection

between the subject of this chapter and the subjects of the next three chapters. The PA connection we outlined here represents more faithfully the train of thought which led us to the later investigations, and may provide a useful contrast to the original connection set forth in the introduction and referenced in the title of this dissertation, namely that of "combining approximations".

3.7 Acknowledgements

BBP was implemented using Joris Mooij's libDAI.[24] The authors would also like to thank Joris Mooij for useful discussions.

3.8 Appendix

3.8.1 BBP derivation

Applying the chain rule to the message representation in section 3.3.2 yields the following equations:

$$\not{d}\psi_i(x_i) = \left(\prod_{\alpha\sim i} m^T_{\alpha i}(x_i)\right) \not{d}\bar{b}_i(x_i) + \sum_{t=1}^{T}\sum_{\alpha\sim i}\left(\prod_{\beta\sim i\setminus\alpha} m^{t-1}_{\beta i}(x_i)\right) \not{d}\bar{n}^t_{i\alpha}(x_i) \quad (3.27)$$

$$\not{d}\psi_\alpha(x_\alpha) = \left(\sum_{i\sim\alpha} n^T_{i\alpha}(x_i)\right) \not{d}\bar{b}_\alpha(x_\alpha) + \sum_{t=1}^{T}\sum_{i\sim\alpha}\left(\prod_{j\sim\alpha\setminus i} n^{t-1}_{j\alpha}(x_j)\right) \not{d}\bar{m}^t_{\alpha i}(x_i) \quad (3.28)$$

$$\begin{aligned}\not{d}n^t_{i\alpha}(x_i) = {} & \delta^T_t \sum_{x_{\alpha\setminus i}}\left(\psi_\alpha(x_\alpha)\prod_{j\sim\alpha\setminus i} n^T_{j\alpha}(x_j)\right)\not{d}\bar{b}_\alpha(x_\alpha) \\ & + (1-\delta^T_t)\sum_{j\sim\alpha\setminus i}\sum_{x_j}\left(\sum_{x_{\alpha\setminus i\setminus j}}\psi_\alpha(x_\alpha)\prod_{k\sim\alpha\setminus i\setminus j} n^t_{k\alpha}(x_k)\right)\not{d}\bar{m}^{t+1}_{\alpha j}(x_j) \quad (3.29)\end{aligned}$$

$$\begin{aligned}\not{d}m^t_{\alpha i}(x_i) = {} & \delta^T_t\left(\psi_i(x_i)\prod_{\beta\sim i\setminus\alpha} m^T_{\beta i}(x_i)\right)\not{d}\bar{b}(x_i) \\ & + (1-\delta^T_t)\sum_{\beta\sim i\setminus\alpha}\left(\psi_i(x_i)\prod_{\gamma\sim i\setminus\alpha\setminus\beta} m^t_{\gamma i}(x_i)\right)\not{d}\bar{n}^{t+1}_{i\beta}(x_i) \quad (3.30)\end{aligned}$$

Here, $\alpha \setminus i \setminus j$ just means $(\alpha \setminus i) \setminus j$.

[24] Mooij, *libDAI 0.2.2: A free/open source C++ library for Discrete Approximate Inference methods*, op. cit.

To derive a sequential update rule, we use a different message representation, which sends only a single message $E^t = (\alpha, i)$ at time t. This representation also includes a damping factor λ, which should be 1 for no damping.

$$\bar{b}_\alpha(x_\alpha) = \psi_\alpha(x_\alpha) \prod_{i \sim \alpha} n_{i\alpha}(x_i) \tag{3.31}$$

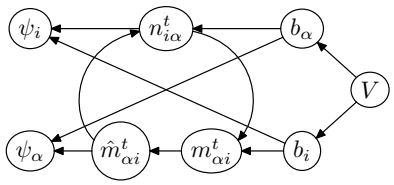
$$\bar{b}_i(x_i) = \psi_i(x_i) \prod_{\alpha \sim i} m_{\alpha i}(x_i) \tag{3.32}$$

$$\bar{n}^t_{i\alpha}(x_i) = \psi_i(x_i) \prod_{\beta \sim i \backslash \alpha} m^t_{\beta i}(x_i) \tag{3.33}$$

$$\bar{\hat{m}}^t_{\alpha i}(x_i) = \sum_{x_{\alpha \backslash i}} \psi_\alpha(x_\alpha) \prod_{j \sim \alpha \backslash i} n^t_{j\alpha}(x_j) \tag{3.34}$$

$$m^{t+1}_{\alpha i}(x_i) = (\hat{m}^t_{\alpha i}(x_i))^{\lambda \delta^{(\alpha,i)}_{E^t}} (m^t_{\alpha i}(x_i))^{1 - \lambda \delta^{(\alpha,i)}_{E^t}} \tag{3.35}$$

Here is a diagram of the dependencies in the new message equations:



Computing adjoints and eliminating t superscripts for messages (having converged), and noting that at convergence $\hat{m}_{\alpha i}(x_i) = m_{\alpha i}(x_i)$, and using pre-

computed quantities T, U, S, and R (equations 3.14, 3.15, 3.16, 3.17) yields:

$$\not{d}\psi_i(x_i) = \not{d}\bar{b}_i(x_i) \prod_{\alpha \sim i} m_{\alpha i}(x_i) + \sum_{t=0}^{T} \sum_{\alpha \sim i} \not{d}\bar{n}^t_{i\alpha}(x_i) T_{i\alpha}(x_i) \tag{3.36}$$

$$\not{d}\psi_\alpha(x_\alpha) = \not{d}\bar{b}_\alpha(x_\alpha) \prod_{i \sim \alpha} n_{i\alpha}(x_i) + \sum_{t=0}^{T} \sum_{i \sim \alpha} \not{d}\bar{\hat{m}}^t_{\alpha i}(x_i) U_{\alpha i}(x_\alpha) \tag{3.37}$$

$$\begin{aligned} \not{d}n^t_{i\alpha}(x_i) = {} & \delta^T_t \sum_{x_{\alpha \setminus i}} \not{d}\bar{b}_\alpha(x_\alpha)\psi_\alpha(x_\alpha) \prod_{j \sim \alpha \setminus i} n_{j\alpha}(x_j) \\ & + \sum_{j \sim \alpha \setminus i} \sum_{x_j} \not{d}\bar{\hat{m}}^t_{\alpha j}(x_j) S_{i\alpha j}(x_i, x_j) \end{aligned} \tag{3.38}$$

$$\begin{aligned} \not{d}m^t_{\alpha i}(x_i) = {} & \delta^T_t \not{d}\bar{b}_i(x_i)\psi_i(x_i) \prod_{\beta \sim i \setminus \alpha} m^t_{\beta i}(x_i) + (1 - \lambda \delta^{(\alpha,i)}_{E^t}) \not{d}m^{t+1}_{\alpha i}(x_i) \\ & + \sum_{\beta \sim i \setminus \alpha} \not{d}\bar{n}^t_{i\beta}(x_i) R_{\alpha i \beta}(x_i) \end{aligned} \tag{3.39}$$

$$\not{d}\hat{m}^t_{\alpha i}(x_i) = \lambda \delta^{(\alpha,i)}_{E^t} \not{d}m^{t+1}_{\alpha i}(x_i) \tag{3.40}$$

Which updates must be performed when $E^t = (\alpha, i)$? Note that $E^t = (\alpha, i) \iff \not{d}\hat{m}^t_{\alpha i}(x_i) \neq 0$. In that case, $\not{d}n^t_{j\alpha}(x_j) > 0$ for $j \sim \alpha \setminus i$. Thus $\not{d}m^t_{\beta j}(x_j)$ must be incremented for $j \sim \alpha \setminus i$ and $\beta \sim j \setminus \alpha$.

By eliminating $\not{d}n_{i\alpha}$, which is mostly zero after initialisation, we only need keep quantities $\not{d}m_{\alpha i}(x_i)$, $\not{d}\psi_i(x_i)$, $\not{d}\psi_\alpha(x_\alpha)$ between messages. The final algorithm is shown in figure 3.5.

Definitions:

$$T_{i\alpha}(x_i) = \prod_{\beta \sim i \setminus \alpha} m_{\beta i}(x_i) \tag{3.41}$$

$$U_{\alpha i}(x_\alpha) = \prod_{j \sim \alpha \setminus i} n_{j\alpha}(x_j) \tag{3.42}$$

$$S_{i\alpha j}(x_i, x_j) = \sum_{x_{\alpha \setminus i \setminus j}} \psi_\alpha(x_\alpha) \prod_{k \sim \alpha \setminus i \setminus j} n_{k\alpha}(x_k) \tag{3.43}$$

$$R_{\alpha i \beta}(x_i) = \psi_i(x_i) \prod_{\gamma \sim i \setminus \alpha \setminus \beta} m_{\gamma i}(x_i) \tag{3.44}$$

Routines:

Send-$n_{i\alpha}\left(f(x_i)\right) =$

$\bar{f}(x_i) = \frac{1}{Z_{n_{i\alpha}}} \left(f(x_i) - \sum_{x_i'} n_{i\alpha}(x_i') f(x_i') \right)$

$\not{d}\psi_i(x_i) \leftarrow \not{d}\psi_i(x_i) + \bar{f}(x_i) T_{i\alpha}(x_i)$

For each $\beta \sim i \setminus \alpha$ do:

$\not{d}m_{\beta i}(x_i) \leftarrow \not{d}m_{\beta i}(x_i) + \bar{f}(x_i) R_{\beta i \alpha}(x_i)$

Send-$m_{\alpha i} =$

$\not{d}\psi_\alpha(x_\alpha) \leftarrow \not{d}\psi_\alpha(x_\alpha) + \lambda \not{d}\bar{m}_{\alpha i}(x_i) U_{\alpha i}(x_\alpha)$

For each $j \sim \alpha \setminus i$ do:

Send-$n_{j\alpha}\left(\lambda \sum_{x_i} \not{d}\bar{m}_{\alpha i}(x_i) S_{j\alpha i}(x_j, x_i)\right)$

$\not{d}m_{\alpha i}(x_i) \leftarrow (1-\lambda) \not{d}m_{\alpha i}(x_i)$

Initialisation:

$$\not{d}\psi_i(x_i) \leftarrow \not{d}\bar{b}_i(x_i) \prod_{\alpha \sim i} m_{\alpha i}(x_i) \tag{3.45}$$

$$\not{d}\psi_\alpha(x_\alpha) \leftarrow \not{d}\bar{b}_\alpha(x_\alpha) \prod_{i \sim \alpha} n_{i\alpha}(x_i) \tag{3.46}$$

$$\not{d}m_{\alpha i}(x_i) \leftarrow \not{d}\bar{b}_i(x_i) \psi_i(x_i) \prod_{\beta \sim i \setminus \alpha} m_{\beta i}(x_i) \tag{3.47}$$

For each i, α do:

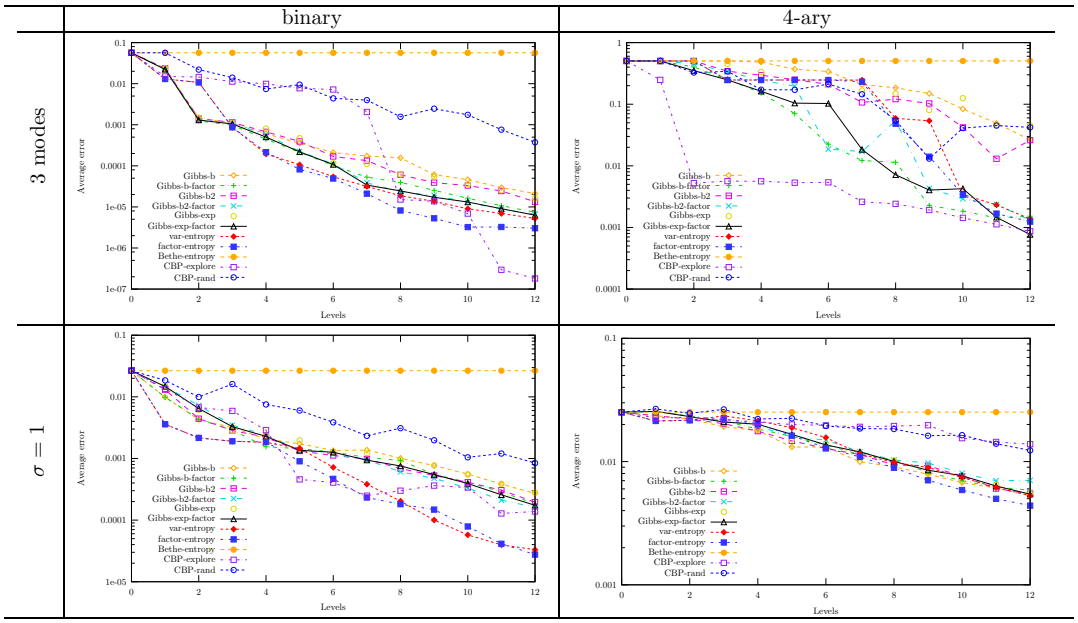
Call Send-$n_{i\alpha}\left(\sum_{x_{\alpha \setminus i}} \not{d}\bar{b}_\alpha(x_\alpha) \psi_\alpha(x_\alpha) \prod_{j \sim \alpha \setminus i} n_{j\alpha}(x_j)\right)$

Main loop:

Call Send-$m_{\alpha i}$ for each message (α, i) sent by BP

Figure 3.5: A sequential BBP algorithm

Random regular graph, 25 variables and 30 factors size 3



8 by 8 square grid

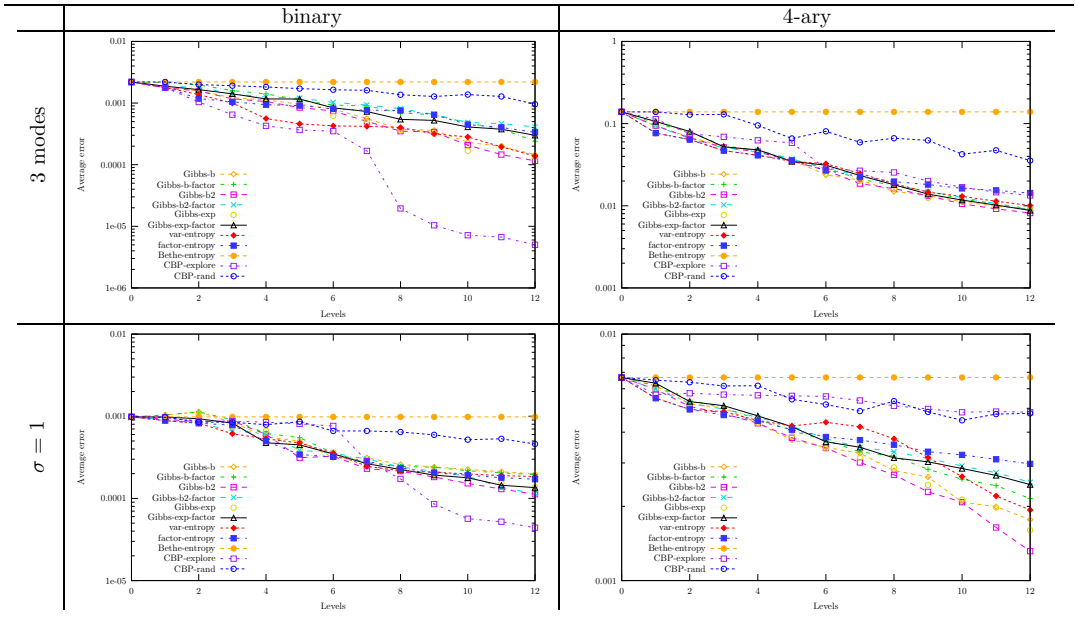


Figure 3.6: Comparison of different cost functions by clamping level: Gibbs-b (CBP-BBP): $V = \sum_i b_i(x_i^\star)$. Gibbs-b2: $V = \sum_i b_i(x_i^\star)^2$. Gibbs-exp: $V = \sum_i \exp(b_i(x_i^\star))$. Gibbs-b-factor: $V = \sum_\alpha b_\alpha(x_\alpha^\star)$ (similarly for Gibbs-b2-factor, Gibbs-exp-factor). var-entropy: $V = \sum_i H(b_i)$. factor-entropy: $V = \sum_\alpha H(b_\alpha)$. Bethe-entropy: $V = F_{\text{Bethe}}$.

Chapter 4

A conditional game for comparing approximations

Abstract

We present a "conditional game" to be played between two approximate inference algorithms. We prove that exact inference is an optimal strategy and demonstrate how the game can be used to estimate the relative accuracy of two different approximations in the absence of exact marginals. We also prove a lower bound on the game's discriminative power, and present experimental results demonstrating its superiority to existing games in statistics.[1]

4.1 Introduction

Our interest in approximate inference is partly motivated by the recognition that an ability to express beliefs is fundamental to intelligence. These beliefs may be manifested either indirectly through decisions as in betting, or directly as probabilities.[2] The problem of statistical inference - to compute such probabilities for a given probabilistic model - is powerful and general. Yet researchers should recognise that "real intelligence" goes beyond statistical inference in many ways. In particular, real intelligence is not just limited to expressing beliefs, but is also able to justify and possibly modify

[1] This chapter is substantially the same as a paper which will be published as Eaton, F. "A conditional game for comparing approximations". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Vol. 15. 2011.

[2] De Finetti, "Probabilism: A critical essay on the theory of probability and on the value of science", op. cit.

its beliefs through communication. The need for this arises not only in cases where two systems have different evidence, but also where they have reached different conclusions from the same evidence. That two systems might arrive at different beliefs about the same model follows from the reality that, because of constraints on resources, inference in most practical applications must get by with approximations. In such applications it is not feasible to establish which approximation is best by simply comparing with exact marginals, which will be unavailable. Any usable method for ranking two approximations would have to be based on some kind of direct comparison. This chapter investigates such a method, based on a two-player game.

4.2 Background

The problem of assessing the accuracy of approximations has been previously considered. In the domain of Monte-Carlo-based inference, techniques exist for determining whether a sequence of samples has converged.[3] For message-passing algorithms such as Belief Propagation, there are heuristics to bound and estimate the accuracy of the final approximation.[4] And when samples from a true distribution are available, as when inference is combined with learning, then the approximation accuracy can be estimated from the log-likelihood of a test set.

These techniques have their uses. However, data points are expensive in some domains, so it is not always possible to validate inference using a test set. And heuristics may be unsuitable for making comparisons between two different types of approximation. In general, evaluating the accuracy of an approximation against itself by some internal metric is bound to be unreliable. Comparing two approximations by self-appraisal will fail when one of the approximations is overconfident due to its having overlooked some important structure in the model, for instance in the case of a sampling run which misses an important but isolated mode.

Situations where the computational effort of multiple humans has been expended in parallel analysis of the same model are common in real life, and humans are able to reclaim this seemingly duplicated effort by resolving their disagreements through argument and debate. For intelligent systems

[3] Propp and Wilson, "Exact sampling with coupled Markov chains and applications to statistical mechanics", op. cit.

[4] J.M. Mooij and H.J. Kappen. "Bounds on marginal probability distributions". In: *Advances in Neural Information Processing Systems 21*. 2009, pp. 1105–1112; S. Shekhar. "Fixing and Extending the Multiplicative Approximation Scheme". MA thesis. University of California, Irvine, 2009.

to accomplish the same kind of cooperation, they would seem first of all to require a way of directly comparing two approximations. To the best of our knowledge, we are the first to propose a formal method for performing such comparisons.

Finally, it is perhaps worth noting that approximate inference competitions, such as the UAI approximate inference competition, currently restrict themselves to medium-sized models for which exact inference is still tractable, because there has been no good way to compare the accuracy of approximations without reference to exact marginals. As acknowledged by Bilmes (2006),[5] in such a competition it would be helpful to be able to evaluate the relative performance of algorithms on large models. Our method provides a straightforward way of carrying out such an evaluation.

4.3 The conditional game

We define a game played on a factor graph, called the "conditional game" (CG). As usual we define a distribution over n variables $x := (x_1, \ldots, x_n)$ (here assumed discrete) as a normalised product of non-negative factors ψ_α (here assumed strictly positive)

$$P(x) = \frac{1}{Z} \prod_\alpha \psi_\alpha(x_\alpha) \tag{4.1}$$

where α indexes a collection of sets of variables.[6]

Play alternates between two players, the "marginal player", MP, and the "conditional player", CP, over a total of n turns. At turn i the MP expresses marginals for variable x_i, say $q_i(x_i)$. The CP then chooses a value for x_i, say x_i^*. The variable x_i is then fixed to take value $x_i = x_i^*$ for the rest of the game. Play finishes when the variables are all fixed, giving a complete assignment $x = x^*$. A quantity which we will call the "value" of the game is then defined in terms of x^* and q:

$$V = \log \frac{\prod_{i=1}^n q_i(x_i^*)}{\prod_\alpha \psi_\alpha(x_\alpha^*)} \tag{4.2}$$

Note that if the approximations q are exact conditionals, i.e. if

$$q_i(x_i) = P(x_i | x_{1:i-1}^*) \tag{4.3}$$

[5] J. Bilmes. *UAI06 Inference Evaluation Results*. Department of Electrical Engineering, University of Washington, Seattle. 2006.

[6] Kschischang, Frey, and Loeliger, "Factor graphs and the sum-product algorithm", op. cit.

then we have

$$V = \log \frac{\prod_{i=1}^{n} P(x_i^*|x_{1:i-1}^*)}{\prod_\alpha \psi_\alpha(x_\alpha^*)} = \log \frac{P(x^*)}{\prod_\alpha \psi_\alpha(x_\alpha^*)} \tag{4.4}$$

$$= -\log Z \tag{4.5}$$

Thus if MP is exact, the choices of CP have no effect on the value V of the game.

4.3.1 From approximations to players

To make the conditional game a game, one player should be trying to maximise V and the other to minimise V. It doesn't matter who does which, as long as the two players are in competition.

Now suppose that MP is trying to maximise V, and CP to minimise it. Because probabilities sum to one, if CP has access to exact conditioned marginals then it is possible for him to guarantee through appropriate choice of x_i^* that

$$q_i(x_i^*) \leq P(x_i^*|x_1^*, \ldots, x_{i-1}^*) \tag{4.6}$$

If MP is not exact, then at least one of these inequalities can be made strict, in which case it follows that $V < -\log Z$. Thus exact marginals, yielding $V = -\log Z$, are the optimal (minimax) strategy for MP.

Given an approximation $Q(x;\psi)$ it is straightforward to derive an MP strategy: at turn i, modify the model ψ to condition on the appropriate variables (perhaps this may be implemented by introducing new factors $\prod_{k=1}^{i-1} \delta(x_k, x_k^*)$), and set q_i to the resulting approximate marginal[7] of x_i under Q

$$q_i(x_i) = Q(x_i|x_{1:i-1}^*) \tag{4.7}$$

$$\equiv Q(x_i; \prod_\alpha \psi_\alpha(x_\alpha) \prod_{k=1}^{i-1} \delta(x_k, x_k^*)) \tag{4.8}$$

Note that for most message passing algorithms, the cost of recomputing marginals after imposing a new condition can be mitigated by reusing the messages between runs. If two parts of the graph are uncorrelated or weakly correlated, then a variable in one part of the graph can be conditioned without affecting the messages in the other part.

[7] In (4.8) we adopt the notation $Q(x;\psi)$ to indicate Q's approximation to a model which is specified by the factors ψ.

Suppose that the conditional player CP trusts a different approximation to Q, call it R. A strategy for CP can be derived which employs R. In this case, CP has multiple options, but it seems sensible for him to choose at turn i:

$$x_i^* = \operatorname*{argmin}_{x_i} \frac{q_i(x_i)}{R(x_i|\ldots)} \tag{4.9}$$

which is guaranteed to satisfy (4.6) if R is exact, and to do so strictly if MP is not exact. Also, it is optimal at each turn, under the assumption that MP might play optimally for the rest of the game.

A general strategy for MP or CP could be arbitrarily complex, for instance attempting to look several moves ahead by simulating the opposing player. This would presumably be more expensive (or error-prone) than simply coming up with a more accurate approximation and using it in the "naive" strategies above. Thus we will assume below that an approximation will always be associated with one of the recommended strategies (including the amendment for CP of section 4.3.3). This allows us to drop the distinction between approximations and players, and to view V as a function of two approximations. We will write $V^+(Q, R)$ for the game value when CP is trying to maximise V using approximation R against MP's Q; and similarly $V^-(Q, R)$ for when CP is minimising V.

We now illustrate the CG with a simple example. The model is the fully-connected graph with four binary variables and six pairwise factors, each with entries $\begin{bmatrix} 0.1 & 1 \\ 1 & 1 \end{bmatrix}$. MP uses Belief Propagation[8] and CP uses Gibbs sampling with 10^3 passes. CP tries to minimise V.

The game is depicted in table 4.1. Shown are the probabilities that a variable takes the value 1, i.e. $q_i(x_i = 1)$. The final variable assignment is $x^* = (1, 1, 0, 0)$. The final value of the game is $V = \log \frac{0.743 \times 0.705 \times (1-0.645) \times (1-0.909)}{0.1} = -1.778$. The true $\log Z$ is 1.723.

4.3.2 A bound

We have seen that if MP plays exact marginals, the game value will be optimal, with $V = -\log Z$. We can also derive a simple bound on V in the case that MP's marginals are not exact. We will assume that CP is trying to minimise V, but results for the opposite case are analogous. Let $p_i(x_i) = P(x_i|x^*_{1:i-1})$ denote the conditioned exact marginals.

[8] Pearl, "Fusion, propagation, and structuring in belief networks", op. cit.

i	x_1	x_2	x_3	x_4	MP		CP
1	?				0.743	$<$	0.798
2	1	?			0.705	$<$	0.738
3	1	1	?		0.645	$>$	0.628
4	1	1	0	?	0.909	$>$	0.908
	1	1	0	0			

Table 4.1: An example game

Theorem 10.

$$V^- \geq -\log Z - \sum_i \max_{x_i} |\log q_i(x_i) - \log p_i(x_i)|$$

Proof. Let $d = \log \prod_{i=1}^n \frac{q_i(x_i^*)}{p_i(x_i^*)}$, then we can write $V = d - \log Z$. We have

$$d \geq \sum_i \min_{x_i} \log \frac{q_i(x_i)}{p_i(x_i)} = -\sum_i \max_{x_i} \log \frac{p_i(x_i)}{q_i(x_i)} \tag{4.10}$$

$$\geq -\sum_i \max_{x_i} |\log q_i(x_i) - \log p_i(x_i)| \quad \square \tag{4.11}$$

Thus, if we can guarantee that all of MP's marginals are within a certain distance (measured between logarithms) from the exact marginals, then we can lower-bound V. The accuracy constraint must hold for both unconditioned and conditioned marginals, but approximations usually become more accurate with conditioning, so this theorem gives some intuition as to the relationship between V and the error of the node marginals. We note, however, that the CG is less concerned about the L_1 error, and more concerned about absolute error in log-marginals, to which we refer as the $L_1^{\log}$ error. For example, estimating 10^{-3} when the true probability is 10^{-4} would give a greater $L_1^{\log}$ error than estimating 0.2 when the true probability is 0.3, even though the L_1 error is greater in the second case.

4.3.3 Variable order

There is nothing special about the order $i = 1, \ldots, n$ in which variables are conditioned at each turn, so it is possible to have CP specify a different

order by choosing a variable as well as a value during his turn. (MP must also be modified so that at each turn he specifies marginals for all variables, and not just for the next variable, which he can no longer predict.) In the new flexible-order setting, simply extending the optimisation of equation 4.9 to variables gives a similar optimality property. Thus at turn t, CP now chooses

$$(i_t, x^*_{i_t}) = \operatorname*{argmin}_{\substack{(j,x_j) \\ j \notin i_{1:t-1}}} \frac{Q(x_j|x^*_{i_{1:t-1}})}{R(x_j|x^*_{i_{1:t-1}})} \tag{4.12}$$

where Q is MP's estimate and R is CP's. The extra freedom for CP allows us to prove a complementary bound to the previous one. Assume, again, that CP wants to minimise V and has access to exact marginals P.

Theorem 11. *If CP is allowed to choose the variable ordering, then he can achieve* $V^- \leq -\log Z - \max_{(i,x_i)} \log \frac{P(x_i)}{Q(x_i)}$

Proof. $V = -\log Z + d$ where

$$d = \sum_{i=1}^{n} \log \frac{Q(x^*_i|x^*_{1:i-1})}{P(x^*_i|x^*_{1:i-1})} \tag{4.13}$$

An optimal CP will force each term of d to be negative (or zero). Taking only the first, we have $d \leq \log \frac{Q(x_1)}{P(x_1)}$. But the variable ordering is now decided by CP, who can choose the first variable to get the tightest bound. He also chooses the variable's value, so

$$d \leq \min_{(i,x_i)} \log \frac{Q(x_i)}{P(x_i)} = -\max_{(i,x_i)} \log \frac{P(x_i)}{Q(x_i)} \tag{4.14}$$

□

4.3.4 The comparison of approximations

Having defined the conditional game, we now describe how this game can be used to compare two approximate inference methods.

The value V of a game is a number typically near $-\log Z$ (with equality in the case of an exact MP). We could declare a "winner" by comparing V to $-\log Z$, but the true value of $-\log Z$ is unknown and intractable. To identify the most accurate of two approximations, it is helpful to have a

score which can be compared to zero. Call the two approximations Q and R and define the "difference score" by

$$S^-(Q,R) = V^-(Q,R) - V^-(R,Q) \tag{4.15}$$

i.e. the difference between two game values, played with approximations switching roles as CP and MP, and CP minimising V. This will be ≥ 0 if Q is exact. We also define S^+ analogously using V^+, that is, where CP is maximising V.

We combine S^+ and S^- to get a "four-way score", based on the outcomes of four games[9]:

$$S_4(Q,R) = S^-(Q,R) - S^+(Q,R) \tag{4.16}$$

The advantage of S_4 can be expressed as follows. The difference score S^- selectively penalises *under*-estimates of small probabilities by MP, while S^+ penalises *over*-estimates. For example, if MP under-estimates 0.01 for $P(x_i = 0)$ when the true probability is 0.1, and CP is trying to maximise V, then CP will be forced to choose the alternate value $x_i = 1$ (to which MP assigns probability 0.99) since he is only looking for over-estimates. The absolute contribution to the error (e.g. d, equation 4.13) will then be $\left|\log \frac{0.99}{0.9}\right| = 0.1$ rather than the much larger $\left|\log \frac{0.01}{0.1}\right| = 2.3$.

Our proposed method has now evolved from a simple two-player game with fixed roles into a more complex ritual incorporating four such games, during which players switch roles and objectives. The final product may seem ad-hoc and inelegant. It may help to draw a comparison to legal procedure, in which a simple building block - the questioning of a witness - is employed in four ways to achieve a "fair trial". The witness may be called by the defence or the prosecution, and may be examined and cross-examined.

Finally, we combine the ideas of Theorems 10 and 11 to prove a simple bound on S_4.

Theorem 12. *Suppose that we are given two approximations Q and R to a true distribution P, with*

$$\sum_t \left| \log \frac{R(x^*_{i_t} | x^*_{i_{1:t-1}})}{P(x^*_{i_t} | x^*_{i_{1:t-1}})} \right| \leq \delta \tag{4.17}$$

[9] If CP uses the rule of equation 4.12 to choose (variable, value) pairs, then the four-way score incorporates four terms corresponding to the values of four games. However, there are only two state configurations x^*, since the configuration which a Q CP chooses when maximising V against a R MP is the same as that chosen by a R CP when minimising V against a Q MP. Thus there are two pairs of terms incorporating the same unnormalised probabilities $\prod_\alpha \psi_\alpha(x^*_\alpha)$. However, the terms in each pair do not cancel out, because they occur with the same sign.

for all x^* and all sequences $i_{1:t}$, while

$$\max_{(i,x_i)} \left| \log \frac{Q(x_i)}{P(x_i)} \right| \geq \epsilon \tag{4.18}$$

Then $S_4(R,Q) \geq \epsilon - 5\delta$.

Proof. Write $S_4(R,Q) = V^-(R,Q) - V^-(Q,R) - V^+(R,Q) + V^+(Q,R)$. We bound each of the terms:

(a) By Theorem 10, $V^-(R,Q) \geq -\log Z - \delta$ and $V^+(R,Q) \leq -\log Z + \delta$.

(b) We bound $V^-(Q,R)$:

$$\begin{aligned}
&V^-(Q,R) + \log Z && (4.19)\\
&= \sum_t \log \frac{Q(x^*_{i_t} | x^*_{i_{1:t-1}})}{P(x^*_{i_t} | x^*_{i_{1:t-1}})} && (4.20)\\
&= \sum_t \left(\log \frac{Q(\ldots)}{R(\ldots)} + \log \frac{R(\ldots)}{P(\ldots)} \right) && (4.21)\\
&\leq \sum_t \log \frac{R(\ldots)}{P(\ldots)} && (4.22)\\
&\leq \delta && (4.23)
\end{aligned}$$

Equation 4.22 follows from the fact that CP will choose $\log \frac{Q}{R}$ to be negative.

(c) For the last term $V^+(Q,R)$, suppose the first condition of the game is $(i_1, x_{i_1}) = (k, x^*_k)$. Let (j, x^*_j) be the maximising assignment in equation 4.18, so that either $\log \frac{Q(x^*_j)}{P(x^*_j)} \geq \epsilon$ or $\leq -\epsilon$. Assume the first case; the proof for the second follows by similarly modifying part (b) above. Now,

$$\begin{aligned}
\log \frac{Q(x^*_k)}{R(x^*_k)} &\geq \log \frac{Q(x^*_j)}{R(x^*_j)} && (4.24)\\
&= \log \frac{Q(x^*_j)}{P(x^*_j)} - \log \frac{R(x^*_j)}{P(x^*_j)} && (4.25)\\
&\geq \epsilon - \delta && (4.26)
\end{aligned}$$

Then as in (b),

$$V + \log Z \tag{4.27}$$
$$= \sum_t \left(\log \frac{Q(\ldots)}{R(\ldots)} + \log \frac{R(\ldots)}{P(\ldots)} \right) \tag{4.28}$$
$$\geq \epsilon - \delta + \sum_t \log \frac{R(\ldots)}{P(\ldots)} \tag{4.29}$$
$$\geq \epsilon - 2\delta \tag{4.30}$$

where equation 4.29 follows from using 4.26 and $i_1 = k$ for the first term of the summation.

Combining (a), (b), and (c) gives $S_4 \geq \epsilon - 5\delta$.

□

In other words, if we can bound the total $L_1^{\log}$ error of one approximation above by δ, and if we know that another approximation does worse than ϵ for the maximum error of one of its variable marginals, and if $\epsilon - 5\delta > 0$, then the first approximation will win against the second one by the four-way score.

This bound is strict and so assumes the worst case scenario for every game. A probabilistic analysis estimating average-case performance given a random distribution of marginal errors might provide a more realistic picture of the CG's effectiveness, but we do not undertake such an analysis here.

4.4 Experiments

4.4.1 Alarm graph

We first present the results of playing the conditional game between five different pairs of approximate inference algorithms, using the implementation in libDAI,[10] running on the "alarm graph" found in libDAI, with 37 variables. The algorithms we consider are:

- **Gibbs** - Gibbs sampling, with 10^5 passes.
- **BP** - Belief Propagation, sequential updates.[11]

[10] J.M. Mooij et al. *libDAI 0.2.5: A free/open source C++ library for Discrete Approximate Inference*. http://www.libdai.org/. 2010.

[11] Pearl, "Fusion, propagation, and structuring in belief networks", op. cit.

- **CBP** - Conditioned Belief Propagation, with 4 levels.[12]
- **TreeEP** - Tree Expectation Propagation.[13]
- **LCBP** - Loop Corrected Belief Propagation.[14]

The L_1 and $L_1^{\log}$ errors of the algorithms are shown in table 4.2. The S_4 scores are shown in table 4.3. We see that the S_4 scores agree with the *average* $L_1^{\log}$ errors on all pairs except BP vs Gibbs, where Gibbs wins even though it has a larger average $L_1^{\log}$ error. But note that the *maximum* $L_1^{\log}$ error of Gibbs is smaller than that of BP, so there is at least one sensible error measure which is consistent with the result of the game in every case.

Method	avg L_1	avg $L_1^{\log}$	max $L_1^{\log}$
LCBP	8.981e-05	5.586e-4	0.01684
TreeEP	0.008652	0.04424	0.5475
CBP	0.01110	0.05355	1.256
BP	0.01627	0.0712988	1.6424
Gibbs	0.02251	0.2111	0.8298

Table 4.2: Errors between approximate and exact variable marginals for different approximations.

S_4 vs:	TreeEP	CBP	BP	Gibbs
LCBP	5.2507	13.795	22.75	12.998
TreeEP		8.383	13.453	3.996
CBP			27.575	3.734
BP				-4.032

Table 4.3: Scores of games between approximations

4.4.2 Generalised Belief Propagation

We might also be interested in measuring the relationship between score and error for multiple models and a larger space of approximations. To this end we used approximations consisting of Generalised Belief Propagation

[12] Eaton and Ghahramani, "Choosing a variable to clamp: approximate inference using conditioned belief propagation", op. cit.

[13] Minka and Qi, "Tree-structured approximations by expectation propagation", op. cit.

[14] Mooij et al., "Loop corrected belief propagation", op. cit.

(GBP)[15] on a fully connected binary pairwise factor graph with triangular regions (regions of size 3).[16] Each approximation was defined by a random set of triangular regions, chosen to be non-singular.[17] As models we used factor graphs of 7 nodes with edge potentials drawn as $\exp(2W)$, with W a standard normal deviate. Figure 4.1 plots the results of playing the CG between 16 random pairs of GBP approximations on each of 120 random models; shown is the four-way score S_4 and the difference in $L_1^{\log}$ error.[18] Figure 4.2 plots the same results but shows difference in L_1 error instead, illustrating that the S_4 score is better at capturing relative $L_1^{\log}$ error than L_1 error.

We will term the "agreement rate" of the CG against a certain error metric as the rate at which the CG correctly identifies the approximation with smallest error. This depends on the particular set of approximations which are being compared (in our case, GBP with random non-singular sets of triangular regions). The agreement rate can be estimated from the fraction of points in the first and third quadrant in figure 4.1 and figure 4.2. For $L_1^{\log}$ error, the estimated agreement rate was 0.754. For L_1 error, it was 0.639.

[15] Yedidia, Freeman, and Weiss, "Generalized belief propagation", op. cit.

[16] For reliable convergence, our implementation used the algorithm of Heskes, Albers, and Kappen (T. Heskes, K. Albers, and B. Kappen. "Approximate inference and constrained optimization". In: *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence*. Vol. 13. 2003, pp. 313–320), which has the same fixed-points as GBP. We ran it with a tolerance of 10^{-7}.

[17] Welling, Minka, and Teh, "Structured region graphs: Morphing EP into GBP", op. cit.

[18] Each point is also reflected about the origin.

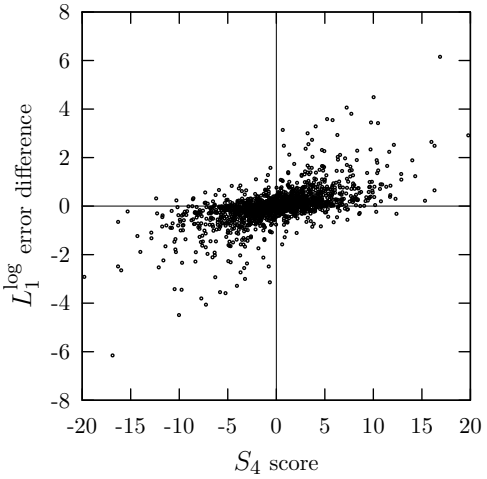


Figure 4.1: Four-way score vs difference in $L_1^{\log}$ error for GBP

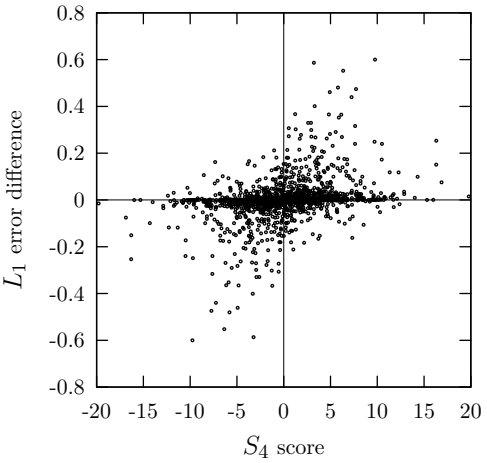


Figure 4.2: Four-way score vs difference in L_1 error for GBP

4.4.3 Comparison to code-length game

We next compare the effectiveness of the conditional game against another simple game, a modification of the "code length game".[19] The outcome of

[19] F. Topsøe. "Information theoretical optimization techniques". In: *Kybernetika* 15.1 (1979), pp. 8–27.

the code-length game (CLG) is defined as follows:

$$\max_{p:\sum_x p(x)=1} \min_{\kappa:\sum_x e^{-\kappa(x)}\leq 1} \mathbb{E}_p \left[\kappa + \sum_\alpha \log \psi_\alpha \right] \tag{4.31}$$

We have added the term $\sum_\alpha \log \psi_\alpha$ to achieve the correct equilibrium for the model. The standard interpretation of this saddle point is that one player chooses a set of code lengths satisfying the Kraft inequality, while another chooses a normalised distribution (P) over symbols. The first player wants to minimise the (modified) expected code length, and the second to maximise it. The equilibrium is at $p = e^\kappa = P$. Note that a sample from this expectation (sign inverted) can be implemented by changing the behaviour of CP in the CG so that he chooses a value randomly from his own distribution $R(x_i|\ldots)$ at each turn. This is not a good strategy for CP in the CG, since in particular it ignores the marginals proposed by MP, but the CLG is a simultaneous game, where each player is unaware of the other's actions, and so in that setting CP (i.e., the distribution player) should act randomly. If the distribution player wants to do well in the CLG in expectation, then his best strategy is to sample as described above. The expected value of the game is

$$\mathbb{E}[V] = \mathbb{E}_R \left[\log \frac{Q}{\prod_\alpha \psi_\alpha} \right] \tag{4.32}$$

$$= \mathbb{E}_R \left[\log \frac{Q}{P} \right] - \log Z \tag{4.33}$$

where Q is MP's approximation. This is equal to $-\log Z$ if Q is exact, and less than or equal to $-\log Z$ if R is exact. Switching roles and subtracting game values yields a score, analogous to the S^- difference score, which can be compared to zero. The drawback of the CLG is that its outcome is stochastic, and so one must average over many trials to get a score of low variance. As a consequence, one might object that a comparison between the CG and CLG is unfair. However, the CLG is the only other game of this type, of which we are aware.

We want to show that the CG is better on average than the CLG at discriminating the error of many similar approximations. For this we used the same GBP approximations as in section 4.4.2 (parametrised by triangular region configurations) on the same distribution over models. We played these approximations against each other in a "single-elimination tournament" (SET). Players are initialised at the leaves of a binary tree of uniform

depth (here 8), and each node represents the winner of a game played between its two children. The "round" of a node is its distance from the leaves. See figure 4.3.

The tournament was repeated with different methods of comparing approximations:

- conditional game (S_4, S^+, S^-);
- code-length game (averaging over 1, 3, and 8 runs);
- exact comparison (comparing actual error of approximations).

The "exact" method is shown only as a reference, as we are ultimately interested in problems for which exact marginals are intractable.

The results are shown in figure 4.4. The error shown in this plot is average $L_1^{\log}$ over variable marginals, averaged over all approximations in the same round, geometrically averaged over 120 random factor graphs generated as above. In both cases one can see that S_4 outperforms S^+ and S^- by a small amount, while the code length game performs poorly. In both cases, the slope of the S_4 curve was close to half of the slope of the exact reference curve.

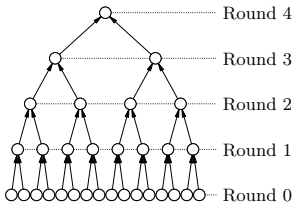


Figure 4.3: Schematic of the single-elimination tournament on a binary tree

It is interesting to see how the agreement rate, defined in section 4.4.2, changes as a function of tournament round. For $L_1^{\log}$ error and S_4 score, the agreement rates (averaged over all the approximations and all the graphs) for tournament rounds one through eight are shown in table 4.4. There is a downward trend for both games, which means that they are having a more difficult time discriminating errors with each new round. This is consistent with the usual state of affairs when a tournament is being played - it is easier to predict the outcome of earlier matches than later ones, since the earlier matches are more likely to involve an uneven pairing of players.

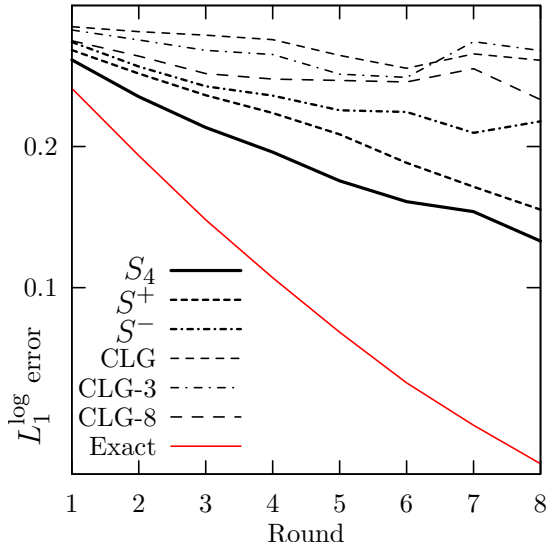


Figure 4.4: Plot of $L_1^{\log}$ error as a function of round, for tournament experiment (round 0 omitted).

Round	1	2	3	4	5	6	7	8
S_4	0.71	0.68	0.67	0.65	0.65	0.63	0.55	0.53
CLG	0.55	0.53	0.53	0.51	0.49	0.56	0.33	0.50

Table 4.4: Agreement rates vs $L_1^{\log}$ error for S_4 and CLG

4.5 Discussion and future work

We have described a technique for comparing two different approximations to a statistical model. The only interface requirement for the approximation algorithms is that they support variable conditioning, i.e. can give estimates of marginals in a conditioned model where a variable is conditioned to take a given value. Some algorithms which satisfy this requirement particularly well are Belief Propagation[20] and instances of Expectation Propagation,[21] and GBP.[22]

The original motivation of this research was to explore ways of moving

[20] Pearl, "Fusion, propagation, and structuring in belief networks", op. cit.
[21] Minka, "Expectation propagation for approximate Bayesian inference", op. cit.
[22] Yedidia, Freeman, and Weiss, "Generalized belief propagation", op. cit.

beyond the dominant approximate inference framework, in which algorithms are only able to express beliefs. It seemed that if one were to be more adventurous, as a natural progression one might seek frameworks in which algorithms are able to defend or modify their beliefs through dialog. We decided that an appropriate prototype for such communication should be a two-player game. This was supported partly by the observation that there is no easy fitness function with which to measure the error of an approximation, but that relative comparisons (such as the code-length game) are possible. We also noticed that two-player games already appear in many places in machine learning, in the form of saddle points $\min_x \max_y f(x, y)$: for example in the Convex-Concave Procedure,[23] Tree-Reweighted Belief Propagation,[24] Boosting,[25] and the EM algorithm.[26]

There is also a well-known (to formal semanticists) two-player game which can be used to define the truth value of a formula in first-order logic. The state of the game is a node in the syntax tree of the formula. Play starts at the root. A "falsifier" chooses branches of conjunctions (AND clauses) which he thinks are false, while a "verifier" chooses branches of conjunctions (OR clauses) which he thinks are true. Upon encountering a negation, they switch roles. (The falsifier and verifier can also instantiate the arguments of $\forall$ and $\exists$ quantifiers, respectively.) The formula is true if and only if the verifier can win.[27] We find this game particularly interesting, although it is not clear what kind of analogy best relates it to the conditional game.

Finally, we note that there is a body of literature which applies iterative, message-passing-like algorithms to look for solutions of games which have a graphical structure, called "graphical games".[28] We have not found a way to make use of it here.

We have made preliminary attempts to harness the conditional game in an approximate inference method, by using it to guide a kind of natural

[23] Yuille and Rangarajan, "The concave-convex procedure", op. cit.

[24] M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky. "Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudomoment matching". In: *Workshop on Artificial Intelligence and Statistics*. Vol. 21. 2003.

[25] Y. Freund. "Boosting a weak learning algorithm by majority". In: *Information and computation* 121.2 (1995), pp. 256–285.

[26] A.P. Dempster, N.M. Laird, and D.B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39.1 (1977), pp. 1–38.

[27] P. Lorenzen and K. Lorenz. *Dialogische logik*. Wissenschaftliche Buchgesellschaft Darmstadt, Germany, 1978.

[28] M. Kearns, M. Littman, and S. Singh. "Graphical models for game theory". In: *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*. 2001, pp. 253–260.

selection between competing approximations. These are described in more detail in chapter 6.

A fundamental drawback of the conditional game is that it requires a complete traversal of all variables in the model, where the algorithm must be re-run once for each variable. This is still faster than the presumably exponential cost of exact inference, but would seem unsuitable for large real-world models. One remedy would be to use approximate inference algorithms that "compile" a model into a form by which conditional and marginal queries can be executed quickly. An example of such an algorithm is described in recent work applying Arithmetic Circuits[29] to approximate inference.[30]

Ideally, it would be possible to devise a game which can be played locally on the nodes of a graphical model, so that inference in different weakly-coupled areas of the model can proceed asynchronously, together with co-evolution towards locally superior approximations. It is not yet clear how this could be done.

In conclusion, we have presented a novel game which can be used for comparing approximate inference algorithms in the absence of exact marginals. We have shown that it has exact inference as an optimal strategy, and we have proven theoretical bounds on its performance in the case where neither player is exact. We have presented experimental results which demonstrate its effectiveness in distinguishing inference algorithms on a graph of moderate difficulty, the alarm graph. We have experimentally demonstrated its superiority to another simple game, the code-length game, for the purpose of comparing approximations based on GBP. We hope that this research will help generate interest in applications, techniques, and formalisms for approximate inference which extend beyond the current paradigm of simply expressing beliefs.

4.6 Acknowledgements

The author would like to thank Iain Murray for useful discussions.

[29] A. Darwiche. "A Differential Approach to Inference in Bayesian Networks". In: *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*. 2000, pp. 123–132.

[30] D. Lowd and P. Domingos. "Approximate Inference by Compilation to Arithmetic Circuits". In: *Advances in Neural Information Processing Systems 23*. 2010, pp. 1477–1485.

Chapter 5

Guided inference: a protocol for learning to do inference

Abstract

We propose a protocol for modelling the exchange of advice between two approximations to a statistical model. In our protocol a "student" advertises marginal probabilities, and a "teacher" chooses an example state to show to the student. The student observes the model's unnormalised joint distribution at the new state. This interaction is repeated over a number of turns. We present results from experiments evaluating the ways in which the teacher might choose states to show the student.

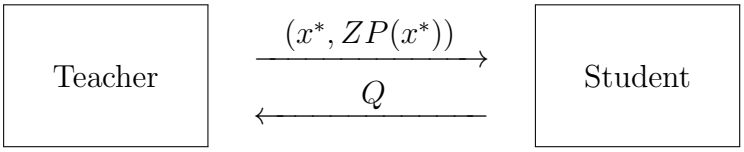
5.1 Introduction

One of the classical ways in which approximate inference can fail to produce good marginals is by failing to find all of the modes - areas of high probability - of a distribution. This is especially true for MCMC methods, which may have difficulty in moving from one mode to another, but it is also true for deterministic methods such as belief propagation, which does well on distributions with one or two modes but not three or more.[1]

In this section we present the results from some experiments which are designed to explore ways in which one approximation can "teach" another about the modes (or other regions of interest) of a distribution. A more

[1] This phenomenon was observed while doing research for chapter 3, although we didn't mention it there.

mature form of such a technology would facilitate the sharing of information between approximations, for instance by making it easier to reuse work between multiple MCMC runs. One might imagine a number of ways in which such an interaction could take place, but we use a simple scenario, in which the "teacher" shows the "student" examples consisting of states (full assignments of all the variables) of the model. The student is allowed to evaluate the full unnormalised joint at these states, but knows nothing else about the model. We shall call this kind of interaction "guided inference".



If the purpose of the conditional game is to be a protocol by which approximations can "argue" about areas of disagreement, the purpose of guided inference could be seen as a (very simple) protocol through which these disagreements can be resolved.

The interaction cycle is formalised in the following pseudocode:

Algorithm 13. *Guided inference*
Repeat for an arbitrary number of turns:
At turn m:

1. *Student proposes a distribution Q based on the example states and unnormalised joints he has seen so far: $\{(x^{*(i)}, ZP(x^{*(i)}))\}_{i=1:m-1}$ (and no other information about the model)*

2. *Teacher selects a new example point $x^{*(m)}$ (perhaps in response to errors in the student's distribution)*

Our model of interaction is based on the key simplifying assumption that the only information which goes from the teacher to the student consists of examples of states of the model. The student has access to unnormalised joint probabilities ZP, but only evaluates them at the states recommended by the teacher; hence it is also possible to imagine the values $ZP(x^*)$ as being transmitted by the teacher together with the states x^*, as in the above diagram. This is meant to capture the extreme of passivity, where the student knows the model specification, but (not knowing where to start with his analysis) only performs computations when prompted with example states received from the teacher. Importantly, the student does not get to know any of the teacher's opinions about the marginals or partition function.

This means that the student is not limited by the accuracy of the teacher (and could learn just as easily from multiple teachers as from one).

In an applied setting, the "teacher" and "student" might both be approximations, and the student would be an approximation Q of a kind which can be parametrised by a set of states so that he learns to be more accurate with every state (example) shown to him by the teacher. But we do not yet concern ourselves with figuring out how an approximate student should make the best use of these examples for learning. In this chapter we restrict our attention to the question of how the teacher should choose examples to present to the student. We are limiting ourselves, in other words, to investigating one half of the guided inference problem, namely, the teacher's strategy. A practical application of the conclusions we derive from our experiments would also require a solution to the second half, namely, a specification of the student's approximation.

Now, if we were to experiment in a setting with an approximate Q, then it would be difficult to know whether to credit some property of the system to the teacher's or the student's approximate inference algorithm (which we are not particularly interested in) or to the teacher's protocol for choosing example points to present to the student (which is what we are interested in). Consequently, in our experiments in this chapter we use two *exact* inference algorithms. The inference of the "teacher" is simply exact, while the student performs "exact" inference (combined with exact sampling) on a simple distribution over models, conditioned to agree with the observations of the unnormalised joint which the teacher has indicated to him.

A consequence of the preliminary nature of this research is that, as in other chapters, we are constrained to study small models on which exact inference is tractable. Our expectation is that these results will generalise to larger models as well.

5.2 Prior work

In this section we relate our work to previous research in machine learning. The problem of learning about a model by observing samples from some or all of its variables is common to most branches of machine learning. When some property of these samples is specified by the learner prior to sampling, in order to optimise learning, then the process is called "active learning"[2] (two related topics are "query learning" and "optimal experiment design").

[2] S. Tong. "Active learning: theory and applications". PhD thesis. Stanford University, 2001.

Active learning is perhaps the closest existing body of research to what we are calling "guided inference". Active learning can be employed to learn the parameters or structure of a model, as with any other form of learning. Typically, an active learner is allowed to specify values of some subset of the variables in a model, and the rest of the variables are then sampled from the true distribution conditioned on these assignments. Alternatively, a learner may have access to a set of unlabelled data points, from which he is allowed to select examples whose labels should be determined by an oracle. The learner presumably chooses these examples in such a way as to optimise his ability to predict labels for the rest of the data. Active learning can be seen as a way of modelling a scientist, who tries to learn about a system of interest by performing a series of experiments in which he constrains some aspects of the system's behaviour and measures the effects of such interventions on other parts of the system.

Guided inference can be seen as a kind of active learning for inference. In the framework of active learning, a learner is trying to learn about a model given data points which are partly specified by the learner and partly drawn at random from a "true distribution". In guided inference, by contrast, the model is considered to be fully specified and the learner is trying to learn how to improve his approximation to this model. He does this by receiving "interesting" states of the model from a "teacher" whose approximation he wishes to emulate. The teacher of guided inference corresponds to the "true distribution" of active learning. Whereas the states in active learning are partly random and this randomness is used to learn about the probability mass assigned by the model to different variable values, in guided inference there is not necessarily any randomness in the states, and the behaviour of the model at the example states is inferred from the unnormalised joint.

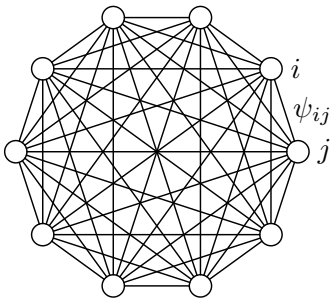
One area of similarity between our work in guided inference and existing work in active learning is the use of a cost function to guide the choice of new states. In active learning, the learner has a sense of the areas of the model about which he would like to be more accurate. His preferences can be expressed in terms of an expected gain for each possible query he can make - perhaps he would like to minimise some form of entropy in his beliefs about the model. In guided inference, the teacher is trying to improve some measure of the student's error, and might employ one of a number of cost functions in deciding which state to show the student at each turn (see the experiments section 5.4).

In the guided inference setting, since approximate inference algorithms are doing the teaching and the learning, one way in which the teacher can choose states for the learner is by playing the conditional game against him.

The conditional game identifies at its conclusion a full configuration, or state, which can be used for this purpose. We have already contrasted the guided inference framework, which seeks to resolve disagreements between approximations, with the conditional game, which seeks to identify and quantify these disagreements. In our experiments, we will show that the conditional game is suitable for use by the teacher in the guided inference framework as well.

5.3 Distributions over models

In this section we describe how the "exact" student of our experiments maintains and reasons about a distribution over models, given a set of examples from the unnormalised joint distribution $ZP(x) = \prod_\alpha \psi_\alpha(x_\alpha)$. We will restrict ourselves to representing distributions over the potentials of fully connected binary pairwise factor graphs. (Thus, we do not consider the problem of reasoning about models of differing structure, e.g. averaging over multiple hypotheses which specify different sized factors or different sparse connectivity.) We have the student represent the potentials of the graph as exponentials of normal random variables, $\psi_{ij}(x_i, x_j) = \exp(\beta W)$ where $W \sim N(0, 1)$.



(5.1)

Initially the potentials are believed by the student to be sampled independently, but when he incorporates his set of observations of the unnormalised joint $\{(x^{*(i)}, ZP(x^{*(i)}))\}_{i=1:m-1}$, then correlations will be introduced in his beliefs. If he represents the log-potentials using a multivariate normal, then the correlations can be represented in a covariance matrix for these quantities, and after each observation the posterior of his beliefs will be in the same class as the prior (i.e., it is a conjugate prior). The observations

$$z^* = ZP(x^*) \tag{5.2}$$

are equivalent to

$$z^* = \prod_{jk} \psi_{jk}(x^*_{jk}) \tag{5.3}$$

$$\log z^* = \sum_{jk} \log \psi_{jk}(x^*_{jk}) \tag{5.4}$$

The quantities $\log \psi_{jk}(x_j, x_k)$ are distributed according to a multivariate normal distribution with mean μ and variance Σ (indexed by $(j, k > j, x_{jk})$ and initialised to $\beta^2 I$) so this is a statement that some subset of the dimensions of this normal distribution should have a certain sum (namely $\log z$). A set of such constraints is in turn a special case of a linear constraint, say

$$B \cdot y = v \tag{5.5}$$

on draws y from a multivariate normal, where B is a matrix and v a vector. More specifically, in our experiments, y is indexed by $(j, k > j, x_j, x_k)$ and represents a vector of log potentials specifying the whole model, while B contains entries which are 0 or 1 according to whether a particular potential entry contributes to a given state, and v is a column vector of the log unnormalised joint entries corresponding to each example $x^{*(i)}$: $v_i = \log z^{*(i)} = \log ZP(x^{*(i)})$.

Conditioning on this linear constraint is equivalent to transforming the mean and the variance of the multivariate normal distribution:

$$\mu' = \mu - \Sigma B^T (B \Sigma B^T)^{-1} (B\mu - v) \tag{5.6}$$

$$\Sigma' = \Sigma - \Sigma B^T (B \Sigma B^T)^{-1} B \Sigma \tag{5.7}$$

Although there is an analytic form for the updates to the distribution parameters in this case, there appears to be no analytic expression for the expected marginals of factor graphs drawn from the distribution. So, when such quantities are needed, we simply draw many sample graphs, compute their marginals, and average together the results. This seems to be more sensible in the log domain (soft-max) since some marginals may be very close to 0 or 1. Averaging sampled marginals in the log domain is equivalent to taking the geometric average and renormalising.

5.4 Experiments

In our experiments, we compare five different ways in which the teacher can choose example states at which to update the student's distribution over

potentials. Below, $Z\tilde{P}(x)$ is a random variable representing a draw from the student's beliefs about $ZP(x)$, and $Z\hat{P}(x)$ represents an estimator, perhaps obtained by averaging these beliefs.

- **max true entry** - States of the factor graph are chosen in decreasing order of the true joint distribution at each state (the "entry" for that state in the unnormalised joint table). In other words, only the states of highest probability mass are shown to the student.
- **max diff entry** - At each turn, all states are examined and the one at which the student's unnormalised joint estimate (the geometric average of the sampled models) is most different from the teacher's (exact) unnormalised joint is chosen. E.g. $\max_x \left| ZP(x) - Z\hat{P}(x) \right|$ where $Z\hat{P}(x) \equiv \exp \mathbb{E}[\log Z\tilde{P}(x)]$.
- **conditional game** - The student's marginals (their geometric average, calculated by sampling) are used to play as MP in a conditional game against the teacher's CP. Whether CP is trying to maximise or minimise the game outcome is decided uniformly at random before each game. The state chosen by the game is used as the next example.
- **uniform random** - A state is chosen uniformly at random.
- **max var log entry** - The state is chosen at which the student is maximally uncertain about the value of $\log ZP(x)$, as measured by the variance of this value over sampled models. I.e. $\max_x \mathrm{Var}(\log Z\tilde{P}(x))$

Note that the first method (MTE) only uses information from the true distribution (the teacher's distribution), without ever querying the student's progress. The second two methods (MDE and CG) compare the true distribution with estimates from the student's distribution. The fourth method (UR) makes no reference to either, and the fifth (MVLE) only uses information from the student.

We tested the 5 methods (MTE, MDE, CG, UR, MVLE) on four different models, all with 10 variables but differing in the variance β of the log-potentials. We explored values of 0.5, 1, 2, and 3 for β. One useful way to visualise the difference in the models produced by these values of β is by making a Zipf plot of the entries in the joint distribution (this means that the entries are sorted and plotted against their rank on a log-log scale; additionally we normalised the values so that the largest is 1). See figure 5.1. The steeper the slope of the line, the more probability mass is placed

on a few dominant states, rather than being spread out across many states. A line with slope -1 has been included for reference.[3]

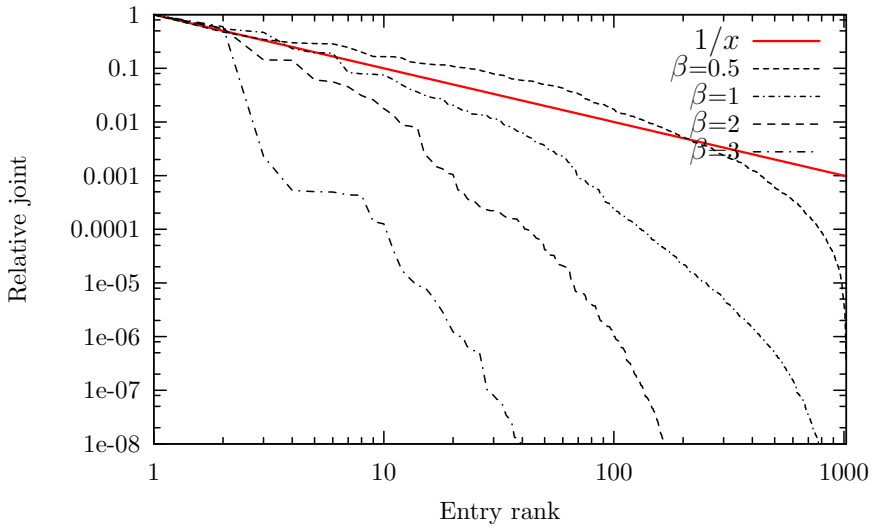


Figure 5.1: Zipf plots for entries in the joint distributions of four typical factor graphs, with reference line

For a fully-connected binary pairwise factor graph of n variables, there are $\frac{n(n+1)}{2}$ parameters in the potentials, thus we expect most methods to require 55 examples to learn our 10-variable example models completely.

The student's distribution over models uses as a prior the same normal parameters from which the true graph was generated. Using other parameters (including putting a normal prior on univariate factors ψ_i, and on a scalar factor for the whole graph) did not produce substantially different results, except for worsening the performance of MTE.

[3] A special case is the slope -1, which corresponds to a relationship $P(x) = \frac{1}{r}$ where r is the rank of the state x. Considering r to be continuous-valued, we note that $\int_0^1 \frac{1}{r^\alpha} \mathrm{d}r$ is ∞ if $\alpha \geq 1$, and $\int_1^\infty \frac{1}{r^\alpha} \mathrm{d}r$ is ∞ if $\alpha \leq 1$. Only when $\alpha = 1$ are both integrals infinite. Thus $\alpha = 1$ can be compared to a model where probability mass is fairly divided between likely and unlikely states. From the plot, we see that the value $\beta = 1$ has a slope which is close to -1 over the first decade, but decreases thereafter.

5.5 Results

The results of running the five methods on each model are shown in figures 5.2 through 5.5. Each figure has three plots. The first plot shows how the L_1 error of the student's marginals decreases as examples are shown to the student; the second plot is the same, but showing $L_1^{\log}$ error. In both cases the student's marginals are calculated by geometrically averaging the marginals of 256 models drawn from the student's distribution, and renormalising. The third plot shows the variance in the student's estimate of the *normalised* joint - a series of draws are made from the student's distribution over models, and the normalised joint distribution is computed from each draw. For each state, the variance of the joint at that state is estimated (unbiased) from the samples, and these quantities are summed over all the states. This "joint variance" is plotted as a function of the number of examples, and is a measure of the student's uncertainty about the true model - in contrast to the other two plots, the joint variance is calculated independently of the teacher. Thus, the joint variance is *not* a measure of the student's performance under a given protocol, but only represents a quantity of interest which can be used to better understand what is going on.

The plots demonstrate a number of consistent relationships between the five methods. First of all we observe that CG seems to have the best overall performance. For small β, CG sometimes performs slightly worse than UR or MVLE, and for large β it sometimes trails MDE, but it is never far off from being the optimal method. By contrast, there are conditions under which each of the other methods performs unacceptably badly. This is perhaps surprising, since most of the other methods (all except UR and perhaps MTE) require examining every entry of the joint and would be prohibitively expensive to implement on large graphs, except in some heuristic form. CG, on the other hand, only requires conditioned marginals, which may be easily obtained from most approximations.

Another interesting observation is that where CG and MDE are in close competition on the error for the high β graphs, the "joint variance" tells a different story. Although the error in students trained by CG and MDE is approximately the same, the joint variance, which is a measure of the student's uncertainty, decreases much more quickly for MDE than CG. This means that even though CG's student is able to perform close to and sometimes better than MDE's in error, this advantage comes in spite of his having a greater uncertainty about his own beliefs. We can infer that the low error of CG's student was enabled by his having taken an average over models

which were sampled from his relatively broader distribution. If, on the other hand, he had only sampled a single model from this distribution, he would have been likely to choose a model with high error.

MVLE is close to UR, although consistently better, and the distance is about the same in L_1 and $L_1^{\log}$ plots. For small β, the CG follows these methods closely, but for large β it begins to do much better.

MTE does surprisingly poorly, and even though the example states it chooses are distinct it appears to be selecting linearly dependent sets of them (perhaps which overlap in many variables) since it doesn't reach zero error after 55 examples. Strangely, for $\beta = 0.5$ and $\beta = 1$, MTE seems to do fairly well at reducing joint variance even while failing to improve the marginals.

Perhaps the most attractive feature of any of the methods is the way MDE manages to achieve near-zero L_1 error in marginals at just over half of the examples required by the other methods for $\beta = 2$ and 3. It is the only method with this desirable feature. It exhibits the same behaviour, but to a much lesser extent, on $\beta = 1$; and not at all on $\beta = 0.5$.

The joint variance plots show some strange behaviour, for instance with $\beta = 0.5$ the joint variance decreases at about the same rate for each of the five methods, even though the marginal error curves of those methods are reasonably diverse. This holds to a lesser extent for $\beta = 1$. Even in the high-β models, the student's own estimate of his uncertainty is a poor indicator of his actual error.

A number of other methods for choosing examples were explored but were found to perform poorly, and are not shown in the plots. The application of the conditional game we used chooses randomly to have CP maximise or minimise the result; if we eliminate the random choice and select each point with CP only maximising or only minimising, then the error does not decrease (in some instances the same state is chosen repeatedly). Choosing an example at random from the exact distribution performs about as well as UR, as does choosing a random state from a model drawn from the student. Choosing a state by selecting the variable and value with the largest $\mathrm{Var}(\sum_{x_{\setminus i}} Z\tilde{P}(x))$, conditioning the variable to that value and recursing, works about as well as MVLE. Having the student's MP use marginals from a sampled $Z\tilde{P}$ rather than an averaged estimator $Z\hat{P}$ gave worse performance, as did using arithmetic rather than geometric averaging of the marginals.

In summary of the above results, we would say that it is surprising that the CG method seems to be the best approach to guided inference, consid-

ering the separate purpose for which it was designed. We were unable to find a method which does uniformly better, even considering those methods which required examining every state in the teacher's and student's approximation. One might be tempted to conclude that the most efficient mode of interaction between a teacher and student, at least in the protocol we have defined, is (as in the real world) that of having an argument.

5.6 Discussion and future work

In the next chapter, we consider the application of simulated evolution to approximate inference. The broader goal of such applications is, as always, to create more efficient approximate inference algorithms. We argue there that suitably flexible approximate inference algorithms should work by simulating a kind of natural selection where candidate approximations are made to cooperate and compete so as to evolve approximations of higher accuracy.

Although we don't make direct use of the results of this chapter in the next one, the present investigations were to some extent motivated by having anticipated such applications of the CG. When one considers a context for interactions in which approximations are made to compete using games, but also to cooperate by sharing information, the question arises of the relationship between these two modes of interaction: Is it (a) possible for an approximation to do well in competitions as a result of having "secret knowledge" which he never has to share with his colleagues? Or (b) does the hidden expertise which allows one approximation to outperform another get revealed as soon as we arrange a competition between them? In case (b), our task of deriving a useful interaction framework would be considerably simplified, by the sufficiency of a single form of interaction to accomplish the goals of both (competitively) *comparing* and (cooperatively) *educating* approximate inference algorithms. In case (b) there would also be an intrinsic motive for each party to participate in such interactions - the winner receives the prestige of winning, while the loser receives useful training. It is of course always possible for an evolutionary framework to provide such motives artificially, by some deliberate innovation of the design, but surely we should prefer such motives to arise as a natural consequence of the interaction itself. We were therefore pleased to discover through the simple experiments presented here that, at least to good approximation, the second case indeed holds - the same competitive interaction (i.e., the CG) accomplishes both comparison and education.

There are some discernible directions in which this research could be

extended. Apparently, for the ideas presented here to become practically useful, at some point it would be necessary to replace the unrealistic exact inference methods used in these experiments with a suitable approximate inference algorithm, one which can be parametrised by states of the model so that it can "learn" from the CG. And as discussed in the previous chapter (in section 4.5), we would like to be able to complete a round of interaction without traversing every variable in the graph. This would be necessary, for instance, if one wanted to conduct a guided-inference-style interaction on only part of a large graph. Finally, the ideal application for guided inference is within some kind of framework for simulated evolution, which we consider in greater detail in the next chapter. Fitting together all of these ideas is likely to be a difficult problem, but the findings of this chapter might encourage us to hope for an elegant solution, one in which multiple design goals are satisfied by the same design element.

5.7 Acknowledgements

The author would like to thank David Duvenaud for useful discussions.

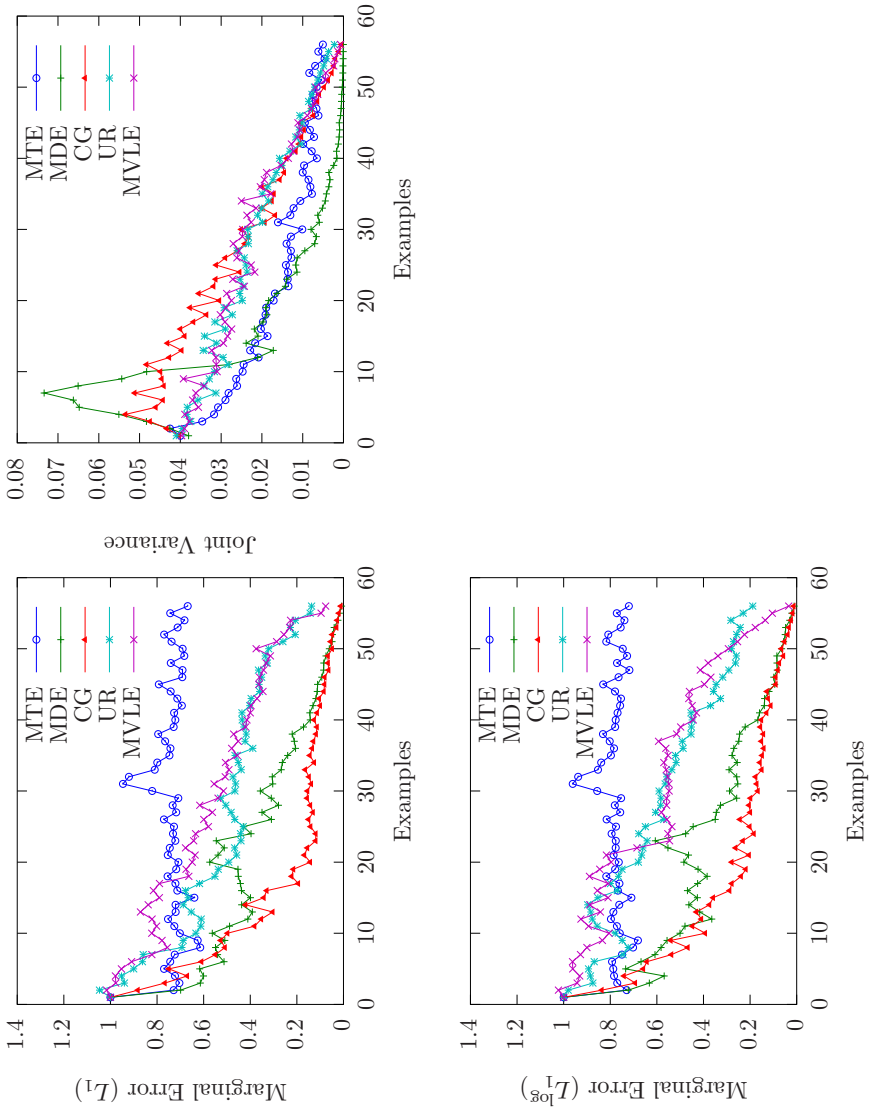


Figure 5.2: Plots showing error and joint variance as a function of example count on a random graph with β=0.5

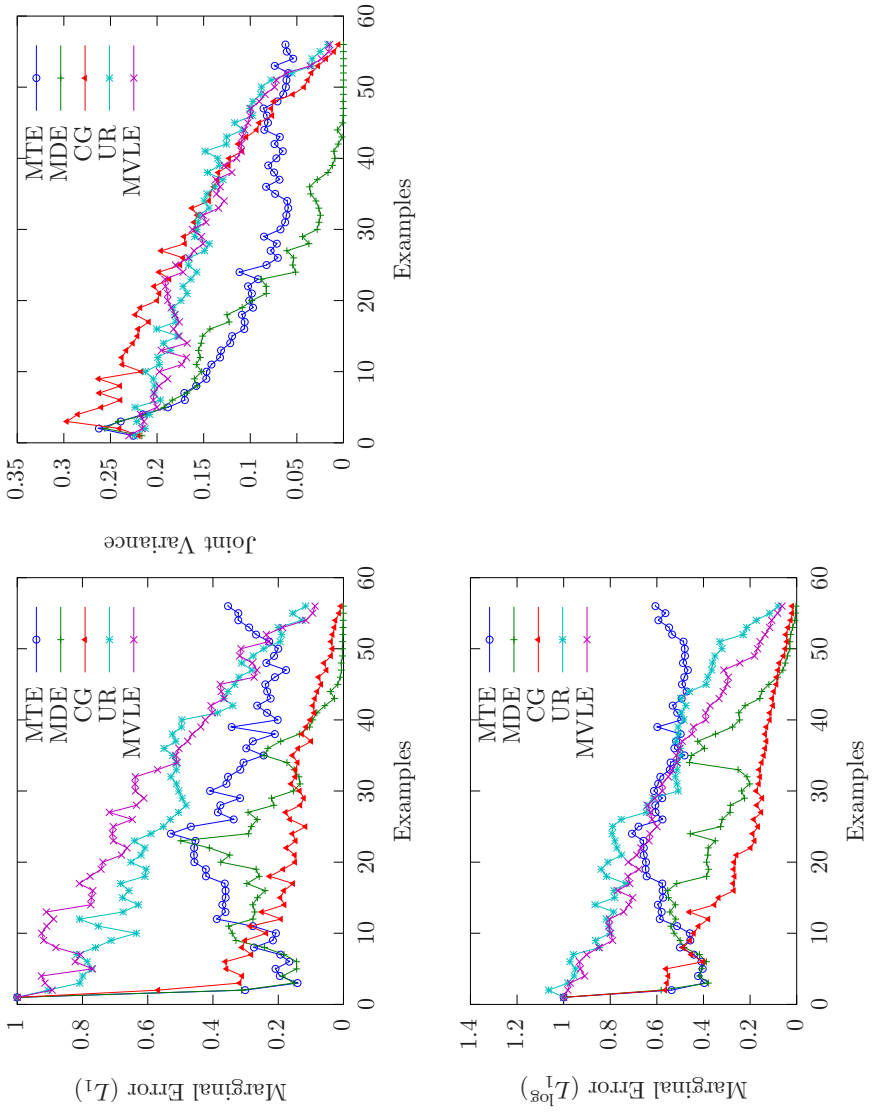


Figure 5.3: Plots showing error and joint variance as a function of example count on a random graph with β=1

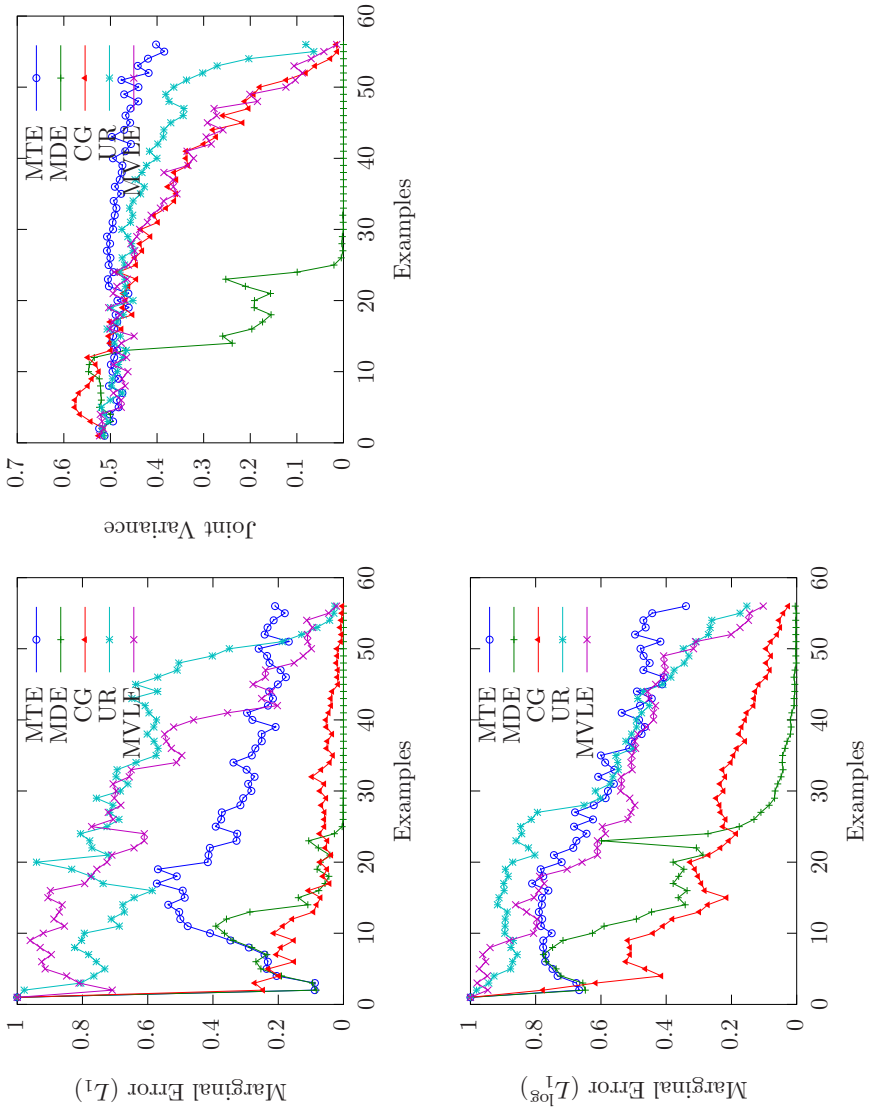


Figure 5.4: Plots showing error and joint variance as a function of example count on a random graph with β=2

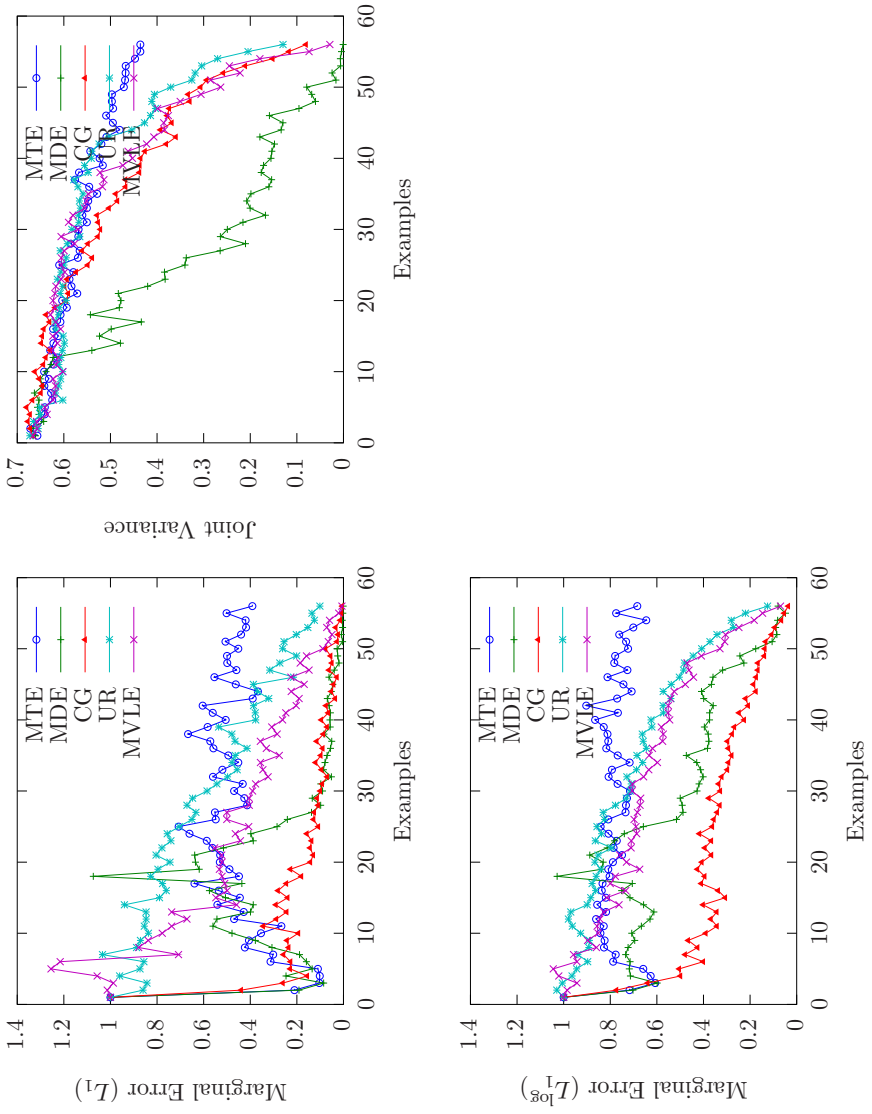


Figure 5.5: Plots showing error and joint variance as a function of example count on a random graph with β=3

Chapter 6

Evolutionary experiments

Abstract

We are interested in algorithms that produce better approximations to a statistical model. Ideally, we should be able to build a system that outputs more accurate approximations over time, much like sampling algorithms do, so that we can obtain within any given time constraint an approximation which is close to the "best possible" for that particular model and that particular constraint. This would be an example of an *anytime algorithm*.

In implementing such a system, it would presumably be useful to have a method for comparing two approximations, like the conditional game. In chapter 4, we showed how to use the conditional game in a single-elimination tournament to select an approximate inference algorithm from among an initial population of candidates. This process could be used to construct an anytime inference algorithm by identifying a sequence of improving approximations from a stream of input candidates. The problem with the tournament method is that it is apparently very wasteful, since it never combines two good approximations but only throws away the losing one. In this chapter we integrate the conditional game with ideas from Genetic Algorithms and biological evolution in an attempt to remedy these shortcomings. The result is a kind of anytime algorithm. Although the performance of our algorithm is not competitive with standard inference algorithms on the models we employ, the outcome of these investigations can be seen as establishing first of all to what extent concepts from Genetic Algorithms can be usefully applied to our problem. Another contribution is a brief exploration of the best strategies for selecting "mates" and producing "offspring" in this evolutionary setting. The discussion section examines at length several potential

improvements to the GA approach which might lead to a more systematic and usable exploitation of these ideas in the future.

6.1 Introduction

We would like to consider the problem of optimising the quality of an approximation. We start with the observation that the nature of interesting yet computationally difficult problems makes them ideal candidates for parallel processing, which means that they can be broken up into smaller pieces to be solved independently without too much intercommunication. Thus, for instance, the definition of the "difficult" complexity class NP invokes the idea of a computer with arbitrary parallelism. Presumably, the subproblems which arise out of an attempt to solve a given problem should inherit something of that problem's nature - if it is difficult, they should be difficult; if it is a problem in approximate inference, perhaps we can also expect them to be problems in approximate inference. The question of how this subdivision of labour should be carried out is closely related to the question of how to combine the solutions resulting from each sub-problem, and hence leads to the subject of this thesis: a general investigation into the ways in which it is possible to combine multiple approximations.

We would like to have an "anytime algorithm", which is an algorithm that outputs gradually-improving approximations the longer it is allowed to run. This definition includes any algorithm that can be tuned to trade time for increased accuracy - every time it finishes, simply rerun it from the beginning with successively more stringent requests for accuracy. The most important part of an anytime algorithm, in other words, is not its "streaming" interface, but its ability to trade time for accuracy. Sampling algorithms like Gibbs sampling, or MCMC methods in general, conform to this definition by outputting a sequence of marginals which converge to ground truth. But there is no reason not to consider a more sophisticated system, which might output a sequence of decompositions or parametrised algorithms, like region configurations for GBP, giving rise to marginals of increasing accuracy.[1] A straightforward approach to the problem of how to effectively combine anytime approximations leads us to consider "evolutionary" simulations.

[1] This accuracy would of course eventually be limited by the class of approximations considered, as it is in our experiments.

Cooperation and competition At this point, we have enumerated three different ways of combining approximations, which fall into two categories:

- **Cooperation**: by partitioning a model among multiple approximations (chapter 3), or by allowing two approximations to share information (chapter 5)
- **Competition**: by comparing the accuracy of two approximations (chapter 4)

In chapter 4, we used the CG in a single-elimination tournament (SET) to choose a winning approximation from some initial population. The SET is a very general approach which can be applied to a stream of input candidates to produce a sequence of approximations whose expected accuracy improves over time, resulting in an anytime algorithm for inference. But it is very slow - it is exponential in the number of rounds played, and the accuracy of the winners of each round improves only gradually with time. Part of this inefficiency is due to the way the tournament combines approximations, by keeping the winner and throwing away the loser. Ideally we would be able to salvage some of the work which was used to produce this losing approximation, by somehow merging its best qualities with those of the winner, forming a kind of cooperative interaction. In the simple SET setup, with competition but no cooperation, we found it was impossible to produce a good approximation efficiently.

On the other hand, if we restricted ourselves to using only cooperative methods of combining approximations, with no competition, we would be stuck with anytime algorithms that output a single sequence of approximations, not being able to compare these approximations with those from possible alternative sequences to notice if an alternative might at some point become superior. An example of a "cooperative" algorithm which produces a sequence of approximations is CBP of chapter 3 - CBP gives a sequence of approximations as we increase the depth of the condition tree, obtained by partitioning a model for cooperative solution between successively smaller sub-models. Other examples are provided by many of the standard approximate inference algorithms - for example MCMC, where a sample can be seen as a crude approximation, and averaging sample statistics as a kind of cooperative combination of these approximations. In several of the experiments in chapter 3 (see figure 3.3, e.g. lower right plot) the accuracy of Gibbs sampling was found to overtake that of CBP after a certain amount of time, even though CBP started out ahead. Selecting the best approximation

from between the two sequences at each instant would necessitate some way of comparing approximations, which we see as a form of competition.

In the preceding two paragraphs, we have argued that neither competition in the absence of cooperation, nor cooperation in the absence of competition, is sufficient to produce an algorithm with the kinds of properties we desire. It seems that we are compelled to investigate algorithms which unify cooperation and competition.

Simulated evolution We would argue that the idea of solving difficult yet parallelisable optimisation problems with methods that combine subproblems using cooperation and competition naturally leads us to consider various analogies to evolution. Because such problems are parallelisable, one may imagine that they are being solved by a "population" of coexisting threads or subproblems. The use of cooperation implies some kind of sharing of information, as in Mendelian inheritance or other forms of communication between individuals or groups. The use of competition implies that such communication is being regulated by a process of selection, so that the propagation or survival of a unit of information depends on some measure of its fitness relative to the rest of the computation. These three elements - a population, the inheritance of some kind of information, and competitive selection - form the ingredients of natural selection and evolution (biological or social) as it is usually understood. As a consequence of these arguments, we will refer here to problem-solving approaches which use both cooperation and competition as *simulated evolution*, whether or not an analogy to the "real world" is made explicit. Researchers have developed a broad range of optimisation strategies based on simulated evolution, both with and without biological motivation, most of which are identified with the field of Genetic Algorithms (GAs). Although these strategies have met with mixed success and are reputed for being inefficient in practice, we hope that our line of reasoning has demonstrated to the reader that the basic idea of simulated evolution arises quite naturally and even necessarily in response to problems such as approximate inference, which are difficult yet parallelisable (NP). The failures of the field of GAs should not, in other words, cause it to be considered as an impractical area of investigation motivated exclusively by hand-wavy or romantic analogies to nature, any more than the early failures of AI should make us consider approximate inference as a toy problem without practical relevance.

On the contrary, we think it is clear that any sufficiently intelligent computer system must have characteristics which enable us to interpret it

in terms of some kind of simulated evolution. Whether it is productive to try to design such a system by taking the existence of such an interpretation as a starting point for the exploration and elaboration of its other characteristics we know not, but in this chapter we adopt the working hypothesis that it is.

Contributions In what follows, we review the basic GA framework and show how it can be applied to approximate inference using the CG. We do not attempt to apply the techniques of chapters 3 or 5 for cooperative interaction, but restrict ourselves to a simple random "crossover" operator following the traditional GA approach.

Next, we present the results of experiments measuring the performance of the system we have developed, and discuss the effects of changing different aspects of the implementation.

Finally, we present our conclusions and propose areas for future work. Although our simulations were not able to achieve good performance, we ascribe this disappointment to shortcomings in the traditional GA approach. We examine several potential remedies. First we discuss how we might be able to harness the types of cooperation explored in earlier chapters (instead of just using "crossover"), and then we explore analogies to the real world which could guide us in developing a better framework.

6.2 Background: Genetic Algorithms

In the design of our evolutionary simulations, we take as a starting point the field of Genetic Algorithms,[2] under which most such research has been conducted.

GAs embody a class of techniques for solving difficult optimisation problems, based on an idealised model of evolution in biological organisms. In the typical GA setup, a fixed-size population of agents competes to optimise a *fitness function*. Each agent is defined by a piece of data called a *genotype*. The fitness function is a real-valued function of an agent's *phenotype*, which is in turn a function of its genotype. The agents try to optimise their average fitness by *reproduction* which involves creating new agents from pairs of parents (combinations of more than two parents could be used, but usually aren't). The genotypes of offspring are determined by applying two operators to the parent genotypes:

[2] J.H. Holland. "Adaptation in natural and artificial systems". In: *Ann Arbor MI: University of Michigan Press* (1975).

- **mutation** - The *mutation* operator applies random mutations to a genotype
- **crossover** - The *crossover* operator combines two genotypes to produce a third, typically by selecting "genes" at random from one or the other input genotypes

In addition to these two operators, an application of genetic algorithms must specify how "mates" are chosen to produce new offspring at each "generation". A typical strategy lets each agent reproduce; in *tournament selection*, for each agent i, for each generation, two potential mates are chosen randomly from the population, and the one with the higher fitness is used to produce an offspring with i. In this way, the probability that an agent will be chosen as mate is $\frac{2k-1}{(n-1)^2}$ where $k \in \{1 \dots n-1\}$ is the rank of the agent within the rest of the population of size n.[3]

A method for removing agents from the population must also be specified, for instance by retaining only the n fittest individuals at each generation.

GAs are commonly criticised for being a slow approach to optimisation, but seem to apply well to very difficult problems such as the travelling salesman problem, for which no easy solution is thought to be possible in general (because it is NP hard). GAs are said to be based on a simplified view of biological evolution, but can also be seen as being derived from nothing more than some common-sense principles based on the properties we would like to have in an algorithm which attempts optimisation over a complex fitness landscape. In particular, we should want such an algorithm to (a) consider a number of possible candidate solutions, to (b) explore the space in the region of each candidate, and to (c) consider combining the best attributes of different solutions to produce new solutions. The GA approach should be regarded as an attempt to guarantee these properties in the simplest possible framework.

6.3 The CG and relative fitness

The chief difficulties we encounter in applying the ideas of GAs to an optimisation process guided by the conditional game stem from the fact that the CG doesn't give a single-argument fitness function by which to evaluate the

[3] A more common strategy in GAs are *fitness proportional selection* which chooses each individual with a probability proportional to his fitness; we do not employ it here.

"absolute" fitness of an approximation (and in fact the original motivation of the game arose from our desire to be able to easily compare approximations in spite of the fact that such a function is intractable). Rather, the game provides an "approximate relative" fitness function which can rank two individuals for accuracy - "approximate", because the ranking only approximately reflects the relationships which we are interested in, for instance between marginal errors.

Adapting the GA methodology to use such a relative fitness[4] rather than absolute fitness is straightforward. For example, in the "tournament selection" strategy described above, the only use made of the absolute fitness values is to compare them against each other, and this comparison can simply be replaced by running the CG. There is no intrinsic need for a consistent ranking to result from the comparison.

The availability of only a relative fitness function is related to what the GA literature calls "population-dependent fitness functions", in which the fitness of each individual is additionally made dependent on the rest of the population.[5] Under certain conditions, including that of the existence of an individual which has maximal fitness in every possible population, one is able to prove convergence results for GA-style evolution with a population-dependent fitness.[6] The fact that exact inference is an optimal strategy for the CG equates to a slightly more general condition, but may still allow us to make similar theoretical guarantees.

We don't attempt such an analysis here, since we are more interested in actual performance. As for the idea of population-dependent fitness, we prefer our model of games played between individuals as more straightforward. The naive approach of computing such a fitness function by comparing each individual against each of the other individuals, for instance by playing the CG against all of them in turn, would seem overly expensive; yet the standard remedy, which uses a random subset of the population as an approximation to some "ideal" population-dependent fitness function,[7] could simply be expressed in terms of the game itself. In other words, a two-player game seems both simpler and more general than a population-dependent fitness

[4]We will omit the word "approximate" and just write "relative fitness" in the future, since non-approximate relative fitness functions induce the same total ordering as some absolute fitness function.

[5]L.M. Schmitt. "Theory of genetic algorithms". In: *Theoretical Computer Science* 259.1-2 (2001), pp. 1–61.

[6]Ibid., Theorem 8.5, p. 49.

[7]C.D. Rosin. "Coevolutionary Search Among Adversaries". PhD thesis. University of California, San Diego, 1997.

function in practice.

At first sight, evolution with relative (or population-dependent) fitness is somewhat closer to the actual process of natural selection, at least as envisioned by Darwin, than is evolution with an absolute fitness function. In natural selection, choice of mate is governed by criteria that are a function of the environment in which the offspring will live, which is in turn largely a function of the rest of the population. Natural selection is often described as a process by which pairs of individuals compete with each other for food sources or reproductive opportunities. Their success, in other words, is determined more as the outcome of a game between two players than as an absolute ranking of individuals. In this view, we could say that evolution proceeds not inexorably "towards" any predetermined goal, as it would if there were an absolute fitness function, but only away from the present state and in a direction determined by the outcome of these competitions. The progress of scientific research has been compared to such a process.[8]

Although a relative fitness function is arguably more natural than an absolute fitness function (according to the biological analogy), it causes some problems in the present context of optimisation via simulated evolution. In situations where there is an absolute fitness function, it is possible to guarantee that the average fitness always increases, since we can be sure to never replace an agent with one that is less fit. With a relative fitness function, this is no longer the case. Since a relative fitness function only gives an approximate ranking of individuals, we may find that their actual (absolute) fitness, as measured by a function which is not in general computationally tractable, fluctuates over time.

For example, it may be that as a population evolves, the average (absolute) fitness improves, but the population loses the ability to compete against a certain type of individual with lower (absolute) fitness; at some point one of these individuals may then appear and outcompete the rest of the population before the ability to counter it is recovered, causing a sudden drop in average fitness. In the GA literature, this is related to the problem of *forgetting* in coevolutionary simulations, in which previously successful individuals are lost because the members of the other population against which they were able to compete suffered extinction. An example of this phenomenon is given by the interactions within the immune system between lymphocytes and parasites: without some special memory mechanism, the immune system might lose the ability to recognise a parasite which it had

[8] T.S. Kuhn. *The structure of scientific revolutions.* University of Chicago press, 1970, Ch 8, p. 171.

managed to eliminate - in the absence of the selection pressure which that parasite had once exerted on the population of antibodies. In the experiments which we present below, we hope to gain a sense of the extent to which our application of the CG to evolutionary simulations is burdened by issues such as these.

6.3.1 Effect of interaction topology

One particular phenomenon to which we devote our attention is the effect of varying the topologies according to which agents are allowed to interact. The reason we used a SET with binary tree topology to evaluate the CG in chapter 4 was that its behaviour is easy to analyse - each game is played between two players who are drawn independently from the same distribution over players. As long as the game's agreement rate (defined on page 85) is above one half, the average error of the players at each level should be an improvement with respect to those of the previous level. Even when the wrong player wins at round n, this only introduces an element drawn from the distribution at round $n-1$, rather than from the initial distribution.

By contrast, if we were to pick the best player using a linear-topology SET, i.e. by playing each contestant in turn against a "reigning champion", then the quality of the resulting "champion" will be reset to a draw from the initial distribution each time the game chooses the wrong winner. Thus, the linear interaction topology seems more prone to a kind of "backsliding" of fitness. The two topologies are contrasted in figure 6.1.

(a) Binary-tree topology (b) Linear topology

Figure 6.1: Diagrams of two tournament topologies for 16 players

In lieu of a more careful analysis we can see from a simple experiment that the binary tree topology is more efficient than the linear topology for the SET on our toy models. The results of the averaging the winner's error for tournaments with 2^n players on binary trees, compared to linear tournaments with the same number of players, for $n = 1$ to 8, is shown in

figure 6.2.

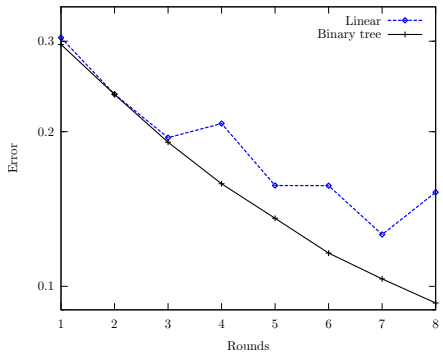


Figure 6.2: Comparison of winners of binary tree tournament and linear tournament. Averaging over 120 random graphs, of 7 nodes with pairwise factors sampled as $\exp(2W)$. Average errors are plotted for the winners of tournaments on 2^n players, with $n = 1$ to 8.

We will say more about this in section 6.6.1, but it is apparently better to conduct optimisation via simulated evolution using topologies that segregate the agents into a number of isolated "ecologies" with only occasional cross-interaction, than to devote the entire population to a single arena where agents are able to compete promiscuously. Note that with an absolute fitness function, the observed difference between the binary-tree topology and the linear topology disappears. With an absolute fitness function, in either topology the contestant with the highest fitness wins. This highlights a difference, which shows up in the absence of mating, between the absolute fitness setting of traditional GAs and our relative fitness setting. In our setting, topology matters even without mating. Topology only has consequences in the traditional setting when mating is taken into account. For example, topology influences the order in which crossover operations take place, which may have various effects on the gene pool. In our setting, interaction topology plays the additional role of preventing cross-contamination between parallel experiments.

6.4 Implementation

We examine three different selection and mating strategies. Each one assumes a population of constant size. Also, note that each one requires four

runs of the CG per reproduction, which simplifies the comparison of different strategies.

- **paired competition** - At each step, two distinct individuals are selected at random and play the CG (computing a 4-way score S_4). A child is produced by cross-over of the two genotypes, and replaces the loser.

- **suitors** - At each step, a "princess" and two "suitors" are chosen at random (without replacement). The princess plays the CG against each suitor as CP, first minimising and then maximising the game value. The difference between the two resulting values is used as a score to rank the suitors, and the single offspring of the princess with the winning suitor (produced by crossover as above) is used to replace the losing suitor.

- **directed crossover** - At each step, a pair of distinct individuals is chosen and the CG is played between them (again, computing S_4). At each turn of each of the four games, the two approximations will tend to report different conditioned marginals for the same variables. These differences are quantified and accumulated. Thus each variable becomes associated with a number measuring the total disagreement, over the course of the four games, between the two players' conditioned marginals for that variable. These numbers are used to select GBP regions containing variables with high disagreement. Some of these regions are then chosen at random and copied from the winner to the loser.[9]

[9]This might be seen as an attempt to implement a sort of transfer of acquired characteristics, see section 6.6.2. The implementation is very ad-hoc, but we specify the details just for completeness. Say the "disagreement" of a variable i in a particular run of the game is given by the ratio of the marginals for the CP and MP at x_i^*; this is what is being minimised or maximised by CP in the choice of i and x_i^* recommended by in equation 4.12. It may be greater or less than 1 as CP is maximising or minimising. The absolute value of the logarithm of this value, summed over all four runs of the game, is called the score of the variable. A variable j is chosen randomly with probability proportional to this score. A region from the winning player which contains variable j is chosen at random and added to a list, in a loop which starts and repeats with probability 0.7 (so the list has a probability 0.3 of being empty, 0.21 of having 1 element, 0.147 of having 2 elements, etc.). The list is appended to the list of regions of the losing player minus those regions containing j, and to the result is appended a random permutation of the regions from the losing player which *do* contain j, followed by a random permutation of all possible regions. A maximal non-singular set of regions is chosen from the final list by adding allowed regions in order of traversal (as in the crossover operation below).

In our simulations, we use a set of triangular regions (regions of size 3) in a GBP approximation to define the genotype of an agent. The permissible region sets were limited to be *non-singular* (NS)[10] and were maximal subject to this constraint[11]. We chose to enforce this constraint of non-singularity because it makes GBP more well-behaved; in particular, it is equivalent to demanding that GBP have only one stationary point when the factors are made uniform. It is straightforward to prove by induction that the number of triangular regions in a maximal NS set in a binary pairwise factor graph of n nodes is $\binom{n-1}{2} = \frac{(n-1)(n-2)}{2}$, out of a total of $\binom{n}{3} = \frac{n(n-1)(n-2)}{6}$ possible regions. Thus for a random initial population of much more than $\frac{n}{3}$ individuals, we expect each triangular region to appear at least once with high probability. On account of this property, it did not seem necessary to use a mutation operator, which otherwise might be needed to encourage more complete exploration of the space of region configurations. In our case, any given configuration could arise via a series of applications of the crossover operator described below. To validate this intuition, we did experiment with a mutation operator for the paired competition strategy and the results were poor (figures 6.5 and 6.9).

The crossover operator (only used in the first two strategies) is defined as follows. Two "parent" region configurations are merged to produce a single "offspring" configuration. The difficult part is making sure that the offspring's configuration is NS. Start by creating a list L containing the union of the two parent region sets, in randomised order. The output C of the crossover operator is constructed by starting with the empty set and adding regions from L one at a time, skipping those additions which would result in a singular C.

We explored two interaction topologies. The "full" topology imposed no constraints and chose mates and suitors randomly from among the population. The "ring" topology considered the members of the population to be arranged in a cycle, and only allowed interactions between adjacent individuals. Interaction on the ring topology took place between individuals at locations 1 and 2 in the first generation, followed by 3 and 4 in the next generation, ..., then $n-2$ and $n-1$, n and 1, 2 and 3, etc. In this way, each individual interacted alternately with each neighbour. For the suitors strategy on the ring topology, the pairs of suitors were chosen in this way but the "princess" was chosen at random from the rest of the population.

[10] Welling, Minka, and Teh, "Structured region graphs: Morphing EP into GBP", op. cit.

[11] A set of triangular regions in a pairwise graph is non-singular if it is empty, or if at least one graph edge is contained in exactly one region and removing this region from the set leads to a non-singular set of regions.

6.5 Experiments

6.5.1 Methods

We tested the above evolutionary simulation on 10 randomly generated binary pairwise graphs of 7 variables with normally-distributed log-potentials (i.e., potentials drawn as $\exp(W)$ where W is a standard normal deviate). For factor graphs of this topology, there are 35 possible regions of size 3, and any maximal non-singular set consists of 15 regions. We ran each of the various strategies starting from random initial populations of size 7, 15, 31, 63, and 127. The marginals, used for playing the CG and for recording $L_1^{\log}$ error, were of the GBP approximation, calculated using the libDAI[12] implementation of HAK[13] with tolerance 10^{-7}.

To assess the progress of a simulation, we use the average error, calculated by averaging (arithmetically) over the population the average $L_1^{\log}$ error in variable marginals (measured against exact marginals) for the estimates produced by each individual's set of regions. This population-average error is then averaged geometrically over the 10 factor graphs. The result is plotted as a function of the iteration number, except in figure 6.6 where the individual graphs are shown separately. The iteration number, labelled as "games" in the plots, is the same as the number of "steps" or "generations" or instances where an offspring is produced, which, in the implementation, technically requires four runs of the CG (to compute the four-way score, or to compare suitors).

The error curves in the first four figures have been smoothed using gnuplot's "csplines" interpolation for readability. All experiments were run to 5000 games.

6.5.2 Results

Here we present the results of our experiments in a series of plots.

In figure 6.3, we compare the performance of the "paired competition" strategy among different population sizes and between the two different topologies. The performance of the binary-tree SET is shown for reference: the vertical position of each SET data point indicates the average error over the players found at a given round, and the horizontal position indicates games played (i.e. $2^{\text{round}-1}$).

[12] Mooij et al., *libDAI 0.2.5: A free/open source C++ library for Discrete Approximate Inference*, op. cit.

[13] Heskes, Albers, and Kappen, "Approximate inference and constrained optimization", op. cit.

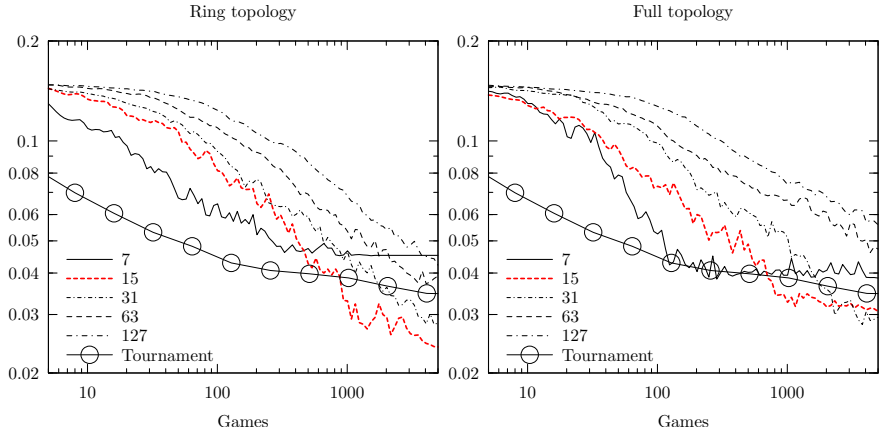


Figure 6.3: Performance of "paired competition" strategy, with different sized populations, compared with single-elimination tournament using four-way score. Left plot is using ring topology, right is full topology. The vertical axis shows average $L_1^{\log}$ error.

We notice that if we shift each evolution curve by $\frac{1}{\text{population}}$, so that the horizontal axis measures games divided by population size, then the curves tend to overlap (the resulting plot, not shown, is not very informative). The general rule seems to be that larger populations are slower to improve, by a factor corresponding to the population size. Thus, if individuals were to reproduce in parallel, this disadvantage would go away. However, the error of large populations decreases further before reaching equilibrium, so they eventually overtake the smaller populations. This is evident where population 7 is overtaken by population 15 in both plots. On the full topology plot, one can also see the point where population 15 is overtaken by population 31, at around 2000-3000 games.

The full topology lines reveal an inferior performance (except for population 7) than those of the ring topology, and they tend to level out sooner. This seems to be a result of diversity decreasing too quickly in the full topology (we give further evidence to support this conclusion starting on page 126).

The tournament line shows a better performance than the evolutionary simulations at first, but is later overtaken by them for population sizes ≥ 15. We find this encouraging for our evolutionary approach, since one of its objectives was to outperform the single-elimination tournament.

Next, we compare the three mating strategies - paired competition, suit-

ors, and directed crossover - using four different population sizes and the ring topology (figure 6.4). There is no clear winner, but "paired competition" seems to converge to a better solution than "suitors" (in population 7 and 15) and decreases more quickly than the other strategies in the two larger populations.

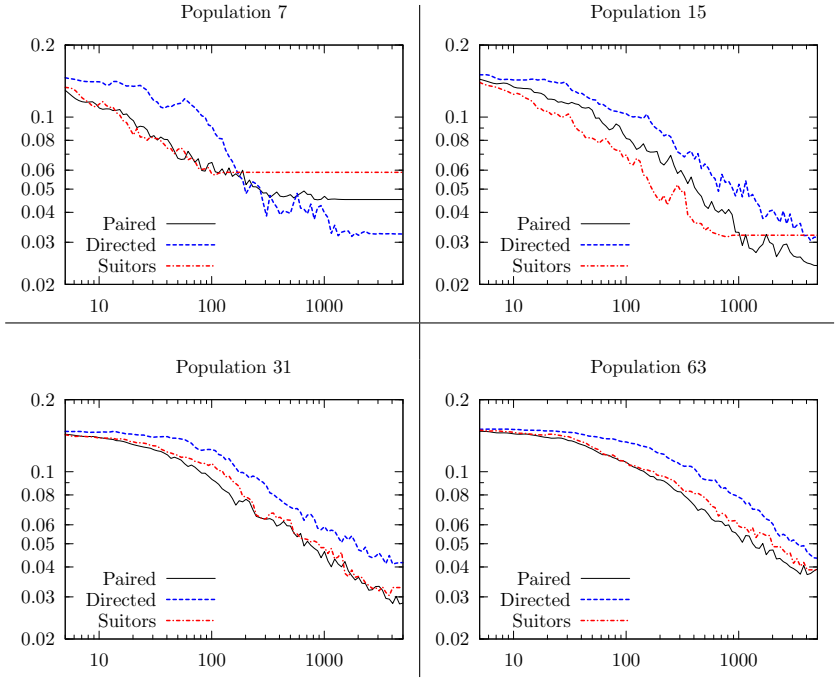


Figure 6.4: Comparison of the three different evolutionary strategies, using the ring topology, for four different population sizes. The plot for a population size of 127 resembles that for 63. The vertical axis shows average $L_1^{\log}$ error.

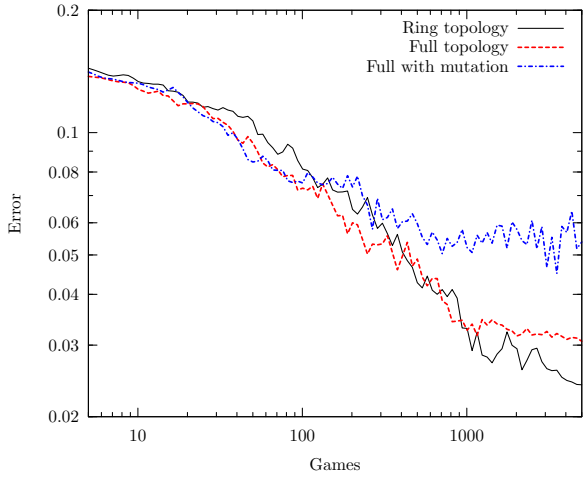


Figure 6.5: Variations on paired competition: full topology, and mutations. Population 15. For the mutation run, at every crossover, with probability $\frac{1}{2}$ a new randomly-chosen region was added to the output configuration (at the expense of an existing region).

In figure 6.5, we show the results of the experiment in which a mutation operator is included. This operator takes the following form: at each reproduction, during crossover, with probability $\frac{1}{2}$, a random region is added to the head of the list L from which the set of offspring regions C is generated (L and C are defined in section 6.4). The specification of this operator is based on what appears to be a standard rule of thumb in GAs, which says that the optimal mutation rate is about one bit per genome per generation.[14] We can see that the performance of the mutation operator simulation matches the performance of the mutation-free simulations at below 100 games, but afterwards it seems to stop decreasing and settles into an area with high error. The mutation operator was only simulated with the full topology and not the ring topology, but as shown on the plot, the results of corresponding simulations without mutation are similar for both full and ring topology. Hence it seems reasonable to infer that the mutation operator would have a similar performance on the ring topology. We did not investigate whether a smaller mutation rate might be more successful.

[14]T. Bäck. "Optimal mutation rates in genetic search". In: *Proceedings of the Fifth International Conference on Genetic Algorithms*. 1993, pp. 2–8.

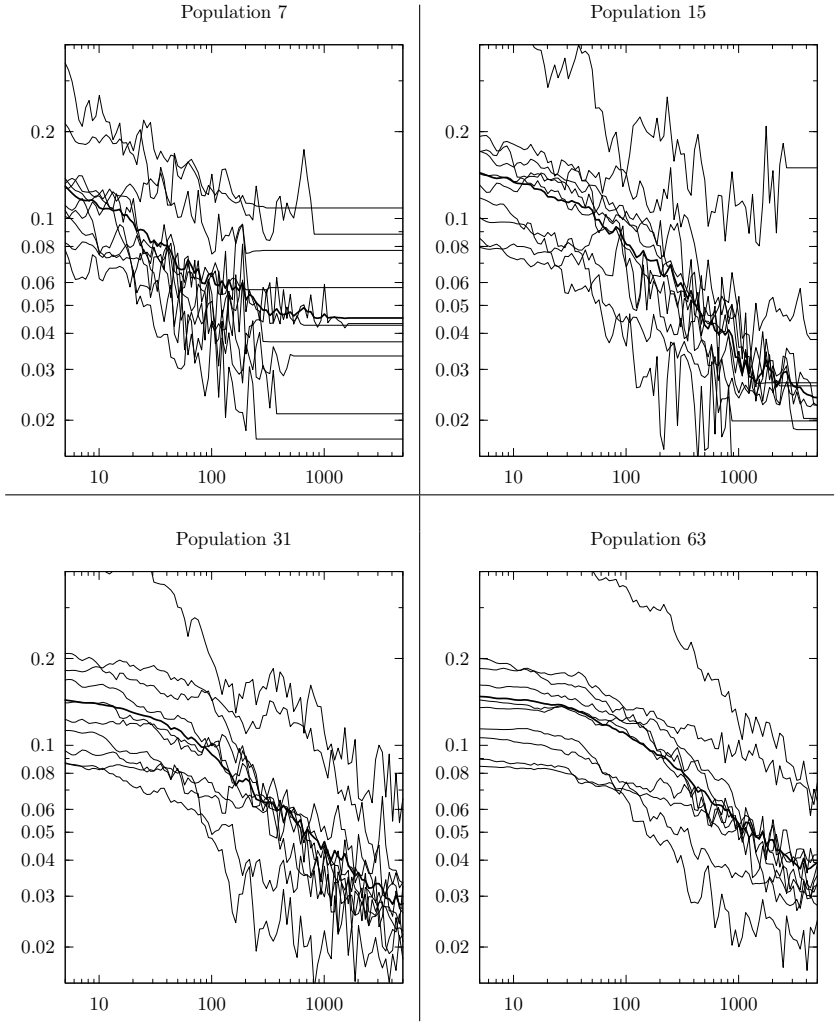


Figure 6.6: Plots of population average $L_1^{\log}$ error curves for 10 individual factor graphs, for 4 different population sizes. All simulations use the ring topology and paired competition strategy. The geometric average of per-model errors is shown in the thick line.

In order to give a sense of the different kinds of behaviour which are found in different simulations, in figure 6.6 we show the individual curves for simulations on each of the 10 factor graphs. The geometric average of the per-model errors is shown in the thick line. Here it is much more apparent that the decrease in average error is not monotonic, as it would be with an

absolute fitness function. One can also see the need to use geometric mean rather than arithmetic mean in the earlier plots, because the errors of the various models deviate from each other by almost an order of magnitude. Had the arithmetic mean been used, the results would be dominated by characteristics of the models with highest error.

One can see that for the smaller population sizes, in many runs the population converges to copies of identical individuals prior to the 5000 game endpoint. This is also shown distinctly in some of the plots below, which include a quantity representing population diversity: refer to figures 6.7, 6.10, 6.11, 6.15, 6.16, and 6.17.

The next 11 plots show more detailed results for individual simulations. The title of each plot lists the selection strategy, topology, and population, followed by the number of the factor graph (1-10) in round brackets. These plots don't use the smoothing which is present in the other plots, so it is possible to see more distinctly what is going on. Each plot is split vertically into three sections. The top section shows curves representing the minimum (thin black line), maximum (blue line), and average (thick black line) error, and also the error of the "mode" or most common genotype (with ties broken arbitrarily; red line). The middle plot, labelled "Accuracy", shows a running average (exponential with decay rate 0.05) of the agreement rate, i.e. the rate at which the game is able to correctly identify the more accurate of two approximations according to the average $L_1^{\log}$ error. The bottom plot shows a measure of the diversity of the individual regions or genes in the population (bottom curve) and of the sets of regions or genotypes (top curve). The genotype diversity is measured as one minus the probability that two random individuals will have the same genotype. The gene diversity is the same quantity, which is to say, one minus the probability that two random regions taken from random members of the population will be the same, but shifted and scaled so that the minimum possible value is zero and the maximum is one (since it is not possible for the population to contain only one gene, or to have no duplicate genes, unlike the situation with whole genotypes).

The gene diversity measure is useful for showing when certain regions become more dominant as the genotypes converge to a single genotype. For instance, in figure 6.7, which plots a directed crossover simulation, we can deduce that at around 3000 games there are only two distinct genotypes in the population, differing by a single region, but it takes 1000 games before one of them becomes extinct, presumably by adopting the correct region from the other. The game accuracy line shows us that the proper individual is always winning during this period. From the minimum error curves, we can see that individuals have existed in the population with small error

but have gone extinct, and the simulation has converged to a sub-optimal solution, even though somewhere between 1000 and 2000 games, according to the mode curve, these were the most common genotypes in the population.

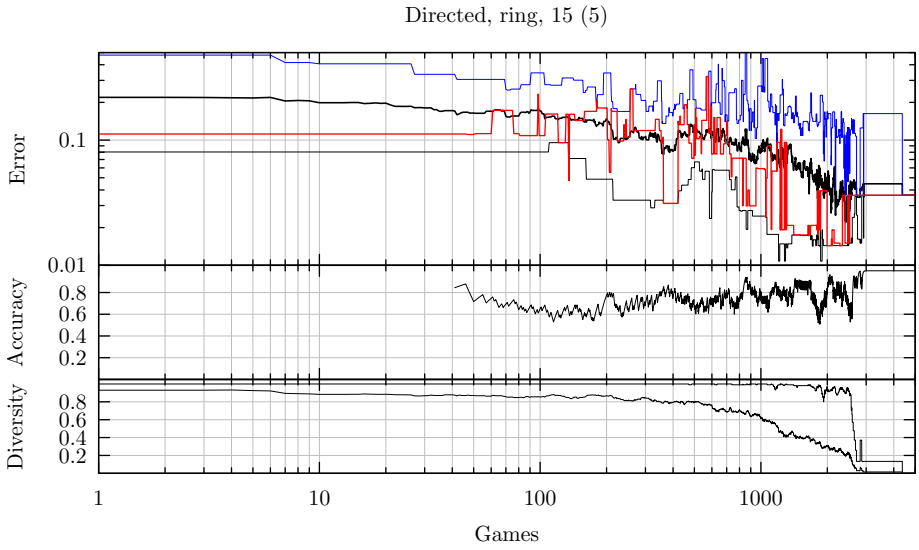


Figure 6.7

The behaviour shown in figure 6.7 is worse than that found in most of the other directed crossover simulations; a more typical example is shown in figure 6.8.

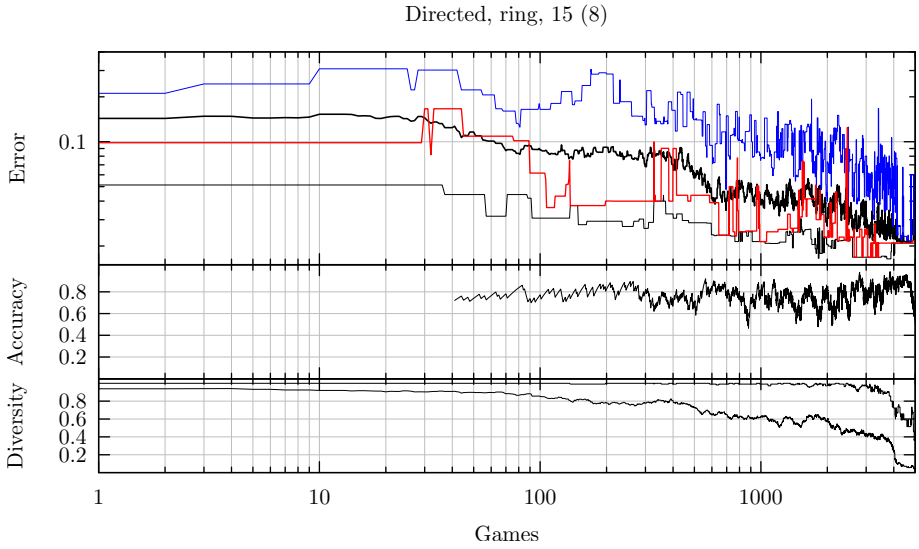


Figure 6.8

In figure 6.9 we can take a closer look at the mutation experiment. There is a big difference between the minimum and maximum error. According to the diversity curve, there is a spot just before 3000 games where the population seems to be converging to one genotype, but then the diversity rebounds.

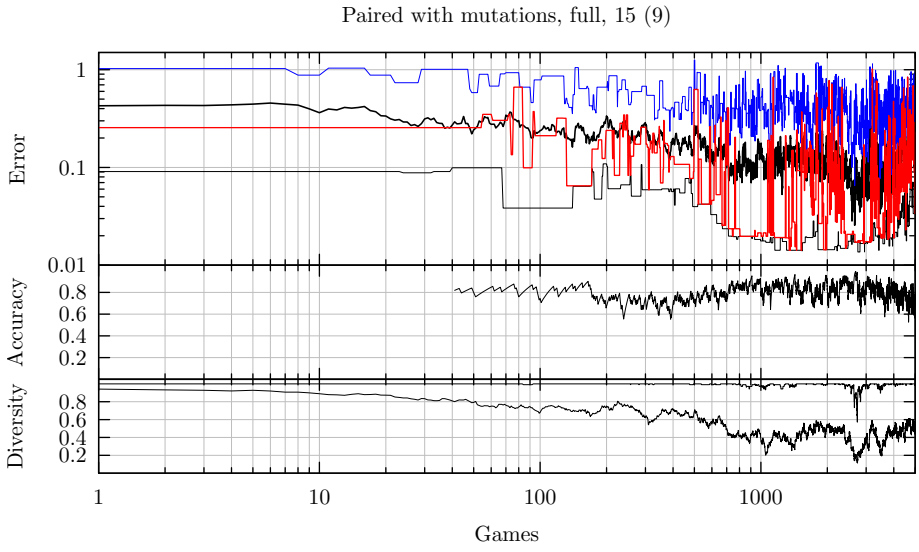


Figure 6.9

Figure 6.10 shows problematic behaviour which seems to afflict the "full" topology. Good results discovered around game 100 are forgotten (minimum error curve) and at the end there is a long stretch (between 1000 and 3000 games) where diversity fluctuates without converging. The behaviour of a simulation on the same model but with ring topology, shown in figure 6.11, is much better. This supports the earlier speculation that sparse topologies reduce "contamination" effects between different groups of individuals (recall the discussion in section 6.3.1, and the experiment shown figure 6.2).

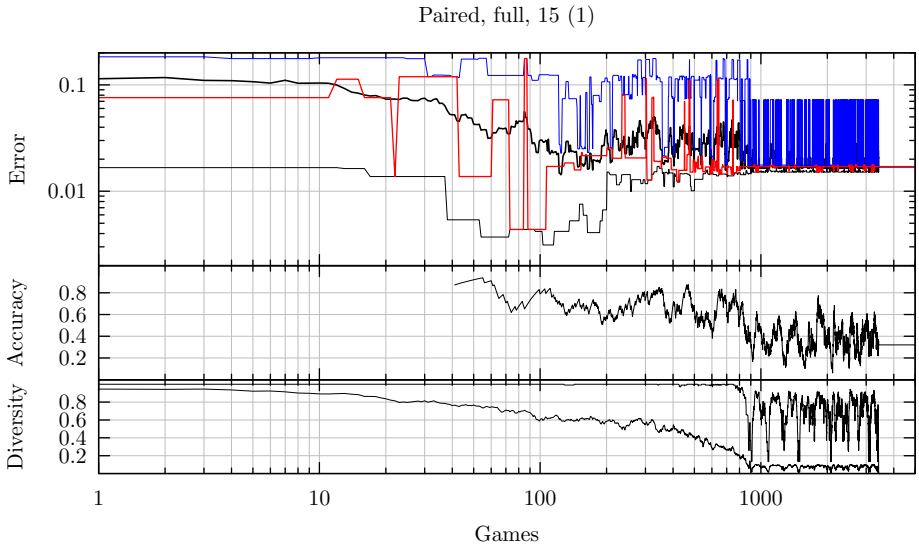


Figure 6.10

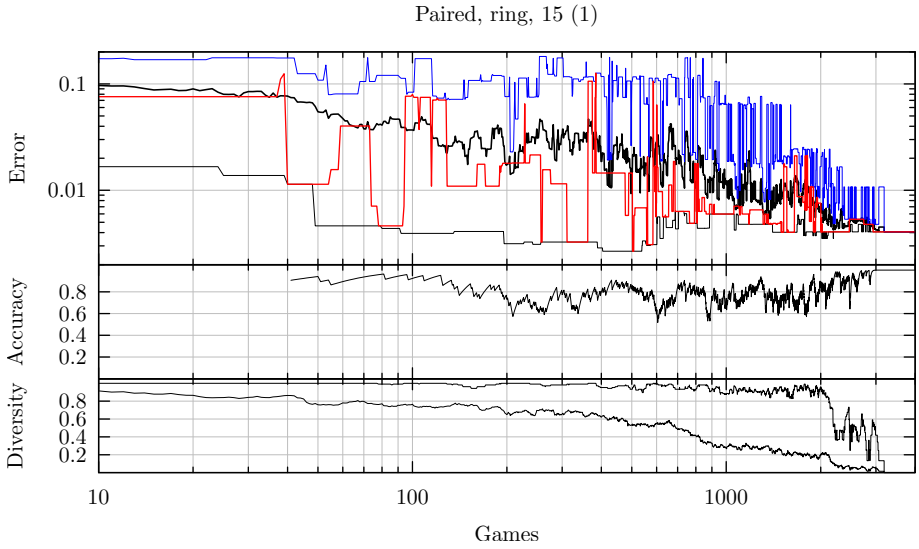


Figure 6.11

Figure 6.12 shows another simulation where, according to the diversity curve, the population seems to be converging to a single genotype with low error, but bounces back to a situation with more diversity. Note that as in figure 6.10, the gene diversity remains low while the genotype diversity fluctuates.

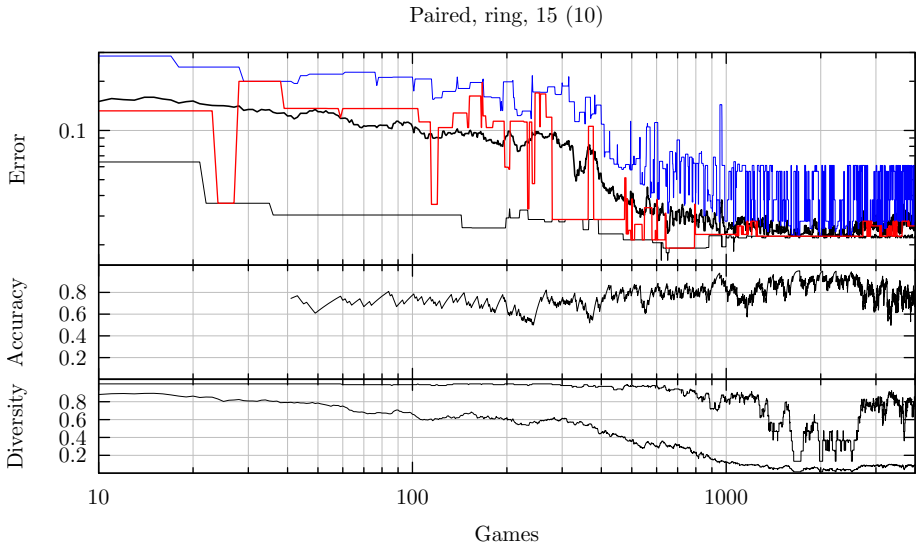


Figure 6.12

Figures 6.13 and 6.14 show examples with higher populations. One can see that there is a good separation between the average, minimum, mode, and maximum error lines. Unfortunately, we did not continue these higher-population simulations to convergence.

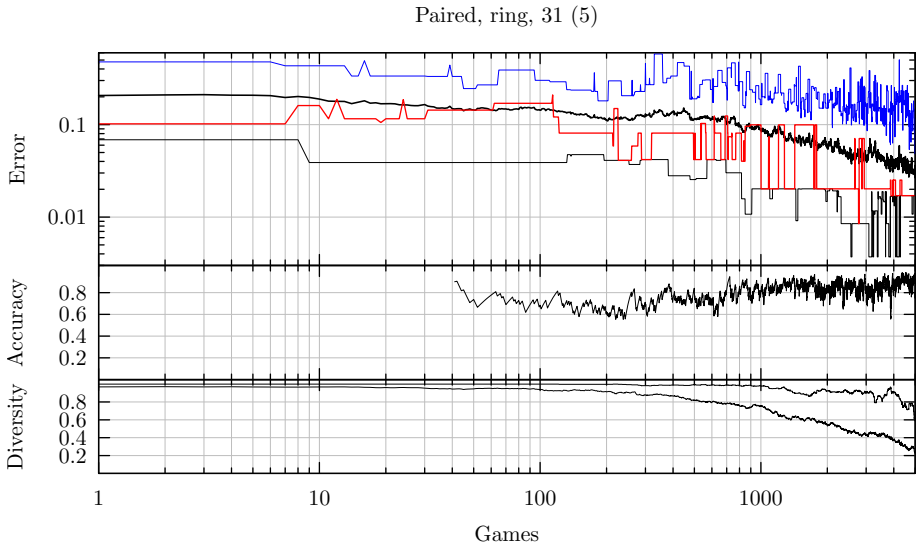


Figure 6.13

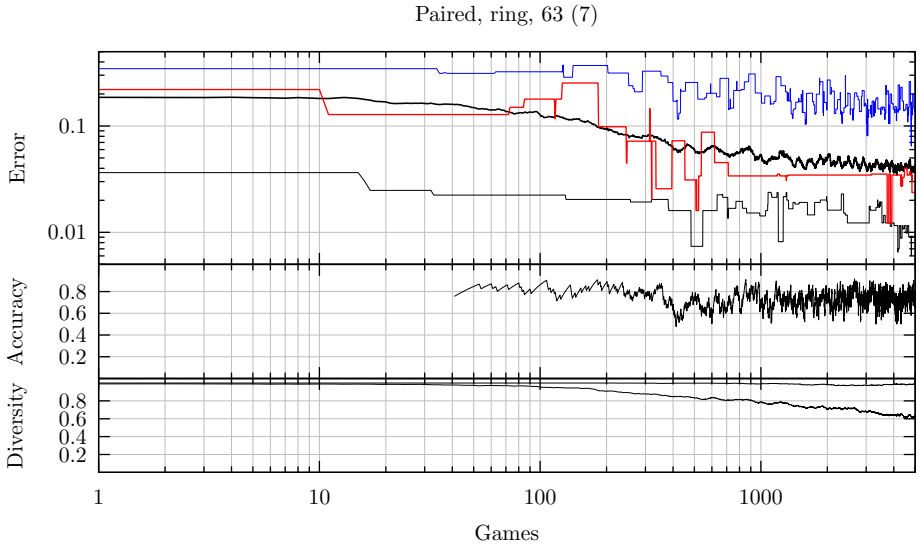


Figure 6.14

Figure 6.15 shows an example with low population. The genotype diversity line shows a fairly rapid decrease prior to convergence.

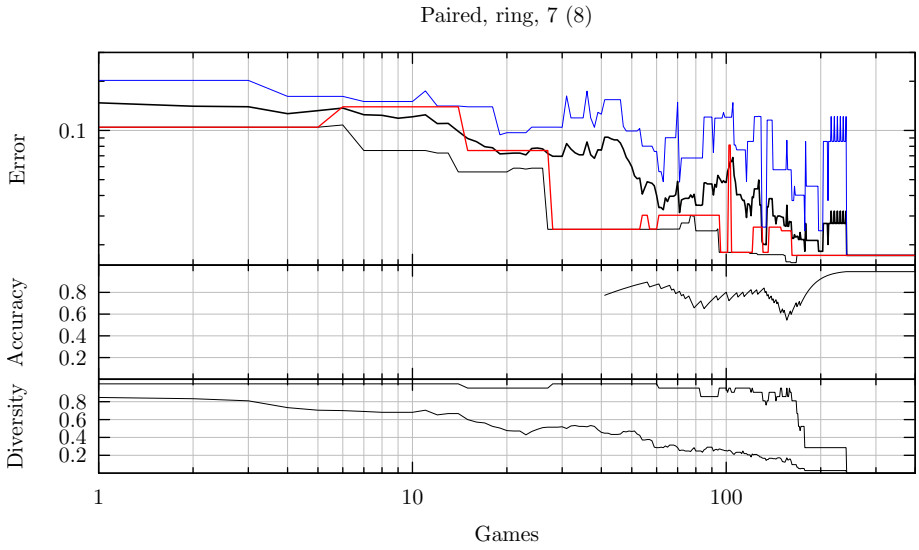


Figure 6.15

Figures 6.16 and 6.17 show two "suitors" simulations. In the first, the minimum error increases over time, and in the second it decreases.

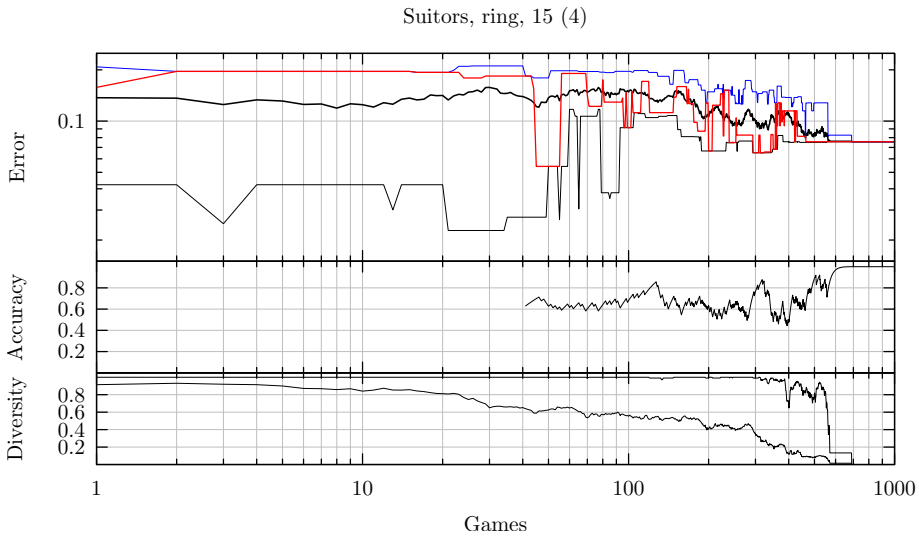


Figure 6.16

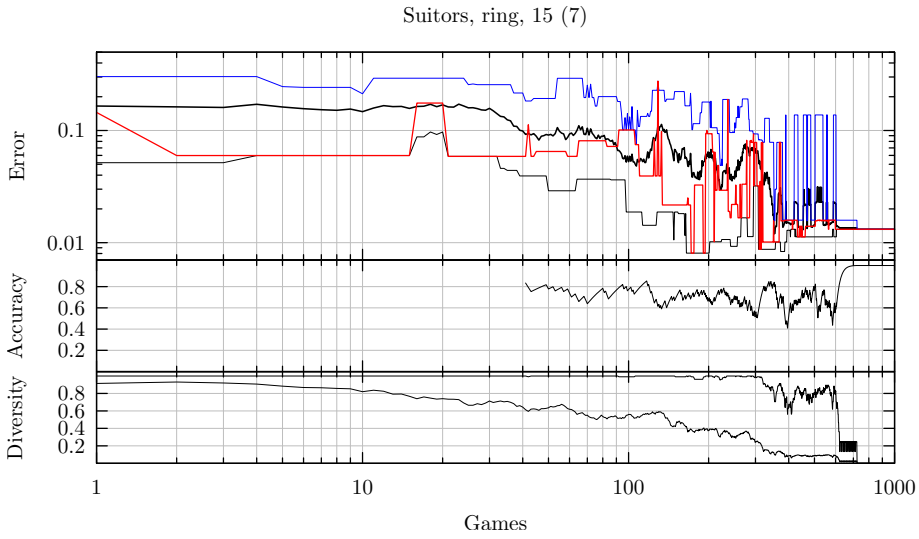


Figure 6.17

6.5.3 Conclusions

We have applied ideas from genetic algorithms to the problem of optimising over the space of approximations to a probabilistic model. We explored three selection strategies, two interaction topologies, and five population sizes, and compared the results of evolutionary simulations with these parameters to the results of the SET described in the previous chapter. Even though they were not expected to be competitive with practical approximate inference algorithms[15], the evolutionary simulations still manifested a decrease in error over time. In addition, for every population size except 7, the evolutionary simulations were able to outperform the tournament (figure 6.3). This is a worthwhile finding, since one goal of our evolutionary simulations was to overcome the basic limitation of the tournament approach (which never combines two good results). And it is also a robust finding: since the tournament has no parameters or alternate implementations, we can be sure that it is impossible to modify it to do better. On the other hand, the tournament does better than evolution for small numbers of games, which suggests that it should be possible to improve the evolutionary simulations so as to create an algorithm which delivers the best of both worlds.

It is somewhat disappointing that the error curves for most simulations only decrease by about a factor of 2 per decade in the steepest sections (e.g. from 100 to 1000). For sampling methods, in which error improves as $\frac{1}{\sqrt{n}}$, the corresponding rate is just a factor of $\sqrt{10} \approx 3$ per decade. Our rate of improvement of error is thus closer to $\frac{1}{\sqrt[3]{n}}$ than to the $\frac{1}{\sqrt{n}}$ of sampling. We are not aware if the $\frac{1}{\sqrt[3]{n}}$ law applies to other classes of approximation than GBP. We note that the accuracy of our simulation is limited by the accuracy of GBP with triangular regions, so unlike with sampling methods the error does not converge to zero. If our framework were modified to include approximations with unbounded region size then obviously this drawback could be overcome (since including a region with every variable results in an "exact" GBP) but the competition would then involve approximations of different time complexity, requiring us to invent a way to appropriately penalise slower approximations.

We found that errors achieved by large populations decrease more slowly than those for small populations, but descend further; and that sparse interaction topologies are more effective than dense ones. This suggests that

[15] It takes about 20 seconds to run the CG with HAK on a seven-node factor graph, on a 2.4 GHz CPU, and we did this 5,000 times per simulation; but exact inference can be performed in milliseconds.

preserving diversity is an important element in achieving good results. On the other hand, using mutation was not found to be an effective way to do this.

There were a number of observed behaviours which we did not fully understand. It would be possible to devote time to analysing the simulations more carefully, to try to achieve a practical grasp of the reasons why, for example, the minimum error rate might not have decreased in a particular simulation. It is not obvious whether it would be more productive to do this, or to undertake new simulations of different topologies or selection strategies, or to try to obtain a better theoretical description of our algorithm. So far, the experiments we have performed have succeeded in demonstrating that the naive application of GA techniques to approximate inference using the CG is feasible and that the results are for the most part well-behaved, even if too time-consuming to be of practical use. In the next section we suggest that in order to achieve significantly better results, more fundamental changes to the GA framework will be necessary. For this reason, we have limited the scope of our experimental analysis to include an account only of the more high-level phenomena, which one might hope to remain relevant even across radical changes to the algorithm.

6.6 Discussion and future work

Our evolutionary simulation experiments were successful in achieving one of their basic goals - to demonstrate the possibility of using such simulations to create an anytime approximate inference algorithm. They also established the superiority of evolutionary methods to tournaments, which never share information cooperatively between two different approximations via operations such as crossover. They succeeded in demonstrating a number of qualitative relationships between different parameter settings, selection strategies, and interaction topologies, which we might hope to be able to inform future research in this area. Although viewed as an inference algorithm these simulations are not at all practical (and in fact showed an inferior convergence rate to sampling methods, at about $\frac{1}{\sqrt[3]{n}}$), yet we might hope that such a simple and compelling idea could somehow be salvaged and improved upon, or perhaps incorporated into a more sophisticated algorithm which overcomes these limitations.

In such a future algorithm we might try to find a way of implementing some of the improvements proposed at the end of the chapter 4. Certainly if we could understand how to avoid playing the conditional game to comple-

tion, then a good deal of computational expense could be saved, especially on large graphs, and in particular when playing against each other two approximations which only differ in a certain portion of the graphical model. With such a modification, it might also be possible to localise agents to different parts of a graph, so that evolution can proceed in parallel when some groups of variables are only weakly coupled. How one should modify the CG to support such partial games is not clear. Given access to an estimate of the partition function of a conditioned model, one can easily derive an estimate of the optimal value of a game which has been played part way, for instance only the first k turns:

$$V \approx \log \frac{\prod_{i=1}^{k} q_i(x_i^*)}{\tilde{Z}_{|x_{1:k}^*}}$$

where $\tilde{Z}_{|x_{1:k}^*}$ is a conditioned partition function estimate. But one also needs to specify who should provide this estimate - whether it is one or the other of the players or a third party - as well as who can decide when to stop playing and under what circumstances they should do so. Furthermore, the benefit of such an optimisation would be limited if each player still privately maintains a full approximation of the entire graph. One would rather want to have "partial players" - each specialising in only part of the graph - playing partial games, which would imply some kind of intimate relationship between the specifications of players, the approximations they entail, the games they play, and also the ways in which these games affect their evolution. These requirements are somewhat difficult and leave a great deal unspecified, but they summarise our vision of how a practical approximate inference algorithm could be built using the simulated evolution concept. Since a partial game is defined by a real number and a "partial assignment" (PA), discussed on page 68 in the context of conditioned belief propagation, one can see how these ideas are related to those described there involving cooperative evolution and PAs.

Leaving aside the question of the finer details of players and interactions, we can identify some broader concerns which were brought into focus by the experiments and seem to apply to evolutionary systems in general, especially those with relative fitness functions like the CG. Before trying to build a more streamlined evolutionary framework, with specialised games and players interwoven as suggested above, one might want to investigate these more general issues carefully and try to understand their implications and possible solutions. Otherwise, one might for instance design a good competition which produces bad results because it is being played between

the wrong players on the wrong field. We conduct such an investigation in the next few subsections.

6.6.1 Regulating interaction topology

We argued in section 6.3.1 that it is important to keep an appropriate amount of segregation between agents in an evolutionary simulation. We supported this argument by comparing the error of approximations selected using SETs on two different topologies: the binary tree topology introduced in chapter 4, and a linear topology. We concluded that the binary tree topology, representing an extreme of segregation, was superior. In the evolutionary simulations, we also observed that the sparser, "ring", topology was superior to the fully connected topology (at least on the statistical models we used). From these observations it is clear that some degree of segregation, i.e. sparsity in interactions, significantly improves performance.

We can think of two forces which might give sparsity a beneficial effect on evolution. First, sparsity promotes diversity, allowing a simulation to concurrently explore multiple areas in the space of approximations. Second, sparsity limits the level of "contaminating" exchange between individuals at different stages of evolution, which might result in a regression of fitness in situations where the wrong player wins a game. Because of the behaviour of the CG, for example as characterised by Theorem 12, the potential for contamination is low when one group has much smaller error than another: the first group will win almost all of the games. One imagines that excess interaction would be more of a problem early in development, where a change leading to a small improvement might be lost before it can evolve into a large improvement.

While we must recognise the value of sparsity, we also note that simulated evolution eventually outperformed the single-elimination tournament. We conclude that the extreme of segregation, as represented by the binary tree SET, was not optimal. Better performance was achieved with some level of mixing, which was in fact the whole purpose of the evolutionary framework. Thus, there is a happy medium to be sought between segregation and integration. How should we locate this optimal degree of interaction sparsity? For an answer to this question, in the tradition of Genetic Algorithms, we might consider looking to illustrations from biology and natural selection.

6.6.1.1 Biology and Genetic Algorithms

The situation where different groups of individuals exist in relative isolation to each other and experience only infrequent interactions is familiar from nature. Isolated ecosystems such as found on islands, and various other ecological niches, are often home to rare and highly specialised animals. Different continents are home to different classes of animals. For example, elephants and giraffes are not found in the wild in South America, even though habitat exists which could support them there. Occasionally a non-native species introduced by humans into a new area does very well, which shows that geography can act as a real barrier to exchange between different ecologies. Such barriers must exert a non-negligible influence on the progress of evolution, since otherwise we would find similar creatures everywhere.

Speciation seems to rely either on isolation or on the existence of different ecological niches to fill. Birds, for example, despite not being significantly isolated from each other, in fact display great diversity. We can, however, account for this by observing that they occupy different niches, having different food sources, predators, and patterns of migration. When a niche provides only a single source of food, and is not partitioned into isolated regions, then a single species will tend to fill it. A good example is Antarctic krill, which is estimated to comprise the greatest biomass of any animal species on the planet. These shrimp-like animals feed only on phytoplankton, and are kept in contact with each other by circulation of waters around Antarctica. Although krill display some very impressive adaptations, it is plausible that their monolithic niche may not be conducive to the evolution of more advanced behaviours. In the same vein, it is easy to imagine that much of the evolution of species such as primates would not have been possible if the Earth had only a single continent, undivided by rivers or mountain ranges.

The problem of regulating interaction topology has long been recognised in the field of Genetic Algorithms as having fundamental importance. However, often the intermediate goals of promoting diversity and preventing fitness regression are addressed more directly. We review some standard approaches to all of these problems, and offer comments and criticisms for each one.

Promoting diversity is most often achieved with *fitness sharing*[16] in

[16]D.E. Goldberg and J. Richardson. "Genetic algorithms with sharing for multimodal function optimization". In: *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*. L. Erlbaum Associates Inc. 1987, pp. 41–49.

which the fitness of each individual is divided by a number quantifying its similarity to other individuals in a population; or *crowding*[17] in which new individuals replace genetically similar ones. A third method is *assortative mating*,[18] in which mate selection prefers individuals with similar genotypes, so that diverse subpopulations can coexist. These methods are simple and no doubt useful, but create problems of their own. They all regulate diversity by direct comparison of genotypes, which seems unnatural - lacking biological analogy, and raising the question of how to measure the significance of a particular set of differences between two genomes. On the other hand, humans and other animals do use phenotypic markers in selecting mates, and it may be useful even if unpleasant to approximate such behaviour artificially with genotype comparisons until we can understand why and how it arises through natural selection.

Regression of fitness is encountered in coevolutionary models. A canonical coevolutionary model is based on a simple view of the immune system, in which antibodies compete to recognise microparasites, which compete to evade antibodies. Such models have for instance been applied to machine learning, where antibodies represent classifiers and parasites represent examples to be classified. A typical method for avoiding fitness regression in such models is the *hall of fame*, in which the fittest individual from each generation is preserved indefinitely, while still being allowed to produce offspring in future generations.[19]

The hall of fame method seems promising but leaves us wondering: how to control the accumulation of immortalised individuals, whether being the fittest individual in a given generation is a good criterion for entry, and how to avoid preserving many similar individuals. It is also difficult to apply to our case in which it is expensive to determine the "fittest" individual, and not even obvious how to best do so. We experimented with (but did not present the results of) similar methods which keep track of the number of games that an agent has won, and use this record to decide who the agent is allowed to mate with or how often it mates. We were not able to achieve much more compelling results with such methods than the ones we have presented earlier.

We also mention a specialisation of fitness sharing to the coevolutionary framework, called *competitive fitness sharing* - essentially, each parasite is

[17] K.A. De Jong. *Analysis of the Behavior of a Class of Genetic Adaptive Systems*. 1975.

[18] L.B. Booker. "Improving the performance of genetic algorithms in classifier systems". In: *Proceedings of the 1st International Conference on Genetic Algorithms*. 1985, pp. 80–92.

[19] Rosin, "Coevolutionary Search Among Adversaries", op. cit.

treated as an independent resource to be shared by antibodies that defeat it.[20]

The more general problem of interaction topology is addressed by the *island model*, in which subpopulations exist on islands and interact only occasionally.[21] Studies have explored ways of automatically adapting the interaction rates in this model. Lardeux[22] proposes a technique for the automatic adjustment of probabilities for moving between islands, in which a transition probability is incremented by a fixed amount when an individual is found to have an above-average fitness, according to which islands his parents came from. This is a promising and creative approach. Again, we do not have an efficient way to detect "above-average" fitness, but could consider doing this incrementing for the winner whenever a game is played. It is difficult to think of an analogy for Lardeux's process in biology, but nonetheless it may be worth experimenting with it in simulation. It is not a complete solution to our regulatory problem, however, in particular since it introduces new fixed parameters such as the size and connectivity of islands, and the amount by which migration probabilities should be modified at each generation.

There are no doubt some stimulating ideas in the GA literature. It is sometimes difficult, however, to see which aspects of the methods are arbitrary and ad-hoc, and which follow from necessity. There may be a tendency when building upon such immature ideas to adopt a particular technique not because it is powerful and general but because it has been tried before and this provides a certain amount of safety to the researcher. The work in this chapter was produced largely in ignorance of the GA literature, with only Mitchell's book as background.[23] On the other hand, such preexisting research is useful in that it provides some common vocabulary and basic tools.

All of the methods we have described introduce new parameters into the simulation which must themselves be optimised. There is no doubt that incorporating some of these methods would yield performance improvements. Yet not even combining all the methods together could be expected to bridge the gap between what we have done so far, and what would constitute a

[20] Ibid.

[21] P.B. Grosso. "Computer simulations of genetic adaptation: Parallel subcomponent interaction in a multilocus model." In: *Dissertation Abstracts International Part B: Science and Engineering* 46.7 (1986).

[22] F. Lardeux and A. Goëffon. "A Dynamic Island-Based Genetic Algorithms Framework". In: *Simulated Evolution and Learning* (2010), pp. 156–165.

[23] M. Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

practical inference algorithm. On the other hand, they would complicate analysis of the simulation results by introducing, through these new parameters, extra and poorly-justified sources of variation.

What we would like to have is a way to simplify the simulation, by automatically determining certain parameters or by adjusting them as we proceed. We are particularly interested in automatic regulation of population size and interaction topology.

Some techniques for doing this regulation are proposed in the field of GAs. The dynamic islands method of Lardeux, which we discussed above, does regulate interaction topology, although in a way which depends on extra parameters and which is unsuitable for our purposes. A paper by Harik and Lobo[24] proposes autodetection of good population size by simulating infinitely many power-of-two sizes in parallel, devoting four times as much time to population sizes which are half as large (so the infinite series converges) and terminating smaller simulations when a larger population has a better average fitness. This is again an interesting idea (similar to Hutter's "fastest and shortest algorithm"[25]) but introduces the problem of how to determine average fitness in our setting, and doesn't tackle the problem of regulating interaction topology.

6.6.1.2 Disease model

We can look to examples from nature to see how interaction topology (a function, for example, of population density) and population size might be automatically regulated to optimise the evolution of biological organisms. Although some of the most important events in evolutionary history, for instance the diversification of placental and marsupial mammals after the Cretaceous-Tertiary extinction event, may have hinged upon large cataclysms such as asteroid impacts; and although the most apparent features of interaction topology are accidental (the shape of continents, the geography of mountain ranges and rivers, etc.) it is doubtful that the kind of regulation we seek could depend on such accidents on a year-to-year basis.

The populations of some species may be limited by the availability of food, or by the overabundance of predators. But the most universal and material limitation on population size and density appears to be the price

[24]G.R. Harik and F.G. Lobo. "A parameter-less genetic algorithm". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. Vol. 1. 1999, pp. 258–265.

[25]M. Hutter. "The fastest and shortest algorithm for all well-defined problems". In: *International Journal of Foundations of Computer Science* 13.3 (2002), pp. 431–443.

of diseases.[26] There are many locations, particularly in temperate climates, where large areas of fertile land exist that cannot be used by humans because of the overabundance of local diseases, such as malaria.[27] These diseases evolve at a certain rate which is determined by the number and density of available hosts. Here is the essential principle:

> The more hosts, the faster the diseases evolve, the more energy is spent by the immune system of the host in keeping up, and the shorter the life expectancy of the host.

The implications of this process stand in opposition to a common belief, which is that to "get" a disease means only to be infected by a member of its species. In fact, we seem to be exposed to many potentially dangerous species of microparasites on an everyday basis, without getting ill. The important factor in illness seems not so much the species of microparasite to which we are exposed, but how far its strain has been able to evolve since our immune system last adapted to it. The most common example is influenza, which is constantly in global circulation yet only occasionally evolves quickly enough to kill large numbers of people.[28] A less well-known example is bubonic plague, whose causative agent, the bacillus Yersinia pestis, is endemic in rodent populations in much of the Americas, in Southeast Asia, and in Africa, but only very rarely makes people ill.[29] The low plague mortality in most decades is more readily explained not by the hypothesis that people never come into contact with rodents, but rather that they are immune to strains in circulation. For another example, syphilis caused widespread suffering (even in monasteries), with horrific flesh-eating symptoms, when first introduced to Europe in 1494.[30] Many historical epidemiologists were sur-

[26] W. McNeil. *Plagues and people*. Jeffrey Norton, 1975.

[27] McNeil, *Plagues and people*, op. cit.; M.J. Echenberg. *Black death, white medicine: bubonic plague and the politics of public health in Colonial Senegal, 1914-1945*. James Currey, 2002.

[28] D.J. Smith et al. "Mapping the antigenic and genetic evolution of influenza virus". In: *Science* 305.5682 (2004), p. 371; C. Langford. "The age pattern of mortality in the 1918-19 influenza pandemic: an attempted explanation based on data for England and Wales." In: *Medical history* 46.1 (2002), p. 1; E.D. Kilbourne. "Influenza pandemics of the 20th century". In: *Emerging infectious diseases* 12.1 (2006), p. 9; K.D. Patterson. *Pandemic influenza, 1700-1900: A study in historical epidemiology*. Rowman & Littlefield Totowa, NJ, USA: 1986.

[29] McNeil, *Plagues and people*, op. cit.; EG Pryor. "The great plague of Hong Kong". In: *J Hong Kong Branch R Asiat Soc* 15 (1975), pp. 61–70.

[30] A. Cunningham and O.P. Grell. *The Four Horsemen of the Apocalypse: religion, war, famine, and death in Reformation Europe*. Cambridge University Press, 2000; McNeil, *Plagues and people*, op. cit.

prised when the bacteria causing syphilis was discovered to be of the same species as that causing yaws, a mild skin disease, endemic to Europe, which could be easily transmitted by handshake.[31] Only the strain - presumably having evolved separately in the New World - was different. The interesting fact that yaws became much less prevalent after the appearance of syphilis can be explained by assuming that the immune system reacts similarly to both diseases.[32] Conversely, recent evidence on the New World side of the "Columbian Exchange" suggests that microparasites such as tuberculosis were already present in the Americas prior to the arrival of relatively more devastating strains from Europe.[33] These examples collectively demonstrate that the concurrent coevolution of hosts and ages-old (rather than newly emergent) disease-causing microparasites plays an important role in pandemics and disease-related mortality, and is closely associated with population density and migration, and other forms of contact between distinct populations. Previous research has also recognised the importance of diseases in guiding natural selection[34] and of promoting diversity (at least in plants[35]).

Such observations bring to mind the immune system model of GAs, which we mentioned earlier. In the immune system model, lymphocytes (or hosts) and parasites coevolve to recognise and evade each other, respectively.[36]

To our knowledge, the present problems of regulating interaction topology and population size are not seen in the GA literature as being any more tractable in the immune system or other coevolutionary frameworks as compared to the standard single-population framework. One relevant concept from GAs is that of "balance", which proposes that the rates of evolution of parasites and lymphocytes should be roughly matched.[37] Heuristics have been proposed to ensure balance, which include adaptively allocating more

[31] McNeil, *Plagues and people*, op. cit.

[32] Cunningham and Grell, *The Four Horsemen of the Apocalypse: religion, war, famine, and death in Reformation Europe*, op. cit.

[33] W.L. Salo et al. "Identification of Mycobacterium tuberculosis DNA in a pre-Columbian Peruvian mummy". In: *Proceedings of the National Academy of Sciences of the United States of America* 91.6 (1994), p. 2091.

[34] J. B. S. Haldane. "Disease and evolution". In: *Ric. Sci. Suppl. A* 19 (1949), pp. 68–76.

[35] J.J. Burdon. *Diseases and plant population biology*. Cambridge University Press, 1987.

[36] Rosin, "Coevolutionary Search Among Adversaries", op. cit.; V. Slavov and N.I. Nikolaev. "Immune network dynamics for inductive problem solving". In: *Parallel Problem Solving Nature*. Springer. 1998, p. 712; J. Paredis. "Coevolution, memory and balance". In: *International Joint Conference on Artificial Intelligence*. Vol. 16. 1999, pp. 1212–1217.

[37] Paredis, "Coevolution, memory and balance", op. cit.

generations to one or the other population. Although our simulation, based on the CG, could have been structured as a coevolution between two types of player (MP and CP), thus fulfilling the immune system framework, we chose to adopt a single-population approach because the specification of both players takes the same form, namely it is an approximate inference algorithm. Yet it would be good to keep in mind the possibility of using such a multi-species framework in future work.

In the meantime, there seems to be nothing in the existing GA research which is able to model the desired regulatory effect of diseases. Although we don't present a fully-specified model of our own in this "future work" section, we would like to describe some principles below which might outline such a model's behaviour and guide its development.

Recall that our original motivation for examining the effect of diseases in the "real world" originated from a desire to better understand how one might automatically regulate interaction topology in simulated evolution, with the goal of maintaining an appropriate level of diversity and limiting regression of fitness.[38]

In taking a closer look at the interaction between diseases and evolution, we first observe that diversity can have a protective effect against diseases. This effect has been recognised in biology.[39] The operating mechanism seems to be that although it is common for microparasites to infect multiple different species or sub-species, it is usually the case that they are better adapted to one or the other of these sub-species. As a result, diversity of hosts limits the scope of potentially severe infections to a particular subset of the population. We also suggest that diseases evolve most quickly when there are high rates of active infection, as for instance during a pandemic; whereas, on the other hand, when infections are spread across many subspecies but

[38]Here we are talking about utilising disease for a particular purpose: controlling the parameters of an evolutionary simulation. In this role, with regard to the optimisation of fitness, diseases would occupy a position secondary to that of the main simulation. But note that, if we imagine that the fitness of our individuals is defined not by some external criterion but by their ability to fight microparasites, as in the GA models of coevolution in the immune system, then apparently in such cases diseases could also interact with these goals directly, and not just through the intermediary of interaction topology. Of course, a model which combines in this way the goals of *fitness* and *health* would still need a mechanism for individuals to die as a result of "sickness" in order for diseases to have the required regulatory effect on sparsity. It may be worth trying to resolve these simple questions for the sake of obtaining a more powerful and streamlined framework. The discussion below applies to both settings, in which diseases are single-purpose or dual-purpose, but assumes the first because it is simpler.

[39]F. Keesing et al. "Impacts of biodiversity on the emergence and transmission of infectious diseases". In: *Nature* 468.7324 (2010), pp. 647–652.

only achieve severity in one of them, then disease evolution as a whole will proceed more slowly. Since we have postulated disease burden to be tied to the rate of evolution of disease, it follows that a diverse population will have a lower disease burden than a uniform one.

We have previously argued that diseases should promote diversity - by restricting population density - and now we have argued that, on the other hand, diversity limits disease incidence. These two mutual influences give rise to a second-order differential equation, with cyclic behaviour. We can describe the cycles of this system qualitatively, by dividing them into four phases, which we have illustrated in the diagram below.

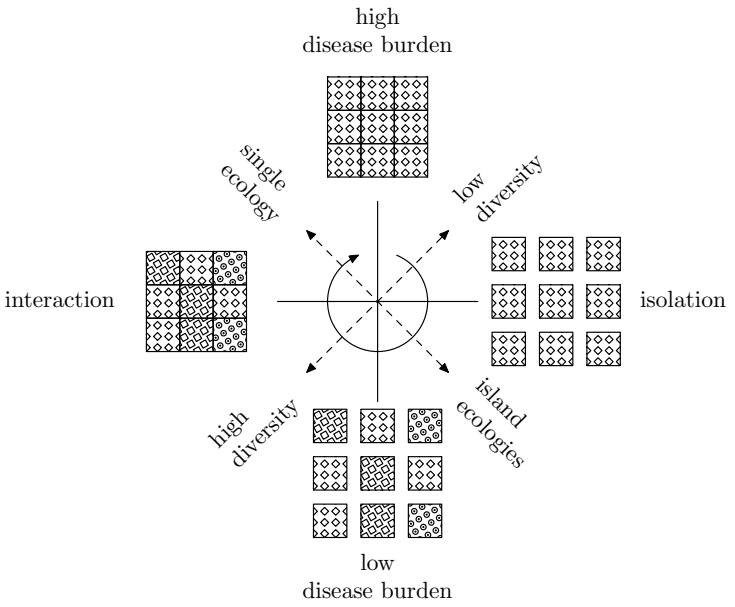
In the first phase (at the top of the diagram), a population has low diversity and lives in a single large community, which promotes the evolution of disease.

In the second phase, the high disease burden causes population density to decrease, and isolated islands to emerge.

In the third phase, populations in these islands evolve, but their isolation makes them take divergent evolutionary paths, thus increasing diversity and lowering disease burden.

In the fourth phase, the lower disease burden increases population (and population density), so that there is more contact between subpopulations.

As a result of this contact, diversity decreases, and the cycle repeats.



(6.1)

This is an interesting self-regulating system, and although it still contains free parameters to quantify the relationships between disease and diversity, perhaps these can be combined into a single parameter describing the frequency of the cycle.

We also note that these cycles which we have elucidated can operate in parallel when there are multiple species, and on multiple levels of a system at once when species can be organised into a taxonomical hierarchy or when diseases are more or less local in their circulation.[40]

Similar cycles can also be envisioned at a political or economic level. William McNeil's popular work on the interactions between microparasites and human populations also outlines a parallel discourse involving "macroparasites", which is to say, empires, governments, and other large institutions.[41] Such actors may play an important role in preserving diversity by regulating the interactions of their subjects. To incorporate such additional regulatory structure in our simulations, it may be useful to consider the purpose and dynamics of both macro- and micro-parasitic systems together.

To our knowledge, artificial or natural models based on the principles we have just described have not yet been proposed in print.

Although we have only made a cursory search, we are not aware of any research in evolutionary biology which recognises our proposed "disease-diversity" cycle.

In the GA literature, there have been a number of published experiments with immune system models, coevolutionary models, and "island models", but these generally adopt a fixed population and have not been combined in the manner we envision for the automatic regulation of interactions between agents.

One criticism of our proposal is that automatic regulation does not imply optimal regulation, and we have not explained why the sparse population

[40]Note that in the simplest form of this model there is no need for restriction on the movement of diseases between subpopulations; diseases can be the same everywhere. In the real world, although many parasites have global circulation, such as influenza, there are also parasites which evolve in more isolated subpopulations, and it may be useful to model diseases with different levels of localised diversity. The basic principles we have outlined apply to either case. But in the more realistic version, it is clear that low population density reduces disease evolution not only by increasing diversity but also by slowing the rate of spread of disease. This is related to the likelihood that susceptibility to disease is itself a trait which can be selected for or against. Higher susceptibility might give a host the advantage of a more up-to-date immune system, even as it hurts his neighbours by subjecting them to greater disease burdens, resulting in a kind of prisoners' dilemma. The presence of more localised diseases could enforce cooperation in such a game.

[41]McNeil, *Plagues and people*, op. cit.

structure which results from the action of transmissible diseases should also happen to be an optimal one from an evolutionary standpoint. To answer this, imagine that disease susceptibility, a natural parameter through which we can tune the influence of diseases on evolution, is a trait that can be selected for. We discussed other aspects of variable disease susceptibility in the previous footnote. Although the subject is a complex one, we can see that, on the surface of it, populations with a sub-optimal susceptibility should, as a result of evolving more slowly than their neighbours, eventually be overtaken and assimilated. Thus, to rough approximation, there should be a selection pressure towards the optimal amount of disease susceptibility.

There may in fact be better ways than the biological analogies we have advanced, in which to reduce the number of free parameters in evolutionary simulations and to increase their autonomy. But whether this "parameter problem" is solved through our method, or somebody else's, we feel that it is an important subject for future research. It is difficult to imagine systems for approximate inference based on artificial evolution becoming efficient without having first solved this problem. And one expects that an approach similar to ours, based on looking at the world around us, should lead to valuable insights. After all, the biological evolution which (as we would argue) produced humans, should also have produced, through the same process of natural selection, sophisticated mechanisms to simplify its task and shorten the time it takes to achieve its goals.

6.6.2 Modelling somatic selection

The subject matter of this section, a theory of evolution associated with Jean Baptiste Lamarck, has remained controversial among evolutionary biologists for over one hundred years. Although relevant to the study of Genetic Algorithms, like the rest of section 6.6 it can be omitted without detracting from an understanding of this chapter's research results. Readers who find themselves displeased by subjects of controversy or scientific debate are encouraged to skip ahead to section 6.6.3.

In the previous section, we proposed ways of addressing shortcomings in the naive approach which is taken by the traditional GA framework to the *choice of population size* and *interaction topology*. In this section, we suggest improvements upon the traditional, naive approach to *crossover* and *mutation*. Recall that crossover and mutation are the operators which, in the traditional GA framework, are used to construct the genotype of an individual from those of its two parents. These are modelled on a dominant view

of evolution and genetics called "Modern Evolutionary Synthesis". Here we discuss an alternative description of evolution, and discuss how it might be applied to GAs.

6.6.2.1 Football

By way of introduction, let us describe an analogy between biological evolution and the formation of teams in some competitive sport such as football[42]. Football teams are comprised of eleven players, and are themselves members of divisions or leagues. At the very lowest level of play are neighbourhood teams or amateurs. As teams become more skillful, they can move up a division, attract a larger audience, start to earn enough income to have a full-time coach, and spend more time practising. They will start to compete against teams from distant localities. With enough success, perhaps just before reaching the top national division, a team will be able to attract a sponsor and become fully professional.

Football teams derive some of their income from advertising, by wearing the logo of a commercial sponsor when they play, and some income comes from ticket sales. Winning or losing only affects income indirectly through these channels. Teams can trade players, but only during specified months, so we can see that there are mechanisms to limit their interactions (as endorsed in the previous subsection). Also, and perhaps more importantly, a certain amount of local or national pride penalises teams which simply consist of the best foreign players or have too much turnover. If a team's income were derived only from winning games, these last two considerations would be less important, so one can see how the ticket sales and sponsorship business model creates a certain amount of diversity by compelling teams to adopt a unique character.

Now imagine viewing teams as agents in an evolutionary simulation, whose genes consist of players. The goal of such a simulation would be to produce an optimal team by exchanging (or copying) players and modifying them in different ways, occasionally playing them against each other to estimate progress or create rankings. One can see immediately that the standard crossover and mutation operations employed by GAs would be terribly inefficient in this task. Random mutation is just as likely to produce an improvement as a regression, and random crossover is just as likely to copy a bad gene as a good one. If there were only a few variables to explore, these techniques might be suitably efficient, but with each player

[42] We are referring to the sport that Americans call "soccer", although the difference is not important in what follows.

being described by (let's say) hundreds of variables, and with eleven players per team, it is no longer feasible to use random walks to explore the space of configurations. The "schema theorem"[43] suggests that if there is a low-dimensional subspace which parametrises consistently better teams, then GA-style evolution will eventually converge to this subspace; but it is unclear how it could do so faster than methods such as Powell's method (a gradient-less version of nonlinear conjugate gradient) which require many fitness evaluations per dimension, thus demanding thousands of games to take a single "step" towards this manifold. Hundreds of such steps might be required for useful progress, so that hundreds of thousands of games would be needed to create a good team.

Yet in the "real world" it is clearly possible to produce good players and teams using a much smaller number of games. It is instructive to think about how such selection might be performed. First of all, players are able to learn and improve their performance not only by trying out random new techniques but also by watching and emulating each other. Such changes can even be implemented and evaluated over the course of a game. Secondly, coaches can watch players and measure their performance, and use these observations in deciding when to acquire a new player, perhaps from a lower-division team, or when to let go of an existing one. To some extent such evaluation depends on a coach's ability to simulate a game in his head, which could be inefficient to model, but there are also simple metrics such as number of goals scored which can serve as a rule of thumb. More careful evaluation might follow the ball over the course of a game - at the moment when one player acquires the ball, what is our estimate of the value of the position of the players; and when he passes the ball or loses it, is the new position stronger or weaker? A better player will tend to improve his team's position by a larger step between the times that he gains and relinquishes control of the ball. Similar reasoning is used to estimate the value of an action in reinforcement learning using dynamic programming. In addition to carrying out such evaluations, one would also want to account for those occasions when a player never touches the ball, but still performs a useful purpose by covering an opposing player or otherwise preventing him from acquiring it. On the other hand, those players who neither touch the ball nor prevent others from touching it can be safely appraised useless. Thus, one can reliably calculate the value of a team's players by observing where the ball goes and imagining where it could go. Valuing players by mentally simulating a whole game with and without each player is not necessary (note

[43] Holland, "Adaptation in natural and artificial systems", op. cit.

this would have similar time complexity to the traditional GA approach of trying out random teams with different combinations of players). We will return to various aspects of this analogy after making some observations about biological evolution.

6.6.2.2 Biological Rerevisionism

The role of crossover and mutation in GAs is based on a view of genetics and natural selection called Modern Evolutionary Synthesis or Neo-Darwinism, which is associated with Charles Darwin (1809-1882) and Gregor Mendel (1822-1884). According to this view, genes are inherited at random from one or the other parent, and undergo random mutations from one generation to the next; natural selection then filters the best individuals out of the resulting chaos. All of the pain and pleasure that an organism experiences during his lifetime, and all of the physical and mental adaptation that he undergoes, or the diseases that he becomes immune to - these experiences and adaptations only contribute to the survival of the individual organism and are not transmitted to his offspring (at least not at the genetic level). The counterpart to this view, which hypothesises the heritability of some such adaptations, is referred to as the Inheritance of Acquired Characteristics (IAC). IAC is most commonly associated with an early proponent of the principle, Charles Darwin's lesser-known predecessor, the French naturalist Jean Baptiste Lamarck (1744-1829). For this reason, IAC is popularly known as Lamarckism or Neo-Lamarckism, although we prefer the broader term IAC since Lamarck espoused other theories as well.

Few people have heard of Lamarck, and fewer still are aware of the fact that Darwin himself was also a proponent of IAC.[44] Darwin first proposed the following mechanism, which he called Pangenesis, in his "The variation of animals and plants under domestication": Cells in the body (which is to say, somatic cells) are continually dividing and being killed (by diseases or toxins, the immune system, physical trauma, etc.). Thus, as their genes experience random mutations, they undergo a process of selection - now referred to as *somatic selection*. Cells with more useful genomes will tend to survive in greater numbers. Darwin suggested that somatic cells may send out "gemmules" which travel to the gonads and integrate successful new genes into the germline, each cell having an equal "vote" in the constitution of the resulting genome. In this way, beneficial mutations which occur in somatic cells could be passed along to offspring. Such a process would

[44] C. Darwin. *The variation of animals and plants under domestication.* Murray, 1868; C. Darwin. *The Descent of Man, and Selection in Relation to Sex.* Murray, 1871.

greatly improve the speed at which genes could evolve: for humans, instead of genes being limited to a populations of billions of individuals producing a new generation every 20 years, they could now undergo selection in a population of 100 trillion cells with new generations occurring for some cells on a daily or weekly basis. Furthermore, there is no reason to consider the mechanism biologically implausible. That evolution would clearly show a strong preference to organisms in which such a mechanism existed argues powerfully in its favour.

Since the 1970s, molecular biologists have been aware of a candidate mechanism for Pangenesis-like functionality, in the form of retroviruses, which have the ability to integrate a genetic payload into a host cell's genome.

Retroviruses were first observed in rare transmissible cancers, starting with the Rous sarcoma virus;[45] but the realisation that viral nucleotide sequences were found in the DNA of all uninfected chicken cells,[46] that the viruses had a tendency to move active genes between cells, and other evidence led to a proposal by Howard Temin - called "the protovirus hypothesis" - that these disease-causing viruses had originated in genes related to normal cellular processes, such as the formation of antibodies.[47] The pathogenic effects of RSV were later found to derive from its genetic payload - a mutant version of a gene regulating cell growth - implying that the sarcoma gave rise to the virus, rather than vice-versa.[48] In his book "Somatic selection and adaptive evolution" (1981), Edward J. Steele, the most prominent of the modern proponents of IAC, elaborated the protovirus hypothesis and connected it to IAC and Lamarck.[49] He explains that an endogenous (i.e. originating internally to the organism) retrovirus (ERV) would be able to move between cells copies of those genes which are being expressed (and therefore undergoing transcription into mRNA) by capturing their mRNA in the viral envelope; the majority of genes, being unexpressed, would be ignored. Through this process, helpful mutations from diversely located areas

[45] P. Rous. "A sarcoma of the fowl transmissible by an agent separable from the tumor cells". In: *The journal of experimental medicine* 13.4 (1911), p. 397.

[46] J. Tooze. *The molecular biology of tumour viruses*. Cold Spring Harbor Laboratory, 1973.

[47] H.M. Temin. "Malignant transformation of cells by viruses." In: *Perspectives in biology and medicine* 14.1 (1970), p. 11.

[48] H.M. Temin. *The DNA provirus hypothesis*. Nobel lecture. 1975; E.J. Steele. *Somatic selection and adaptive evolution: on the inheritance of acquired characters*. University of Chicago Press, 1981.

[49] Steele, *Somatic selection and adaptive evolution: on the inheritance of acquired characters*, op. cit.

of the genome, each improving the function of a particular type of somatic cell, could be incorporated together into a recipient germline cell's genome for transmission to offspring. Steele also proposes that gene exchange between pairs of somatic cells should occur using the same mechanism, and he hypothesises organ-specific retroviruses to control such exchange.[50] In other research, Steele shows that an enzyme utilised in DNA repair, called DNA polymerase eta, makes use of reverse transcription.[51] This suggests that imported retroviral genetic payloads could be incorporated specially by somatic cells under stress into their flagging genomes, providing a mechanism whereby cells performing unsatisfactorily could emulate their neighbours.

With the advent of the Human Genome Project,[52] we now know that 8% of the genome codes for ERVs.[53] ERVs are expressed most abundantly in the epididymis[54] and are also operative in the placenta,[55] suggesting a role in reproduction, and blocking their action makes pregnancy impossible.[56] Sperm possess a mechanism allowing reverse-transcription of foreign RNA into heritable DNA sequences existing outside of chromosomes.[57] Thus, a fully functional mechanism for IAC is to be found in humans and in other animals (and perhaps even in plants as well), which is actually very similar to the pangenesis mechanism proposed by Darwin in 1868, with retroviruses taking the place of "gemmules".

There is, in addition, considerable evidence that IAC actually takes

[50] Ibid.

[51] E.J. Steele. "DNA polymerase-eta as a reverse transcriptase: implications for mechanisms of hypermutation in innate anti-retroviral defences and antibody SHM systems." In: *DNA repair* 3.7 (2004), p. 687.

[52] F.S. Collins et al. "New goals for the US human genome project: 1998-2003". In: *Science* 282.5389 (1998), p. 682.

[53] R. Belshaw et al. "Long-term reinfection of the human genome by endogenous retroviruses". In: *Proceedings of the National Academy of Sciences of the United States of America* 101.14 (2004), p. 4894.

[54] A.A. Kiessling, R. Crowell, and C. Fox. "Epididymis is a principal site of retrovirus expression in the mouse". In: *Proceedings of the National Academy of Sciences of the United States of America* 86.13 (1989), p. 5109; R. Crowell and A. Kiessling. "Endogenous retrovirus expression in testis and epididymis". In: *Biochemical Society Transactions* 35 (2007), pp. 629–633.

[55] S. Mi et al. "Syncytin is a captive retroviral envelope protein involved in human placental morphogenesis". In: *Nature* 403.6771 (2000), pp. 785–789.

[56] K.A. Dunlap et al. "Endogenous retroviruses regulate periimplantation placental growth and differentiation". In: *Proceedings of the National Academy of Sciences* 103.39 (2006), p. 14390.

[57] C. Pittoggi et al. "Generation of biologically active retro-genes upon interaction of mouse spermatozoa with exogenous DNA". In: *Molecular reproduction and development* 73.10 (2006), pp. 1239–1246.

place in a variety of organisms. Darwin himself[58] documented a variety of anecdotes suggesting the possibility of IAC in both animals and humans. Decades of experiments performed by Paul Kammerer (1880-1926) uncovered a number of instances where by simulating environmental changes he could directly induce the acquisition of new traits in animals such as salamanders, olms, newts, toads, and sea squirts; Kammerer was then able to demonstrate the heritability of such adaptations[59]. Soviet geneticist Trofim Lysenko found that winter wheat varieties could be made to grow in spring, by cold-treating their seeds in a process called "vernalisation"; he also found that these changes were heritable.[60] Some learned behaviours are thought to be heritable as well, as was asserted by Pavlov.[61] William McDougall conducted experiments with rats at Harvard between 1927 and 1933, which demonstrated improvements over 34 generations in learning rate for a maze task, despite the fact that he used only the worst-performing individuals for breeding (to give a negative selection bias).[62] McDougall's results were replicated[63] successfully[64] by Wilfred Agar at the University of Melbourne.[65]

[58]Darwin, *The variation of animals and plants under domestication*, op. cit.

[59]For example, Kammerer was able to coerce ovoviviparous fire salamanders to become viviparous, and viviparous alpine salamanders to become ovoviviparous. He found these changes to be heritable. Kammerer's life and scientific accomplishments were memorialised in the Soviet film *Salamandra* (1928). (A. Koestler. *The case of the midwife toad*. Hutchinson (London), 1971)

[60]R. Amasino. "Vernalization, competence, and the epigenetic memory of winter". In: *The Plant Cell Online* 16.10 (2004), p. 2553.

[61]Koestler, *The case of the midwife toad*, op. cit.

[62]W McDougall. "An experiment for the testing of the hypothesis of Lamarck". In: *Brit. J. Psychol.* 17 (1927), pp. 267–304.

[63]WE Agar et al. "Fourth (final) report on a test of McDougall's Lamarckian experiment on the training of rats". In: *Journal of Experimental Biology* 31 (1954), pp. 307–321.

[64]Agar, despite using a lower shock voltage than McDougall for training, found a statistically significant improvement in his trained colony. See table 4 in Agar 1954. A t-test comparing 1 to the ratio of percent membership in the first performance class for training versus control gives a p-value of 0.955, for membership in performance class 10 (reversing the comparison) the p-value is 0.917. Including only generations 25-50, where training can be expected to have had a greater effect, gives p-values of 0.913 and 0.999, respectively. There are 13 data points in total.

[65]It is difficult to imagine a biological mechanism which could be responsible for the heritability of learned behaviours. We note, however, that the immune system is estimated to generate about 10 billion antibodies (L.J. Fanning, A.M. Connor, and G.E. Wu. "Development of the immunoglobulin repertoire". In: *Clinical immunology and immunopathology* 79.1 (1996), pp. 1–14), which cross-react in a regulatory network (N.K. Jerne. "Towards a network theory of the immune system." In: *Annales d'immunologie*. Vol. 125. 1-2. 1974, p. 373) whose complexity could come close to that of the nervous system (with 100 billion neurons); because the primary structure of antibodies is heritable, a correspondence be-

Steele's IAC research gives special attention to the immune system, since the highest rates of natural somatic evolution among the various cells in the body are found in lymphocytes. This evolution of lymphocytes underlies the process by which the immune system learns to recognise antigens - called "somatic hypermutation" (SHM). In SHM, lymphocytes divide, survive or perish, and undergo mutations in their antibody-encoding genes. SHM plays an indispensable role in the mechanism by which an animal recovers from an infection, for example.

A number of experiments have shown that acquired changes to the immune system are heritable. We outline a few of the major results. Through a series of experiments in rabbits, Guyer and Smith[66] showed that acquired autoimmune diseases could be inherited. In experiments where a (male or female) rabbit or rat is inoculated with a particular antigen prior to mating, the offspring are (upon inoculation with the same antigen) found to have antibodies with idiotype[67] expression levels that show characteristics of maternal[68] and paternal[69] influence. Some take this to suggest that the somatic genes which give antibodies their specificity - called v-genes, since they code for antibody "variable regions" - are directly heritable. One promising mechanism for v-gene heritability would involve transferring such genes into the germline from circulating lymphocytes using retroviruses. In fact, a well-studied retrovirus which is found in lymphocytes has most of its genetic variability concentrated in its envelope proteins, in five regions totalling about 117 amino acids, while the length of an antibody heavy chain v-gene is about 110 amino acids long, a near match.[70] One might imagine that this particular retrovirus could be related to some immune-specific ERV respon-

tween the nervous system and the immune system could explain the heritability of learned behaviours.

[66]MF Guyer and EA Smith. "Studies on cytolysins. I. Some prenatal effects of lens antibodies". In: *Journal of Experimental Zoology* 26.1 (1918), pp. 65–82; MF Guyer and EA Smith. "Studies on cytolysins. II. Transmission of induced eye-defects". In: *Journal of Experimental Zoology* 31.2 (1920), pp. 171–223.

[67]The idiotype of an antibody refers to those aspects of its shape not related to antigenic specificity.

[68]M. Wikler et al. "Immunoregulatory role of maternal idiotypes. Ontogeny of immune networks." In: *The Journal of experimental medicine* 152.4 (1980), p. 1024; J.C. Olson and G.A. Leslie. "Inheritance patterns of idiotype expression: maternal-fetal immune regulatory networks". In: *Immunogenetics* 13.1 (1981), pp. 39–56.

[69]C.A. Cooper-Willis et al. "Influence of paternal immunity on idiotype expression in offspring". In: *Immunogenetics* 21.1 (1985), pp. 1–10.

[70]S. Modrow et al. "Computer-assisted analysis of envelope protein sequences of seven human immunodeficiency virus isolates: prediction of antigenic epitopes in conserved and variable regions." In: *Journal of Virology* 61.2 (1987), p. 570.

sible for the transfer of antibody genes between pairs of lymphocytes and between lymphocytes and germline cells[71]. We also remark that the transfer of blood between a post-inoculation rabbit and a naive one causes the recipient to produce antibodies with the same idiotypic markers as the donor.[72] Recent research by Steele exhaustively analyses asymmetries in DNA nucleotide mutation probabilities, reconstructed from published antibody gene sequences, to argue that certain kinds of mutation must be occurring in an RNA intermediary - implying a (presumably retroviral) reverse transcription step in which antibody genes are shared between lymphocytes.[73] Even without such a detailed analysis of cellular mechanisms, there are a number of basic characteristics of v-genes which are difficult to explain using a Neo-Darwinian model of mutation: for instance, the presence of many homologous copies of a v-gene in the same genome, and the concentration of variability into three distinct subregions.[74]

6.6.2.3 Simulated evolution

We have brought together a number of biological facts in this section because they are not elsewhere described well enough for our purposes. Mainstream biology gives an alternative interpretation to the evidence we have presented, but at the same time it is clear that there is no reason to avoid trying to implement the same mechanisms in simulation which we could plausibly

[71] The presence of antibodies to HIV is also associated with antibodies to ERVs (K.E. Garrison et al. "T cell responses to human endogenous retroviruses in HIV-1 infection". In: *PLoS Pathog* 3.11 (2007), e165).

[72] G. Urbain-Vansanten et al. "Synthesis of antibodies and immunoglobulins bearing recipient allotypic markers and donor idiotypic specificities in irradiated rabbits grafted with allogeneic cells from hyperimmune donors." In: *Annales d'immunologie*. Vol. 130. 3. 1979, p. 397; J. Urbain et al. "Sharing of idiotypic specificities between different antibody populations from an individual rabbit". In: *European Journal of Immunology* 5.8 (1975), pp. 570–575.

[73] E.J. Steele and J.W. Pollard. "Hypothesis: Somatic hypermutation by gene conversion via the error prone DNA→RNA→DNA information loop". In: *Molecular Immunology* 24.6 (1987), pp. 667 –673; Robert V. Blanden et al. "The signature of somatic hypermutation appears to be written into the germline IgV segment repertoire". In: *Immunological Reviews* 162.1 (1998), pp. 117–132; E.J. Steele et al. "Computational analyses show A-to-G mutations correlate with nascent mRNA hairpins at somatic hypermutation hotspots". In: *DNA repair* 5.11 (2006), pp. 1346–1363; E.J. Steele. "Mechanism of somatic hypermutation: critical analysis of strand biased mutation signatures at A: T and G: C base pairs". In: *Molecular immunology* 46.3 (2009), pp. 305–320.

[74] T.T. Wu and E.A. Kabat. "An analysis of the sequences of the variable regions of Bence Jones proteins and myeloma light chains and their implications for antibody complementarity". In: *The Journal of Experimental Medicine* 132.2 (1970), p. 211.

hypothesise in nature. In fact, computer scientists are not restricted to making analogies to nature, and some methods in the field of GAs are based on a supposedly "Lamarckian" view of evolution while at the same time denying or ignoring the relevance of IAC to biology.[75]

In our own evolutionary simulations, we included a selection strategy that we called "directed crossover" (this term has a different meaning in GAs), which was intended to capture some small fraction of the abilities of IAC. In our version, each variable could be compared to a cell, and the set of GBP regions to genes in the genome. Within a winner of the CG, the "success" of a cell was measured by how greatly the conditioned marginals of the corresponding model variable differed from those of the losing opponent. The crossover operation then involved copying random "genes" (GBP regions) from successful cells (variables) which were "active" in those cells (i.e., the GBP regions contained the corresponding variables) from the winner to the loser.

The performance of our directed crossover was not distinguished from the other methods; it was slightly less efficient overall, but not enough so as to discourage us from pursuing the ideas of IAC in a more refined form. Indeed, the implementation we chose was simplistic enough that even had it performed much worse, this would not have deterred us from recommending a more sophisticated implementation of the ideas of IAC. We regret that we must leave such an implementation to future work.

Although our own version of IAC left much to be desired, at the same time it is not easy to be enthusiastic about the methods which have been proposed in the GA literature. The "Lamarckian" methods of Ross, Whitley, and other authors generally function by directing the simulation to evaluate a sequence of specific mutations, somewhat like coordinate ascent, in a *local search* after each reproduction, and select the best individual from among the results of each one.[76] In other words, a new individual is required in order to evaluate each proposed change. Due to their time complexity, from a biological standpoint these methods are closer in spirit to cloning of individuals than to somatic selection of cells. We suggest that one should try to bring IAC to the field of GAs by first devoting some attention to

[75] D. Whitley, V. Gordon, and K. Mathias. "Lamarckian evolution, the Baldwin effect and function optimization". In: *Parallel Problem Solving Nature from PPSN III* (1994), pp. 5–15; B.J. Ross. "A Lamarckian Evolution Strategy for Genetic Algorithms". In: (1999).

[76] Whitley, Gordon, and Mathias, "Lamarckian evolution, the Baldwin effect and function optimization", op. cit.; Ross, "A Lamarckian Evolution Strategy for Genetic Algorithms", op. cit.

the more plausible and parsimonious IAC mechanisms which have been put forward by professional biologists, such as Darwin and Steele.

The same criticisms that we have levelled against the "Lamarckism" of GAs also hold for many implementations of the closely related "Baldwin effect". In the Baldwin effect, local search is used to optimise each new individual, but the results are not incorporated back into the genome.[77] Whitley found that the Baldwin effect can sometimes outperform "Lamarckian" methods (Lamarckian in the GA sense).[78] The poor time complexity of both methods, requiring a number of fitness evaluations proportional to the size of the genome, may be slightly remedied by more intelligent "local search" methods which could for example identify good search directions by efficiently computing a gradient rather than performing coordinate ascent. Yet even then, the GAs version of Lamarckism (and the related Baldwin effect) fall short of the power of the Pangenesis/retroviral mechanism proposed by Darwin and Steele, in which a type of cell that makes heavy use of a specialised gene can evaluate and then broadcast helpful mutations specific to that gene throughout the body. To highlight the contrast between the plausible capabilities of IAC and the impoverished mechanisms of existing GA research, we outline some ideas about how an optimisation method using IAC-based natural selection could be constructed. To this end we now return to the football analogy with which we opened this section.

We hope that it is apparent at this point that many of the behaviours we identified in the process of training and assembling a good football team, which may have seemed complex and beyond the capacity of simple biological mechanisms to implement, actually have straightforward biological analogies: when a football player emulates the behaviour of a better player, we can compare this to a cell under stress incorporating retroviruses from successful, reproducing cells, and updating its genome to emulate theirs. When one team acquires a good player from another team, we can for example imagine an embryo inheriting a successful combination of the genes of its parents, which have been evaluated and aggregated into a pair of germline cells through the Pangenesis/retroviral mechanism.

There is one part of the football analogy whose biological counterpart we have not yet discussed, and which stands out as still lacking an obvious mechanism. This is the process by which it is possible to evaluate a player by watching as the ball passes in and out of his possession. The observation that

[77] G.E. Hinton and S.J. Nowlan. "How learning can guide evolution". In: *Complex systems* 1.1 (1987), pp. 495–502.

[78] Whitley, Gordon, and Mathias, "Lamarckian evolution, the Baldwin effect and function optimization", op. cit.

one might evaluate the potential or strength of an arrangement of players and ball on the field (which could be quantified as the probability of scoring the next goal, for example) at the instant of time when the ball comes into the control of a given player, and again when it leaves his control, and then credit him with the difference, is similar in spirit to an idea proposed by Eric Baum in "Toward a model of intelligence as an economy of agents".[79] In that paper, Baum considers solving a reinforcement learning problem using a population of agents. At each step, agents bid for control of the system, and the winning agent gets to choose the next action which the system takes. He then receives any reward from the environment, plus the amount of the next agent's bid to receive control of the system. Thus, money is conserved within the system, and tends to accumulate with those players who are skillful in taking actions that result in rewards or which leave the system in a more highly-esteemed state than it was found. This accumulation of money is then used to guide a kind of evolution among players.

In both Baum's proposal and our football analogy, there is a system which traces out a one-dimensional sequence over time - consisting of actions in one, and possession of the ball in the other; and an objective for this system - to get rewards, or to score goals. By attributing responsibility for the system at any given time to one of a collection of cooperating agents or players, one can define a conserved quantity representing progress towards this goal which is then distributed between these agents and used to apportion credit between them for the system's overall performance. The sequential, one-dimensional nature of such a credit-assignment framework seems to be the key idea. We are not sure to what this notion might correspond in biology, but note that a familiar conserved quantity is energy, and that if the energy from food or the air were somehow to flow through the body in one-dimensional paths, then the manner in which cells are able to use this energy could serve as a guideline for determining their fitness and deciding how much they should be allowed to reproduce. Such a process is not to our knowledge proposed by Darwin or Steele, but neither do their theories explain very well how fitness is determined for the purpose of somatic selection, for example how tumours and other malignancies are avoided.

The sequential control pattern outlined above can easily be pictured within the conditional game as well. In this case, control over the system

[79] E.B. Baum. "Toward a model of intelligence as an economy of agents". In: *Machine Learning* 35.2 (1999), pp. 155–185.

would represent the ability to clamp a given variable to a given value, and the final objective would be the game's value. If one is to use the more sophisticated and powerful framework of adaptive evolution in trying to evolve a population of players who can play the conditional game, an important problem would be to understand how to partition credit for a player's performance among his "genes". It may be well to keep in mind Baum's economy of agents model, and the above discussion, in trying to address this problem.

The essential difference between our evolutionary ideas and those of the traditional GA framework, is that in our framework, as in the Pangenesis/retroviral model of IAC, evolution happens simultaneously on multiple levels of an overlapping hierarchy ("overlapping" in the sense that the structure is more like a DAG than a tree). In retroviral IAC, most of the natural selection which takes place does so at the level of cells, which are allowed to replicate in proportion to their service to the organism, and which communicate their improvements among each other and to the germline cells. The next higher level of natural selection is that of organisms, and sometimes philosophers and social scientists talk about evolution on higher levels such as societies as well. A solution to the problem of appropriately promoting the survival of cells that are useful to the whole organism, which is a kind of credit assignment problem, is presupposed by IAC. It may in turn be helpful to frame this problem in terms of reinforcement learning (or sequential decision theory) as proposed above. The full credit assignment problem, requiring an exact computation of the fitness of the optimal organism, would be intractable, and casting credit assignment in the sequential control framework with some approximate notion of reward would only facilitate an approximate solution. Evolution on higher levels such as that of organisms could then be seen as a necessary consequence of the imperfect and approximate nature of the values assigned by such a framework to cells and genes (consider the problem of cancer, for example). Thus, the mortality of organisms is related to communication between their cells. We note that just as cells can exchange retroviruses, organisms can also exchange information as well. This information can be communicated either by observation of another's behaviour, or through language, or yet again in the form of retroviruses (which can be transferred as a corollary of sexual reproduction).

The overlapping hierarchy of communication between organisms and cells in IAC can be seen as analogous to the interwoven structure of interactions in an evolutionary simulation, where we saw that the (cycle-free) tree-structured SET was sub-optimal, as well as the fact that diseases exist in a (presumably overlapping) hierarchy of successively more localised

species (mentioned on page 145 in section 6.6.1.2). We imagine that part of designing a successful optimisation framework based on IAC would be gaining a better understanding of how these various hierarchies fit together in a natural and seamless manner, and to what extent the levels of such hierarchies can be fluid or emergent versus granular or primary.

6.6.2.4 Contemporary wisdom

We have permitted ourselves to engage in some healthy speculation about the biological mechanisms which might underlie a type of evolution which is more efficient and powerful than the one espoused by most biologists. Were the IAC concept and the Pangenesis/retroviral mechanism to win general support among evolutionary biologists, those who study GAs would certainly suffer some embarrassment on account of having neglected to consider one of the most basic processes driving adaptation. We can also surmise that these theories, implying the possibility of horizontal transmission of genetic and immunological information, could be attended by certain political difficulties. Thus it is with a sigh of relief that we inform the reader of the numerous reasons which can be given for disregarding the evidence we have presented in favour of IAC.

Biologists universally reject the possibility of the inheritance of acquired characteristics. The scientific consensus among biologists is that all viral fragments present in the human genome are biologically inactive, mere fossil remnants from infections which occurred millions of years ago. A few of these infections may have been able to serve a useful purpose - for instance, placental cells are held together by retroviral envelope proteins, and this helps them create an impermeable barrier between the blood of the mother and that of the foetus.[80] This is why retroviruses are expressed on the placenta. Transmission of immunity to offspring is made possible by the sharing of antibodies, which are able to move across the placenta; heredity of idiotypic markers is found to occur because of the immunological network, which guides the recipient immune system in evolving antibodies with the same idiotypes as the donor.[81] Horizontal transmission uses the same mechanism, although paternal heredity of immunity is discredited by most immunologists.[82] As for the possibility of IAC in other organs, Richard Dawkins (1941-) points out that the genome is complex and more like a recipe than

[80] Mi et al., "Syncytin is a captive retroviral envelope protein involved in human placental morphogenesis", op. cit.

[81] J. Flegr. *Evolutionary biology, 2nd edition.* Academia Prague, 2009.

[82] Ibid.

a set of attributes. Expecting it to be able to capture the changes occurring through use and disuse of an organ is like expecting to be able to modify a cake recipe so that the cake comes out of the oven with one slice missing.[83] Many animal experiments have confirmed the impossibility of IAC. For instance, Francis Galton (1822-1911), cousin of Darwin, performed experiments in which blood from one variety of rabbit was transfused into another; the offspring of the recipient failed to show any of the donor's traits.[84] August Weismann (1834-1914) performed experiments in which he cut off the tails of rats, but their offspring still had tails even after many generations of such modification, thus proving the existence of what became known as the "Weismann barrier", a tissue barrier which prevents genetic information from somatic cells from passing into the germline.[85] As for the experiments of McDougall, we should note that Agar's 20-year reproduction of McDougall's experiment, in which he used proper controls, found that trainability of the trained and control colony both fluctuated up and down together, and showed clear seasonal variations as well. This suggests that the improvements observed by McDougall were due to a general improvement in the health of his colony and not to IAC, a possibility which he would have detected had he used controls.[86] We also note that McDougall was a proponent of eugenics and worked to establish parapsychology as a respected scientific discipline.[87] As for Paul Kammerer, he was exposed as a fraud by Gladwyn Kingsley Noble, who discovered the black nuptial pads on the feet of Kammerer's midwife toad specimen to contain India ink - Kammerer was so embarrassed that he committed suicide in the forest of Schneeberg.[88] Kammerer also believed in spontaneous generation.[89] And Lysenko stands accused of scientific fraud of an even more serious nature. He obstructed Soviet genetics for decades with his Marxism-based dogmas of IAC, he contributed to Soviet malnutrition with his opposition to hybrid corn and support of failed agricultural policies costing over a billion rubles,

[83] R. Dawkins. *The selfish gene*. Paladin, London, 1976.

[84] M.G. Bulmer. *Francis Galton: pioneer of heredity and biometry*. Johns Hopkins University Press, 2003.

[85] Flegr, *Evolutionary biology, 2nd edition*, op. cit.

[86] Agar et al., "Fourth (final) report on a test of McDougall's Lamarckian experiment on the training of rats", op. cit.

[87] E. Asprem. "A nice arrangement of heterodoxies: William Mcdougall and the professionalization of psychical research". In: *Journal of the History of the Behavioral Sciences* 46.2 (2010), pp. 123–143.

[88] Koestler, *The case of the midwife toad*, op. cit., p. 13.

[89] W. Reich. "The discovery of the orgone. Vol. I. The function of the orgasm; sex-economic problems of biological energy." In: (1942), p. 26.

and he even used the secret police to eliminate his scientific opponents.[90]

In spite of these misfortunes, we think that IAC is an important idea whose implications might inspire some productive developments in the field of GAs. Even if it is doomed to languish forever in margins of biology, its concepts and mechanisms could still be used to improve the performance of simulated evolution to the point of enjoying wide practical applicability. We also raise the tentative suggestion that the GA field could consider other analogies, such as the simple one we have sketched involving football, which are less encumbered and easier to investigate than biological evolution.

6.6.3 Conclusion

In this section we have reviewed the lessons, successes, and shortcomings of our attempt to harness the CG in evolutionary optimisation, and we have discussed at some length our ideas about principles and techniques which might play a role in more serious projects of the same nature. Most of the ideas we discussed have not been explained elsewhere, which is why we considered it necessary to set them down here. Perhaps a long "future work" section is also appropriate to a chapter which presents such disappointing results.

We hope that the foregoing material was stimulating. However, given all of the issues we raised and the analogies we drew upon, there may be some doubt in the mind of the reader as to whether it is possible - without getting lost in an endless progression of increasingly complex concerns and remedies - to define a flexible self-regulating evolutionary system which can apply well to both small and large inference problems.

On the other hand, in the introduction (section 6.1) we argued that the only way to create an accurate and autonomous approximate inference algorithm is to incorporate notions of cooperation and competition, as we have tried to do, to result in some kind of genetic algorithm-like system. There seems to be no reason to avoid attempting to improve upon the naive, GA-inspired framework we documented in the experiments of this chapter, until we are able to obtain a useful approximate inference algorithm. A good benchmark is the rate at which evolutionary simulations are able to improve the error of an approximation; simply outperforming or even replicating the $\frac{1}{\sqrt{n}}$ rate of sampling methods would be a worthy goal. Furthermore, even though explaining the notions of this section required numerous analogies, they are at their core rather simple, so that we might imagine that they

[90] B.M. Cohen. "The descent of Lysenko". In: *Journal of Heredity* 56.5 (1965), p. 229; B.M. Cohen. "The demise of Lysenko". In: *Journal of Heredity* 68.1 (1977), p. 57.

could be unified in a mathematically elegant manner. And perhaps the problems of complexity should be smaller when applying evolution to discrete approximate inference than to more general tasks involving language or theorem-proving in a higher-order logic, where a system could conceivably have the power to exploit arbitrary symmetries in the input problems or even to reason about its own behaviour. If we restrict our attention to the more concrete discrete approximate inference framework, we think it should be possible to identify a reasonable, circumscribed set of rules and principles by which evolutionary optimisation could be conducted in a practical yet general manner. The limited framework would be a useful target for research into evolutionary systems which, after having been validated in the simpler setting of discrete approximate inference, could then be used as a foundation for more ambitious projects.

6.7 Conclusion

This final chapter concludes our investigation into the subject of combining approximate inference algorithms. We surveyed the results and insights of previous chapters, and argued from first principles that the final outcome of our approximate inference philosophy should be the construction of approximate inference algorithms based on a kind of artificial evolution. We found that it was practicable to harness the conditional game of chapter 4 within a simple Genetic Algorithms framework in optimising the quality of an approximation. Although we were not able to produce a good inference algorithm using this approach, we were not very surprised to discover that our promising yet elusive objective did not yield to a simple application of the traditional framework, and we drew valuable lessons from the failed attempt. We regretted having to leave the rest of this task to future researchers, but we could not resist elaborating our thoughts about some directions in which one might try to look for a better solution. As is common in the related field of Genetic Algorithms, we found ourselves obligated to draw some comparisons to nature in order to explain and justify our philosophy for tackling this complex subject. We considered the possibility that, just as we must understand our own minds in order to understand artificial intelligence, before we can really understand simulated evolution we must try to understand biology. This outlook motivated us to delve into the interaction between pathogenic microparasites and biodiversity in evolution, and even to examine the relationship between the evolution of various multicellular organisms and the processes of somatic adaptation in individual

organisms. We hope that it is possible to develop and refine these notions in future work, and eventually to produce a practical inference algorithm based on evolutionary principles.

6.8 Acknowledgements

The author would like to thank Victor Carlsen for explaining the ins and outs of professional football.

Chapter 7

Summary

In this thesis we have explored different ways of combining approximate inference algorithms. In the introduction we defined approximate inference and argued that because it is not always feasible to hand-craft an approximation method by validating it with data samples or by some other metric, more attention should be paid to the setting where approximate inference must be evaluated using internal, rather than external criteria - what we called the "pure approximate inference" setting. We conjectured that new advances in approximate inference will depend on new ways of combining approximate inference algorithms with each other, and we suggested four forms which such combination might take:

1. *Subdividing an inference problem among multiple approximate algorithms and combining the results.* In chapter 3, we looked at performing inference by divide-and-conquer. The inference algorithm we used was BP, but other message passing algorithms, such as GBP, could be harnessed in the same way. By applying back-propagation to the message updates we derived a simple heuristic for choosing the condition variable by which to partition a model. We found that the results were competitive with the time and accuracy of existing algorithms.

2. *Comparing the accuracy of two approximations.* Chapter 4 proposed a "conditional game" which can be used to compare the accuracy of two approximations. We showed that this game can correctly distinguish between five standard inference algorithms by accuracy on a standard example graph. We also used it to rank GBP approximations parametrised by different region configurations, via a single-elimination tournament, and we saw that the conditional game out-

performed another simple game, the code-length game. Applications of the conditional game were further explored in the next two chapters.

3. *Transferring information from one approximation to another.* Chapter 5 looked at a protocol which we called "guided inference", in which one inference algorithm attempts to teach another by giving it a series of example states at which to evaluate the model's unnormalised joint distribution. We implemented several methods of selecting these states and found that the best performance was given by the method in which the teacher selects states by playing the conditional game with the student.

4. *Optimising over the space of approximations.* Chapter 6 explored the use of the conditional game as a way of selecting mates in a simulated evolution experiment, similar to Genetic Algorithms. In our experiment, the accuracy of members of the population progressed at a rate of about $1/\sqrt[3]{n}$. We found that evolution was eventually able to outperform the (cooperation-less) single-elimination tournament at producing good approximations.

The conditional game (the second of the projects listed above) solved an important open problem concerning how we should evaluate the relative performance of our inference algorithms on large graphs. All four projects make valuable contributions to the study of approximate inference. We hope that this research provides a structure upon which future research into approximate inference could be built, and as a collection of new problems and of directions which merit further investigation.

Acknowledgements

I wish to thank my supervisor, Zoubin Ghahramani, for providing genuine and conscientious feedback at every stage of my investigations, and for cheerfully drawing upon his own valuable experience to enrich my understanding of Machine Learning.

I wish to thank my parents for being supportive and understanding during my six years as a graduate student.

I wish to thank Joris Mooij for his libDAI software without which this research would have been impossible; and Joris Mooij, Iain Murray, David Duvenaud, and Victor Carlsen for useful discussions relating to some of the chapters. Also, Nathan Bowler, James DeMeo, Drew Frank, Kenji Fukumizu, Inmar Givoni, Alex Ihler, Bert Kappen, Bill Redfearn, Chung-chieh Shan, Martin Szummer, Yee-Whye Teh, Max Welling, and members of the Computational and Biological Learning laboratory at the University of Cambridge deserve my gratitude for miscellaneous feedback and insights. Finally, I wish to thank the many friends and colleagues, and especially my examiners Sean Holden and Tom Heskes, who took the time to read and comment on drafts of this thesis.

During my five years in Cambridge I was supported by grants from Microsoft Research, Trinity Hall, the Cambridge Overseas Trust, and from the British government through an Overseas Research Studentship.

Glossary

agreement rate The rate, according to a given error metric, at which the Conditional Game correctly identifies the approximation with smallest error (our terminology), 86

AI Artificial Intelligence, 1

anytime algorithm An algorithm which can return an approximate solution to a problem even if it is stopped before it has found an exact solution., 108

applied approximate inference The application of approximate inference to problems where a model is learned from data (our terminology), 10

BBP Back-belief-propagation, an application of reverse-mode automatic differentiation (i.e. back-propagation) to the Belief Propagation messages and beliefs (our terminology), 55

Bethe Free Energy A free energy function whose stable points are fixpoints of Belief Propagation, 23

BP Belief Propagation, a simple approximate inference algorithm, which uses message passing and is exact on trees, 19

CBP Conditioned Belief Propagation - the application of Belief Propagation to a model which has been partitioned into smaller models through variable conditioning (our terminology), 54

CCCP The Convex Concave Procedure, a convergent double-loop algorithm for finding local minima of the Bethe Free Energy, which is derived by writing the energy as a difference of two convex functions, 17

CG The Conditional Game, a two-player game for comparing approximations (our terminology), 76

CP Conditional Player - in the CG, the player who conditions variables (our terminology), 76

crossover An operator used in Genetic Algorithms which combines two parent genotypes to produce an offspring genotype, 113

EP Expectation propagation, an approximate inference algorithm which is based on a generalisation of "assumed-density filtering", 17

factor A function which is used to define interactions between some subset of variables in a factor graph, 15

factor graph A way of specifying a statistical model as a product of local functions of the variables, or "factors", 15

FBP Fractional Belief Propagation, an approximate inference algorithm which generalises BP. It can be derived by introducing fractional over-counting numbers into the Bethe Free Energy and minimising the resulting expression, 17

fitness function A way of measuring the fitness of individuals in an evolutionary simulation based on Genetic Algorithms, 112

fitness proportional selection A selection strategy used in Genetic Algorithms which chooses an individual with a probability proportional to his fitness, 113

GA Genetic Algorithms is a way of solving optimisation problems using evolutionary simulations which apply crossover and mutation operators to produce new individuals, and use a problem-specific fitness function for selection, 111

GBP Generalised Belief Propagation, an algorithm which passes messages to find local minima of the Kikuchi Free Energy, 19

Gibbs sampling The simplest MCMC method, 25

guided inference An protocol for sharing information between two approximate inference algorithms (our terminology), 92

HAK Heskes-Albers-Kappen, a convergent double-loop algorithm for finding local minima of the Bethe Free Energy, 84

IAC The inheritance of acquired characteristics, a kind of evolution in which somatic adaptations are heritable, 149

idiotype The idiotype of an antibody refers to those aspects of its shape not related to antigenic specificity, 153

Kikuchi Free Energy A free energy function whose stable points are fixpoints of Generalised Belief Propagation, 24

Lamarckism Another name for IAC, 149

lymphocyte A type of cell employed by the immune system, which involved in producing antibodies and recognising antigens, 115

MCMC Markov Chain Monte Carlo, a class of approximate inference algorithms which averages samples produced using a Markov Chain, 18

MF Mean field, an approximate inference algorithm which derives message updates from the assumption that every variable in a model is independent, 17

microparasite A parasitic microorganism, 138

Modern Evolutionary Synthesis The dominant view of evolution, based on natural selection and Mendelian inheritance, 149

MP Marginal Player - in the CG, the player who proposes marginals (our terminology), 76

mutation An operator used in Genetic Algorithms which applies random mutations to a genotype, 112

NP The class of decision problems which are polynomial-time soluble on an NTM, 44

NP-complete A problem is NP-complete if it is in NP and any other problem in NP can be reduced to it in polynomial time, 44

NP-hard A problem is NP-hard if problems in NP are polynomial time reducible to it, 45

NTM A TM which is allowed to make "non-deterministic" choices - the input is accepted if some unspecified set of choices leads to an "accept state", 44

P The class of decision problems which are soluble on a TM in time bounded by a *polynomial* function of the input length, 44

PA partial assignment: an assignment of values to a subset of a model's variables (our terminology), 30

Pangenesis A mechanism of adaptation proposed by Darwin to explain the inheritance of acquired characteristics, 149

phenotype The external characteristics of an organism, a consequence of its genotype and the environment, 112

pure approximate inference The application of approximate inference to problems where a model is fully specified without using data, for instance from rules or axioms (our terminology), 10

relative fitness A fitness function which provides a relative comparison of two individuals, 114

retrovirus A kind of virus which has the ability to integrate a genetic payload into a host cell's genome, 150

reverse-mode automatic differentiation A method for computing the gradient of a function in time proportional to evaluating it (also known as backpropagation), 56

SET Single-elimination tournament, a protocol for choosing a winner from a set of contestants by arranging competitions between pairs of them, in which contestants are retired as soon as they lose a game, 87

SHM Somatic Hypermutation - a mechanism by which the immune system learns to recognise antigens, in which lymphocytes divide, survive or perish, and undergo mutations in their antibody-encoding genes, 152

simulated evolution The use of algorithms for solving difficult yet parallelisable optimisation problems using cooperation and competition, perhaps based on analogies to the principles of biological evolution, natural selection, and adaptation (our terminology), 111

TM Turing Machine: an idealised computer, 44

tournament selection A selection strategy used in Genetic Algorithms which compares individuals chosen at random from the population, 113

TRW-BP Tree Reweighted Belief Propagation, 17

UAI Uncertainty in Artificial Intelligence, a machine learning conference which holds an approximate inference competition, 76

XOR Exclusive or, a function of some number of bits which outputs whether an even number of its arguments are set, 35

Bibliography

Agar, WE et al. "Fourth (final) report on a test of McDougall's Lamarckian experiment on the training of rats". In: *Journal of Experimental Biology* 31 (1954), pp. 307–321.

Amasino, R. "Vernalization, competence, and the epigenetic memory of winter". In: *The Plant Cell Online* 16.10 (2004), p. 2553.

Asprem, E. "A nice arrangement of heterodoxies: William Mcdougall and the professionalization of psychical research". In: *Journal of the History of the Behavioral Sciences* 46.2 (2010), pp. 123–143.

Bäck, T. "Optimal mutation rates in genetic search". In: *Proceedings of the Fifth International Conference on Genetic Algorithms*. 1993, pp. 2–8.

Barahona, F. "On the computational complexity of Ising spin glass models". In: *Journal of Physics A: Mathematical and General* 15 (1982), p. 3241.

Baum, E.B. "Toward a model of intelligence as an economy of agents". In: *Machine Learning* 35.2 (1999), pp. 155–185.

Belshaw, R. et al. "Long-term reinfection of the human genome by endogenous retroviruses". In: *Proceedings of the National Academy of Sciences of the United States of America* 101.14 (2004), p. 4894.

Bethe, H.A. "Statistical Theory of Superlattices". In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 150.871 (1935), pp. 552–575.

Bilmes, J. *UAI06 Inference Evaluation Results*. Department of Electrical Engineering, University of Washington, Seattle. 2006.

Blanden, Robert V. et al. "The signature of somatic hypermutation appears to be written into the germline IgV segment repertoire". In: *Immunological Reviews* 162.1 (1998), pp. 117–132.

Booker, L.B. "Improving the performance of genetic algorithms in classifier systems". In: *Proceedings of the 1st International Conference on Genetic Algorithms*. 1985, pp. 80–92.

Braunstein, A., M. Mezard, and R. Zecchina. "Survey propagation: an algorithm for satisfiability". In: *Random Structures and Algorithms* 27.2 (2005), pp. 201–226.

Braunstein, A. and R. Zecchina. "Survey propagation as local equilibrium equations". In: *Journal of Statistical Mechanics: Theory and Experiment* (2004).

Bulmer, M.G. *Francis Galton: pioneer of heredity and biometry*. Johns Hopkins University Press, 2003.

Burdon, J.J. *Diseases and plant population biology*. Cambridge University Press, 1987.

Chertkov, M., V. Gomez, and H. Kappen. *Approximate inference on planar graphs using loop calculus and belief progagation*. Tech. rep. Los Alamos National Laboratory (LANL), 2009.

Cohen, B.M. "The demise of Lysenko". In: *Journal of Heredity* 68.1 (1977), p. 57.

— "The descent of Lysenko". In: *Journal of Heredity* 56.5 (1965), p. 229.

Collins, F.S. et al. "New goals for the US human genome project: 1998-2003". In: *Science* 282.5389 (1998), p. 682.

Cook, S.A. "The complexity of theorem-proving procedures". In: *Proceedings of the third annual ACM symposium on Theory of computing*. ACM. 1971, pp. 151–158.

Cooper, G.F. "The computational complexity of probabilistic inference using Bayesian belief networks". In: *Artificial intelligence* 42.2-3 (1990), pp. 393–405.

Cooper-Willis, C.A. et al. "Influence of paternal immunity on idiotype expression in offspring". In: *Immunogenetics* 21.1 (1985), pp. 1–10.

Crowell, R. and A. Kiessling. "Endogenous retrovirus expression in testis and epididymis". In: *Biochemical Society Transactions* 35 (2007), pp. 629–633.

Cunningham, A. and O.P. Grell. *The Four Horsemen of the Apocalypse: religion, war, famine, and death in Reformation Europe*. Cambridge University Press, 2000.

Dagum, P. and M. Luby. "Approximate probabilistic reasoning in Bayesian belief networks is NP-hard". In: *Artificial Intelligence* 60 (1993), pp. 141–153.

Darwiche, A. "A Differential Approach to Inference in Bayesian Networks". In: *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*. 2000, pp. 123–132.

Darwiche, A. "Conditioning methods for exact and approximate inference in causal networks". In: *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence*. 1995.

— "Inference in Bayesian Networks: A Historical Perspective". In: *Heuristics, Probability and Causality. A Tribute to Judea Pearl*. College Publications,

— "Recursive conditioning". In: *Artificial Intelligence* 126.1-2 (2001), pp. 5–41.

Darwin, C. *The Descent of Man, and Selection in Relation to Sex*. Murray, 1871.

— *The variation of animals and plants under domestication*. Murray, 1868.

Davis, M., G. Logemann, and D. Loveland. "A machine program for theorem-proving". In: *Communications of the ACM* 5.7 (1962), pp. 394–397.

Davis, M. and H. Putnam. "A computing procedure for quantification theory". In: *Journal of the ACM (JACM)* 7.3 (1960), pp. 201–215.

Dawkins, R. *The selfish gene*. Paladin, London, 1976.

De Finetti, B. "Probabilism: A critical essay on the theory of probability and on the value of science". In: *Erkenntnis* 31.2 (1989), pp. 169–223.

De Jong, K.A. *Analysis of the Behavior of a Class of Genetic Adaptive Systems*. 1975.

Dempster, A.P., N.M. Laird, and D.B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39.1 (1977), pp. 1–38.

Domke, Justin. "Implicit Differentiation by Perturbation". In: *Advances in Neural Information Processing Systems 23*. 2010, pp. 523–531.

Duane, AD et al. "Hybrid monte carlo". In: *Physics letters B* 195.2 (1987), pp. 216–222.

Dunlap, K.A. et al. "Endogenous retroviruses regulate periimplantation placental growth and differentiation". In: *Proceedings of the National Academy of Sciences* 103.39 (2006), p. 14390.

Eaton, F. and Z. Ghahramani. "Choosing a variable to clamp: approximate inference using conditioned belief propagation". In: *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*. Vol. 5. 2009, pp. 145–152.

Eaton, F. "A conditional game for comparing approximations". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Vol. 15. 2011.

Echenberg, M.J. *Black death, white medicine: bubonic plague and the politics of public health in Colonial Senegal, 1914-1945*. James Currey, 2002.

Elidan, G., I. McGraw, and D. Koller. "Residual belief propagation: Informed scheduling for asynchronous message passing". In: *Proceedings of the Twenty-second Conference on Uncertainty in AI (UAI), Boston, Massachussetts*. Vol. 6. 6.4. 2006, pp. 6–4.

Fanning, L.J., A.M. Connor, and G.E. Wu. "Development of the immunoglobulin repertoire". In: *Clinical immunology and immunopathology* 79.1 (1996), pp. 1–14.

Feder, T. "Network flow and 2-satisfiability". In: *Algorithmica* 11.3 (1994), pp. 291–319. ISSN: 0178-4617.

Fisher, M.E. "On the dimer solution of planar Ising models". In: *Journal of Mathematical Physics* 7 (1966), p. 1776.

Flegr, J. *Evolutionary biology, 2nd edition*. Academia Prague, 2009.

Fortnow, L. "The status of the P versus NP problem". In: *Communications of the ACM* 52.9 (2009), pp. 78–86.

Freund, Y. "Boosting a weak learning algorithm by majority". In: *Information and computation* 121.2 (1995), pp. 256–285.

Gallager, RG. *Low Density Parity Check Codes. Number 21 in Research monograph series*. 1963.

Garrison, K.E. et al. "T cell responses to human endogenous retroviruses in HIV-1 infection". In: *PLoS Pathog* 3.11 (2007), e165.

Gelfand, A.E. and A.F.M. Smith. "Sampling-Based Approaches to Calculating Marginal Densities". In: *Journal of the American Statistical Association* 85.410 (1990), pp. 398–409.

Geman, S. and D. Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images". In: *IEEE transactions on pattern analysis and machine intelligence* 6.6 (1984), pp. 721–741.

Geweke, J. "Bayesian Inference in Econometric Models Using Monte Carlo Integration". In: *Econometrica* 57.6 (1989), p. 1317.

Globerson, A. and T.S. Jaakkola. "Approximate inference using planar graph decomposition". In: *Advances in Neural Information Processing Systems 19* 19 (2007), p. 473.

Goldberg, D.E. and J. Richardson. "Genetic algorithms with sharing for multimodal function optimization". In: *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*. L. Erlbaum Associates Inc. 1987, pp. 41–49.

Griewank, A. "On automatic differentiation". In: *Mathematical Programming: Recent Developments and Applications* (1989), pp. 83–108.

Grosso, P.B. "Computer simulations of genetic adaptation: Parallel subcomponent interaction in a multilocus model." In: *Dissertation Abstracts International Part B: Science and Engineering* 46.7 (1986).

Guyer, MF and EA Smith. "Studies on cytolysins. I. Some prenatal effects of lens antibodies". In: *Journal of Experimental Zoology* 26.1 (1918), pp. 65–82.

— "Studies on cytolysins. II. Transmission of induced eye-defects". In: *Journal of Experimental Zoology* 31.2 (1920), pp. 171–223.

Haldane, J. B. S. "Disease and evolution". In: *Ric. Sci. Suppl. A* 19 (1949), pp. 68–76.

Harik, G.R. and F.G. Lobo. "A parameter-less genetic algorithm". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. Vol. 1. 1999, pp. 258–265.

Hastings, W.K. "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika* 57.1 (1970), p. 97.

Heskes, T. "Stable fixed points of loopy belief propagation are minima of the Bethe free energy". In: *Advances in Neural Information Processing Systems 15*. MIT Press. 2003, p. 359.

Heskes, T., K. Albers, and B. Kappen. "Approximate inference and constrained optimization". In: *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence*. Vol. 13. 2003, pp. 313–320.

Heskes, T. et al. "Approximate inference techniques with expectation constraints". In: *Journal of Statistical Mechanics: Theory and Experiment* 2005 (2005), P11015.

Heskes, Tom. "Convexity arguments for efficient minimization of the Bethe and Kikuchi free energies". In: *Journal of Artificial Intelligence Research* 26 (2006), pp. 153–190.

Hinton, G.E. and S.J. Nowlan. "How learning can guide evolution". In: *Complex systems* 1.1 (1987), pp. 495–502.

Holland, J.H. "Adaptation in natural and artificial systems". In: *Ann Arbor MI: University of Michigan Press* (1975).

Horvitz, E.J., H.J. Suermondt, and G.F. Cooper. *Bounded conditioning: Flexible inference for decisions under scarce resources*. Tech. rep. Stanford University. Medical Computer Science. Knowledge Systems Laboratory, 1990.

Hutter, M. "The fastest and shortest algorithm for all well-defined problems". In: *International Journal of Foundations of Computer Science* 13.3 (2002), pp. 431–443.

Impagliazzo, R. and R. Paturi. "Complexity of k-SAT". In: *Computational Complexity, 1999. Proceedings. Fourteenth Annual IEEE Conference on*. IEEE. 2002, pp. 237–240.

Jaynes, E.T. and G.L. Bretthorst. *Probability theory: the logic of science*. Cambridge University Press, 2003.

Jensen, F.V., K.G. Olesen, and S.K. Andersen. "An algebra of Bayesian belief universes for knowledge-based systems". In: *Networks* 20.5 (1990).

Jerne, N.K. "Towards a network theory of the immune system." In: *Annales d'immunologie*. Vol. 125. 1-2. 1974, p. 373.

Jordan, M.I. et al. "An introduction to variational methods for graphical models". In: *Machine learning* 37.2 (1999), pp. 183–233.

Jung, K. and D. Shah. "Inference in binary pair-wise markov random field through self-avoiding walk". In: (2006).

Kasteleyn, P.W. "Dimer statistics and phase transitions". In: *Journal of Mathematical Physics* 4 (1963), p. 287.

Kearns, M., M. Littman, and S. Singh. "Graphical models for game theory". In: *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*. 2001, pp. 253–260.

Keesing, F. et al. "Impacts of biodiversity on the emergence and transmission of infectious diseases". In: *Nature* 468.7324 (2010), pp. 647–652.

Kiessling, A.A., R. Crowell, and C. Fox. "Epididymis is a principal site of retrovirus expression in the mouse". In: *Proceedings of the National Academy of Sciences of the United States of America* 86.13 (1989), p. 5109.

Kikuchi, R. "A Theory of Cooperative Phenomena". In: *Physical Review* 81.6 (1951), pp. 988–1003.

Kilbourne, E.D. "Influenza pandemics of the 20th century". In: *Emerging infectious diseases* 12.1 (2006), p. 9.

Kim, J.H. and J. Pearl. "A computational model for causal and diagnostic reasoning in inference systems". In: *Proceedings of the 8th International Joint Conference on Artificial Intelligence*. 1983, pp. 190–193.

Kimura, K., K. Taki, and S.S.K.G.K. Kikō. *Time-homogeneous parallel annealing algorithm*. Institute for New Generation Computer Technology, 1991.

Klaua, D. "Über einen Ansatz zur mehrwertigen Mengenlehre". In: *Monatsberichte der Deutschen Akademie der Wissenschaften Berlin* 7 (1965), pp. 859–867.

Koestler, A. *The case of the midwife toad*. Hutchinson (London), 1971.

Kschischang, F.R., B.J. Frey, and H.A. Loeliger. "Factor graphs and the sum-product algorithm". In: *IEEE Transactions on information theory* 47.2 (2001), pp. 498–519.

Kuhn, T.S. *The structure of scientific revolutions*. University of Chicago press, 1970.

Langford, C. "The age pattern of mortality in the 1918-19 influenza pandemic: an attempted explanation based on data for England and Wales." In: *Medical history* 46.1 (2002), p. 1.

Lardeux, F. and A. Goëffon. "A Dynamic Island-Based Genetic Algorithms Framework". In: *Simulated Evolution and Learning* (2010), pp. 156–165.

Lauritzen, S.L. and D.J. Spiegelhalter. "Local computations with probabilities on graphical structures and their application to expert systems". In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1988), pp. 157–224.

Lorenzen, P. and K. Lorenz. *Dialogische logik*. Wissenschaftliche Buchgesellschaft Darmstadt, Germany, 1978.

Lowd, D. and P. Domingos. "Approximate Inference by Compilation to Arithmetic Circuits". In: *Advances in Neural Information Processing Systems 23*. 2010, pp. 1477–1485.

Mansinghka, V. et al. "Exact and Approximate Sampling by Systematic Stochastic Search". In: 5 (2009).

McDougall, W. "An experiment for the testing of the hypothesis of Lamarck". In: *Brit. J. Psychol.* 17 (1927), pp. 267–304.

McNeil, W. *Plagues and people*. Jeffrey Norton, 1975.

Meteopolis, N. and S. Ulam. "The monte carlo method". In: *Journal of the American Statistical Association* 44.247 (1949), pp. 335–341.

Metropolis, N. et al. "Equation of state calculations by fast computing machines". In: *The journal of chemical physics* 21.6 (1953), p. 1087.

Mézard, M. and G. Parisi. "Mean-field equations for the matching and the travelling salesman problems". In: *EPL (Europhysics Letters)* 2 (1986), p. 913.

— "Mean-field theory of randomly frustrated systems with finite connectivity". In: *EPL (Europhysics Letters)* 3 (1987), p. 1067.

Mézard, M. and R. Zecchina. "Random K-satisfiability problem: From an analytic solution to an efficient algorithm". In: *Physical Review E* 66.5 (2002), p. 56126.

Mi, S. et al. "Syncytin is a captive retroviral envelope protein involved in human placental morphogenesis". In: *Nature* 403.6771 (2000), pp. 785–789.

Minka, T. "Divergence measures and message passing". In: *Microsoft Research, Cambridge, UK, Tech. Rep. MSR-TR-2005-173* (2005).

Minka, T. and Y. Qi. "Tree-structured approximations by expectation propagation". In: *Advances in Neural Information Processing Systems 16*. 2004, p. 193.

Minka, T. and J. Winn. "Gates: A graphical notation for mixture models". In: (2008).

Minka, T.P. "Expectation propagation for approximate Bayesian inference". In: *Uncertainty in Artificial Intelligence*. Vol. 17. 2001, pp. 362–369.

— "Expectation propagation for approximate Bayesian inference". In: *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence*. Vol. 17. 2001, pp. 362–369.

Mitchell, M. *An introduction to genetic algorithms*. MIT press, 1998.

Modrow, S. et al. "Computer-assisted analysis of envelope protein sequences of seven human immunodeficiency virus isolates: prediction of antigenic epitopes in conserved and variable regions." In: *Journal of Virology* 61.2 (1987), p. 570.

Montanari, A. and T. Rizzo. "How to compute loop corrections to the Bethe approximation". In: *Journal of Statistical Mechanics: Theory and Experiment* 10 (2005), P10011.

Mooij, J.M. *libDAI 0.2.2: A free/open source C++ library for Discrete Approximate Inference methods*. `http://mloss.org/software/view/77/`. 2008.

Mooij, JM. "Understanding and improving belief propagation". PhD thesis. Radboud Universiteit Nijmegen, 2008.

Mooij, J.M. and H.J. Kappen. "Bounds on marginal probability distributions". In: *Advances in Neural Information Processing Systems 21*. 2009, pp. 1105–1112.

Mooij, JM and HJ Kappen. "On the properties of the Bethe approximation and loopy belief propagation on binary networks". In: *Journal of Statistical Mechanics: Theory and Experiment* 2005 (2005), P11012.

Mooij, J.M. et al. *libDAI 0.2.5: A free/open source C++ library for Discrete Approximate Inference*. http://www.libdai.org/. 2010.

Mooij, J.M. et al. "Loop corrected belief propagation". In: *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*. 2007.

Nakanishi, K. "Two- and three-spin cluster theory of spin-glasses". In: *Physical Review B* 23.7 (1981), pp. 3514–3522.

Neal, R.M. and University of Toronto. Department of Computer Science. *Probabilistic inference using Markov chain Monte Carlo methods*. 1993.

Olson, J.C. and G.A. Leslie. "Inheritance patterns of idiotype expression: maternal-fetal immune regulatory networks". In: *Immunogenetics* 13.1 (1981), pp. 39–56.

Opper, M. and O. Winther. "Expectation consistent approximate inference". In: *The Journal of Machine Learning Research* 6 (2005), pp. 2177–2204.

Paredis, J. "Coevolution, memory and balance". In: *International Joint Conference on Artificial Intelligence*. Vol. 16. 1999, pp. 1212–1217.

Patterson, K.D. *Pandemic influenza, 1700-1900: A study in historical epidemiology*. Rowman & Littlefield Totowa, NJ, USA: 1986.

Pearl, J. *Causality: models, reasoning, and inference*. Cambridge University Press, 2000.

— "Evidential reasoning using stochastic simulation of causal models". In: *Artificial Intelligence* 32.2 (1987), pp. 245–257.

— "Fusion, propagation, and structuring in belief networks". In: *Artificial intelligence* 29.3 (1986), pp. 241–288.

— *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

— "Reverend Bayes on inference engines: A distributed hierarchical approach". In: *Proceedings of the AAAI National Conference on AI*. 1982, pp. 133–136.

Peierls, R. "On Ising's model of ferromagnetism". In: *Mathematical Proceedings of the Cambridge Philosophical Society*. Vol. 32. 03. Cambridge University Press. 1936, pp. 477–481.

Pelizzola, A. "Cluster variation method in statistical physics and probabilistic graphical models". In: *J. Phys. A: Math. Gen* 38 (2005), R309–R339.

Peterson, C. and J.R. Anderson. "A mean field theory learning algorithm for neural networks". In: *Complex systems* 1.5 (1987), pp. 995–1019.

Pittoggi, C. et al. "Generation of biologically active retro-genes upon interaction of mouse spermatozoa with exogenous DNA". In: *Molecular reproduction and development* 73.10 (2006), pp. 1239–1246.

Pólya, G. *Mathematics and Plausible Reasoning: Patterns of plausible inference*. Princeton University Press, 1954.

Propp, J.G. and D.B. Wilson. "Exact sampling with coupled Markov chains and applications to statistical mechanics". In: *Random structures and Algorithms* 9.1-2 (1996), pp. 223–252.

Pryor, EG. "The great plague of Hong Kong". In: *J Hong Kong Branch R Asiat Soc* 15 (1975), pp. 61–70.

Reich, W. "The discovery of the orgone. Vol. I. The function of the orgasm; sex-economic problems of biological energy." In: (1942).

Richardson, T. "Markov properties for acyclic directed mixed graphs". In: *Scandinavian Journal of Statistics* 30.1 (2003), pp. 145–157.

Richardson, T.S. "A factorization criterion for acyclic directed mixed graphs". In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press. 2009, pp. 462–470.

Rosin, C.D. "Coevolutionary Search Among Adversaries". PhD thesis. University of California, San Diego, 1997.

Ross, B.J. "A Lamarckian Evolution Strategy for Genetic Algorithms". In: (1999).

Roth, D. "On the hardness of approximate reasoning". In: *Artificial Intelligence* 82.1-2 (1996), pp. 273–302.

Rous, P. "A sarcoma of the fowl transmissible by an agent separable from the tumor cells". In: *The journal of experimental medicine* 13.4 (1911), p. 397.

Salo, W.L. et al. "Identification of Mycobacterium tuberculosis DNA in a pre-Columbian Peruvian mummy". In: *Proceedings of the National Academy of Sciences of the United States of America* 91.6 (1994), p. 2091.

Schmitt, L.M. "Theory of genetic algorithms". In: *Theoretical Computer Science* 259.1-2 (2001), pp. 1–61.

Selman, B., H. Kautz, and B. Cohen. "Local search strategies for satisfiability testing". In: (1993).

Selman, B., H. Levesque, and D. Mitchell. "A new method for solving hard satisfiability problems". In: *Proceedings of the tenth national conference on artificial intelligence*. 1992, pp. 440–446.

Shekhar, S. "Fixing and Extending the Multiplicative Approximation Scheme". MA thesis. University of California, Irvine, 2009.

Slavov, V. and N.I. Nikolaev. "Immune network dynamics for inductive problem solving". In: *Parallel Problem Solving Nature*. Springer. 1998, p. 712.

Smith, D.J. et al. "Mapping the antigenic and genetic evolution of influenza virus". In: *Science* 305.5682 (2004), p. 371.

Steele, E.J. "DNA polymerase-eta as a reverse transcriptase: implications for mechanisms of hypermutation in innate anti-retroviral defences and antibody SHM systems." In: *DNA repair* 3.7 (2004), p. 687.

— "Mechanism of somatic hypermutation: critical analysis of strand biased mutation signatures at A: T and G: C base pairs". In: *Molecular immunology* 46.3 (2009), pp. 305–320.

— *Somatic selection and adaptive evolution: on the inheritance of acquired characters*. University of Chicago Press, 1981.

Steele, E.J. and J.W. Pollard. "Hypothesis: Somatic hypermutation by gene conversion via the error prone DNA→RNA→DNA information loop". In: *Molecular Immunology* 24.6 (1987), pp. 667 –673.

Steele, E.J. et al. "Computational analyses show A-to-G mutations correlate with nascent mRNA hairpins at somatic hypermutation hotspots". In: *DNA repair* 5.11 (2006), pp. 1346–1363.

Sudderth, Erik, Martin Wainwright, and Alan Willsky. "Loop Series and Bethe Variational Bounds in Attractive Graphical Models". In: *Advances in Neural Information Processing Systems 20*. Cambridge, MA: MIT Press, 2008, pp. 1425–1432.

Sutton, C. and A. McCallum. "Improved dynamic schedules for belief propagation". In: *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*. 2007.

Temin, H.M. "Malignant transformation of cells by viruses." In: *Perspectives in biology and medicine* 14.1 (1970), p. 11.

— *The DNA provirus hypothesis*. Nobel lecture. 1975.

Tong, S. "Active learning: theory and applications". PhD thesis. Stanford University, 2001.

Tooze, J. *The molecular biology of tumour viruses*. Cold Spring Harbor Laboratory, 1973.

Topsøe, F. "Information theoretical optimization techniques". In: *Kybernetika* 15.1 (1979), pp. 8–27.

Turing, A.M. "Computing machinery and intelligence". In: *Mind* 59.236 (1950), pp. 433–460.

Urbain, J. et al. "Sharing of idiotypic specificities between different antibody populations from an individual rabbit". In: *European Journal of Immunology* 5.8 (1975), pp. 570–575.

Urbain-Vansanten, G. et al. "Synthesis of antibodies and immunoglobulins bearing recipient allotypic markers and donor idiotypic specificities in irradiated rabbits grafted with allogeneic cells from hyperimmune donors." In: *Annales d'immunologie*. Vol. 130. 3. 1979, p. 397.

Von Neumann, J. "Various techniques used in connection with random digits". In: *Applied Math Series* 12.36-38 (1951), p. 1.

Wainwright, Martin, Tommi Jaakkola, and Alan Willsky. "A New Class of Upper Bounds on the Log Partition Function". In: *Proceedings of the Eighteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-02)*. 2002, pp. 536–54.

Wainwright, M.J., T.S. Jaakkola, and A.S. Willsky. "Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudomoment matching". In: *Workshop on Artificial Intelligence and Statistics*. Vol. 21. 2003.

Watanabe, Y. and K. Fukumizu. "Graph zeta function in the Bethe free energy and loopy belief propagation". In: *Advances in Neural Information Processing Systems 22* (2009), pp. 2017–2025.

— "Loop series expansion with propagation diagrams". In: *Journal of Physics A: Mathematical and Theoretical* 42 (2009), p. 045001.

Weiss, Y. and W.T. Freeman. "Correctness of belief propagation in Gaussian graphical models of arbitrary topology". In: *Neural computation* 13.10 (2001), pp. 2173–2200.

— "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs". In: *Information Theory, IEEE Transactions on* 47.2 (2002), pp. 736–744.

Weitz, D. "Counting independent sets up to the tree threshold". In: *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*. ACM. 2006, pp. 140–149.

Welling, M. "On the choice of regions for generalized belief propagation". In: *Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence*. 2004, pp. 585–592.

Welling, M., T. Minka, and Y.W. Teh. "Structured region graphs: Morphing EP into GBP". In: *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence*. 2005, pp. 609–616.

Welling, M. and Y.W. Teh. "Belief optimization for binary networks: A stable alternative to loopy belief propagation". In: *Uncertainty in Artificial Intelligence* (2001).

— "Linear response algorithms for approximate inference in graphical models". In: *Neural computation* 16.1 (2004), pp. 197–221.

Whitley, D., V. Gordon, and K. Mathias. "Lamarckian evolution, the Baldwin effect and function optimization". In: *Parallel Problem Solving Nature from PPSN III* (1994), pp. 5–15.

Wiegerinck, W. and T. Heskes. "Fractional belief propagation". In: *Advances in Neural Information Processing Systems 15*. MIT Press. 2003, p. 455.

Wikler, M. et al. "Immunoregulatory role of maternal idiotypes. Ontogeny of immune networks." In: *The Journal of experimental medicine* 152.4 (1980), p. 1024.

Winn, J. and C.M. Bishop. "Variational message passing". In: *Journal of Machine Learning Research* 6.1 (2006), p. 661.

Wolpert, D.H. and W.G. Macready. "No free lunch theorems for optimization". In: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), pp. 67–82.

— *No free lunch theorems for search*. Tech. rep. SFI-TR-95-02-010. Santa Fe Institute, 1995.

Wu, T.T. and E.A. Kabat. "An analysis of the sequences of the variable regions of Bence Jones proteins and myeloma light chains and their implications for antibody complementarity". In: *The Journal of Experimental Medicine* 132.2 (1970), p. 211.

Yedidia, J.S., W.T. Freeman, and Y. Weiss. "Bethe free energy, Kikuchi approximations and belief propagation algorithms". In: *Advances in Neural Information Processing Systems 12* 13 (2000).

Yedidia, JS, WT Freeman, and Y. Weiss. "Constructing free-energy approximations and generalized belief propagation algorithms". In: *IEEE Transactions on Information Theory* 51.7 (2005), pp. 2282–2312.

Yedidia, J.S., W.T. Freeman, and Y. Weiss. "Generalized belief propagation". In: *Advances in Neural Information Processing Systems 13* (2001), pp. 689–695.

Yedidia, J.S., W.T. Freeman, and Y. Weiss. "Understanding belief propagation and its generalizations". In: *Exploring Artificial Intelligence in the New Millennium* (2003), pp. 239–236.

Yuille, A. L. and A. Rangarajan. "The Concave-Convex Procedure (CCCP)". In: *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press, 2002, pp. 1033–1040.

Yuille, A.L. and A. Rangarajan. "The concave-convex procedure". In: *Neural Computation* 15.4 (2003), pp. 915–936.

Zadeh, L.A. "Fuzzy sets". In: *Information and control* 8.3 (1965), pp. 338–353.