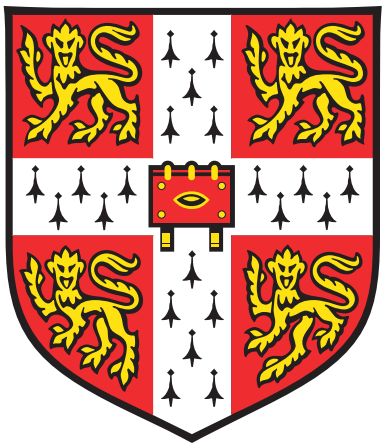
Approximate Inference in Graphical Models

Philipp Hennig

Robinson College

dissertation submitted in candidature for the degree of
Doctor of Philosophy, University of Cambridge

Inference Group
Cavendish Laboratory
University of Cambridge



14 November 2010

Declaration

I hereby declare that my dissertation entitled "Approximate Inference in Graphical Models" is not substantially the same as any that I have submitted for a degree or diploma or other qualification at any other University.
I further state that no part of my dissertation has already been or is concurrently submitted for any such degree of diploma or other qualification.
Except where specifically indicated in the text, this dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration.
This dissertation does not exceed sixty thousands words in length.

Date: 14 November 2010 Signed: _____

Philipp Hennig, Cambridge

Abstract

Probability theory provides a mathematically rigorous yet conceptually flexible calculus of uncertainty, allowing the construction of complex hierarchical models for real-world inference tasks. Unfortunately, exact inference in probabilistic models is often computationally expensive or even intractable. A close inspection in such situations often reveals that computational bottlenecks are confined to certain aspects of the model, which can be circumvented by approximations without having to sacrifice the model's interesting aspects. The conceptual framework of graphical models provides an elegant means of representing probabilistic models and deriving both exact and approximate inference algorithms in terms of local computations. This makes graphical models an ideal aid in the development of generalizable approximations. This thesis contains a brief introduction to approximate inference in graphical models (Chapter 2), followed by three extensive case studies in which approximate inference algorithms are developed for challenging applied inference problems. Chapter 3 derives the first probabilistic game tree search algorithm. Chapter 4 provides a novel expressive model for inference in psychometric questionnaires. Chapter 5 develops a model for the topics of large corpora of text documents, conditional on document metadata, with a focus on computational speed. In each case, graphical models help in two important ways: They first provide important structural insight into the problem; and then suggest practical approximations to the exact probabilistic solution.

Acknowledgments

I would like to thank all members of the Inference Group at the Cavendish Laboratory in Cambridge for their support during the preparation of this thesis. In particular, I am grateful to Carl Scheffler, Philip Sterne, Keith Vertanen, Emli-Mari Nel, Tamara Broderick, Oliver Stegle and Christian Steinrücken for engaging discussions and thoughtful comments over the past years, as well as for making this time as much fun as it has been.

I feel privileged to have had the chance to work alongside, interact with, and learn from a number of members of the extended Cambridge machine learning community. Among them are Tom Borchert, John Cunningham, Marc Deisenroth, Jürgen van Gael, Zoubin Ghahramani, Katherine Heller, Ferenc Huszár, Gergji Kasneci, David Knowles, Simon Lacoste-Julien, Máté Lengyel, Tom Minka, Shakir Mohamed, Iain Murray, Peter Orbanz, Juaquin Quiñonero-Candela, Carl Rasmussen, Yunus Saatçi, Richard Samworth, Anton Schwaighofer, Martin Szummer, Yee Whye Teh, Ryan Turner, Sinead Williamson and John Winn.

I am especially indebted to David Stern and Ralf Herbrich, both of Microsoft Research, for repeatedly and actively offering their time and expertise, often after hours, to help me with final preparations of papers and talks, and for suggesting interesting experiments and extensions.

My work was supported through a grant from Microsoft Research Ltd. Besides the financial support, I am also grateful to Microsoft for access to the great researchers and technical possibilities in its Cambridge research laboratories.

My deepest gratitude goes to my two supervisors: To David MacKay, for innumerable enlightening comments and constructive criticism as well as for being a shining example of a scientist's boldness and freedom. And to Thore Graepel, for relentlessly supporting and challenging me throughout the last three years. This thesis would not have been possible without either of them.

Contents

List of Figures

Chapter 1

Introduction

1.1 Probability

1.1.1 A Calculus of Uncertainty

The essence of human existence is constant interaction with the world, aiming to steer and shape our surroundings to our benefit. To this end, we constantly form hypotheses about the structure of the world, and the rules that govern it, act according to our theories, and then change our beliefs in the light of new experience. The process of updating one's belief about the validity of a hypothesis, and about the values of parameters describing the hypothesis, is known as *inference*, or *learning*. Biological learning systems have been very successful adapting to their environments in this way, and still outperform machines in many areas. In an attempt to replicate some of the abilities of biological systems, the discipline of *machine learning* has emerged over the past two decades, bringing together researchers from a diverse range of fields, such as computer science, engineering, physics, mathematics, neuroscience and biology, rallying around the central idea that the behaviour of a system should be governed to a large degree by examples (data), rather than explicit predetermined rules.

An unavoidable consequence of learning from finite data is incompleteness of the acquired knowledge, because some areas of the data space will not have been visited or experience might otherwise be limited. This incompleteness is associated with some level of uncertainty, and quantifying this uncertainty is essential for good decision making. Probability theory provides an ideal, mathematically rigorous framework to represent and manipulate uncertain information. In fact, any rational inference paradigm consistent with common sense (as defined by three rather incontrovertible axioms) can be mapped to probability theory [Cox, 1946]. In an extension of formal logic, where statements A are either *true* or *false*, the probabilistic framework introduces the concept of a probability $p(A) \in [0, 1]$ of

statement A being true[1] The probability of both statements A and B being true is denoted by the *joint* probability $p(A, B)$, the probability of A being true if B is *known* to be true is the *conditional* probability $p(A \mid B)$. The means to manipulate probabilities is provided by a calculus arising directly from Cox's axioms, which can be subsumed in the two central rules [Jaynes and Bretthorst, 2003]

$$p(X) = \sum_Y p(X, Y) \qquad \text{and} \qquad p(X, Y) = p(Y \mid X)p(X) \tag{1.1}$$

known as the *sum* and *product rule*, respectively. The sum symbol is meant to represent the sum over all possible values for Y. An important corollary of these results is *Bayes' theorem*

$$p(X \mid Y) = \frac{p(Y \mid X)p(X)}{\sum_X p(Y \mid X)p(X)} \tag{1.2}$$

which relates the *posterior* probability of X after observation of Y to the *prior* probability of X, the *likelihood* of X (the conditional probability of Y given X) and the *evidence* for Y under the probabilistic *model* we use to evaluate conditional probabilities.

$$\text{posterior} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}} \tag{1.3}$$

This text uses a somewhat sloppy notation to increase readability, in which the probability of a particular variable X having the concrete value x will simply be written as $p(x)$, instead of $p(X = x)$. This applies to both discrete variables and real-valued variables; in the latter case, the sums in Equations (1.1) and (1.2) have to be replaced with integrals.

The above framework, sometimes called "inferring *inverse probabilities*", is most widely known as the *Bayesian* interpretation of probability. Although named in honour of early works by a non-conformist priest [Bayes, 1763], the theory was first formally developed by Laplace [1774].[2] Cox [1946] and Kolmogorov [1933] established the fundamental nature of these rules of reasoning. Nevertheless, the Bayesian paradigm has been under fierce debate throughout the 20th century, mostly based on philosophical [see Jaynes and Bretthorst, 2003, for an opinionated review] and technical [e.g. Walker, 2004] issues involving the prior. In recent years, though, the Bayesian framework has increasingly found acceptance within the statistics and machine learning communities, mostly thanks to successful experimental demonstration of its powerful abilities in the description of complicated data and prior knowledge. Some important strengths of the Bayesian formulation

[1] In principle, probabilities can be defined as members of any space isomorphic to the real line [Jaynes and Bretthorst, 2003], but $[0, 1]$ has become the established space. Appendix C contains an example of the inconveniences caused by this choice in some cases.

[2] However, Laplace did not yet state Bayes' theorem explicitly.

are

- ▷ Bayesian methods allow quantitative statements about uncertainty. This allows reasoning about the information conveyed by data about the value of a latent variable, and about the confidence with which future data can be predicted.

- ▷ In contrast to classical statistical methods, which distinguish between random data and deterministic parameters that are not random variables, every variable in a Bayesian model can be assigned probabilities. This makes it possible to design hierarchical probabilistic models, describing very general 'hypotheses over hypotheses'.

- ▷ Because probabilistic models use probability measures to spread a finite amount of probability mass over all possible outcomes, they do not suffer from the problem of 'over-fitting' that can plague estimative methods. Bayesian methods can thus deal consistently with arbitrary, even infinite numbers of free parameters in a model (the latter case is known as Bayesian *nonparametric* inference).

- ▷ From the standpoint of classical statistics, Bayesian methods can be shown to be consistent and to converge optimally fast (subject to minor technical restrictions[3]) [Le Cam, 1973, Ibragimov and Has'minskii, 1981, Ghosal et al., 2000]. This has allowed researchers to concentrate their efforts on the development of flexible models and efficient inference methods without having to worry about theoretical guarantees.

The body of literature on the subject of probabilistic inference, although still expanding rapidly, has become too large even to be characterized by individual exemplary works. Good introductions to the applied aspects of the field can be found in the textbooks by Jaynes and Bretthorst [2003], MacKay [2003] and Bishop [2006].

[3]More precisely: Consider a dataset $\{X_i\}$ generated by a probability measure $P_\theta(\{X_i\})$ parametrized by a parameter θ from a parameter set Θ. If the prior over Θ puts nonzero mass on every sufficiently small open neighbourhood of the true value θ_0, and if Θ is a subset of a finite-dimensional Euclidean space, and the functional relationship $\theta \to P_\theta(\{X_i\})$ is sufficiently regular, then the posterior distribution on θ converges with the optimal rate [Le Cam, 1973, Ibragimov and Has'minskii, 1981]. The situation is less clear-cut when Θ is infinite dimensional [Cox, 1993, Ghosal et al., 2000]. Note that such analyses of both Bayesian and Frequentist methods assume the likelihood function to be correct, which is a causal assumption about the generative process that is often incorrect in real-world applications, such as the ones studied in this thesis.

1.2 Numerical Complexity Mandates Approximations

There is also one aspect of Bayesian modeling that is still seen as a weakness by many applied researchers, and that provides the motivation for this thesis: Although Bayes' rule provides a unique and in some sense straightforward way to do inference, obtaining the posterior for a particular choice of prior is often expensive, or even intractable. Research in Bayesian methods addresses this issue in two ways:

1. Priors that allow closed-form, or computationally cheap, evaluation of the posterior without sacrificing generality. Some Bayesian nonparametric methods fall into this category, including Gaussian process algorithms [Williams and Rasmussen, 1996] and, more recently, random processes over discrete probability spaces [Beal et al., 2002, Blei et al., 2004, Griffiths and Ghahramani, 2006, Roy and Teh, 2009].

2. Approximation methods that capture important characteristics of the posterior at tractable computational cost, such as Markov Chain Monte Carlo methods [Neal, 1996], and approximate inference methods for graphical models. The latter are the methods that this text will focus on. See Chapter 2 for an introduction and further references.

1.2.1 The Classic Answer

Classic statistical research takes a disparate approach: Noting the numerical complexity mentioned above, researchers in this field construct deterministic *estimators*, and then take care to prove several desirable characteristics of these estimators, such as the absence of *bias*, good rates of convergence, and bounds on the error between the "true" and estimated value of a parameter. Carefully constructed estimators can perform very well. But their serious drawback is a lack of generalization: Even slight changes in the structure of a problem can invalidate the derivation of an estimator and require the attention of a specialized researcher starting from scratch. Since not every applied researcher can afford the luxury of their own statistician, estimators are often marketed as "black boxes" to the applied fields, despite being usually subject to several technical restrictions. This creates a considerable risk of misuse [Jaffe and Spirer, 1987, Altman, 1982, 1994].

1.2.2 Provability Is Not Everything

From the Bayesian viewpoint, complex models are not a weakness, but a reflection of the complexity of the modeled system. The ability to capture intricate non-linear

relationships may come at the price of losing analytic mathematical formulation. This structural complexity also means that it is often very difficult to provide explicit proof of the correctness of a particular approximate algorithm, even though the underlying exact algorithms might be known to be optimal. This situation is similar to that in the natural sciences, where there are epistemological limits to the discoverability of the "correct" model [Popper, 1934, Hume, 1739], and predictions are made not just on the basis of assumptions, but also as a result of extensive approximations. Experimental evaluation has to replace mathematical proofs in many cases. In fact, modern machine learning might be seen as an extension of the effort of physics to explain the inanimate world, to ever more complex systems, and to systems dominated by human behaviour (the field is thus also sometimes termed more generally as *probabilistic data analysis*). Although provable performance is of course a virtue for any method, absence of a proof should not keep us from using approximate algorithms known empirically to perform better. In recent years several approximate algorithms with unknown general performance have become established tools of machine learning, because they have been shown to perform better than simpler algorithms with provable characteristics. In some cases (such as loopy belief propagation [Frey and MacKay, 1998] and expectation propagation [Minka, 2001]), the approximations are even known to fail in certain cases, and are still used for their superior capacity in those cases where they do not fail.

1.2.3 About This Text

This thesis presents a series of applied inference schemes. Although solving largely independent concrete problems, the chapters are linked by their use of approximate inference techniques within the framework of *graphical models*. These are a particularly expressive form of notation exposing the factorization properties of multivariate probability distributions. The framework itself was developed largely within the statistics and machine learning community, starting with seminal work by Pearl [1988], Lauritzen and Spiegelhalter [1988] and Frey [1998]. Approximations using graphical models [Frey and MacKay, 1998, Minka, 2001, Winn and Bishop, 2006] allow tractable inference in highly structured models.

This thesis makes extensive use of graphical models, less as a theoretical framework, but as a robust toolbox for the applied researcher. It utilizes graphical models to present and describe the often intricate structure of a series of probabilistic models, and to discover and study approximate solutions to inference problems where exact posterior distributions are intractable.

Chapter 2 presents a short introduction to graphical models, and to popular approximate inference methods, all of which will be used in the following chapters. Chapter 3 derives the first probabilistic game tree search algorithm, allowing the

update of probabilistic beliefs on the solution of an EXPTIME-complete problem, from individual data points, in linear time. Chapter 4 presents an approximate Bayesian inference scheme for the evaluation of psychometric questionnaires, allowing inference on individual evaluation ranges for every item in the questionnaire *and* every respondent to the questionnaire. Chapter 5 derives a lightweight inference algorithm for the popular *Latent Dirichlet Allocation* topic model for corpora of text documents, which allows the model to be conditioned on features of individual documents and which can run on very large datasets in a single pass.

Chapter 2

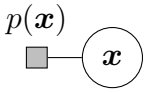
Graphical Models

At the core of probability theory lies the idea that knowledge about the values of quantities can be "spread out" over a measurable space, rather than having to be confined to a single exact value. As a direct consequence, inference in probabilistic models requires probability mass to be summed up over ranges or, in the case of continuous spaces, to be integrated. Unfortunately, integration is a tricky art, and analytical solutions to the integrals over general probability distributions are the exception rather than the rule. Hence much of applied inference relies on the use of a small set of parametric distributions amenable to integration under certain atomic operations, such as multiplication.

In multivariate models, the complexity of integration can be immense. The computational complexity of integrating, even numerically, a general multivariate probability density $p(\boldsymbol{x})$ on a variable $\boldsymbol{x} \in V^D$ in a D dimensional space $V^D = \bigotimes_D V$ rises exponentially with D, because the volume over which we need to integrate rises exponentially with D. This is known as the *curse of dimensionality* [Bellman, 1957].
However, if $p(\boldsymbol{x})$ has structure, in particular if certain dimensions of $p(\boldsymbol{x})$ are conditionally independent of certain dimensions given others, then the problem might be much easier than in the general case. In the most extreme instance, if all dimensions are independent, $p(\boldsymbol{x}) = \prod_d p(x_d)$, then the cost of integration is only $\mathcal{O}(D)$.
With a bit of luck (and foresight during model design), the univariate marginals $p(x_d)$ might also be more susceptible to analytic integration. Of course, such a fully *factorizing* model is also less interesting, as it corresponds to the assumption that none of the elements of $\boldsymbol{x}$ have any influence on each other.
Between this linear and the general exponential-cost regime lies a wide spectrum of conditional independence structures. Particularly interesting cases are hierarchical models in which the elements of $\boldsymbol{x}$ have a natural ordering (known as *directed acyclic* models). Complexity is not just determined by conditional independence

either; the actual functional form of the relationships between different elements of $\boldsymbol{x}$ also has influence on computational cost. For example, some elements of $\boldsymbol{x}$ might act as switches, controlling functional influence of other elements upon others (this is the case in mixture models). Such structure can have influence on the computational complexity of inference.

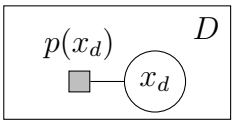
2.1 Factor Graphs



$$\boldsymbol{x} \sim p(\boldsymbol{x})$$

Figure 2.1: In *factor graph* notation, probabilistic variables are denoted by hollow circles. Functions and probability distributions are denoted by gray squares called *factors*. Edges between nodes denote membership in a functional relation.

Factor graphs are bipartite graphs providing a symbolic representation of the functional relationships between elements of multivariate probability distributions, where the term "relationship" is meant in the sense that the expression $y = f(x, z)$ defines a relationship f between the variables (x, y, z).



$$\int \prod_{d=1}^{D} p(x_d)\,\mathrm{d}x_d = \prod_{d=1}^{D} \int p(x_d)\,\mathrm{d}x_d$$

Figure 2.2: In factor graph notation, identical copies of functional relationships are denoted by a black rectangle around the copied group of variables and factors, called a *plate*. A parameter in a corner of the plate denotes the number of copies.

Figures 2.1 and 2.2 introduce the main components of factor graphs: *factors* (functional relationships between groups of nodes); *variable nodes* and *plates* (copies of groups of nodes). In addition, other graphs in this text will also contain *observed variables*, denoted by black circles. Factor graphs have three main *raisons d'être*:

- ▷ they provide an expressive graphical language which can help reveal structure in multivariate probability distributions
- ▷ often, inference algorithms can be read off mechanically from the factor graph (and in fact there exist compilers that can generate machine code directly from factor graphs [Minka and Winn, 2008])
- ▷ graph theoretical concepts (such as whether the graph is *planar*, or even a *tree*) can be used to describe certain algebraic characteristics of the underlying multivariate model and make general statements about solvability and computational complexity.

The development of a theory of graphical models in general, and factor graphs in particular, has been a community effort and the available literature is now too vast to give a full account. Seminal works were contributed by Pearl [1988], Lauritzen and Spiegelhalter [1988] and Frey et al. [1997]. More recently, interest in *approximate* inference algorithms operating on graphs led to important contributions by Frey and MacKay [1998] Minka [2001] and Winn and Bishop [2006]. Good introductions to graphical models can be found in Bishop [2006] and Frey [1998].

2.1.1 Other Graphical Models

Besides factor graphs, there are two other forms of graphical models in general use, both of which predate factor graphs historically. They are known as *directed graphical models*, or *Bayesian Networks*, and *undirected graphical models*, or *Markov Random Fields (MRFs) / Markov Networks*. In both kinds of models, there are no factor nodes. Instead, variable nodes are connected by arrows and lines, respectively, when they share functional relationships.
Undirected graphical models will not feature in this text. Directed graphical models are of particular utility in the design phase of a probabilistic model, as they are often better suited to depict motivations and convey the intuition behind probabilistic models, and consequently will be used for this purpose regularly throughout this text.

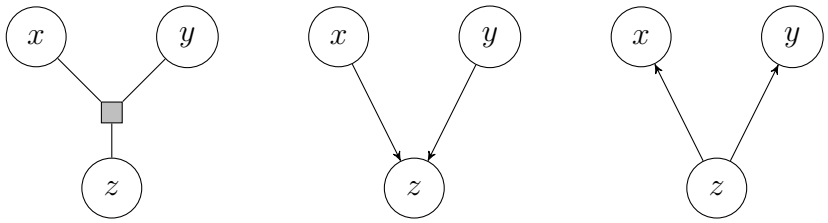


Figure 2.3: Advantages of directed graph notation: The factor graph on the left may be expressed by either of the directed graphs in the middle and on the right. However, in the directed graph in the middle, x and y are dependent conditional on z; in the graph in the right, x and y are *in*dependent conditional on z. The factor graph does not make this structure explicit.

Another advantage of directed graphical models is that they allow conditional independence structure to be read off from the graph (Figure 2.3), which is not possible in general in factor graphs. Conversely, factor graphs are more expressive than directed graphs in depicting functional relationships (Figure 2.4): Not all distributions that can be encoded as a factor graph can be encoded as a directed graphical model, and often one directed graphical model can encode different models which would be distinguishable as separate when written as a factor graph [Bishop, 2006]. Hence, algorithmic constructions will invariably be accompanied by both factor graphs and directed graphs in this thesis.

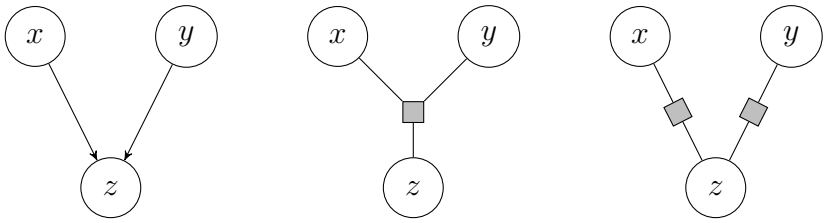


Figure 2.4: Advantages of factor graph notation: The directed model might represent either of the two factor graphs.

2.1.2 Outlook

The remainder of this chapter proceeds as follows: Section 2.2 introduces the central concept of inference in graphical models — message passing — and constructs the most important exact algorithm for inference on the marginals of nodes in graphs, the sum-product algorithm. Since this thesis focuses on approximate inference, the rest of the chapter (Section 2.3) introduces some important general approximate algorithms for inference which leverage the graphical view: Expectation Propagation (2.3.2), variational bounds (2.3.3), local Laplace approximations 2.3.4, Markov Chain Monte Carlo algorithms (2.3.5) and other less formalized methods (2.3.6). All of these methods use convenient classes of parametric probability distributions known as exponential families, introduced in Section 2.3.1.

2.2 The Sum-Product Algorithm

One of the main strengths of the factor graph representation is that inference becomes almost mechanical. This is true in particular if the graph is a *tree*, in which case there is an exact algorithm for finding the *marginal* distribution of all variables in the graph individually, known as the *sum-product* algorithm. Technical derivations can be found in Kschischang et al. [2001] and Bishop [2006]; this section gives a brief introduction.

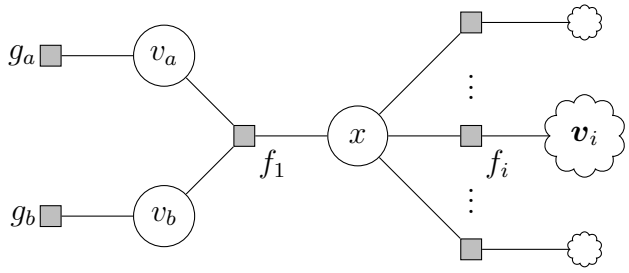


Figure 2.5: A factor graph for the derivation of the sum-product algorithm. The clouds on the right are placeholders for subgraphs.

Consider the tree-structured factor graph shown in Figure 2.5. Assume there is a potentially large number K_i of variables, jointly denoted $\boldsymbol{v}_i$, connected to the

factors f_i to the right of x, indicated by the clouds in Figure 2.5. We will assume that the conditional dependencies between these variables retain the tree structure. For notational clarity, we will denote $(v_a, v_b) = \boldsymbol{v}_1$. If we subsume the normalization into a constant Z, then the graph represents the multivariate distribution

$$p(x, v_a, v_b, \{\boldsymbol{v}_{i>1}\}) = Z^{-1} f_1(x, v_a, v_b) p_{g_a}(v_a) p_{g_b}(v_b) \prod_{i>1} f_i(x, \boldsymbol{v}_i) p_{g_i}(\boldsymbol{v}_i) \tag{2.1}$$

Assume that we are interested in the marginal on x. For a general function $f(\alpha, \beta, \gamma)$ with finite integral (i.e. any function that can be turned into a probability distribution), we have

$$\int \frac{f(\alpha, \beta, \gamma)}{\int f(\alpha, \beta, \gamma)\, \mathrm{d}\alpha\, \mathrm{d}\beta\, \mathrm{d}\gamma} \mathrm{d}\beta\, \mathrm{d}\gamma = \frac{\tilde{f}(\alpha)}{\int \tilde{f}(\alpha)\, \mathrm{d}\alpha} \tag{2.2}$$

(using an implicitly defined "marginal function" $\tilde{f}$ whose exact form is irrelevant). To get a marginal on x, we can thus work with the unnormalized joint distribution $\tilde{p}(x, \{\boldsymbol{v}_i\})$ and normalize over each variable individually only at the end. So we can write the marginal as

$$\begin{aligned} \tilde{p}(x) &= \int \cdots \int \prod_i p_{f_i}(x \mid \boldsymbol{v}_i) g_i(\boldsymbol{v}_i)\, \mathrm{d}\boldsymbol{v}_i \\ &= \prod_i \int f_i(x, \boldsymbol{v}_i) g_i(\boldsymbol{v}_i)\, \mathrm{d}\boldsymbol{v}_i \\ &= \prod_i m_i(x). \end{aligned} \tag{2.3}$$

Note that the step from the first line to the second in Equation (2.3) is only possible because of the structure of the graph. The terms of the product in the second line are functions only of x. They can thus be interpreted as *messages* $m_i(x)$, *sent to the node for* x *from the factors* f_i (Figure 2.6). This is a helpful paradigm, because it allows to speak about the process of deriving local marginals in terms of local objects: the messages.

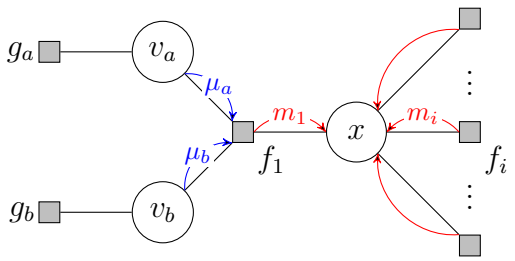


Figure 2.6: Messages m_i sent from the neighbouring factor nodes to x, and messages μ_j from variables to one of the factors.

But what is the actual form of the messages $m_i(x)$? The factor f_i is connected

to K_i variables nodes v_k^i other than x. Because the graph is a tree, these nodes are themselves connected to separate sub-graphs of their own, containing other variables $\boldsymbol{w}_k^i$, connected through a (potentially factorizing) distribution $h_k^i(v_k^i, \boldsymbol{w}_k^i)$. So we can write

$$\begin{aligned} m_i(x) &= \int \mathrm{d}v_1^i \cdots \int \mathrm{d}v_{K_i}^i f_i(x, v_1^i, \ldots, v_{K_i}^i) \prod_k^{K_i} \left[\int h_k^i(v_k^i, \boldsymbol{w}_k^i) \, \mathrm{d}\boldsymbol{w}_k^i \right] \\ &= \int \mathrm{d}v_1^i \cdots \int \mathrm{d}v_{K_i}^i f_i(x, v_1^i, \ldots, v_{K_i}^i) \prod_k^{K_i} \mu_k^i(v_k^i) \end{aligned} \tag{2.4}$$

In the second line, we have noted that the product again contains terms which are local objects, only depending on v_k^i, because all other variables are integrated out, and have denoted these terms as *messages $\mu_k^i(v_k^i)$ from the variable nodes v_k^i to the factor f_i*. In its general form, Equation (2.4) is difficult to parse, so consider specifically the example in Figure 2.6. The subgraph connected to f_1 contains the variables v_a and v_b, each of which are is connected to only one other factor, g_a and g_b. We thus have

$$m_1(x) = \int \mathrm{d}v_a \int \mathrm{d}v_b \; f_i(x, v_a, v_b) g_a(v_a) g_b(v_b) \tag{2.5}$$

and are already done; the messages from the variables are simply given by g_a and g_b. In the more general case of Equation (2.4), it might seem like evaluating the messages μ_k^i is complicated, because it involves integrating out all other variables connected to this node. However, the sub-graph connected to this node is a tree itself, containing other factors, so h_k^i separates into other terms $h_{k,1}^i, \ldots h_{k,H}^i$. This is where an inductive argument can take hold: In fact, μ_k^i is of analogous form to Equation (2.3), and the message can be constructed simply by multiplying the incoming messages from all factors connected to v_k^i, except for f_i:

$$\mu_k^i(v_k^i) = \prod_{\ell=1}^{H} m_{h_{k,\ell}^i}(v_k^i) \tag{2.6}$$

We summarize: Evaluating the marginal of any variable node in a tree-structured factor graph involves a series of localized computations represented by messages.

- ▷ Messages from variables v_i to factors f_k are formed by *multiplying* all incoming factor-to-variable messages into v_i, other than that from f_k (denote that neighbourhood set by $\mathrm{ne}(v_i) \setminus f_k$):

$$\mu_{v_i \to f_k}(v_i) = \prod_{\ell \in \mathrm{ne}(v_i) \setminus f_k} m_{f_\ell \to v_i}(v_i) \tag{2.7}$$

▷ Messages from factors f_k to variables v_i are formed by *integrating out (summing over)* all variables connected to f_k, except for v_i itself (denoted by $\mathrm{ne}(f_k) \setminus v_i$):

$$m_{f_k \to v_i}(v_i) = \prod_{h \in \mathrm{ne}(f_k) \setminus v_i} \int \mu_{v_h \to f_k}(v_h) \, \mathrm{d}v_h \tag{2.8}$$

Using this scheme, calculating marginals in factor graphs reduces to a mechanical process of summing and multiplying probability distributions. At the leaves of the graph, the induction can be anchored by defining that the messages from leaf nodes are

$$\mu_{x \to f}(x) = 1 \qquad \text{and} \qquad m_{f \to x}(x) = f(x) \tag{2.9}$$

as already used in Equation (2.5). Another appealing property is the fact that message passing to gain marginals on all variables in the graph is linear in the number of variables. To see this, choose any variable in the graph and define it to be the root of the tree. Sending messages from the leaves to the root is possible because all necessary messages are available locally. Once the root has received all messages, send messages to the leaves, which is now possible because all nodes have their necessary incoming messages. The result is a set of marginals (products of incoming messages) on all variables in the tree, where every node was involved in message passing twice.

Conditioning on Data

Equation (2.2) already established that normalization of marginals can be performed locally. So far it was assumed that all variables are described by probability measures (i.e. they are latent). In real applications, a subset of the variables will invariably have been observed, or otherwise set to some fixed value. Extending the message-passing paradigm to include this situation is straightforward from a theoretical standpoint: We simply connect all observed variables to "pin-down" factors containing Dirac δ distributions to fix values. Because $\int \delta(x - x_0) \, \mathrm{d}x = x_0$, the effect of this on the resulting algorithm is even less complicated: Instead of evaluating integrals over variables, their values are set to fixed values.

General Graphs

The sum-product algorithm is applicable only to trees. However, it is possible to extend it to general graphs by constructing a tree through combination of *cliques* of nodes (non-tree-structured subgraphs) into joint nodes of a meta-graph, known as the *junction tree*, on which the sum-product algorithm can then be run [Lauritzen and Spiegelhalter, 1988]. This scheme will be used implicitly in chapters 4 and 5 of this thesis. Unfortunately, inference within the cliques now involves high-

dimensional integrals again, and the overall computational cost will be exponential in — and hence dominated by — the size of the largest clique in the graph.

2.2.1 The Sum Product Algorithm is No Panacea

The graphical model paradigm and the sum-product algorithm leverage the conditional independence structure of multivariate distributions to find low-cost ways of evaluating marginals. Unfortunately, they do not solve all challenges of applied inference. Some problems remain:

Integration Constructing factor-to-variable messages still involves multidimensional integrations, albeit of lower dimensionality. Possible approximate approaches include representing the messages by a set of samples (Section 2.3.5), or by a "nonparametric" representation on a grid of fixed resolution, i.e. essentially a histogram. Another popular approach, which will be used widely in this thesis, is to choose the distributions involved carefully such that analytical integration is possible, or to approximate the resulting distributions with others of simpler, parametric structure (Section 2.3.2).

Conditional Dependence Even taking all factorization properties into account, many models will still retain large cliques of dependent variables which have to be represented by joint vector-valued nodes in the graph to retain tree structure. Inference involving such variables can still be subject to the curse of dimensionality. Even in the easiest case, if the relationships are linear, inference will involve solving linear systems of equations. Leaving some technical issues aside [Golub and Van Loan, 1996], solving such systems has computational cost cubic in the clique size. This can still be too slow for many real-world applications. Often, the best remaining options will then be to *construct* independent approximations (Section 2.3.3), or simply to *assume* independence in the marginals and use certain heuristics to deal with the resulting defects (Section 2.3.6).

At their core, these issues reflect the fact that factorization simplifies integration, but does not make it trivial: We are still left with integrals, and integration, both analytic and approximate, is still an art, not a mechanical process. There are no general analytic solution strategies for integrals. Applied inference remains a challenging field. The remainder of this thesis will revolve around approximate techniques, as outlined above, which can make hard inference problems tractable without sacrificing the crucial aspects of the problem itself. In effect, this entire thesis is a collection of approximations and design tricks. This is not necessarily a deficiency: Entire disciplines in other fields, e.g. condensed matter physics, could arguably be described as collections of highly developed integration tricks. The

hope is that the algorithms presented in the following sections and the remaining chapters of this thesis can convey concepts necessary for applied inference problems in general.

2.3 Approximate Inference Methods

One could think that approximation in probabilistic models is a completely unstructured field: We build a model, realize that inference is intractable, and then have to somehow pluck an ad-hoc approximation out of thin air. In fact, there are several more structured approaches available. A lot of problems can be prevented from the start by using convenient probability distributions. Distributions forming *exponential families* provide parametric descriptions with several favourable properties. The next section will give a brief introduction to this approach. Although exponential families in themselves are exact mathematical representation of beliefs, they are presented here in a section on approximate methods; because from a strict Bayesian point of view, they already represent a form of approximation, in the sense that the designing human represents her or his internal beliefs in an "alphabet" of parametric distributions approximating the mental beliefs which are never explicitly stated.

2.3.1 Exponential Families

An exponential family is a set of distributions $q(\boldsymbol{x})$ over the variable $\boldsymbol{x} \in \mathbb{R}^D$, parameterized by a set of parameters $\boldsymbol{\eta} \in \mathbb{R}^F$, of the general form

$$q(\boldsymbol{x}; \boldsymbol{\eta}) = g(\boldsymbol{\eta}) \exp\left[\boldsymbol{\eta}^\mathsf{T} \boldsymbol{u}(\boldsymbol{x})\right] = \exp\left[\boldsymbol{\eta}^\mathsf{T} \boldsymbol{u}(\boldsymbol{x}) + \log g(\boldsymbol{\eta})\right] \tag{2.10}$$

with functions $g : \mathbb{R}^F \to \mathbb{R}$ and $\boldsymbol{u} : \mathbb{R}^D \to \mathbb{R}^F$. In this, the so-called *canonical* representation, the elements of $\boldsymbol{\eta}$ are the *natural parameters* of the distribution. The elements of $\boldsymbol{u}(\boldsymbol{x})$, or linear combinations thereof, are known as the *sufficient statistics* of the distribution. Not all functions of the form (2.10) are integrable, the definition should be understood to contain the implicit assumption that the distribution is in fact normalizable and normalized (this might also imply that both $\boldsymbol{x}$ and/or $\boldsymbol{\eta}$ might only be well defined on a subspace of $\mathbb{R}^D$). Under this requirement, the function value $g(\boldsymbol{\eta})$ can be interpreted as the normalization constant, because

$$g(\boldsymbol{\eta}) \int \exp\left[\boldsymbol{\eta}^\mathsf{T} \boldsymbol{u}(\boldsymbol{x})\right] \mathrm{d}\boldsymbol{x} = 1. \tag{2.11}$$

Example: The Gaussian Distribution The family of all normal distributions forms an exponential family, because it can be written as

$$\begin{aligned}\mathcal{N}(x;\mu,\sigma^2) &= \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)\\ &= \exp\left[\begin{pmatrix}\mu/\sigma^2\\ 1/\sigma^2\end{pmatrix}^{\mathsf{T}}\begin{pmatrix}x\\ -\frac{1}{2}x^2\end{pmatrix} - \frac{1}{2}\left[\frac{\mu^2}{\sigma^2}+\log(2\pi\sigma^2)\right]\right]\end{aligned} \tag{2.12}$$

its natural parameters are the *precision-adjusted mean* μ/σ^2 and the *precision* σ^{-2}. Note that the normalization function

$$g(\mu/\sigma^2,\sigma^{-2}) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2}\frac{\mu^2}{\sigma^2}\right) \tag{2.13}$$

has a slightly different form than the term normally interpreted as the normalization constant of the Gaussian. The Gaussian's sufficient statistics are often called the *sample mean* and *sample variance* (as well as the number of samples), for the following reason: If a dataset consists of N data points x_i generated from a Gaussian distribution, then their conditional probability given the natural parameters is

$$\begin{aligned}p(D\,|\,\mu,\sigma^2) &= \prod_i^N \mathcal{N}(x_i;\mu,\sigma^2)\\ &= \exp\left[\begin{pmatrix}\mu/\sigma^2\\ 1/\sigma^2\end{pmatrix}^{\mathsf{T}}\begin{pmatrix}\sum_i x_i\\ -\frac{1}{2}\sum_i x_i^2\end{pmatrix} - \frac{N}{2}\left[\frac{\mu^2}{\sigma^2}+\log(2\pi\sigma^2)\right]\right]\\ &= \exp\left[\begin{pmatrix}\mu/\sigma^2\\ 1/\sigma^2\end{pmatrix}^{\mathsf{T}}\begin{pmatrix}N\bar{x}\\ -\frac{N}{2}(S+\bar{x})\end{pmatrix} - \frac{N}{2}\left[\frac{\mu^2}{\sigma^2}+\log(2\pi\sigma^2)\right]\right]\end{aligned} \tag{2.14}$$

with

$$\bar{x} = \frac{1}{N}\sum_i x_i \quad \text{and} \quad S = \frac{1}{N}\sum_i (x_i-\bar{x})^2. \tag{2.15}$$

Exponential families have several algebraic aspects that make them particularly well suited for use in structured probabilistic models; the following paragraphs point out some of these properties.

Closure Under Multiplication and Exponentiation The members of one exponential family form an associative algebra over the positive real numbers, in the sense that two members of the family can be multiplied together and with a positive real number, returning another unnormalized distribution which also lies

in the family. They are also closed under exponentiation: In general, we find

$$\begin{aligned} a{\cdot}q(\boldsymbol{x};\boldsymbol{\eta})^{\alpha}q(\boldsymbol{x};\boldsymbol{\eta}')^{\beta} \qquad & \forall a \in \mathbb{R}_+ \quad \forall \alpha, \beta \in \mathbb{R} \\ &= \exp\left[(\alpha\boldsymbol{\eta} + \beta\boldsymbol{\eta}')^{\mathsf{T}}\boldsymbol{u}(\boldsymbol{x}) + \alpha \log g(\boldsymbol{\eta}) + \beta \log g(\boldsymbol{\eta}') + a\right] \\ &= a\frac{g^{\alpha}(\boldsymbol{\eta})g^{\beta}(\boldsymbol{\eta}')}{g(\alpha\boldsymbol{\eta} + \beta\boldsymbol{\eta}')} \cdot q(\boldsymbol{x}; \alpha\boldsymbol{\eta} + \beta\boldsymbol{\eta}') \end{aligned} \tag{2.16}$$

Consider for example the multiplication of two Gaussian distributions, $\mathcal{N}(x;\mu_1,\sigma_1^2)$ and $\mathcal{N}(x;\mu_2,\sigma_2^2)$. From Equation (2.12) above, we see that multiplying Gaussians involves adding the precision-adjusted mean and precision. We can also find the constant scaling factor under multiplication (starting from Equation (2.13) and leaving out a few lines of straightforward algebra) to be

$$\frac{g(\mu_1/\sigma_1^2, \sigma_1^{-2})g(\mu_2/\sigma_2^2, \sigma_2^{-2})}{g(\mu_1/\sigma_1^2 + \mu_2/\sigma_2^2, \sigma_1^{-2} + \sigma_2^{-2})} = \frac{1}{\sqrt{2\pi(\sigma_1^2 + \sigma_2^2)}} \exp\left(-\frac{1}{2}\frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}\right) \tag{2.17}$$

Which gives the widely known formula

$$\begin{aligned} &\mathcal{N}(x;\mu_1,\sigma_1^2)\mathcal{N}(x;\mu_2,\sigma_2^2) = \\ &\qquad \mathcal{N}(\mu_1;\mu_2,\sigma_1^2+\sigma_2^2)\mathcal{N}\left[x; \left(\frac{\mu_1}{\sigma_1^2} + \frac{\mu_2}{\sigma_2^2}\right)\left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right)^{-1}, \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right)^{-1}\right] \end{aligned} \tag{2.18}$$

So multiplication within exponential families amounts to summation of the natural parameters. This is crucial because multiplication is such a central operation in message passing algorithms, and exponentiation will be a convenient trick in some approximate schemes (Section 2.3.2). Of course, if we can multiply by adding natural parameters, we can also divide distributions, by subtracting natural parameters — as long as we are careful not to break integrability and end up with an improper distribution. Up to that restriction, exponential families do indeed form Abelian groups under multiplication.

Analytic Conjugate Priors on the Parameters In Bayesian inference, conjugate priors are a much-used tool in the construction of analytic inference algorithms. A conjugate prior $\pi(\eta;\omega)$ to the likelihood $\lambda(x \,|\, \eta)$ is a probability distribution over η, parametrized by some parameters ω, such that the posterior on η (the normalized product of π and λ) is a member of the same functional family as the prior:

$$\frac{\lambda(x \,|\, \eta)\pi(\eta;\omega)}{\int \lambda(x \,|\, \eta)\pi(\eta;\omega)\,\mathrm{d}\eta} = \pi(\eta;\omega') \tag{2.19}$$

Such forms are crucial for efficient inference because they allow data to be incorporated into a belief analytically. For exponential families, conjugate priors can be constructed analytically, up to normalization: Assume that we have access to N

data points $\boldsymbol{x}_n$ generated from a distribution with uncertain parameters $\boldsymbol{\eta}$ of the form (2.10). We define a prior distribution r on the parameters $\boldsymbol{\eta}$ of the form

$$\begin{aligned} r(\boldsymbol{\eta}; \boldsymbol{\chi}, \nu) &= f(\boldsymbol{\chi}, \nu) g(\boldsymbol{\eta})^{\nu} \exp\left[\boldsymbol{\eta}^{\mathsf{T}} \boldsymbol{\chi}\right] \\ &= \exp\left[\begin{pmatrix} \boldsymbol{\chi} \\ \nu \end{pmatrix}^{\mathsf{T}} \begin{pmatrix} \boldsymbol{\eta} \\ \log[g(\boldsymbol{\eta})] \end{pmatrix} + \log f(\boldsymbol{\chi}, \nu) \right]. \end{aligned} \tag{2.20}$$

Note that this distribution itself forms an exponential family, with the natural parameters $\boldsymbol{\omega} \equiv (\boldsymbol{\chi}, \nu)$, the sufficient statistics $\boldsymbol{u}(\boldsymbol{\eta}) = (\boldsymbol{\eta}, \log[g(\boldsymbol{\eta})])$, and the normalization constant $f(\boldsymbol{\omega})$. Multiplying the likelihood (2.10) with this prior, we get the unnormalized posterior

$$\begin{aligned} p(\boldsymbol{\eta} \,|\, \boldsymbol{\chi}, \boldsymbol{\nu}, \{\boldsymbol{x}_n\}) &\propto g(\boldsymbol{\eta})^{\nu+N} \exp\left[\boldsymbol{\eta}^{\mathsf{T}} \left(\sum_n \boldsymbol{u}(\boldsymbol{x}_n) + \boldsymbol{\chi} \right)\right] \\ &\propto r\left(\boldsymbol{\eta}; \boldsymbol{\chi} + \sum_n \boldsymbol{u}(\boldsymbol{x}_n), \nu + N\right) \end{aligned} \tag{2.21}$$

which is of the same form as (2.20). This construction of course does not provide the normalization constant f of either prior or posterior, which can be difficult to evaluate in practice. Its technical nature also means that it does not necessarily provide the most convenient parametrization for the conjugate prior. For example, consider again the Gaussian distribution from Equation (2.12). Equation (2.20) gives a conjugate prior of the unnormalized form

$$\begin{aligned} r(\mu, \sigma^2; \boldsymbol{\chi}, \nu) &\propto (2\pi\sigma^2)^{\nu/2} \exp\left[\frac{1}{\sigma^2}\left(\mu\chi_1 + \chi_2 + \mu^2\chi_3\right)\right] \\ &\propto \frac{1}{(2\pi)^{\nu/2}} (\sigma^{-2})^{\nu} e^{\chi_2 \sigma^{-2}} e^{\frac{(\mu\chi_1 + \mu^2\chi_3)}{\sigma^{-2}}} \end{aligned} \tag{2.22}$$

which is in fact an unnormalized form of the well-known normal-gamma prior for the parameters of a Gaussian. The usual parametrization of the normal-gamma distribution is

$$r(\mu, \sigma^{-2} \,|\, a, b, \alpha, \beta) = \frac{\beta^{\alpha}\sqrt{b}}{\Gamma(\alpha)\sqrt{2\pi}} (\sigma^{-2})^{\alpha - 1/2} e^{-\beta\sigma^{-2}} e^{-\frac{b(\mu-a)^2}{2\sigma^2}} \tag{2.23}$$

so we can identify the linear transformation $\nu = \alpha - 1/2$, $\chi_1 = ab$, $\chi_2 = -\beta$, $\chi_3 = -b/2$ to establish an isomorphism between our derivation and the standard form. The remaining dissimilarities between Equations (2.22) and (2.23) are functions of the parameters only and can thus be subsumed into the normalization constant.

Estimation Through Differentiation For exponential families, the expected value of the sufficient statistics is closely related to the gradient of the normal-

ization constant. To see this, we differentiate Equation (2.11) with respect to $\boldsymbol{\eta}$ on both sides:

$$0 = [\nabla g(\boldsymbol{\eta})] \int \exp\left[\boldsymbol{\eta}^\mathsf{T} \boldsymbol{u}(\boldsymbol{x})\right] \mathrm{d}\boldsymbol{x} + g(\boldsymbol{\eta}) \int \exp\left[\boldsymbol{\eta}^\mathsf{T} \boldsymbol{u}(\boldsymbol{x})\right] \boldsymbol{u}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \tag{2.24}$$

Using (2.11) for the first term and (2.10) for the second, we get

$$0 = [\nabla g(\boldsymbol{\eta})] \cdot g^{-1}(\boldsymbol{\eta}) + \mathsf{E}[\boldsymbol{u}(\boldsymbol{x})] \tag{2.25}$$

and hence

$$\mathsf{E}[\boldsymbol{u}(\boldsymbol{x})] = -\nabla \log g(\boldsymbol{\eta}). \tag{2.26}$$

So the expected value of the sufficient statistics equals the negative log derivative of the natural parameters. This is a somewhat obscure result at first sight, but it will become important in the derivation of several approximate methods in the following sections. The significance arises because for many of the widely used exponential families at least one of these terms is readily evaluated; so being able to express one in terms of the other is helpful.

Expressive Forms Many of the most popular parametric distributions in applied statistics do indeed form exponential families. Among them are the (univariate and multivariate) Gaussian, and the Gamma and Wishart distributions (defined on positive real numbers and positive definite matrices, respectively). The Multinomial distribution on categorical variables, and the Dirichlet distribution over discrete probabilities (including their special two-dimensional or single-count cases known as the Bernoulli, Discrete, Binomial and Beta distributions), as well as the Poisson and Exponential distributions, the von-Mises distribution, and several others, also form exponential families.

There are additional beneficial aspects of exponential family distributions which are beyond the scope of this text. For example, they are in some sense easy to extend to nonparametric Bayesian models [Orbanz, 2009]. The upshot for the practitioner of approximate inference is that it is generally a good idea to use exponential family distributions when designing probabilistic models, as this will already avoid many problems arising immediately when using general, less structured distributions. The diversity of available exponential family distributions means that they are applicable in many situations; and they can be combined (using graphical models) to form very expressive hierarchical models.

2.3.2 Expectation Propagation

Minimizing KL-Divergence — Gaussian Moment Matching

Even when approximations become necessary, exponential family distributions can still be helpful. Intractability will typically be due to nonlinear factors in the graph or the connection of variables with nonconjugate marginals. When using the sum-product algorithm, both of these problems can often be addressed through local approximations: At a given problematic variable node x, the marginal

$$p(x) = \prod_i m_i(x) \tag{2.27}$$

will consist of a product over messages $m_i(x)$ that do not give a simple parametric form when multiplied. We can then try to find a projection of the marginal into an exponential family $q(x)$, and hope that the resulting approximate marginal still captures the "important" aspects of the exact marginal. How to choose this projection is a matter of mathematical convenience and what the approximation is supposed to achieve (for example, whether the approximation should give a good representation of the exact distribution's overall width, or whether is should give a good representation of a certain region of the distribution).

One popular method is to minimize the Kullback–Leibler divergence $D_{\mathsf{KL}}(p\|q)$ of q from p, which leads to an algorithm known as *Expectation Propagation (EP)* [Minka, 2001]. The KL-divergence [Kullback and Leibler, 1951] is also known as the relative entropy and defined as

$$D_{\mathsf{KL}}(p\|q) = \int p(x) \log \frac{p(x)}{q(x)} \,\mathrm{d}x = -\int p(x) \log q(x) \,\mathrm{d}x - \mathsf{H}[p(x)] \tag{2.28}$$

where $\mathsf{H}[p(x)] \equiv -\mathsf{E}_p(\log p)$ is the entropy of p. Since we want q to be in an exponential family, i.e. of form (2.10), we can evaluate (2.28) to

$$D_{\mathsf{KL}}(p\|q) = -\log g(\boldsymbol{\eta}) - \boldsymbol{\eta}^{\mathsf{T}} \mathsf{E}_p[\boldsymbol{u}(\boldsymbol{x})] + \text{constants.} \tag{2.29}$$

To minimize the divergence, take the gradient with respect to $\boldsymbol{\eta}$ and set to zero, giving

$$-\nabla \log g(\boldsymbol{\eta}) = \mathsf{E}_p[\boldsymbol{u}(\boldsymbol{x})]. \tag{2.30}$$

Now we can use Equation (2.26) on the left side to find

$$\mathsf{E}_q[\boldsymbol{u}(\boldsymbol{x})] = \mathsf{E}_p[\boldsymbol{u}(\boldsymbol{x})]. \tag{2.31}$$

So, to minimize KL-divergence from p to q, we need to match the expected sufficient statistics of q to their expected value under p. The most widely used exponential

family for expectation propagation is the Gaussian family. Extending the univariate form of Equation (2.12) to the general, D-dimensional case, the exponential family form of the multivariate Gaussian is

$$\begin{aligned}\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Lambda}^{-1}) &= \frac{|\boldsymbol{\Lambda}|^{1/2}}{(2\pi)^{D/2}}\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Lambda}(\boldsymbol{x}-\boldsymbol{\mu})\right] \\ &= \exp\left\{-\frac{1}{2}\left[\boldsymbol{x}^{\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{x}-2\boldsymbol{x}^{\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{\mu}+\boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{\mu}-\log|\boldsymbol{\Lambda}|+D\log 2\pi\right]\right\} \\ &= \exp\left\{-\frac{1}{2}\begin{pmatrix}\boldsymbol{\Lambda}\boldsymbol{\mu}\\ \operatorname{vec}\boldsymbol{\Lambda}\end{pmatrix}^{\mathsf{T}}\begin{pmatrix}-2\boldsymbol{x}\\ \operatorname{vec}\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}\end{pmatrix}+\log g(\boldsymbol{\Lambda}\boldsymbol{\mu},\boldsymbol{\Lambda})\right\}\end{aligned} \tag{2.32}$$

where $\operatorname{vec}\boldsymbol{A}$ denotes some arbitrary but consistent way of stacking matrix $\boldsymbol{A}$ into a vector[1]. Analogous to the univariate case, the canonical parameters of the multivariate Gaussian are the *precision* matrix $\boldsymbol{\Lambda}$ (the inverse of the covariance matrix) and the *precision-adjusted mean* vector $\boldsymbol{\Lambda}\boldsymbol{\mu}$; and the sufficient statistics of the multivariate Gaussian are $\boldsymbol{x}$ and $\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}$. Hence, for Gaussian EP, we want to choose a Gaussian approximation $q(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Lambda}^{-1})$ such that its first two moments (mean $\mathsf{E}[\boldsymbol{x}]$ and variance $\mathsf{E}[\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}]-[\mathsf{E}\boldsymbol{x}][\mathsf{E}\boldsymbol{x}]^{\mathsf{T}}$) match the first two moments of the true marginal p.

Approximating Individual Messages

In message passing algorithms, we will deal with marginals consisting of a product of messages:

$$p(x) = Z^{-1}\prod_i f_i(x). \tag{2.33}$$

Because the messages are the fundamental objects of the sum product algorithm, we will want the approximating Gaussian to be also a product of Gaussians

$$q(x) = \tilde{Z}^{-1}\prod_i \tilde{f}_i(x) \qquad \text{with } \tilde{f}_i(x) = \mathcal{N}(x;m_i,v_i) \tag{2.34}$$

The naïve thing to do would be to match the moments of each unnormalized distribution $f_i(x)$ to the moments of $\tilde{f}_i(x)$. However, we are ultimately interested in the marginals, not the messages. So it would be better if we could ensure that the approximate marginal minimizes KL-divergence to the exact marginal. Unfortunately, matching the moments of the product of complicated messages f_i can be challenging — splitting the resulting marginal into meaningful messages even more so. Minka [2001] found a surprisingly simple and elegant intermediate approach, based on the ease with which exponential family distributions can be multiplied

[1]This operation is well defined here, because we can write, using the sum convention [Einstein, 1916], $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{x} = x_i\Lambda_{ij}x_j = \Lambda_{ij}(\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}})_{ij}$, so any consistent way of mapping the indices $ij \in [1,\dots,D]\times[1,\dots,D]$ to a new index $k \in [0,\dots,D^2]$ works.

and divided (Section 2.3.1, Equation (2.16)). He proposes an iterative scheme, in which, for any j, the message $\tilde{f}_j$ is chosen such that

$$q'(x) \propto \tilde{f}_j(x) \prod_{i \neq j} \tilde{f}_i(x) \tag{2.35}$$

is as close to

$$f_j(x) \prod_{i \neq j} \tilde{f}_i(x) \tag{2.36}$$

as possible in KL-divergence. At runtime of the algorithm, we have available a collection of approximate messages $\tilde{f}_i(x)$, as well as an approximate marginal $q(x)$ (if some messages are not yet available, they are replaced with uninformative degenerate messages with natural parameter vector $\mathbf{0}$. In the Gaussian case, this corresponds to setting the precision matrix to $\mathbf{0}$). To update the message $\tilde{f}_j(x)$, we calculate the *cavity distribution*

$$q^{\setminus j}(x) \equiv \frac{q(x)}{\tilde{f}_j(x)}. \tag{2.37}$$

Note that this division of distributions is a well defined operation, corresponding to subtracting natural parameters (Section 2.3.1). Then, we calculate the expected values of the sufficient statistics (for Gaussians: the first two moments) of the product

$$f_j(x) q^{\setminus j}(x). \tag{2.38}$$

These statistics (moments) give an approximation $q'(x)$. Minka introduces an operator proj for this operation, which allows the intuitive notation

$$q'(x) = \text{proj}\left[f_j(x) q^{\setminus j}(x)\right]. \tag{2.39}$$

The distribution q' is the new approximate marginal and defines the approximate message as

$$\tilde{f}_j(x) = \frac{q'(x)}{q^{\setminus j}(x)} = \frac{\text{proj}[f_j(x) q^{\setminus j}(x)]}{q^{\setminus j}(x)} \tag{2.40}$$

The process is repeated until convergence, which can be measured as follows: Because the message is in the exponential family, the update from the old message $\tilde{f}_j^{\text{old}}$ to the new message $\tilde{f}_j^{\text{new}}$ can be written as

$$\Delta \tilde{f}_j = \frac{\tilde{f}_j^{\text{new}}}{\tilde{f}_j^{\text{old}}}. \tag{2.41}$$

Since this update is itself an element of the exponential family, it is described by a natural parameter vector, which can be used to measure convergence through

any arbitrary norm.[2] It is interesting to note in passing that messages themselves do not need to be proper distributions (see Section 2.2), as long as the marginal is normalizable. Indeed, in some applications of EP, "Gaussian" messages with negative precision are not unusual.

Power EP

One interpretation for EP is as a form of approximate numerical integration: The normalization constant $Z_q = \int \prod_i \tilde{f}_i(x)\,\mathrm{d}x$ (which is easy to evaluate thanks to Equation (2.16) and Equation (2.11)) is an approximation to the local partition function $Z_p = \int \prod_i f_i(x)\,\mathrm{d}x$. Sometimes, however, even the individual integrals required to evaluate $\mathrm{proj}[f_j(x)q^{\backslash j}(x)]$ can be intractable. In that case, an extension of EP called *Power EP* [Minka, 2004] can offer hope: For many functions $f_j(x)$ for which $\int f_j(x)q^{\backslash j}\,\mathrm{d}x$ is difficult to evaluate, there exists a scalar n_j such that $\int f_j^{n_j}(x)q^{\backslash j}(x)$ is tractable. As a simple example, consider $f(x) = 1/x$. The integral

$$\int x^{-1}\mathcal{N}(x;\mu,\sigma^2)\,\mathrm{d}x \tag{2.42}$$

is much more challenging than

$$\int x\mathcal{N}(x;\mu,\sigma^2)\,\mathrm{d}x = \mu \tag{2.43}$$

In the Power EP algorithm, we introduce new functions $h_i(x) = f_i^{n_i}(x)$. Then, the algorithm proceeds almost identically to EP: Calculate the cavity distribution

$$q^{\backslash j}(x) = q(x)/\tilde{h}_j(x); \tag{2.44}$$

project

$$q'(x) = \mathrm{proj}\left[h_j(x)q^{\backslash j}(x)\right]; \tag{2.45}$$

and store the message

$$\tilde{h}_j^{\mathrm{new}}(x) = \frac{q'(x)}{q^{\backslash j}(x)}; \tag{2.46}$$

but now update the marginal to the new distribution

$$q^{\mathrm{new}} = q^{\mathrm{old}}(x)\left(\frac{\tilde{h}_j^{\mathrm{new}}(x)}{\tilde{h}_j^{\mathrm{old}}(x)}\right)^{1/n_j} \tag{2.47}$$

[2] because the natural parameters are elements of $\mathbb{R}^M$ for some M, and all norms on finite-dimensional real vector spaces are equivalent, the choice of norm is arbitrary. If there is an ε_a such that the parameter vectors $\boldsymbol{\eta}_\Delta$ of the updates become and remain, for all subsequent iterations, smaller than ε_a under the norm $\|\cdot\|_a$, i.e. $\|\boldsymbol{\eta}_\Delta\| < \varepsilon_a$, then for any other norm $\|\cdot\|_b$, there exists an ε_b such that the updates' parameters become and remain smaller than ε_b under $\|\cdot\|_b$. Hence, only the numerical value of the stopping condition ε has to be chosen sensibly.

where the exponent n_j denotes exponentiation of exponential family distributions, as defined in Equation (2.16). Minka [2004] shows that this iterative scheme has the same fixed points as an implementation of EP which does not minimize KL-divergence, but instead the more general α-*divergence* [Amari, 1985] for $\alpha = 1/n_i$. The α-divergence is defined as

$$D_\alpha(p\|q) = \frac{1}{\alpha(1-\alpha)} \int \alpha p(x) + (1-\alpha)q(x) - p^\alpha(x)q^{1-\alpha}(x)\,\mathrm{d}x \tag{2.48}$$

It satisfies $D_\alpha(p\|q) \geq 0$, with equality if and only if $p = q$. For $\alpha \to 1$, it reduces to[3] $D_{\mathsf{KL}}(p\|q)$; for $\alpha \to 0$, it equals $D_{\mathsf{KL}}(q\|p)$. For $\alpha = 0.5$ it gives the Hellinger distance, for $\alpha = 2$ the χ^2 distance. Changing $\alpha \to 1 - \alpha$ swaps the position of p and q [Minka, 2005].

Damping Messages

A drawback of EP is that it is not guaranteed to converge [Minka, 2004]. The messages passed around the graph constitute a dynamic system that is not always stable. One approach to this problem, which is not guaranteed to be effective, but has been empirically found to be helpful in many cases, is to "damp the messages" [Minka, 2004], i.e. to weaken the iterative updates to the messages in a way that does not change the fix-points of the algorithm. For the general case of Power EP (for standard EP, set $n_j = 1$), we use a scalar $0 < \gamma \leq 1$ to construct the projected marginal as in Equation (2.45), then create a new message as

$$\tilde{h}_j^{\text{new}}(x) = \left(\tilde{h}_j^{\text{old}}(x)\right)^{1-\gamma} \left(\frac{q'(x)}{q^{\setminus j}(x)}\right)^\gamma = \tilde{h}_j^{\text{old}}(x) \left(\frac{q'(x)}{q(x)}\right)^\gamma \tag{2.49}$$

and update the marginal to

$$q^{\text{new}}(x) = q^{\text{old}}(x) \left(\frac{\tilde{h}_j^{\text{new}}(x)}{\tilde{h}_j^{\text{old}}(x)}\right)^{n_j} = q^{\text{old}}(x) \left(\frac{q'(x)}{q(x)}\right)^{\gamma n_j}. \tag{2.50}$$

EP: Summary

EP and Power EP provide a flexible approximate scheme for message passing which can double as an approximate integration method. Power EP iteratively minimizes

[3] To see this, note that both numerator and denominator of Equation (2.48) are smooth functions and vanish for $\alpha \to 1$ and $\alpha \to 0$, so L'Hôpital's rule applies. Differentiating both numerator and denominator with respect to α gives

$$\lim_{\alpha\to 1} D_\alpha(p\|q) = \lim_{\alpha\to 1} \frac{-\int \log(p/q)p^\alpha q^{1-\alpha}\,\mathrm{d}x}{1-2\alpha} = D_{\mathsf{KL}}(p\|q).$$

because $\alpha \to 1 - \alpha$ swaps positions of p and q, the other statement follows directly.

the local α-divergence

$$D_\alpha\Big(f_j(x)\prod_{i\neq j}\tilde{f}_i(x)\Big\|\prod_i \tilde{f}_i(x)\Big) \tag{2.51}$$

with $\alpha = (2/n_i) - 1$, standard EP sets $n_i = 1$, corresponding to minimizing the local KL-divergence from p to q. Note that, since this KL-divergence contains the term $-\int p(x)\log q(x)\,\mathrm{d}x$, it becomes large if $q(x)$ puts small mass on values of x where $p(x)$ is large. Conversely, regions where $p(x)$ vanishes do not influence $D_{\mathsf{KL}}(p\|q)$. This means *EP prefers broad approximations*. This is in stark contrast to the other approximations following in this section, and is an important characteristic determining the use cases for EP.

The two biggest issues with EP are

- ▷ EP still requires analytic evaluation of certain integrals (albeit much easier ones than the ones required for the full joint distribution). This can make EP difficult to apply (see Chapter 3 and Appendix A for derivations of just one specific factor and the technical stretches required)
- ▷ EP is not guaranteed to converge (but often does so anyway)

In this thesis, EP is used extensively in Chapter 3 and Chapter 4. Appendices A and B contains derivations for EP updates on one specific functional factor.

2.3.3 Variational Inference

The previous section established EP, an approximation based on the minimizing the KL-divergence $D_{\mathsf{KL}}(p\|q)$ from p to q on some *local* objective. Minimizing KL-divergence in this direction leads to a "broad" approximation, because the divergence becomes large if p puts mass on a region where q vanishes. This section considers in some sense the opposite approach (but see the alternative motivation at the end of this section): Given the joint distribution $p(\boldsymbol{x}, \boldsymbol{\theta})$ of a set of observable variables (data) $\boldsymbol{x}$ and latent parameters $\boldsymbol{\theta}$, we try to minimize the KL-divergence

$$D_{\mathsf{KL}}[q(\boldsymbol{\theta})\|p(\boldsymbol{\theta}\,|\,\boldsymbol{x})] = -\int q(\boldsymbol{\theta})\log\frac{p(\boldsymbol{\theta}\,|\,\boldsymbol{x})}{q(\boldsymbol{\theta})}\,\mathrm{d}\boldsymbol{\theta} \tag{2.52}$$

between an approximating distribution q for the parameters and their exact posterior distribution. This leads to an approximation attempting to fit the shape of p in some region, while ignoring other regions. This behaviour can be desirable for example in mixture models, which often have combinatorial symmetries: They put identical mass on equivalent realizations. In such situations, the approximation should pick out one good realization and represent a belief over it well. In contrast,

EP would construct a very broad approximation that covers all equivalent realizations, but does not necessarily capture the structure of any individual realization well.

Inspecting Equation (2.52), it would seem that minimizing this objective involves having to evaluate the exact posterior, which would beg the question, as that is exactly what we are trying to avoid. However, there is a "shortcut", which involves an algebraic trick: Notice that, using any arbitrary proper distribution $q(\boldsymbol{\theta})$, we can use Bayes' rule to expand the log model evidence (which is a fixed number) as

$$\begin{aligned}\log p(\boldsymbol{x}) &= \underbrace{\int q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{x}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} \,\mathrm{d}\boldsymbol{\theta}}_{\mathcal{L}[q(\boldsymbol{\theta})]} + \underbrace{\int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta} \mid \boldsymbol{x})} \,\mathrm{d}\boldsymbol{\theta}}_{D_{\mathsf{KL}}[q(\boldsymbol{\theta}) \| p(\boldsymbol{\theta} \mid \boldsymbol{x})]} \\ &= \mathcal{L}[q(\boldsymbol{\theta})] + D_{\mathsf{KL}}[q(\boldsymbol{\theta}) \| p(\boldsymbol{\theta} \mid \boldsymbol{x})]\end{aligned} \tag{2.53}$$

where we have defined a *variational bound* $\mathcal{L}(q)$, also known as the *variational free energy* in statistical physics, on the log evidence. To see that this is indeed a lower bound, note that the KL-divergence is non-negative, a statement known as Gibbs' inequality[4]. Hence, to minimize the KL-divergence, maximize $\mathcal{L}(q)$ instead. This is potentially much easier, because the joint is often available in closed form.
This is known as a *variational approximation*, because maximizing the bound involves functional derivatives of the functional $\mathcal{L}$, and the branch of mathematics concerned with functional derivatives is known as the *calculus of variations*.

Alternate Motivation The derivation of the bound by Equation (2.53) may seem somewhat unsatisfactory, and indeed it of course leaves out certain technical requirements on the approximating distribution. It is also possible to motivate the variational approximation in an alternate way inverse to the argument of the previous section: To find a good approximation q, construct a bound on the log evidence of the model, note that it involves the positive definite KL-divergence, and maximize the bound. The connection between EP and variational approximations provided by the different directions of KL-divergence they minimize can provide a helpful intuition when deciding which of the two to use (see Section 2.3.7). However, it also contains a pitfall: Note that variational inference directly minimizes KL-divergence, i.e. using the actual joint distribution, not some factorised approximation of it, while EP only minimizes a local KL-divergence. This difference is less benign than it might seem: Using variational inference, one can at least rely on having the right objective (even if the found bound might well be loose), while

[4]It is easy to prove Gibbs' inequality [e.g. MacKay, 2003], using Jensen's inequality [Jensen, 1906]. Consider the convex function $f(u) = 1/u$ and $u = p(x)/q(x) > 0$. Then $D_{\mathsf{KL}}(q\|p) = \mathsf{E}_q[f(u)] \geq f[\mathsf{E}_q(u)] = f(\int q\frac{p}{q}) = -\log \int p = -\log 1 = 0$

with EP, one always has to worry somewhat about the local approximations missing important aspects of the product of the factors.

Factorized Variational Approximations

If we pose no restrictions on q, the bound $\mathcal{L}$ reaches a unique maximum if $q(\boldsymbol{\theta}) = p(\boldsymbol{\theta} \mid \boldsymbol{x})$, where $D_{\mathsf{KL}}(q\|p) = 0$. Since the assumed intractability of this distribution is the reason to use an approximation in the first place, this most general form is of little use. To find approximations of easier to track structure, we can impose factorization properties on q over K disjoint subsets $\boldsymbol{\theta}_k$ of $\boldsymbol{\theta}$:

$$q(\boldsymbol{\theta}) = \prod_{k=1}^{K} q_k(\boldsymbol{\theta}_k) \tag{2.54}$$

In line with the overall point of this chapter that factorization leads to local computations, this factorized form for q allows us to derive a local bound which is a function of only one q_j, simply by inserting (2.54) into $\mathcal{L}$ from (2.53). Writing q_k as shorthand for $q_k(\boldsymbol{\theta}_k)$ we get

$$\begin{aligned}
\mathcal{L}(q) &= \int \prod_k q_k \left[\log p(\boldsymbol{x}, \boldsymbol{\theta}) - \prod_\ell \log q_\ell \right] \mathrm{d}\boldsymbol{\theta} \\
&= \int q_k \underbrace{\left[\int \log p(\boldsymbol{x}, \boldsymbol{\theta}) \prod_{\ell \neq k} q_\ell \, \mathrm{d}\boldsymbol{\theta}_\ell \right]}_{\equiv \mathsf{E}_{\ell \neq k}[\log p(\boldsymbol{x}, \boldsymbol{\theta})]} \mathrm{d}\boldsymbol{\theta}_k - \int q_k \log q_k \, \mathrm{d}\boldsymbol{\theta}_k + \text{const.} \\
&= \int q_k \cdot \mathsf{E}_{\ell \neq k}[\log p(\boldsymbol{x}, \boldsymbol{\theta})] \, \mathrm{d}\boldsymbol{\theta}_k + \mathsf{H}[q_k] + \text{const.}
\end{aligned} \tag{2.55}$$

The last line of this equation has the form (up to sign) of a KL-divergence between $q_k(\boldsymbol{\theta}_k)$ and the distribution $\tilde{p}$ that satisfies the relation

$$\log \tilde{p}(\boldsymbol{\theta}_k) = \mathsf{E}_{\ell \neq k}[\log p(\boldsymbol{x}, \boldsymbol{\theta})] + \text{const.} \tag{2.56}$$

Hence, $\mathcal{L}$ is maximized when $q_k = \tilde{p}$, i.e.

$$\begin{aligned}
\log q_k(\boldsymbol{\theta}_k) &= \mathsf{E}_{\ell \neq k}[\log p(\boldsymbol{x}, \boldsymbol{\theta})] + \text{const.} \\
q_k(\boldsymbol{\theta}_k) &\propto \exp\left(\mathsf{E}_{\ell \neq k}[\log p(\boldsymbol{x}, \boldsymbol{\theta})] \right)
\end{aligned} \tag{2.57}$$

If the expectation of log probabilities in this expression is easy to evaluate (here again, using exponential family distributions for model design can greatly reduce the computational complexity), then Equation (2.57) directly leads to an iterative optimization scheme: For each variable group $\boldsymbol{\theta}_k$, maximize the local bound, then repeat. If we choose the $\boldsymbol{\theta}_k$ to correspond to one variable in the graphical model

each, this scheme is known as *Variational Message Passing (VMP)* [Winn and Bishop, 2006]. To find good approximations, however, it can sometimes be a better idea to adapt the graphical representation to the approximation, rather than the other way round (see Chapter 5).

2.3.4 Laplace Approximations

The approximate inference methods described in the previous two sections can provide expressive approximations; but they both require the evaluation of some integrals involving the exact distribution p in some way. Even though the integrals in question are much simpler than the integration required to evaluate the exact posterior, they can still be challenging in some cases. This section introduces an approximate scheme, known as a *Laplace approximation*, which only involves derivatives of p. Since integration is harder than differentiation, this approximation can be applied to a larger class of distributions. Unfortunately this simplicity comes at the cost of a considerable defect, which means that this scheme should only be applied with care.

Since conditional dependence structure will not be important for the following derivations, we will abandon the separation of variables and parameters, and consider, for simplicity, a probability distribution $p(\boldsymbol{x})$ of some multidimensional variable $\boldsymbol{x}$.

"Laplace approximation" is a grand name for a second order Taylor expansion of $\log p(\boldsymbol{x})$ around a mode $\boldsymbol{\mu}$ of $p(\boldsymbol{x})$ (which, because the logarithm is a strictly monotonic function, is identical to the mode of $\log p(\boldsymbol{x})$): To find a mode, differentiate once and set to zero

$$\nabla \log p(\boldsymbol{\mu}) = \left.\frac{\partial \log p(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{\mu}} = 0 \tag{2.58}$$

if $\log p(\boldsymbol{x})$ is analytic in a neighbourhood of $\boldsymbol{\mu}$, then it can be expanded in this neighbourhood to

$$\begin{aligned}\log p(\boldsymbol{\mu}+\boldsymbol{x}) &= \log p(\boldsymbol{\mu}) + \underbrace{(\boldsymbol{x}-\boldsymbol{\mu})^\mathsf{T}\nabla \log p(\boldsymbol{\mu})}_{=0 \text{ by eq. (2.58)}} - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\mathsf{T}\boldsymbol{\Lambda}(\boldsymbol{x}-\boldsymbol{\mu}) + \mathcal{O}[(\boldsymbol{x}-\boldsymbol{\mu})^3] \\ &= -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\mathsf{T}\boldsymbol{\Lambda}(\boldsymbol{x}-\boldsymbol{\mu}) + \text{const} + \mathcal{O}[(\boldsymbol{x}-\boldsymbol{\mu})^3]\end{aligned} \tag{2.59}$$

where $\boldsymbol{\Lambda}$ is the negative Hessian matrix with elements

$$\Lambda_{ij} = -\left.\frac{\partial^2 \log p(\boldsymbol{x})}{\partial x_i \partial x_j}\right|_{\boldsymbol{x}=\boldsymbol{\mu}}. \tag{2.60}$$

There is an exponential family whose natural parameters are linearly related to $\boldsymbol{\mu}$

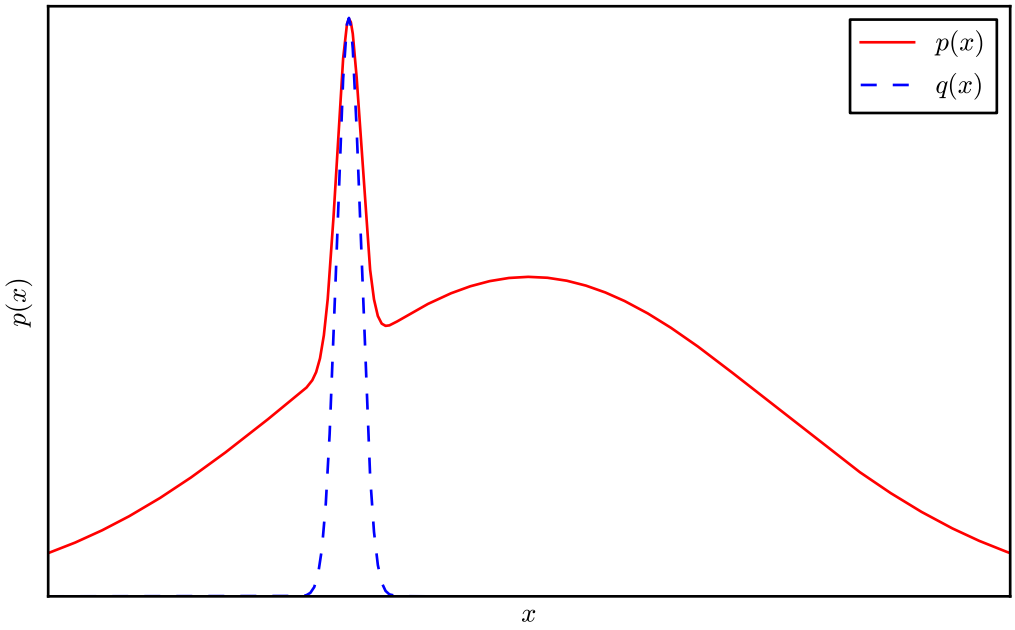


Figure 2.7: Defect of the Laplace approximation: because modes are a local feature of the distribution rather than a global one, approximations based solely on the structure of the distribution around the node can lead to bad representation of the true distribution $p(x)$ by the approximation $q(x)$.

and $\boldsymbol{\Lambda}$: the Gaussian

$$\log \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) = -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^{\mathsf{T}} \boldsymbol{\Lambda} (\boldsymbol{x} - \boldsymbol{\mu}) + \frac{1}{2} \left(\log |\boldsymbol{\Lambda}| - D \log 2\pi \right) \tag{2.61}$$

so the Laplace approximation consists of approximating $p(\boldsymbol{x})$ with a Gaussian $q(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$ whose mean is set to the location of a mode of p, and whose precision matrix (inverse covariance matrix) is set to the negative Hessian of $\log p$ at its mode.

From this construction, the main strength of this approximation scheme is immediately obvious: Laplace approximations require very little algebraic structure of p. It suffices to be able to differentiate p twice. It is a particularly convenient approximation if we can find the mode and Hessian analytically, but if everything fails, a numerical optimization scheme, such as Newton–Raphson iterations or even numerical differentiation can be used as well. At this level, virtually every distribution p which can be evaluated at all becomes fair game. Unfortunately, there is a major drawback: Modes are a local feature and need by no means represent the overall distribution well (see Figure 2.7). If the distribution has heavy tails, is multimodal, is strongly asymmetric, or if its shape is badly represented by the Hessian at the mode, the Laplace approximation can be arbitrarily bad, and this defect may not even be easy to spot during the derivation! Hence, before resorting

to this form of approximation, it should be ensured that the approximated function does have an overall shape that can at least be roughly represented by a Gaussian. See Sections 5.2.2 and 5.3.1 as well as Appendix C for an application where this approach arguably makes sense.

2.3.5 Markov Chain Monte Carlo

Another approach to approximate inference, quite different from the three methods presented in the previous sections, is to replace analytical parametric distributions with samples, leading to *Monte Carlo* algorithms. If exact sampling from the distribution in question is not possible — which is the typical situation — sampling algorithms based on ergodic random walks through the distribution, known as *Markov Chain Monte Carlo (MCMC)* algorithms provide an approximate answer. MCMC methods will not feature prominently in this thesis; good introductions into this vast field can be found in Murray [2007] and MacKay [2003, chapters 29 and 30]. But approximate sampling can be a great tool for approximate inference, for two main reasons.

- ▷ MCMC algorithms are guaranteed by construction to sample from the exact posterior in the limit of large sample sizes
- ▷ they tend to be easy to implement for general probabilistic models

They also have a few important drawbacks which are mostly to do with practical issues of implementation rather than with theoretical guarantees:

- ▷ finding and fixing bugs in MCMC algorithms can be difficult, as the results are stochastic by nature
- ▷ more importantly, diagnosing convergence of a MCMC sampler is challenging, and there is no generally applicable criterion for convergence. A very bad MCMC sampler, performing a random walk in a confined region of the distribution, can look dangerously similar to a very good MCMC sampler from the point of view of convergence diagnostics, such as autocorrelation measures. The time required for convergence is also difficult to predict from the model structure alone. In models where a particular sampling scheme mixes well, it can in fact provide the most computationally efficient solution. But if the algorithm is badly designed and mixes slowly, sampling can be an exceedingly expensive solution.

The fact that MCMC methods only show up on the fringes of this thesis should not be interpreted as a statement about their usefulness, but as a conscious decision to focus on parametric analytic approximations.

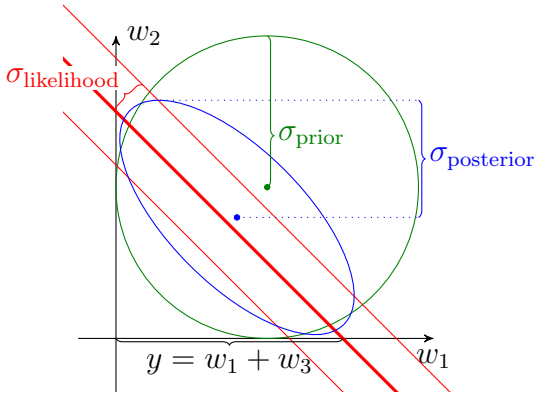


Figure 2.8: Sketch of the "explaining away" phenomenon. An independent Gaussian prior (green circle) over the weights $\boldsymbol{w}$ turns into an anti-correlated posterior (blue ellipsis) upon observing noisy information about the value of their sum $y = w_1 + w_2$ (red band, with thick mean and light lines at one standard deviation).

2.3.6 Assuming Independence

A final, drastic measure to simplify inference is to simply assume certain parts of the model to be independent, even though they are not. Since every interesting aspect of probabilistic inference relies on dependences between variables, this is obviously not a good general approach. But there are a few special situations in which a set of variables is "all but independent" in such a way that they can in fact be treated as independent. The most important such case is sparse Gaussian regression: Assume that we get to observe a noisy data point $y \in \mathbb{R}$ assumed to be generated in a linear way from weights $\boldsymbol{w} \in \mathbb{R}^D$, over which we have a Gaussian prior.

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \qquad p(y \,|\, \boldsymbol{w}, \boldsymbol{\phi}, \tau) = \mathcal{N}(y; \boldsymbol{\phi}^\mathsf{T}\boldsymbol{w}, \tau) \tag{2.62}$$

(see Section 5.2.3 for the more general case with $\boldsymbol{y} \in \mathbb{R}^K$). The posterior on $\boldsymbol{w}$ can be found by "completing the square" [e.g. Bishop, 2006, §2.3.3], and is given by

$$\begin{aligned} p(\boldsymbol{w} \,|\, y, \phi) &= \mathcal{N}(\boldsymbol{w}; \boldsymbol{\Psi}\left[\tau^{-1}\boldsymbol{\phi}y + \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right], \boldsymbol{\Psi}) \\ \text{where } \boldsymbol{\Psi} &= \left(\boldsymbol{\Sigma}^{-1} + \tau^{-1}\boldsymbol{\phi}\boldsymbol{\phi}^\mathsf{T}\right)^{-1} \\ &= \boldsymbol{\Sigma} - \frac{\boldsymbol{\Sigma}\boldsymbol{\phi}\boldsymbol{\phi}^\mathsf{T}\boldsymbol{\Sigma}}{\tau + \boldsymbol{\phi}^\mathsf{T}\boldsymbol{\Sigma}\boldsymbol{\phi}} \end{aligned} \tag{2.63}$$

(The last line follows from the matrix inversion lemma. See also Equation (C.17) and the footnote associated with it.) A typical situation in applications is that D is very large and $\boldsymbol{\phi}$ is a sparse vector with very few non-zero entries.

For example, consider the task of predicting user preferences for certain types of music from the users' past rankings of records belonging to certain classes. (This is just an example, and should not be considered a particularly good solution to this kind of task. See Bennett and Lanning [2007] for a review of good approaches

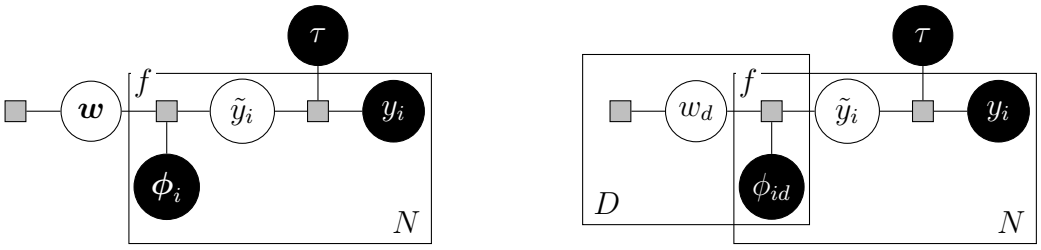


Figure 2.9: On factorized regression: **Left:** fully connected, exact regression factor graph. **Right:** factor graph for the approximate assumption of independence. The "sum factor" is marked by an f.

to such tasks). Here, $\boldsymbol{\phi}$ might be a binary vector that is zero everywhere, except for a one at the ID of a particular user, and a one at one other index, indicating that "this is a user from North America".

This setup poses a computational challenge: On the one hand, because $\boldsymbol{\phi}$ has more than one nonzero entry, the posterior on $\boldsymbol{w}$ is correlated — even if the prior has a diagonal covariance matrix $\boldsymbol{\Sigma} = \mathrm{diag}(\boldsymbol{\sigma^2})$, the outer product $\boldsymbol{\phi}\boldsymbol{\phi}^\mathsf{T}$ induces correlations (nonzero off-diagonal elements). This effect is famously known as *explaining away* [Pearl, 1988]; Figure 2.8 shows a simple sketch providing an intuition. On the other hand, it would be computationally intractable to try to model the entire $D \times D$ correlation matrix: If $D \sim 10^7$ (a typical user number for popular contemporary web services), then a single precision representation of $\boldsymbol{\Psi}$ would require Petabytes of storage.

The alternative is to enforce the assumption that the posterior beliefs on individual weights w_i be independent: $p(\boldsymbol{w}) = \mathcal{N}[\boldsymbol{w}; \boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma^2})]$. This approach has been an established way to bring down computational cost for some time [Maybeck, 1982, Opper, 1996]. Figure 2.9 shows a factor graph reflecting this assumption.

Under this assumption of independence, the message from the elements of $\boldsymbol{w}$ to each data point y_i — the "likelihood" term of Equation (2.62) — has a diagonal covariance and can thus be evaluated with low cost:

$$\begin{aligned} m_{f\to y_i}(y_i) &= \int f(\tilde{y}_i \mid \boldsymbol{\phi}_i, \boldsymbol{w}) \mathcal{N}(\tilde{y}_i; y_i, \tau) \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}, \boldsymbol{\sigma^2}) \,\mathrm{d}\tau \,\mathrm{d}\boldsymbol{w} \\ &= \int \mathcal{N}\left[y_i; \sum_d \phi_{id} w_d, \tau\right] \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}, \boldsymbol{\sigma^2}) \,\mathrm{d}\boldsymbol{w} \\ &= \mathcal{N}\left[y_i; \sum_d \phi_{id}\mu_d, \tau + \sum_d \phi_{id}^2 \sigma_d^2\right]. \end{aligned} \tag{2.64}$$

Because the marginals on w_d are considered independent, we can find the messages to the individual weights simply by re-arranging the terms (because the Gaussian

is symmetric around the mean: $\mathcal{N}(x; m, v) = \mathcal{N}(m; x, v)$). This gives

$$m_{f \to w_d}(w_d) = \mathcal{N}\left[w_d; \frac{1}{\phi_{id}}\left(y_i - \sum_{d' \neq d} \phi_{id'}\mu_{d'}\right), \frac{1}{\phi_{id}^2}\left(\tau + \sum_{d' \neq d} \phi_{id'}^2 \sigma_{d'}^2\right)\right] \tag{2.65}$$

The drawback of this approach is that we have ignored a correction that would show up in the fully connected model. Over time, the beliefs on w_d will thus become too confident. A simple but ad-hoc approach to rectify this behaviour is known as *exponential forgetting*: When iterating through the dataset, at each y_i, before sending the messages to the w_d, add a small constant to the uncertainty (variance) on each w_d with $\phi_{id} \neq 0$:

$$\sigma_d^2 \leftarrow \sigma_d^2 + \epsilon \tag{2.66}$$

where ϵ is a free parameter chosen through some heuristics. There are motivations for this approach — for example, one can interpret it as modeling "drift" of the correct values for w_d from one data point to the other — but they have the whiff of *ex post facto* motivations. A more honest motivation is that this factorization assumption is simply so much cheaper computationally that one is willing to accept small inaccuracies. As long as $\boldsymbol{\phi}$ are sparse and pairwise dissimilar (in the sense that $\sum_i \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\mathsf{T}$ is a matrix dominated by its diagonal), the resulting defects might be unproblematic. If additional knowledge about the structure of the $\boldsymbol{\phi}_i$ is available, e.g. if co-occurrences of features are limited to a sub-space, block-diagonal approximations can provide a good trade-off between the fully factorized posterior of Equation (2.63) and the fully factorized marginals arising from Equation (2.65).

2.3.7 Comparison of Approximation Schemes

All the approximate inference algorithms presented in the preceding sections are generally applicable and fit well with the graphical models framework. Yet they each have their advantages and shortcomings. Table 2.1 contains a compact overview of the these aspects. Because of the table's space limitations, the following list provides some background on the individual entries:

EP provides broad, "zero-avoiding" [Bishop, 2006] approximations (i.e. q tries to be non-zero everywhere where p is non-zero). This is advantageous if we are searching for a approximation of the type "estimate with error bar" — an unstructured Gaussian approximation covering the entire spread of the underlying exact, structured distribution (such as in Chapter 3). The same characteristic is a problem in distributions with combinatorial symmetry if we are only interested in one of the modes. The most important example of such combinatorial symmetries is the case of mixture models.

Variational Inference is in some sense the counterpart to EP, providing a "zero-

enforcing" [Bishop, 2006] approximation. Hence, it is a good candidate for inference in mixture models (such as in Chapter 5), but not well suited for structured broad distributions with several nodes, whose width we would like to approximate (such as the max function of Appendix A).

Laplace Approximations can be used in two different ways: As an overall approximation for the evidence of a big multivariate model, or to construct a lightweight approximation for the message between two variables within a larger graphical model. In the former case, the mode and Hessian are usually found numerically. The second case can be found in Chapter 5 and Appendix C, where it is used to provide a link between two parts of a graphical model in which different approximate schemes (factorized message passing and variational inference) produce two different exponential family distributions (multivariate Gaussian and Dirichlet distributions, respectively).

MCMC sampling algorithms have the advantage of providing a nonparametric representation of a distribution through samples, and converge to the exact posterior in the limit of many samples. They are thus a "gold-standard" to which other, faster approximations can be compared. The single big drawback of MCMC methods is that, unless great care is taken to ensure good mixing, they can mix badly and take exceedingly long to converge — and that this behaviour is difficult to discover. This problem is particularly pronounced in hierarchical models of complicated structure, particularly when a large amount of data is available and the posterior is thus highly concentrated in several "disconnected" regions.

Exact Sampling Methods mostly refers to rejection sampling algorithms, such as adaptive rejection sampling [Gilks and Wild, 1992, Wild and Gilks, 1993]. Aside from analytical exact sampling schemes, which are rarely available, a rejection sampler with high acceptance rate provides arguably the best possible representation of a joint posterior. Unfortunately, in high-dimensional models any rejection sampler has low acceptance rate [MacKay, 2003, §29.3] and is thus of little help. However, in some cases it can be helpful to sample individual dimensions exactly. For example, single dimensions in a larger Gibbs sampling scheme might be sampled by a method like adaptive rejection sampling, when the conditional distributions are not of closed exponential family form.

method	requirements	strengths	weaknesses	good applications	bad applications
EP	$\int f(\theta \mid x) q(\theta) \, \mathrm{d}\theta$	broad approximation	convergence not guaranteed; can smooth out modes	non-linear factors (e.g. max, step-function, ...)	mixture models
Variational Inference	$\int q(\theta) \log p(\theta, x) \, \mathrm{d}\theta$	mode-finding (symmetry-breaking)	weak tails, ignores minor modes	mixture models	non-linear factors
Laplace	$\nabla p^*(\boldsymbol{\theta} \mid \boldsymbol{x}), \frac{\partial^2 p^*(\boldsymbol{\theta} \mid \boldsymbol{x})}{\partial \theta_i \partial \theta_j}$	lightweight	approximation based on local feature of the distribution (mode) only	linking approximations	distributions where node is no good representation
MCMC	evaluation of unnormalized distribution	exact in the limit of many samples	convergence/mixing not measurable	gold standard for other models to compare to; models with complex structure	highly hierarchical models (unless well mixing); applications where computational cost matters
Exact MC	good proposal distribution	exact, known convergence rate	hard to design; bad proposal leads to high rejection rate	challenging 1D subparts of a model	high-dimensional models (curse of dimensionality)
Assumed Independence	none	can drastically reduce cost	can be arbitrarily wrong	almost independent distributions	strongly correlated distributions

Table 2.1: Overview over strengths and weaknesses of the approximate inference schemes introduced in this chapter.

Chapter 3

Inference on Optimal Play in Games

The work presented in this chapter was carried out in collaboration with **Thore Graepel** and **David Stern**, both of Microsoft Research Ltd. All mathematical derivations, algorithmic implementations and experimental evaluations were performed by the author of this thesis.
A shorter version of this chapter was published as [Hennig et al., 2010]: *Coherent Inference on Optimal Play in Games*; P. Hennig, D. Stern and T. Graepel; proceedings of the 13th International Conference on Artificial Intelligence and Statistics, 2010. Journal of Machine Learning Research: W&CP **9**

Abstract

Round-based games are an instance of discrete planning problems. Some of the best contemporary game tree search algorithms use random roll-outs as data. Relying on a good policy, they learn on-policy values by propagating information upwards in the tree, but not between sibling nodes. This chapter presents a generative model and a corresponding approximate message passing scheme for inference on the optimal, off-policy value of nodes in smooth AND/OR trees, given random roll-outs. The crucial insight is that the distribution of values in game trees is not completely arbitrary. We define a generative model of the on-policy values using a latent score for each state, representing the value under the random roll-out policy. Inference on the values under the optimal policy separates into an inductive, pre-data step and a deductive, post-data part. Both can be solved approximately with Expectation Propagation, allowing off-policy value inference for any node in the (exponentially big) tree in linear time.

3.1 Introduction

Games are one of the oldest problem of artificial intelligence research, going back to early works by Wiener [1948] and Shannon [1950]. Most early research was concerned with Chess. Following the success of *Deep Blue* [Campbell et al., 2002] against Gary Kasparov, interest has shifted to the Asian board game *Go*. Compared to Chess, Go has a much larger number of possible positions and is not as amenable to the design of hand-crafted evaluation functions.

Many round-based two-player games, like Chess and Go, can be represented, up to transpositions, by graphs with the structure of AND/OR trees [Nilsson, 1971]. These are trees with binary leaf nodes, in which branch nodes at alternate depths are assigned the OR (maximum) value of their children or AND (minimum) value of their children, respectively, reflecting the adversarial efforts of two competing players trying to achieve either binary value as the outcome of the game. If the branching factor b is constant, a tree of depth d contains b^d nodes. For Go, b and d are on the order of 200, so it seems finding the optimal path through the game should be intractable (In fact, finding an optimal path through the Go game tree has been shown to be EXPTIME-complete [Robson, 1983][1]). Yet humans *can* find good paths through the Go tree in finite time.

A crucial property of games that humans use is that the tree has structure: Winning positions are not distributed uniformly among the tree's leaves; they are clustered in the sense that leaf nodes that are close to each other (in terms of the graph) tend to have similar outcomes. This structure can be modeled by a latent *score*, representing the amount by which one player is 'ahead' or 'behind'. Random play leads to a random walk, typically changing the evaluation by small increments. Critical moves, changing the score drastically, are a rare occurrence in the tree overall.

Although it is not usually mentioned explicitly, this smoothness is a crucial intuition behind Monte Carlo algorithms for 'best first' tree search, like UCT [Kocsis and Szepesvári, 2006], which have been very successful recently. These algorithms repeatedly play *roll-outs*—random descents through the tree to a leaf, generated by a relatively weak or even uniformly random policy. The search tree is expanded asymmetrically from the root, based on the frequency of wins and losses in the roll-outs. If wins and losses were distributed uniformly at random among the leaves, the roll-out results would be almost completely uninformative [Pearl, 1985]. The best contemporary Go machines use UCT as part of their method [Gelly and Silver, 2008]. However, UCT-like methods base their value estimates directly on average

[1] EXPTIME problems are problems solvable by a deterministic Turing machine in $\mathcal{O}(2^{p(n)})$ where $p(n)$ is a polynomial of n. EXPTIME-complete problems are such problems, such that every other EXPTIME problem has a polynomial time many-to-one reduction to it. In particular, all NP problems are a subset of EXPTIME problems.

outcomes of tree-descents, making them dependent on a good exploration and roll-out policy. They also do not propagate information laterally in the tree, although, thanks to the smoothness of the tree, the value of a node does contain information about the value of its siblings.

This chapter constructs an explicit generative model for the value of game tree nodes under the *random* roll-out policy. Finding the value under the *optimal* policy would amount to solving a min-max optimization problem of complexity $\mathcal{O}(b^d)$ if all nodes were observed. However, for best-first search algorithms, it will be shown how an approximate closed form for the unobserved parts of the tree can be constructed and used to derive an approximate message passing scheme. The resulting algorithm tracks a joint posterior belief over the *optimal* values of *all* nodes in the tree. It incorporates a new roll-out at depth k from the root in $\mathcal{O}(kb)$ time (using heuristics, this can be brought to $\mathcal{O}(k)$, the complexity class of UCT), arriving at an intermediate set of local marginals that is sufficient to evaluate the posterior of an arbitrary node at depth ℓ in the tree (something classic Monte Carlo algorithms cannot do) with cost $\mathcal{O}(\ell)$. The algorithm can be interpreted as an instance of Bayesian off-policy reinforcement learning [Watkins and Dayan, 1992, Duff, 2002], inferring the optimal policy from samples generated by a non-optimal policy. Our method might be applicable, to varying degree, to other tree-structured optimization problems, if they exhibit a functional relationship between steps; c.f. the metric (though not tree-structured) sets of bandits considered by Kleinberg et al. [2008].

The main research contributions are the generative model for the score of game positions, the formulation of a probabilistic best-first tree search algorithm, and a demonstration of Expectation Propagation on min-max trees.

Probabilistic approaches to game tree search have been suggested before [see e.g. Baum and Smith, 1997, Russell and Wefald, 1991, Palay, 1985, Stern et al., 2007]. These works concentrated on guiding the search policy. Here we focus on efficient and consistent off-policy inference for the entire tree. This is valuable because a coherent posterior over off-policy values provides meaningful probabilistic training data for Bayesian algorithms attempting to learn a generalizing evaluation function based on features of the game state.

3.2 Methods

This section defines the problem (3.2.1), then develops the algorithm in several steps. We define the generative model of on-policy and off-policy values (3.2.2), show how to perform inference in this model by way of example (3.2.3), and finally combine the results into an explicit algorithm (3.2.4).

3.2.1 Problem Definition

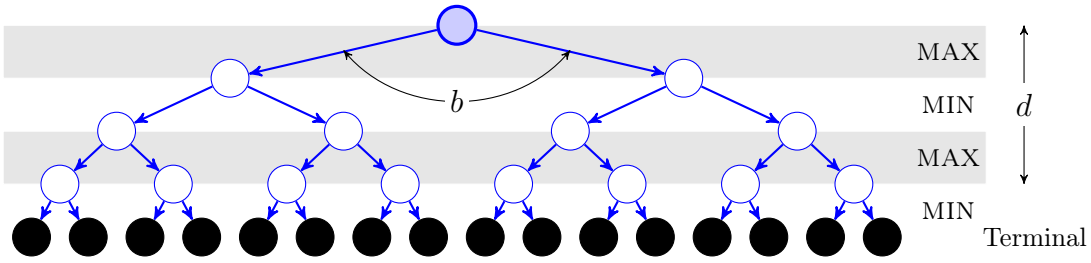


Figure 3.1: A zero-sum, two-player, round-based game represented by a game tree. Terminal positions are shown as filled nodes, nonterminal positions as hollow circles. The meaning of the branching factor b (here $b = 2$) and tree depth d (here $d = 4$) is indicated by schematic annotations.

We consider a tree-structured graph defining a round-based, loop-free, zero-sum game between two opposing players, MAX and MIN with binary outcomes 1 and -1 — "win" and "loss" from MAX's point of view (Games with real-valued outcomes are in fact an *easier* variant of this problem, because the scores g_t of terminal nodes can be observed directly, rather than just their signs). MAX is defined to be the first player to move.

The task is to predict the outcome of the game, assuming optimal play by both players, from any position in the tree. The only type of data available (at request) is the length $\ell \in \mathbb{N}$ and result $r_i \in \{-1; 1\}$ of random *roll-outs* of the game starting at node i. A roll-out is a path through the tree to a terminal node, generated by a policy choosing stochastically (not necessarily uniformly) among available moves in each encountered position. (The policies of contemporary UCT algorithms do not choose moves uniformly at random. See Section 3.3.4 for experimental evidence that our model can still be approximately valid in this case.)

3.2.2 Generative Model

Two kinds of evaluations for nodes in the game tree will be central to the analysis: The value under the *non-optimal* roll-out policy, known as the *on-policy* value

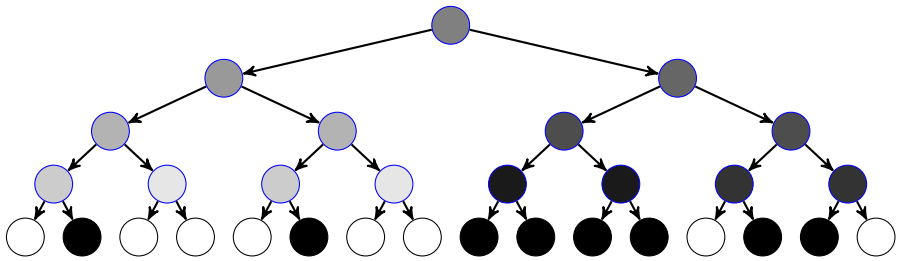


Figure 3.2: Conceptual sketch of game tree smoothness. The binary win/loss values of terminal positions (shown as black/white nodes) are the result of incremental changes in the game score over the course of the game, indicated by grayness levels.

in reinforcement learning, will be called the *score* for clarity. The *optimal* value achieved by two hypothetical ideal players is known as the *off-policy* value in reinforcement learning. The following two definitions jointly form a generative model $\mathcal{M}$ for the latent scores $G = \{g_i\}$ and optimal values $V = \{v_i\}$ of tree nodes i.

Definition: The *score* g_i of node i models the value of i under random play. It is a real number such that

- ▷ for any terminal position t, $\text{sign}(g_t) = r_t$, where r_t is the binary result of the game at t. The likelihood for g_t is thus a step function (denoted θ):

$$p(r_t \,|\, g_t) = \theta(r_t g_t). \tag{3.1}$$

- ▷ the score of child node c of node i is generated from g_i by a zero mean, unit variance Gaussian step (see also note below):

$$p(g_c \,|\, g_i, \mathcal{M}) = \mathcal{N}(g_c; g_i, 1). \tag{3.2}$$

- ▷ the prior for the score of the root node is Gaussian $p(g_0) = \mathcal{N}(\mu_0, \sigma_0^2)$ (one is free to choose μ_0 and σ_0, although $\mu_0 = 0$ is the obvious choice).

Thus, scores of sibling nodes are independent given their parent's score, and the prior distribution of the value during a roll-out is a Brownian random walk. For binary results, the scale factor of the steps is arbitrary, and so is the choice of 1 here. In the simpler case where the actual real value of a terminal position is observed (which will not be considered further here), this step size obtains a meaning, and should then be learned. How to do this will follow straightforwardly from the following derivations in Section 3.2.3: It involves inferring the variance of values at the leaf nodes (which is trivial from the observed data), and dividing that number by the depth of the tree.

Definition: The *off-policy value* v_i of node i is the true value of node i under optimal play by both players. That is

- ▷ for terminal positions t, we have $v_t = g_t$.

- ▷ for non-terminal positions i with children $\{c\}_i$

$$v_i = \begin{cases} \max_c\{v_c\} & \text{if MAX plays at } i \\ \min_c\{v_c\} & \text{if MIN plays at } i. \end{cases} \tag{3.3}$$

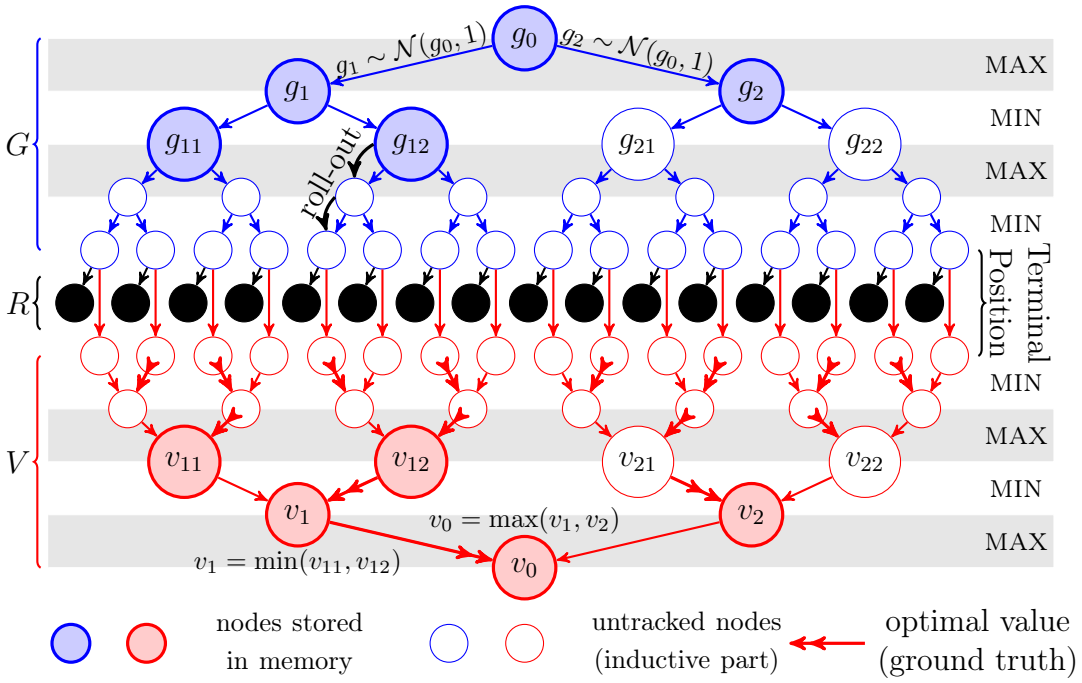


Figure 3.3: Generative model (Bayesian network, directed graphical model) for the scores G, the roll-out results R and the optimal values V of an example game represented by a binary tree of depth 4. Shown is the situation after four search-descents into the tree, the blue and orange shaded nodes represent the representation of the exploratory tree in the algorithm's memory. The most recent roll-out (of length 2) is shown as two black curved arrows, previous roll-outs (from nodes (1), (2) and (11)) are not shown. The lower half of the diagram shows the generative process of optimal values v. In a minor deviation from the standard notation for directed graphical models, ground truth identity of the optimal paths through the tree (not observed by the algorithm) is indicated by thick arrows with double heads (i.e. optimal play consists of MAX moving from node 0 to node 1, then MIN moving to 12, etc.). This is only for intuition, the generative model itself just consists of the nodes and arrows. Note that, while the v nodes are shown below the g nodes here for readability, in the text 'up' and 'down' refer to the game tree structure, with parent nodes v_i being 'above' children v_{ij}.

We note in passing that OR-trees (i.e. tree-structured optimization problems, and non-adversarial games like Solitaire) are a trivial variant of this formulation.

Figure 3.3 shows the full generative model for a small tree of $b = 2$ and $d = 4$, as a directed graphical model. Note that the only observed data is the sign (the binary value) of terminal positions (black nodes); all other variables are latent.

In some games — such as Chess, see e.g. Campbell et al. [2002] — noisy value estimates for non-terminal positions might be available from some evaluation function. This situation will not be studied here, but in many cases it might be possible to include such information in a principled way. In fact, if the observation noise can be considered Gaussian with known mean and variance, incorporating such data is a straightforward case of belief propagation (see Section 2.2).

3.2.3 Inference

We will use the results from 2 to derive an approximate message passing algorithm to perform inference both on the scores and values, using Gaussian Expectation Propagation (Section 2.3.2) to project the messages to the normal exponential family.

Inspecting Figure 3.3, it might seem like inference in the tree would call for messages among all b^d nodes. In this section, we will show that this can be avoided, because the messages from unobserved parts of the tree can be derived *a priori*, in jointly $\mathcal{O}(bd)$ time.

We assume the learner acquires data in a best-first manner: At any given point in time, it tracks an asymmetric but contiguous tree in memory which includes the root. Additional nodes are added to the boundary of the stored tree by requesting a roll-out from that node (Figure 3.3, shows the situation after four roll-outs, with four (blue/orange shaded) nodes already added to the memory representation). The message passing is performed in parallel with the search process. The resulting message passing schedule is quite complex. For clarity, we will use an example descent through the small tree of Figure 3.3.

On-Policy Inference on g

Each search descent begins at the root node 0. As the descent passes through the tracked part of the tree, a policy π chooses among available children, potentially based on the current beliefs over their v. For our example, say the policy chose the descent $0 \to 1 \to 12$. At each step, we update the message from the current node to the chosen child. The message out of g_1 in the direction of g_{12} is

$$m_{\mathrm{pa}(1)}(g_1) \prod_{j \in \mathrm{ch}(1) \backslash 12} m_{g_j}(g_1) \equiv \mathcal{N}(\mu_{1\backslash 12}, \sigma^2_{1\backslash 12}) \tag{3.4}$$

where ch(1)\12 is the set of child nodes of 1 excluding node 12, and pa(1) is the parent node (i.e. 0) of 1. And the message into g_{12} is

$$\begin{aligned} m_{g_1}(g_{12}) &= \int p(g_{12}|g_1) p(g_1|g_{\mathrm{pa}(1)}, \{g_{\mathrm{ch}(1)\backslash 12}\}) \, \mathrm{d}g_1 \\ &= \mathcal{N}(\mu_{1\backslash 12}, \sigma^2_{1\backslash 12} + 1) \end{aligned} \tag{3.5}$$

Assume node 12 just reached is not part of the stored tree yet. To add it to the tree, we request a roll-out starting from 12, which turns out to be of length $\ell = 2$ and have result $r = +1$ (see Figure 3.3). The data thus gained is a likelihood $p(r|g_t)$ of the score of the terminal position t at the end of the roll-out, which is a step-function. We can generate a prior over g_t as a message from the current

marginal over g_{12} by integrating out the ℓ intermediate steps (see Equation (3.5)):

$$p(g_t \mid g_{12}) = \mathcal{N}(\mu_{12}, \sigma_{12}^2 + \ell) \tag{3.6}$$

giving a posterior over g_t which is a truncated Gaussian. To get the EP message back to g_{12}, we need the function that calculates the moments of this truncated Gaussian distribution (this function will be denoted $f_{\text{trG}}^{\text{EP}}$ in Algorithm 1). The calculation is straightforward, because for Gaussians in general

$$\int_0^\infty x\mathcal{N}(x;\mu,\sigma^2) = \mu\Phi\left(\frac{\mu}{\sigma}\right) + \sigma\phi\left(\frac{\mu}{\sigma}\right) \text{ and}$$
$$\int_0^\infty x^2\mathcal{N}(x;\mu,\sigma^2) = (\mu^2+\sigma^2)\Phi\left(\frac{\mu}{\sigma}\right) + \mu\sigma\phi\left(\frac{\mu}{\sigma}\right)$$

where $\phi(x) = \mathcal{N}(x;0,1)$ is the standard Gaussian and $\Phi(x) = \int_{-\infty}^{x} \phi(y)\,\mathrm{d}y$ is the cumulative Gaussian. This result was used previously for EP by Herbrich et al. [2007]. To finally arrive at the message $m_{r_{12}}(g_{12})$ from the roll-out to g_{12}, we need to apply $f_{\mathcal{N}(0,\ell)}$ again to the resulting EP message. Note that the message contains no g other than g_i. It is thus possible to perform inference on g_i using exactly two messages: One from its parent node, and one from the outcome of the roll-out.

To incorporate the new knowledge from this roll-out into all ancestor nodes of g_{12}, we pass messages of analogous form to Equation (3.5) back *up* the tree. This obviously does not propagate the information through the whole tree, but it leads to a situation where the score g_i of any node i in the tree can be evaluated in linear time, simply by performing one descent from the root towards i, updating messages analogously to Equation (3.5) downwards during the descent.

There is one more pitfall to avoid: Consider the next time a search descent passes through node 12, which currently lies at the boundary of the tracked tree. This leads to the addition of a child node of 12 and a roll-out from there. Now the roll-out result associated with 12 has to be dealt with in a consistent way. Simply keeping the corresponding message in the marginal is not correct: Because the roll-out necessarily passed through one of i's children, information from that child would otherwise be counted twice. There are two other options: If information about the course of the roll-out was stored, the corresponding message can be moved down along the path of the roll-out to the boundary of the search tree. If the amount of roll-outs played is too large to store the paths of all roll-outs, it becomes necessary to remove the information gained from the roll-out at node 12 from the marginal. This removal corresponds to 'dividing' the corresponding message out of the marginal as discussed in Sections 2.3.1 and 2.3.2. In collecting the experimental results reported in Section 3.3.2, we opted for this latter, more memory-conservative (but slightly information-wasting) approach, to facili-

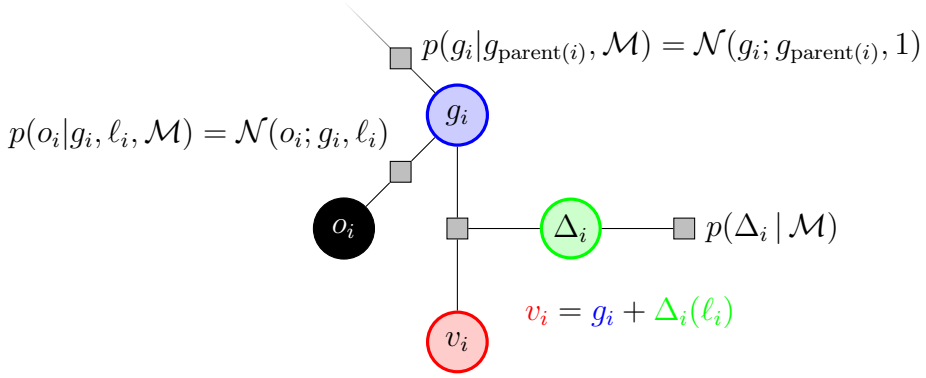


Figure 3.4: Inference on optimal values v_i separates into two independent parts: The value g_i of playing to node i, and the value Δ_i of playing *optimally* from node i onwards. Inference on g_i is *deductive*, because it uses the roll-out data o_i, while inference on Δ_i uses the model only, and is thus *inductive*.

tate comparison to contemporary Monte Carlo tree search algorithms.

Inference on Optimal Values v

The previous paragraph sketched the message passing inference leading to a consistent posterior over scores G. How can these beliefs be used to obtain a posterior over the values V? We will use the definitions of Section 3.2.2 to again derive a message-passing scheme running parallel to the search process.
First, consider again node $i = 12$ in Figure 3.3, at the boundary of the stored tree. The value under optimal play is the sum of the two independent variables (see factor graph in Figure 3.4)

$$v_i = g_i + \Delta_i \tag{3.7}$$

where Δ_i is the optimal reachable *increment* to the score of i. So the inference breaks up into a deductive part (on g_i, as solved in the previous section) and an inductive part (on Δ_i). If the next move after i is controlled by MAX (replace max with min in the opposite case), then

$$\Delta_i = \max_{j \in \text{children}(i)} \{\xi_j + \Delta_j\} \tag{3.8}$$

where $\xi_j \sim \mathcal{N}(0, 1)$ is the unknown Brownian step to the score of node j. Deriving a belief over Δ_i for any node i, which is ℓ_i steps from a terminal position, is a recursive problem which depends only on ℓ_i and the branching factor b: Assuming MIN gets the last move (with straightforward variations in other cases), Δ_i is

$$\Delta_i(\ell_i, b) = \begin{cases} 0 \\ \max_{j=1\ldots b} \{\Delta_j(\ell_i - 1, b) + \xi_j\} \\ \min_{j=1\ldots b} \{\Delta_j(\ell_i - 1, b) + \xi_j\} \end{cases} \tag{3.9}$$

for $\ell = 0$, for $\ell \bmod 2 = 0$ and for $\ell \bmod 2 = 1$, respectively.

Similarly to the situation in the previous section, the beliefs generated by this recursive operation are not Gaussian themselves. To perform the EP approximation, we need the function calculating the moments of the maximum or minimum over Gaussian variables (this function will be denoted $f^{\mathrm{EP}}_{\max\,/\,\min}$ in Algorithm 1). The necessary derivations are lengthy and have thus been moved to Appendix A, where these moments are derived for the case of the maximum of two Gaussian variables, and it is shown how to combine such binary comparisons iteratively into an approximate posterior belief over the maximum of a finite set of such variables. The corresponding messages for the minimum of variables is a trivial variant, because $\min_i\{x_i\} = -\max_i\{-x_i\}$.

Using this approximation, we arrive at a recursive operation in closed form, which can be used to derive the message from $\Delta(\ell)$ for all ℓ up to a pre-defined depth. Perhaps surprisingly, this lookup table for optimal increments can be constructed prior to data acquisition, once for the entire game tree, in $\mathcal{O}(bd)$ time (as opposed to the $\mathcal{O}(b^d)$ cost of probing the entire tree), which is easily tractable even for massive game trees like that of 19×19 Go. To see this, note that the optimal value increments from nodes one level above the leafs to the leafs are the maximum (or minimum) of b unit-variance, zero-mean Gaussian random variables. The above approximation gives a new Gaussian approximation for the belief over the value of this maximum (or minimum) that is identical for *all* these nodes. So the fact that there are b^{d-1} of these nodes is irrelevant for this question.

In my simple, non-optimized implementation, constructing this table takes about 2 minutes on a contemporary desktop machine, for a tree of Go-like dimensions (10^{400} nodes). This step is a parametrized version of the Monte Carlo technique known as *density evolution* (see e.g. [MacKay, 2003, §47.5] and Richardson and Urbanke [2008, §4.5]). See Section 3.3.3 for an experimental analysis of the quality of the approximation.

We sum the independent variables g_i and Δ_i, using the exact function

$$f_{\sum\mathcal{N}}\left[\mathcal{N}(\mu_a, \sigma_a^2), \mathcal{N}(\mu_b, \sigma_b^2)\right] = \mathcal{N}(\mu_a + \mu_b, \sigma_a^2 + \sigma_b^2). \tag{3.10}$$

For a node j that does not lie on the boundary of the stored tree, v_j is given by Equation (3.3). Marginals $p(v_{c_j})$ are available for all children c_j of j in this case. Hence, an approximate Gaussian message to v_j from its children can be found using the same method as above. However, it is important to note that the children of v_j are correlated variables, because they are all of the form shown in Equation (3.7), sharing the contribution g_j. The EP equations derived in Appendix A include the

correlated case. Using approximate Gaussian messages

$$q(v_k) = \mathcal{N}(\mu_k, \sigma_k^2) = \mathcal{N}(\mu_{g_j} + \mu_{\Delta_k}, \sigma_{g_j}^2 + \sigma_{\Delta_k}^2) \tag{3.11}$$

for each child k of j, the correlation coefficient[2] $\varrho_{k_1 k_2}$ between two children k_1 and k_2 is

$$\varrho_{12} = V^{-1}(\sigma_{g_i}^2 - \mu_{g_i}\mu_{\Delta_{k_1}} - \mu_{g_i}\mu_{\Delta_{k_2}} - \mu_{\Delta_{k_1}}\mu_{\Delta_{ik_2}}) \qquad \text{where}$$
$$V \equiv (\sigma_{g_i}^2 + \sigma_{\Delta_{k_1}}^2 - 2\mu_{g_i}\mu_{\Delta_{k_1}})^{1/2} \cdot (\sigma_{g_i}^2 + \sigma_{\Delta_{k_2}}^2 - 2\mu_{g_i}\mu_{\Delta_{k_2}})^{1/2}$$

3.2.4 Algorithm

Algorithm 1 sums up the message-passing scheme presented in the previous sections. It defines a recursive function that descends through the tracked tree to a leaf, passing messages downward (the operator $f_{\mathcal{N}(0,1)}(p)$ refers to Equation (3.5)). To choose the part of the tree to explore, the algorithm uses a generic policy π (line 3), whose precise form can be arbitrary. In particular, the policy may or may not make use of the algorithm's value estimates (see Section 3.2.6 below). At the boundary of the stored tree, the algorithm performs a roll-out (line 14), then passes g and v messages upwards to the root. The actual top-level search algorithm repeatedly calls this function, accumulating more and more data, at roughly constant computational cost per call (apart from the small increase in cost caused by the growth of the stored tree). The notation $\mathrm{pa}(i)$ and $\mathrm{si}(i)$ refers to the parent of i and the set of siblings of (and including) i, respectively. The function STORED accesses a one-dimensional array of stored inductive messages from the unexplored parts of the tree, as discussed in Section 3.2.3.

[2]The correlation coefficient ϱ_{ij} between two Gaussian variables i and j is defined by $\mathrm{cov}\,(ij) = \varrho_{ij}\sqrt{\mathsf{Var}\,(i)\,\mathsf{Var}\,(j)}$.

Algorithm 1 Bayesian Best-First Tree Search

1: **procedure** DESCENT(i)
2: **if** i previously visited **then**
3: $c \leftarrow \pi(i)$ ▷ policy π chooses child c to explore
4: $p(g_c) \leftarrow p(g_c)/m_{g_i}(g_c)$ ▷ update message to c
5: $m_{g_i}(g_c) \leftarrow f_{\mathcal{N}(0,1)}[p(g_i)/m_{g_c}(g_i)]$
6: $p(g_c) \leftarrow p(g_c) \cdot m_{g_i}(g_c)$
7: $m'_c(g_i) \leftarrow \text{DESCENT}(c)$ ▷ continue descent (returns g message from child)
8: $p(g_i) \leftarrow p(g_i)/m_c(g_i) \cdot m'_c(g_i)$ ▷ update marginals
9: $m_c(g_i) \leftarrow m'_c(g_i)$
10: **else**
11: $p(g_{\text{pa}(i)}) \leftarrow p(g_{\text{pa}(i)})/m_{r_{\text{pa}(i)}}(g_{\text{pa}(i)})$ ▷ divide out roll-out from parent's marginal
12: $m_{r_{\text{pa}}}(g_{\text{pa}(i)}) \leftarrow \mathcal{N}(0, \infty)$
13: [lines 3 to 5] ▷ update message from parent to g_i, identical to above
14: $(r_i, \ell_i) \leftarrow \text{ROLL-OUT}(i)$ ▷ do roll-out
15: $m_{r_i}(g_i) \leftarrow f_{\mathcal{N}(0,\ell_i)}\left[f^{\text{EP}}_{\text{trG}}(r_i, f_{\mathcal{N}(0,\ell_i)}[p(g_i)])\right]$ ▷ build message from roll-out result to i
16: $p(g_i) \leftarrow p(g_i) \cdot m_{r_i}(g_i)$
17: $p(v_i) \leftarrow f_{\sum \mathcal{N}}[p(g_i), \text{STORED}(\ell_i)]$ ▷ generate marginal for v_i
18: **end if**
19: $m_i(g_{\text{pa}(i)}) \leftarrow f_{\mathcal{N}(0,1)}(p(g_i)/m_{\text{pa}(i)}(g_i))$ ▷ Calculate messages to parent's g and v
20: $p(v_{\text{pa}(i)}) \leftarrow f^{\text{EP}}_{\max\,/\,\min}(\{p(v_k)\}_{k \in \text{si(i)}}, p(g_i))$
21: **return** $m_i(g_{\text{pa}(i)})$
22: **end procedure**

3.2.5 Replacing a Hard Problem with a Simple Prior

Figure 3.5 provides an intuition for the computational simplification achieved with the probabilistic model and inference algorithm introduced in the previous sections. Finding the exact answer to the tree search problem is an exponentially hard problem. But finding a good path through the tree — estimating the value of a position under optimal play — is a much simpler problem *if* we allow ourselves the luxury of a probabilistic model for the tree structure. With this model, inferring a *belief* (rather than an exact statement) over the value of any node in the tree has only linear computational cost. This also provides some insight into why humans can play exponentially hard problems like Go, despite their ostensibly incredible structural complexity: Humans do not predict entire games from the start: They think ahead a few moves, and assume that the game proceeds "as usual" (i.e. as described by some simple model) from there on. In classic tree search algorithms, this intuition is encoded as a hard evaluation function by the designer of the algorithm. In games where such an evaluation is hard to obtain, such as Go, random roll-outs combined with the inference algorithm presented above provide a means of *learning* the evaluation function.

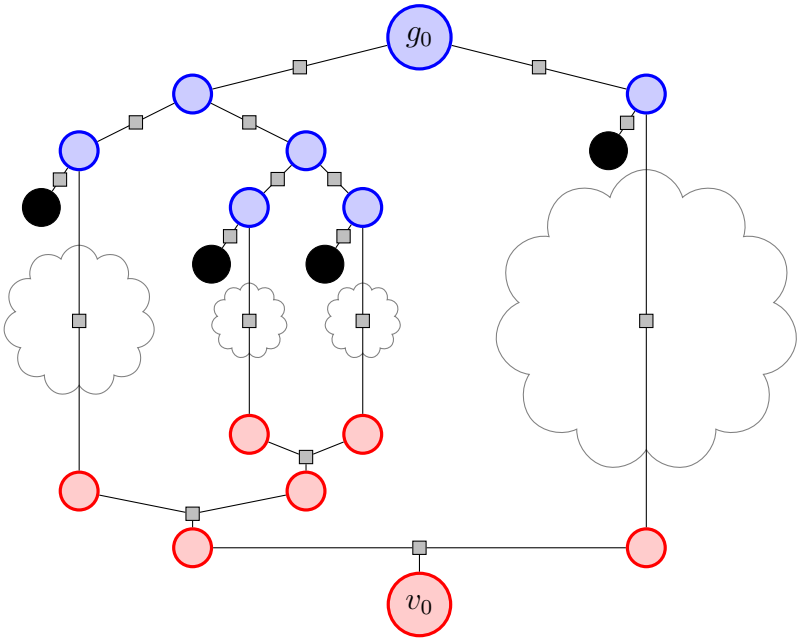


Figure 3.5: Factor graph illustrating the advantage of a probabilistic model over classic tree search algorithms: the exponentially large tree from Figure 3.3 (indicated by gray clouds) has been replaced, literally, with black boxes modelling its effect on optimal play. The exponentially hard problem of identifying the optimal path through the tree is replaced with the linear-cost problem of inferring a *belief* over the optimal path.

3.2.6 Exploration Policies

While the inference process itself is independent of the chosen policy π in Algorithm 1, the policy is crucial to make the roll-outs informative about the optimal path (this is a general feature of off-policy reinforcement learning). Many possible policies are available, among them the point-estimate based UCT [Kocsis and Szepesvári, 2006], greedy choice among samples [Thompson, 1933], and information gain [Dearden et al., 1998]. An approach that has received some renewed interest recently [Kolter and Ng, 2009] is 'optimistic' exploration based on weighted sums of mean and variance of the value estimates. That is, given approximate beliefs $q(v_i) = \mathcal{N}(\mu_i, \sigma_i^2)$ over the optimal value of children i, the policy chooses as

$$\pi_{\text{optimistic}}[q(\boldsymbol{v})] = \arg\max_{\text{children } i}(\mu_i + \beta\sigma_i) \tag{3.12}$$

with a parameter β controlling between exploration and exploitation. In our experiments, we used this optimistic exploration where applicable, but it should be understood that this is essentially an arbitrary choice, and the 'right' choice of policy is still a matter of open debate in the reinforcement learning community.

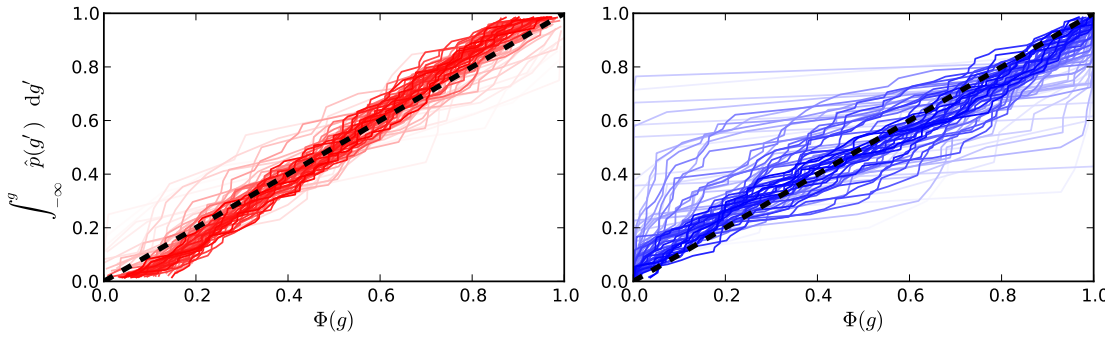


Figure 3.6: Quantile-quantile plot of 71 empirical distributions of g, centered on their mean, at varying depth from the root, for one random path through Go, against the standard normal distribution $\Phi(x)$ (black dotted line). **Left:** Random roll-out policy. **Right:** Smart roll-out policy. Color intensity of the plots decays linearly with depth from the root. Note that the distributions far from the root are based on increasingly small sample sizes.

3.3 Results

We performed experiments to answer several questions arising with regard to the described inference model: Is the model of Brownian motion applicable for real games, like Go (3.3.1)? Is the EP approximation on the roll-out results effective (3.3.2)? Does the recursive min/max approximation in the inductive part of the inference produce a reasonable approximation of the true MIN/MAX values (3.3.3)? How robust is the model to mis-match between generative model and ground truth (3.3.4)? And could the algorithm be used as a standalone searcher (3.3.5)?

3.3.1 Structure of Go Game Trees

We generated one random path through the tree of 9×9 Go. At each level in this path, roll-outs from all legal moves i at this position were generated (1000 roll-outs from each i). Depending on the depth from the root, there were between 81 and 0 such legal positions. We stopped the game after 71 steps, when there were less than 5 legal moves available. The average length $\bar{\ell}_i$ of the roll-outs and the empirical frequency $\hat{p}(\text{win}|i)$ of a win for the MAX player from i under a random roll-out policy was stored. This implicitly defines the value of g_i under the model through

$$p(\text{win}|g_i) = \int_0^\infty \mathcal{N}(g_t, g_i, \ell_i)\, \mathrm{d}g_t = \Phi\left(\frac{g_i}{\sqrt{\ell_i}}\right) \tag{3.13}$$

where we replace $p(\text{win}|g_i)$ and ℓ_i with empirical averages. With sufficiently many samples, $p(\text{win}|g_i)$ and thus g_i can be evaluated up to negligible error. The generative process defined as part of our model $\mathcal{M}$ then leads to the statement that the $\{g_i\}$ of sibling nodes in the tree are distributed like the standard normal distribution around their parent's value. Figure 3.6 shows Q-Q plots of these empirical

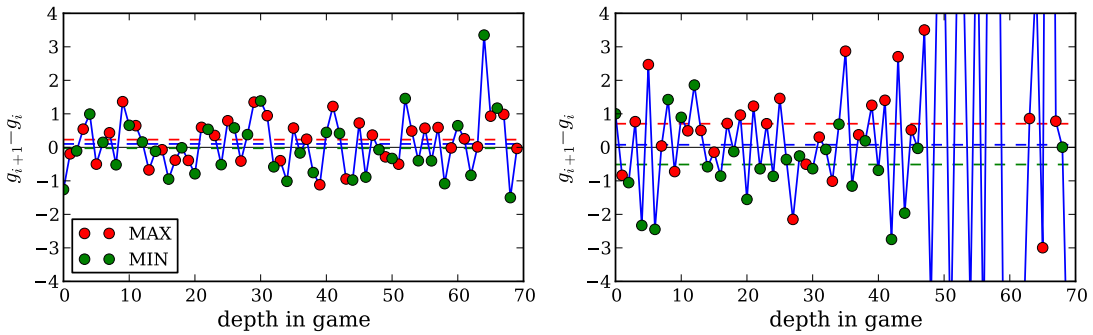


Figure 3.7: Change to means of positions' values during roll-outs, as a function of depth in tree (i.e. this plot shows the effects explicitly removed in Figure 3.6). **Left:** random roll-out policy. **Right:** Smart roll-out policy. Nodes controlled by MIN player in green, nodes controlled by MAX player in red. Connecting blue lines for visual aid only. Means over entire dataset, MAX's moves only, and MIN's moves only as dashed blue, red and green lines, respectively (For the smart policy, all these averages only include data points up to move 48). Note the strong amplitudes at the end of the game for the smart policy. The data points in this region, cut off in this plot to aid readability, reach values of up to -14/+31.

distributions against Φ for depths $d = 0$ to 71 from the root. A standard normal would lie on the diagonal in this plot; weaker tailed distributions are steeper, heavier tailed ones flatter. The left plot shows results from a uniform roll-out policy (only excluding illegal and trivially bad 'suicide' moves, which are not technically illegal, but would never be played by a human player, and are cheap to test for). Given the limited sample sizes, especially towards the end of the game, the empirical distributions are strikingly similar to $\Phi(x)$.

Contemporary Monte Carlo search algorithms use 'heavy roll-outs', i.e. policies that produce less random, more informative results. Clearly, the generative part of our model will be more and more invalid the smarter the policy — the limit of a perfect policy would repeatedly generate only a single perfect roll-out, and it is unlikely that this roll-out would conform to the assumptions for randomly generated games made in the generative model. To examine how drastic this effect is, we repeated the above experiment with a smart policy, similar to the published parts of MoGo [Gelly et al., 2006] (Figure 3.6, right). The results do develop heavier tails, particularly deep in the tree, towards the end of the simulated game. The reason is that in this late phase, only few good moves remain, and the roll-outs under the smart policy become very similar to each other. Based on Figure 3.6, one could argue that the Gaussian generative model remains an acceptable approximation. Alternatively, one could of course search for a better generative model for any particular roll-out policy used; example approaches might include heaver-tailed distributions like the Laplace distribution.

The analysis so far has only considered the distribution of children *relative* to their parent nodes, and explicitly excluded the drift of the mean value g_i from one

node to the next chosen child. Figure 3.7 shows the relative difference $g_{i+1} + g_i$ between subsequent moves, as a function of the depth in the tree, again for the uniformly random and "smart" roll-out policy. For the random policy, subsequent means have no clear tendency towards one particular direction. An analysis of variance shows not enough evidence for the rejection of the null hypothesis that either player's increments come from the same distribution (p-value 0.19). For the smart policy, a clearer tendency to choose moves favouring the player's desired outcomes is apparent. More importantly, in the final twenty moves of the game, the step sizes become extreme, as it becomes possible for the policy to choose very good moves based on its heuristic alone. This effect is also reflected in the extreme step-shaped cumulative density functions in visible in Figure 3.6. In this late phase of the game, the generative model presented here arguably becomes invalid under this particular smart roll-out policy. Note, however, that this very deviation from the model also suggests that this part of the game tree has a simpler structure that can in fact be modeled well with heuristics. The more benign tendency to choose moves changing the score towards the direction favoured by the current player evident in the earlier phase of the game can be modeled straightforwardly by introducing a player-dependent bias-term in the mean for the generative step in Equation (3.2).

3.3.2 Inference on the Generators

A good way to evaluate the quality of Bayesian models is the (log) likelihood they assign to ground truth in known test environments. To do so, we generated 500 artificial trees of $b = 2$, $d = 18$ from $\mathcal{M}$. The inference model was implemented as presented in Algorithm 1. A time step corresponds to one descent into the tree, ending with a roll-out at a previously unexplored node. For the descent through previously visited nodes, an optimistic policy was used (see Section 3.2.6), choosing greedily among children i based on $\mu_{v_i} + 3\sigma_{v_i}$. In addition, the value of the root node's score was assumed to be 0 with high precision. Figure 3.8 shows the log likelihood assigned to the ground truth of both g and v at distances 1, 2 and 3 from the root, as a function of the number of roll-outs performed

As expected, the likelihoods rise on average during learning. They saturate when the majority of the nodes is expanded, because the (binary) roll-out results do not contain sufficient information to determine g, and thus v, to arbitrary precision. Nodes deeper in the tree saturate at smaller likelihoods because they receive information from fewer offspring nodes. They also start out with a smaller likelihood because their priors contain more uncertainty. Note that the message passing causes the beliefs to develop simultaneously at all three depths, even though the nodes at greater depths are not explored until several descents after initialization.

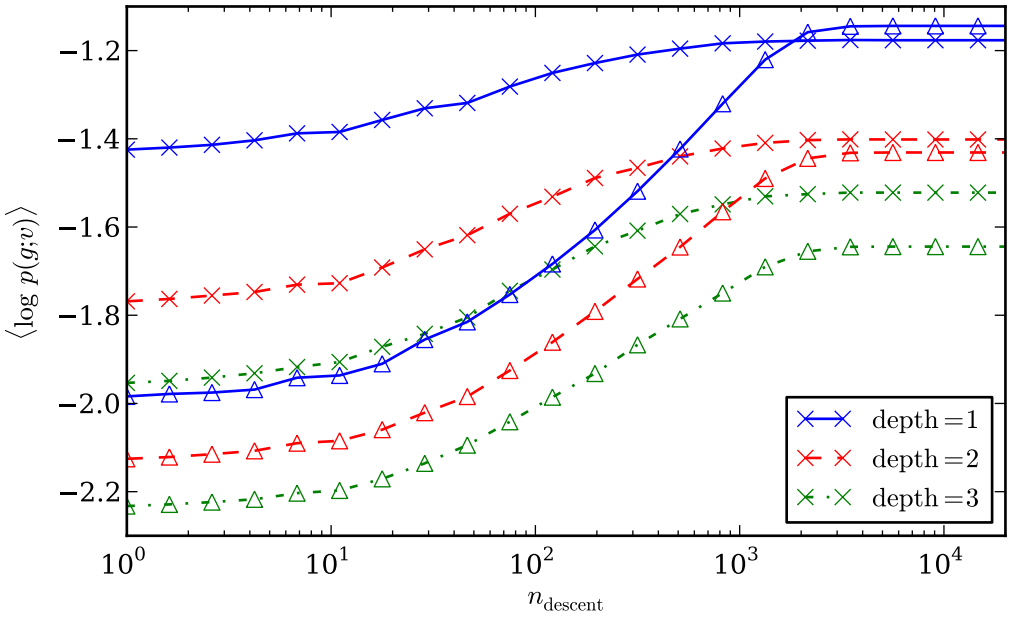


Figure 3.8: Log likelihood of the ground truth value of g_i (crosses) and v_i (triangles) under the beliefs of the model, at varying depth from the root in artificial trees, as a function of number of roll-outs performed. Averages over all nodes at those depths.

3.3.3 Recursive Inductive Inference on Optimal Values

To evaluate the quality of the inductive part of the approximation, 1000 artificial game trees were generated and solved by explicit min/max search as in the preceding section. Figure 3.9 compares empirical ground truth of Δ and values predicted by pre-data inference, for all nodes at two different distances ℓ from the leaves, in 1000 artificial game trees. Despite the repeated application of the Gaussian approximation to non-Gaussian beliefs, there is good agreement between predictions and ground truth.

3.3.4 Errors Introduced by Model Mismatch

For the last two experiments, the artificial game trees were generated by the generative model $\mathcal{M}$, and we showed in Section 3.3.1 that the real-world game Go is in fact approximated well by this model. However, other games might be less well approximated by $\mathcal{M}$. As a tentative test of the severity of the errors thus introduced, the Bayesian searcher was tested on 500 generated p-game trees [Kocsis and Szepesvári, 2006] with $b = 2, d = 18$. These trees are also generated by a latent stochastic variable, but with a different generative model, choosing ξ_i uniformly from $[-1, 0]$ if the player is MIN and uniformly from $[0, 1]$ if the player is MAX. We performed a best-first search in these trees (results not shown), and compared to the performance on trees for which $\mathcal{M}$ is the correct model. The performance

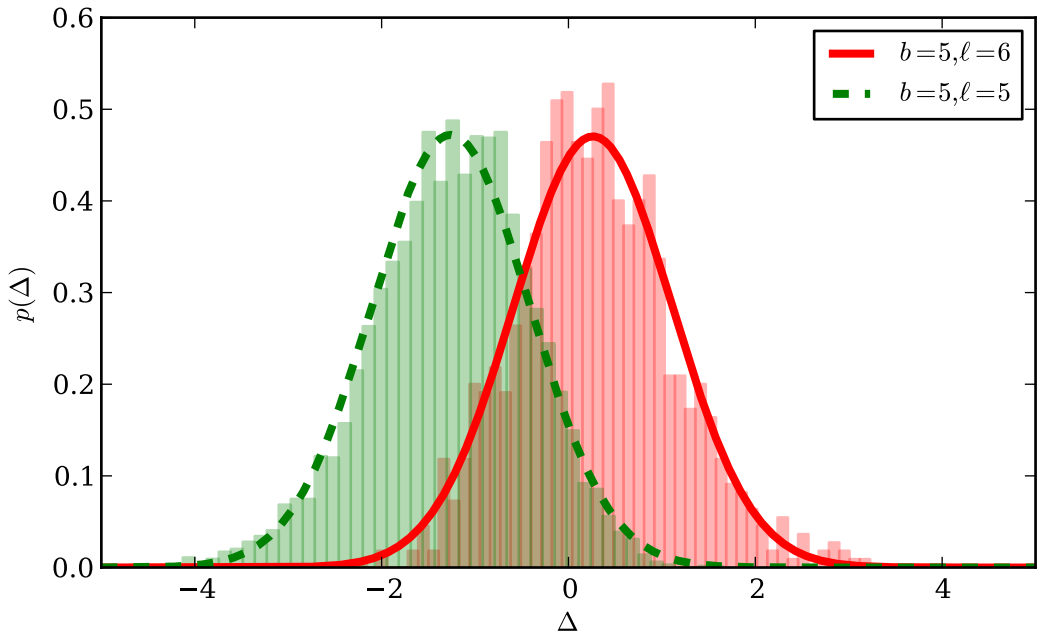


Figure 3.9: Empirical histogram and model predictions (lines) for the optimal future value increment Δ (see Equation (1.1)), for $p(\Delta|\ell = 6, b = 5)$ (first move for MAX, last move for MIN) and $p(\Delta|\ell = 5, b = 5)$ (first and last move for MIN). Model predictions as lines.

on these two types of trees during the search was very similar (i.e. a corresponding plot looks very much like Figure 3.10), except for a globally slightly higher chance of the model misclassifying nodes into winning and losing nodes. At least in this particular case, the model mismatch causes only minor decay in performance.

3.3.5 Use as a Standalone Tree Search

It is tempting to interpret the presented inference algorithm as a standalone search method. Experiments on artificial game trees (Figure 3.10) suggest that the resulting algorithm does not necessarily improve on a vanilla UCT searcher, and that performance depends strongly on the policy used. The intent of the presented algorithm is not to develop a good tree searcher, but to provide a consistent posterior from which generalizing evaluation functions can be learned.

3.4 Conclusion

We have presented a generative model for game trees, and derived an approximate message-passing scheme for inference on the optimal value of game positions, given samples from random roll-outs. Inference separates into two tractable deductive and inductive parts, and allows evaluation of the marginal value of any node in the

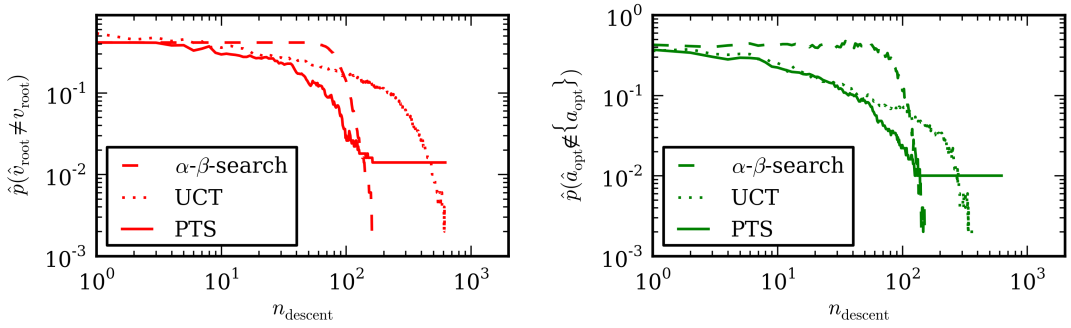


Figure 3.10: Performance in a tree search task. Comparison between α-β search, standard UCT and a probabilistic tree search algorithm (PTS) based on our model, as a function of number of evaluated leaf nodes / descents. Averages over a set of 500 artificially generated game trees from the model, with $b = 5$, $d = 4$ (larger trees are very challenging for α-β). **Left:** Probability of the models predicting the correct binary value (win/loss) of the root node of the tree (α-β assigns random values with uniform probability before convergence, the exact value afterwards. The smooth development of the corresponding curve is an effect of averaging). Note that, due to the optimistic policy, the probabilistic algorithm is not guaranteed to actually explore the entire tree and thus sometimes does not observe enough data to converge, leading to the about 1% error rate in the limit of many descents. **Right:** Probability of the algorithm's policy *not* choosing the correct optimal next move from the root.

tree in linear time. Similar to the way humans think about games, the computational cost of our algorithm depends more directly on the number of data collected than on the size of the game tree.

Like any model, the assumption of Brownian motion is imperfect. Neither does it catch all the details of any one game, nor does it necessarily apply to all games. But it provides a quantitative concept of a smoothly developing game between competing players. Our experimental results suggest that errors introduced by model mismatch are not severe, and that this is a meaningful model for Go. We argue that it lies at a 'sweet spot' between unstructured priors and non-probabilistic, rule-based methods, which can over-fit.

Retaining a posterior does not necessarily improve on point-based methods in the search for the optimal next move. Nevertheless, the results presented here are valuable in two ways: humans use sophisticated feature-based concepts to play games, not full tree search. Emulating this behavior with a Bayesian algorithm, learning an evaluation function of features of the game position, requires probabilistic beliefs over the optimal values, which our algorithm provides, decoupling the task of designing an evaluation function from that of data acquisition. Secondly, game trees are discrete planning tasks (hierarchical bandit problems). Our results indicate that such problems can exhibit structure (in this case, correlation) even if they might ostensibly look unstructured, offering potential for considerable performance increase.

3.5 *Addendum:* Related Subsequent Work

The work presented in the preceding sections of this chapter was carried out in 2009 and, after an initial unsuccessful submission to NIPS 2009, published in AISTATS 2010 (see citation at beginning of chapter). Three months after the publication of our paper, Tesauro et al. [2010] published a paper at UAI 2010 titled *Bayesian inference in Monte-Carlo tree search* in which they independently arrived at very similar results to ours. Their paper focused on guiding tree search more than on providing good absolute predictions of minimax values, but the algorithm shares many core aspects of our work, including the Gaussian moment matching for inference on minimax values.

The main difference between the two works is that Tesauro et al. do not rely on a generative model for the game tree, but rather describe a tree search algorithm closer to the standard form of UCT. They construct beliefs over the generative score g_i of leaf nodes of the search tree by replacing our inductive step with inference from the binary roll-out results. For this inference, they use a local Beta posterior (moment matched to get Gaussian approximations). They also construct an explicit policy for exploration of the tree, using the current beliefs over minimax values of leaf nodes of the search tree. After a node has been visited N times, their policy explores the child node i with minimax belief $p(v_i) = \mathcal{N}(v_i; \mu_i; \sigma_i^2)$ that maximizes the score

$$s_i = \mu_i + \sqrt{2 \log N} \sigma_i \tag{3.14}$$

The disadvantage of this approach is that it only allows inference on parts of the tree which have already been incorporated into the search tree. The estimates will also be of larger uncertainty as they do not incorporate the additional knowledge about the tree's structure available in our model. Both of these issues are of no concern to Tesauro and colleagues, because their work focuses on guiding the tree search, rather than providing high quality absolute value estimates. Finally since their scheme does not take the depth of the remaining game tree below a leaf node of the search tree into account, one could also conjecture that it might lead to inefficient exploration if different children of a given node have drastically differing remaining trees below them.

However, their algorithm also has some important advantages over ours. Most importantly, it is guaranteed to converge to the exact minimax values as the number of sampled roll-outs approaches infinity, and the authors provide proof-sketches for these guarantees in their paper. The proofs are simple enough to be sketched here in a few lines, and provide some insight into why proving a similar statement is more challenging with regard to our algorithm.

To prove that an algorithm inferring the Bernoulli probabilities π_i of the roll-out results at leaf nodes i of the search tree (not the full game tree), and the minimax

value under these probabilities for nodes within the search tree, converges to the exact minimax probabilities π_i^* for all nodes in the search tree in the limit of large sample numbers, consider the following argument.

- ▷ For the leaf nodes: Using any prior distribution assigning nonzero mass to all $0 \leq \pi_i \leq 1$ (e.g. a Beta distribution) and n_i binary samples with Bernoulli likelihoods, the posterior converges towards the correct π_i^* with the optimal convergence rate of $\mathcal{O}(1/\sqrt{n_i})$. This is a general characteristic of Bayesian inference.

- ▷ Theorem 1 (Tesauro et al.): *Ergodic policies lead to convergence.* Consider a fixed finite bandit tree with binary reward leaf nodes and priors as in the previous point, and assume that no two sibling nodes have the exact same minimax pay-off rates. Then for any policy which visits every leaf an unbounded number of times, the minimax posteriors of all nodes converge to Dirac δ-distributions at the exact minimax values. Proof: by induction, starting from the point made above for leaf nodes. If a node's children all collapse to correct δ's, due to the finite separation between children's values, the parents' values also collapse. This also applies under Gaussian approximation, because the approximation errors vanish as the input Gaussians' variances vanish.

- ▷ Theorem 2 (Tesauro et al.): For the policy of Equation (3.14), the algorithm converges to the exact minimax reward probabilities everywhere in the search tree in the limit of $n_i \to \infty$ for all leaves i. Proof: Show that policy (3.14) is ergodic. This is straightforward because for any node j which is never selected over its siblings i, the variance σ_j^2 will remain constant and thus the score s_j will rise without bound. Hence, the policy visits every node an unbounded number of times. This also applies for the Gaussian approximation, because the approximations variance is strictly positive if the children's variances are positive.

By contrast, our algorithm does not infer roll-out reward probabilities, but optimal values from a model. Of course, in applications where the model is wrong, the inferred values can be arbitrarily wrong. If the model is correct, however, we argue that because our algorithm is based on a probabilistic construction, it is exact up to the effects of the Gaussian approximations made in the message passing. The effects of this approximation are subtle (see also Appendix A), and it is thus difficult to make exact statements about its quality (apart from the trivial convergence characteristics mentioned in the proofs above). As in the other chapters of this thesis, this should not be seen as a defect of the approximation, but as a symptom of the complexity of the underlying model: If it were easy to make general structural

statements about the exact beliefs, we would not need the approximations to begin with.

Chapter 4

Approximate Bayesian Psychometrics

The work presented in this chapter is the result of a collaboration with **Michal Kosinski**, of the Department of Social and Developmental Psychology at the University of Cambridge, who collected the presented data, and **David Stern** and **Thore Graepel**, both of Microsoft Research Ltd. Except where the work of others is explicitly cited, all mathematical derivations and algorithmic implementations are the work of the author of this thesis.

Abstract

This chapter applies approximate Bayesian inference methods to a problem from psychometrics. Specifically, we provide a Bayesian methodology for analyzing psychometric questionnaire data in which people answer tens to hundreds of questions on a five-point ordinal scale. Taking the established item response theory as a starting point, we develop a Bayesian model of ordinal user-item responses and propose an inference algorithm, based on Expectation Propagation, to infer marginal distributions on underlying traits, such as "extraversion" or "neuroticism". We present results on a subset of a very large scale psychometric data set collected on Facebook and illustrate the advantages of the proposed method in terms of predictive accuracy, computational speed, and the availability of error bars in the inferred traits.

4.1 Introduction

Psychometrics, the business of measuring, describing and classifying the human mind, has traditionally relied on the use of statistical methods. The complicated

variations of human behavior, the stochastic noise induced by the often limited sample size, and the unavoidable biases in the selection of experimental subjects make for challenging inference. This might account for the fact that explicitly structured generative models of experimental data have not been applied as widely in psychology as in other fields. This work is an initial attempt at showing how methods from machine learning might be helpful for data description and inference in psychology.

Psychometrics is a subfield of psychology, concerned with the design and evaluation of psychological measurements, such as questionnaires and personality tests. Psychological tests are usually designed based on accepted models of the human mind. That means questions (called *items*) are chosen specifically to measure an explicit latent quantity, such as "extraversion", "agreeableness", "neuroticism", etc. Because such traits are abstract concepts, they are typically defined implicitly through associated responses to items (such as "Do you see yourself as a careless person?" or "Are you inventive?"). The items themselves are often carefully designed to "load" on a single trait only. That is, ideally, every item should co-vary with only one trait. This focus on formulated traits raises the scientific question of how well these latent dimensions describe the actual data, in particular when compared to other models. However, from the scientists' point of view, the latent personality traits are not nuisance parameters here, so integrating them out for prediction is not usually the desired path. Whether this focus on ease of interpretation (rather than information content) and binary item loadings is a good design decision by the community (or even is achievable at all) is a question far beyond the scope of the technical work presented here. For the purpose of this chapter, the modeling assumptions of the test designers will be taken at face value (see Section 4.2 for specific definitions).

Many psychological tests use discrete ordinal answers; for example a range of N choices ranging from "completely disagree" through "indifferent" to "completely agree" (known as *Likert scales* [Likert, 1932]). From these discrete answers, real-valued latent variables associated with personality traits of the user are inferred (such latent scores are often called the *ability* of the respondent in the associated trait, even in tests which are not supposed to be judgmental). This discretized setup is not as simple as it might seem at first sight, as the way in which the ordinal responses are interpreted varies both between individual items and between individual respondents. Apart from scaling and location, classical test theory [Novick, 1966] does not take effects specific to items or respondents into account. The more modern *item response theory* [e.g. Hambleton and Swaminathan, 1990] uses a special case of generalized logistic regression (see Section 4.2), allowing for varying "difficulty" among items, but does not consider variations in the way users interpret the Likert scale. Here, we will construct a probabilistic model for ordinal

responses which, in addition to variations between items, also accounts for varying nonlinear scalings of the Likert response levels among *respondents*. We introduce an approximate inference scheme using Expectation Propagation, which returns an approximate belief over the latent meaning of an answer on a unified scale for all individual respondents.

Bayesian methods have so far occupied a niche among the psychometric literature, and are usually confined to Markov Chain Monte Carlo techniques [Lord, 1986, Patz and Junker, 1999, Arima, 2006], and such treatments are often accompanied by warnings about the computational cost of these methods. With the advent of social networking sites, psychometric tests have become popular among internet users, producing large quantities of data: The experiments presented in Section 4.4 were carried out on a data set comprising $10,000$ respondents, which is only a subset of a much larger data set collected on Facebook.com (the total number of respondents is approximately 4 million). While these large amounts of available data can potentially lead to improvements in test reliability and design, it also creates challenging inference tasks. Approximate Bayesian inference can provide inference results of high quality in acceptable computational time from such large amounts of data, improving on maximum likelihood results by introducing an explicit representation of parameter uncertainty.

As pointed out several times already in this thesis, approximate inference algorithms have the perceived disadvantage of not providing provable performance from a Frequentist point of view. It is thus important to point out that there is no unequivocal "ground truth" in psychometry; so the usual Frequentist framework of proving convergence toward ground truth in the limit of large amounts of data does not seem meaningful. The human mind is too complex to be described by a small number of parameters, so any simple low-dimensional regression model will be patently wrong, whether it be based on point estimates or Bayesian beliefs. In such a situation, insisting on provable convergence towards some ad-hoc population has little value. If anything, such requirements limit the expressiveness of new models by imposing unnecessarily hard requirements on their algebraic structure. We argue that, since the ultimate goal of psychometry is to predict human behaviour, the only fair comparison of descriptive models and associated inference algorithms is given by predictive performance on a test set.

The following Section 4.2 formally introduces the item response problem, and gives a brief introduction to some state-of-the-art methods (a thorough overview of the entire field would be far beyond the scope of this thesis). Section 4.3 extends the classic models to a novel generative probabilistic description of users' responses to items based on low-dimensional latent traits. Section 4.3.2 derives an approximate inference algorithm based on Expectation Propagation. Section 4.4 gives empirical evidence of the increased expressiveness of this scheme relative to maximum

likelihood estimation in contemporary item response models.

4.2 Item Response Theory

The Likert scale is a ubiquitous format for psychological tests, and indeed many other forms of questionnaires. In its general form, it involves a range of R ordered discrete answers, ranging from a strong disagreement to a strong agreement with given statements. The machine learning community is accustomed to this form of data from the widely studied Netflix movie ranking data set [Bennett and Lanning, 2007], where the $R = 5$ discrete answers correspond to "stars" rating a user's satisfaction with a particular movie.

We assume that a given questionnaire consisting of I items has been answered by U respondents (or "users"), providing a data set of $U \times I$ responses $r_{ui} \in \{1, \ldots, R\}$. The goal of inference is to assign a latent score (ability) $x_{uc} \in \mathbb{R}, c \in [1, \ldots, C]$ to $C \leq I$ different *traits* for each user. It is usually assumed that there exists a function $c(i) : [1, \ldots, I] \to [1, \ldots, C]$ assigning items of the questionnaire to only one trait (as pointed out in Section 4.1, this assumption may be questionable in reality. But it is ubiquitous in the literature, and will be taken for granted for the purpose of this work). Since the responses r_{ui} are ordinal, some relationship between r_{ui} and $x_{uc(i)}$ is required. The definition of this relationship usually involves an intermediate real-valued latent variable y_{ui} which is related to $x_{uc(i)}$ in some straightforward way (e.g. through Gaussian noise), and which is assumed to generate the item-specific response r_{ui}.

For example, in the popular *Big Five* factorization model, the item $i =$ "I am always the center of attention at parties." is supposed to test for a trait $c =$ "Extraversion", one of five such traits, and the answer to this item is not assumed to contain any information about the other four traits, labeled "Conscientiousness", "Openness", "Agreeableness" and "Neuroticism". The Big Five model is a longstanding framework of character classification, and was developed over an extended period by a number of authors. Seminal papers include McDougall [1932] and [Cattell, 1943]. See Digman [1990] and Barrick and Mount [1991] for historical reviews. An important detail about the history of this model is that it is in fact based on statistical analysis, rather than philosophical deduction: College students were asked to describe peers with English adjectives; co-occurring clusters of words were identified by factor analysis and then named. Early models had many more than 5 clusters, and the decision to limit $C = 5$ seems to be based on analytical convenience as much as on statistical evidence.

A problem with the Likert scale is that users' responses do not only depend on the latent trait itself, but also on personal interpretation of both the Likert scale's and the items' meaning. Figure 4.1 shows empirical histograms of the marginal

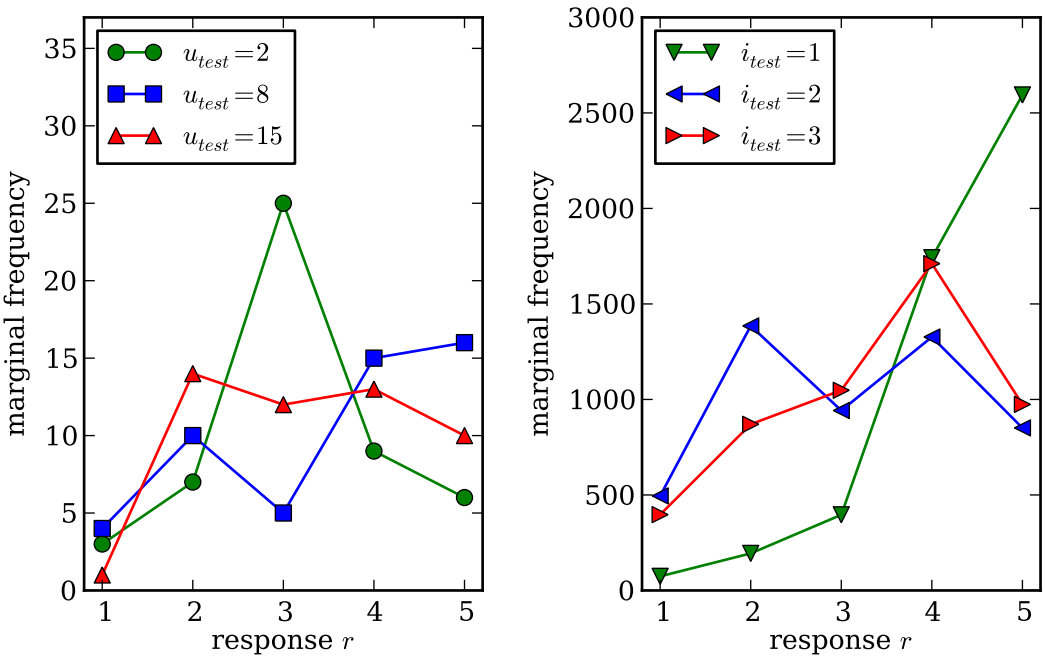


Figure 4.1: Marginal frequencies of reply categories in the dataset studied in Section 4. Connecting lines for visual aid only. **Left:** Marginal frequency of replies from *users* 1,2, and 3 from the dataset. **Right:** Marginal frequencies for *items* 1,2, and 3 in the dataset. Note the strong variations in the frequencies between users, and that none of the histograms follows a bell curve.

frequencies of the Likert answers 1 to 5 for three users to 100 items (Figure 4.1 left) and for three items (right) from 5000 users. It is immediately evident that the users make very varied use of the responses, in a nonlinear way: Some users prefer the "conservative" statements 2 and 4, others use response 5 much more often than any other option, others again do not use response 1 at all. A similar behaviour emerges in the marginals on the items.

4.2.1 Contemporary Item Response Models

A full overview over the psychometric literature is beyond the scope of this work. For the purpose of the model presented here, the relevant conceptual development in that literature is a trend toward more expressive and flexible descriptions, first through an explicit model for the Likert response scale, then through the addition of parameters specific to item and response. We we will here review three important models in this development. Our contribution is then the addition of parameters specific to every *respondent* and response.

Classic test theory [Lord, 1959, Novick, 1966] presupposes some means of turning respondent's u response r_{ui} to item i into an *empirical score* $y_{ui} \in \mathbb{R}$. This mapping is assumed to be provided by the test's designer. The task left for the statistician is then to infer a so-called *true score* ξ_{ui} for further analysis. For example, a frequent

assumption is a linear relationship between true score and empirical score through

$$\xi_{ui} = \bar{y}_i + n_i(y_{ui} - \bar{y}_i), \tag{4.1}$$

where $\bar{y}_i$ is the average empirical score of all respondents, and n_i is the *reliability* of item i, i.e. a measure of variance.

Since the mapping from responses to scores is left to the test designer, it might be tempting to simply use a linear or logistic scale for Likert type responses, mapping directly from a discrete ordered response to a real-valued score. However, as Figure 4.1 shows, the empirical answer frequencies exhibit considerable structure. It thus seems desirable to learn some of this structure directly from the data.

An important development in this direction is the *Rasch* Model [Rasch, 1960], and a generalized version of it known as the *generalized partial credit model* [Muraki, 1992] (GPCM). These models are instances of logistic regression. Each item i is assigned a difficulty d_i, and each response type $r = 1, \ldots, R$ is represented by an item-specific threshold t_{ri}. In the GPCM, the probability of person u answering r_{ui} to item i is given by a logistic distribution parameterized by the latent value y_{ui}:

$$\begin{aligned} p(r_{ui} = 2, \ldots, R \,|\, y_{ui}) &= \frac{\exp\left(\sum_{\tilde{r}=1}^{r_{ui}} d_i(y_{ui} - t_{\tilde{r}i})\right)}{1 + \sum_{m=1}^{R} \exp\left(\sum_{\tilde{r}=1}^{m} d_i(y_{ui} - t_{\tilde{r}i})\right)} \qquad \text{and} \\ p(r_{ui} = 1 \,|\, y_{id}) &= \frac{1}{1 + \sum_{m=1}^{R} \exp\left(\sum_{\tilde{r}=1}^{m} d_i(y_{ui} - t_{\tilde{r}i})\right)} \end{aligned} \tag{4.2}$$

The Rasch model is the restricted case where $d_i \equiv 1$ for all items i. The parameters are usually set to maximum likelihood (ML) values. While these models can account for variations in the distribution of responses over items, they evidently cannot take into account variations between individual users.

4.3 A Generative Model

This section will construct a probabilistic model for ordinal answers. A compact directed graphical model is shown in Figure 4.2. Instead of the logistic link function used in the GPCM, we will use a multivariate probit (i.e. cumulative Gaussian) link function, because it has algebraic advantages when used with the Expectation Propagation approximation. We will pair this link function with linear regression on a combined model of both item-specific and user-specific thresholds.

Quite similar to the GPCM, we will use a set of $L = R - 1$ thresholds $\boldsymbol{h}_{ui} = (h_{ui}^{\ell})_{\ell=1,\ldots,L}$ to describe the generative process of the responses $r_{ui} = 1, \ldots, L + 1$. In contrast, however, we will assume that these thresholds are also a function of the respondent, not only of the item. Specifically, we assume there is a set of L

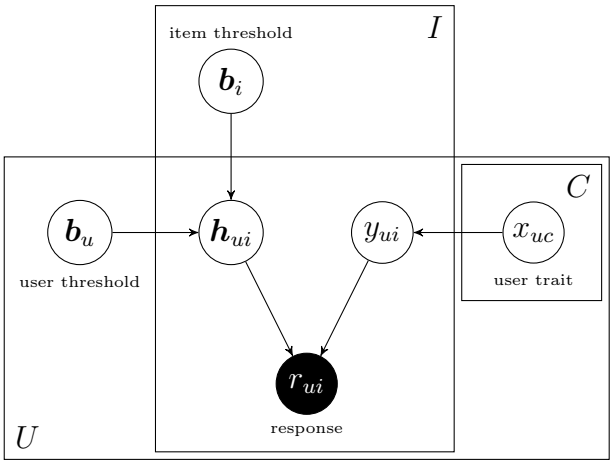


Figure 4.2: Directed graphical model for ordinal regression on user and item thresholds and user traits. Priors and intermediate variables are left out in this representation for clarity. See Figure 4.3 for a more explicit factor graph.

thresholds $\boldsymbol{b}_u$ for each user and similarly a set of item-specific thresholds $\boldsymbol{b}_i$ for item i, such that the overall threshold is given by

$$\boldsymbol{h}_{ui} = \boldsymbol{b}_u + \boldsymbol{b}_i. \tag{4.3}$$

These thresholds are latent variables, and we construct a prior over their values by combining a general Gaussian prior with the explicit requirement that they be ordered:

$$p(\boldsymbol{b}_u) \propto \mathcal{N}(\boldsymbol{b}_u; \boldsymbol{\mu}_u, \boldsymbol{\Sigma}_u) \prod_{\ell=2}^{L} \theta(b_u^\ell - b_u^{\ell-1}), \tag{4.4}$$

where θ here and throughout the rest of the chapter is Heaviside's step function, which equals 1 for positive arguments and 0 elsewhere. The normalization constant of this prior is nontrivial, but the approximate algorithm derived in Section 4.3.2 will construct an approximate Gaussian belief, which is easily normalized.

As in the previous section, the latent variable $x_{uc} \in \mathbb{R}$, $c \in [1, \ldots, C]$ represents the respondent's traits. It generates the latent opinion / ability y_{ui} on item i through the addition of Gaussian noise with variance τ^2

$$p(y_{ui}|x_{uc}) = \mathcal{N}(y_{ui}; x_{uc}, \tau^2). \tag{4.5}$$

Similarly, we also assume that the actual threshold in the user-item combination ui is a noisy version $\tilde{\boldsymbol{h}}_{ui}$ of $\boldsymbol{h}_{ui}$:

$$\tilde{\boldsymbol{h}}_{ui} \sim \mathcal{N}(\boldsymbol{h}_{ui}, \beta^2 \mathbf{I}_L), \tag{4.6}$$

where $\mathbf{I}_L$ represents the L-dimensional identity matrix. The respondent's discrete answer r_{ui} represents the statement that y_{ui} lies between h_{ui}^{r-1} and h_{ui}^r, or above

or below all thresholds for $r_{ui} = R = L + 1$ and $r_{ui} = 1$, respectively:

$$p(r_{ui} \mid \tilde{\boldsymbol{h}}_{ui}, y_{ui}) = \left[\prod_{\ell=1}^{r-1} \theta(y_{id} - \tilde{h}_{ui}^{\ell})\right] \left[\prod_{\ell=r}^{L} \theta(\tilde{h}_{ui}^{\ell} - y_{id})\right] \tag{4.7}$$

4.3.1 Expressiveness of the Model

By defining a generative model and a prior for its parameters, we have implicitly defined a prior distribution over the responses. It is clear that this prior does not put finite mass on all possible response processes. For example, the additive relationship in Equation (4.3) allows either of the thresholds to "veto" the other: There is no way for the model to encode a situation in which a specific respondent never uses, say, reply number 3 (implying that $b_u^2 = b_u^3$), because the item thresholds (assuming at least some of the other respondents do use reply 3) will enforce a finite width of this region. So, even though the model is more expressive than a model which does not take respondent-specific effects into account at all, it is not fully general.

Other simple models one might consider instead of the additive interaction in Equation (4.3) have defects of their own: For example a multiplicative interaction exhibits a vetoing effect similar to that described above. We thus consider the model defined above, while far from perfect, still a considerable improvement over existing models.

It is also clear that a further generalization, to a situation in which every individual respondent-item *pair* has its own thresholds, is challenging because it would remove all identifiability. The degree to which responses were described by the latent trait over the latent free parameters for the thresholds would become critically dependent on the priors over these different parameters, and great care would have to be taken to analyze and find good priors for this situation.

4.3.2 Approximate Inference

Exact inference in the probabilistic model constructed in this section is intractable in two ways: First, the prior (4.4) on the thresholds $\boldsymbol{b}_u$ and $\boldsymbol{b}_i$ has no closed form, because it corresponds to a multivariate Gaussian distribution constrained to a sub-space. Similarly, the posterior on $(\boldsymbol{h}_{ui}, y_{ui})$ given the response r_{ui}, from the likelihood (4.7), is intractable as it too is a restriction of a Gaussian. This is the general form of the probit problem [Ashford and Sowden, 1970], which is known to be intractable [Huguenin et al., 2009].

To circumvent this problem, we will use Expectation Propagation [Minka, 2001] (Section 2.3.2) to construct approximate Gaussian posteriors. Even using this approach, inference remains potentially challenging, because the plates in Figure 4.2

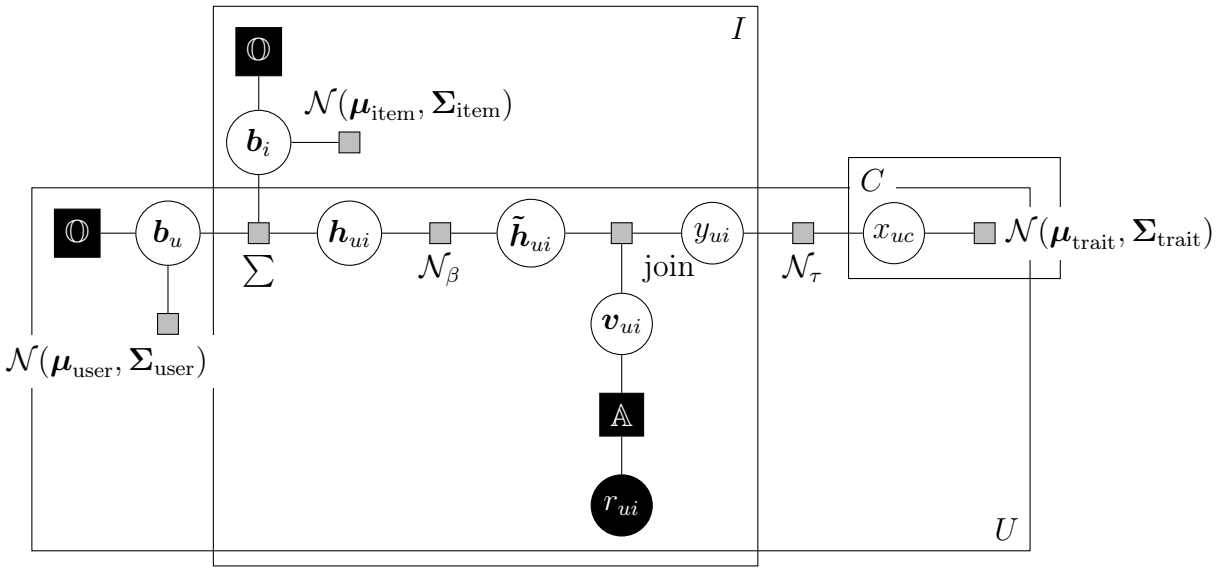


Figure 4.3: Factor graph for the user-item response model. To reduce clutter, the fixed noise parameters β and τ have not been drawn as observed variables. Their influence is indicated by labels underneath the corresponding factors. The factors marked $\mathbb{O}$ and $\mathbb{A}$ represent structured subgraphs. The factor $\mathbb{O}$ enforces ordinality among thresholds, the factor $\mathbb{A}$ encodes the precise meaning of the answer r. To improve readability, these sub-graphs are shown separately in Figure 4.4.

introduce correlations between different respondent and item thresholds, so that we would in principle have to track a joint approximate Gaussian posterior over all variables in the model. For large datasets, such as the one studied in Section 4.4, with many thousands of users, memory requirements alone make this approach impractical. Instead, we will make an approximation to the posterior in which the beliefs on the variables drawn as individual nodes in Figure 4.2 are independent. That is, we will only retain covariance terms between the elements of individual thresholds $\boldsymbol{b}_u$ and $\boldsymbol{b}_i$. Figure 4.3 shows a factor graph from which a message passing algorithm will be derived in the following sections. The subgraphs for the ordinal factors have been replaced with placeholders, which are shown in detail in Figure 4.4. Note that the factor graph contains a large number of loops. In such situations, Expectation Propagation is not guaranteed to converge, but is empirically known to often perform well anyway [Minka, 2001], similar to other loopy inference methods [Frey and MacKay, 1998]. Inference is performed in an iterative way, cycling through the data set several times. At each iteration, the message previously incorporated in the marginal from this data point is "divided out" of the marginal to avoid double counting (see Section 2.3.2).

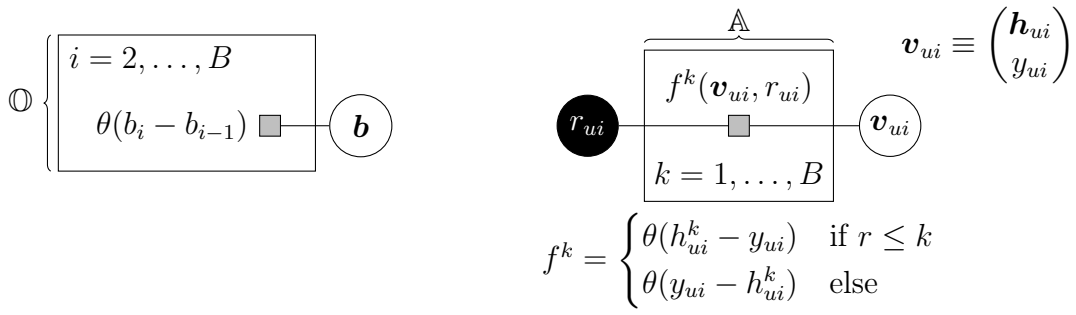


Figure 4.4: Factor graphs for the ordinal (**left**) and response-assignment (**right**) inference, denoted by $\mathbb{O}$ and $\mathbb{A}$ in Figure 4.3, respectively.

Ensuring Proper Ordering of the Boundaries

We begin by constructing approximate Gaussian marginals on the boundaries $\boldsymbol{b}_u$ (the treatment for $\boldsymbol{b}_i$ is entirely analogous). From Figure 4.4, an approximate Gaussian marginal on $\boldsymbol{b}_u$ can be constructed from a set of Gaussian EP messages representing the effect of the step-functions $\theta(b_u^\ell - b_u^{\ell-1})$. This step factor has been used for similar purposes before in Herbrich et al. [2007] and in Stern et al. [2009]. A technical report by [Minka, 2008] describes a particularly efficient way of updating the corresponding EP message, based on the observation that the integral

$$Z_\ell = \int \theta(\boldsymbol{\pi}_\ell^\mathsf{T} \boldsymbol{b}_u) \mathcal{N}(\boldsymbol{b}_u; \boldsymbol{\mu}_u, \boldsymbol{\Sigma}_u) \, \mathrm{d}\boldsymbol{b}_u \tag{4.8}$$

with a vector $(\boldsymbol{\pi}_\ell)_k = \delta_{k\ell} - \delta_{k(\ell-1)}$ (using Kroneckers' δ) depends on $\boldsymbol{\mu}_u$ and $\boldsymbol{\Sigma}_u$ only through rank 1 derivatives

$$\nabla_\mu \log Z_\ell = \alpha_\ell \boldsymbol{\pi}_\ell \qquad \nabla_\Sigma \log Z_\ell = \frac{1}{2}(\alpha_\ell^2 - \beta_\ell) \boldsymbol{\pi}_\ell \boldsymbol{\pi}_\ell^\mathsf{T} \tag{4.9}$$

with some scalar terms α_ℓ and β_ℓ. Detailed derivations are reproduced in Appendix B for completeness. This allows the individual updates to be performed with computational cost $\mathcal{O}(L^2)$, rather than the $\mathcal{O}(L^3)$ a naïve implementation would require. For the purpose of the following derivations, it suffices to know that there exists an algorithm constructing a joint approximate Gaussian marginal with dense covariance matrix $\hat{\boldsymbol{\Sigma}}_u$ on $\boldsymbol{b}_u$, which approximates the second moments of the distribution (4.4) and, in later stages of the algorithm, of the posterior marginal on $\boldsymbol{b}_u$ (and analogously for $\boldsymbol{b}_i$).

Factorized Regression

We perform straightforward Gaussian belief propagation to construct an intermediate marginal on $\tilde{\boldsymbol{h}}_{ui}$ (i.e. a message from this variable to the factor labeled "join" in Figure 4.3): Because the approximate marginals on the thresholds are Gaussian

$\mathcal{N}(\boldsymbol{b}_u; \hat{\boldsymbol{\mu}}_u, \hat{\boldsymbol{\Sigma}}_u)$ and $\mathcal{N}(\boldsymbol{b}_i; \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i)$, the message can be evaluated analytically to

$$\begin{aligned} m(\tilde{\boldsymbol{h}}_{ui}) &= \int \mathcal{N}(\tilde{\boldsymbol{h}}_{ui}; \boldsymbol{b}_u + \boldsymbol{b}_i, \beta^2 \mathbf{I}_L) \mathcal{N}(\boldsymbol{b}_u; \hat{\boldsymbol{\mu}}_u, \hat{\boldsymbol{\Sigma}}_u) \mathcal{N}(\boldsymbol{b}_i; \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i) \, \mathrm{d}\boldsymbol{b}_i \, \mathrm{d}\boldsymbol{b}_u \\ &= \mathcal{N}(\tilde{\boldsymbol{h}}_{ui}; \hat{\boldsymbol{\mu}}_u + \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_u + \hat{\boldsymbol{\Sigma}}_i + \beta^2 \mathbf{I}_L) \equiv \mathcal{N}(\tilde{\boldsymbol{h}}_{ui}; \boldsymbol{\mu}_{h_{ui}}, \boldsymbol{\Sigma}_{h_{ui}}) \end{aligned} \tag{4.10}$$

Similarly, the message from y_{ui} in this direction is $\mathcal{N}(y_{ui}; \mu_{x_{uc}}, \sigma^2_{x_{uc}} + \tau^2)$. This is using the mean and variance of the marginal on the trait x_{uc}, where $c = c(i)$ is the trait responsible for this item. From these two marginals, we construct a marginal on their joint $\boldsymbol{v}_{ui} \equiv (\boldsymbol{h}_{ui}, y_{ui})$ in the trivial way, by stacking them:

$$p(\boldsymbol{v}_{ui}) = \mathcal{N}\left[\boldsymbol{v}_{ui}; \begin{pmatrix} \boldsymbol{\mu}_{h_{ui}} \\ \mu_{y_{ui}} \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{h_{ui}} & 0 \\ 0 & \sigma^2_{y_{ui}} \end{pmatrix}\right] \tag{4.11}$$

Since the conditional (4.7) of the response is again a product of step functions, EP inference on the marginal of $\boldsymbol{v}_{ui}$ can be performed in a similar way to the ordinal inference in Section 4.3.2, establishing a marginal on $\boldsymbol{v}_{ui}$ with a dense co-variance matrix $\boldsymbol{\Sigma}_{ui}$. Now that there is a set of messages from r_{ui} towards $\boldsymbol{v}_{ui}$ in Figure 4.3, the direction of message passing can be reversed: The messages from $\boldsymbol{v}_{ui}$ to $\tilde{\boldsymbol{h}}_{ui}$ and y_{ui} can be constructed by simply separating the elements of the mean and variance of the marginal of $\boldsymbol{v}_{ui}$, because for Gaussians in general (see also Equation A.13)

$$p(x, y) = \mathcal{N}\left[\begin{pmatrix} x \\ y \end{pmatrix}; \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{xy} & \Sigma_{yy} \end{pmatrix}\right] \qquad \Longrightarrow \qquad p(x) = \mathcal{N}(x; \mu_x, \Sigma_{xx}) \tag{4.12}$$

The new message to x_{uc} is constructed in an analogous way to the presentation in Equation (4.10): Adding Gaussian noise of variance τ^2, the factorized messages to the thresholds $\boldsymbol{b}_u$ and $\boldsymbol{b}_i$ are found by re-arranging the conditional in Equation (4.10)

$$\mathcal{N}(\tilde{\boldsymbol{h}}_{ui}; \boldsymbol{b}_u + \boldsymbol{b}_i, \beta^2 \mathbf{I}_L) = \mathcal{N}(\boldsymbol{b}_u; \tilde{\boldsymbol{h}}_{ui} - \boldsymbol{b}_i, \beta^2 \mathbf{I}_L) = \mathcal{N}(\boldsymbol{b}_i; \tilde{\boldsymbol{h}}_{ui} - \boldsymbol{b}_u, \beta^2 \mathbf{I}_L) \tag{4.13}$$

and marginalizing once more, as in Equation (4.10). This completes the derivation of all approximate Gaussian messages necessary for Expectation Propagation in this model.

Algorithmic Implementation

For a large dataset, the number of messages passed in the scheme laid out in the previous sections is considerable, and the message-passing schedule can have a marked effect on the speed of convergence and numerical stability. In our implementation, the algorithm begins with a small loop over the ordinality constraints.

Once this loop has converged, the algorithm iterates over all pairs (u, i) of user u responding to item i with answer r_{ui}, stopping at regular intervals to update the ordinal messages, and loops over the data set in this fashion several times until the messages have converged. Convergence can be measured in several ways. In our implementation, we track the maximal change to the logarithm of the determinant of the precision matrix of any message in the algorithm. We also damp the messages slightly, as described in Section 2.3.2, which increases performance slightly. The algorithm typically converges within less than 10 iterations, even for large datasets.

Free Parameters and Identifiability

The approximate probabilistic algorithm has two obvious free parameters β and τ. To fix these in our experiments, the evidence of the training set was evaluated for a small number of settings and maximized, leading to the values $\tau = 3.0$ and $\beta = 0.2$. The parameters of the priors jointly define the scale and location of the latent space, which is arbitrary. If the priors' means are set to zero, there is only one further meaningful degree of freedom, which is the *relative* width of the prior on x_{ui} to the effective prior on $\boldsymbol{h}_{ui}$ defined through the priors on $\boldsymbol{b}_u$ and $\boldsymbol{b}_i$. Similar to the treatment of the other free parameters, a number of settings were tested based on the assigned evidence, and the priors where chosen to be $p(\boldsymbol{b}_i) = p(\boldsymbol{b}_u) = \mathcal{N}(\boldsymbol{b}_{u/i}; \mathbf{0}, \mathbf{I}_4)$, where $\mathbf{I}_4$ is the four-dimensional unit matrix, and $p(x_{uc}) = \mathcal{N}(x_{uc}; 0, 1)\ \forall c$. So the overall number of free parameters to choose by maximum evidence is three, which should be compared to the $U + I(L + 1)$ free parameters fitted by maximum likelihood in the GPCM.

In the literature on generalized regression, it is often pointed out that models such as the one presented here are not fully "identified", because under the transformation $\boldsymbol{h}_{ui} \to \boldsymbol{h}_{ui} + k\mathbf{1}$ and $x_{ui} \to x_{ui} + k$, i.e. under addition of a constant to all involved quantities, the model produces the same predictions. In the GPCM, the different structure of the last line of Equation (4.2) is a solution to this issue, enforcing identifiability. However, since the algorithm presented here retains beliefs rather than point estimates, producing predictions by integrating out the latent beliefs, this degree of freedom is not a problem, as long as the latent distribution is proper. The simple independent Gaussian prior mentioned above suffices for this task. There is never a need to identify any specific solution for the latent parameters.

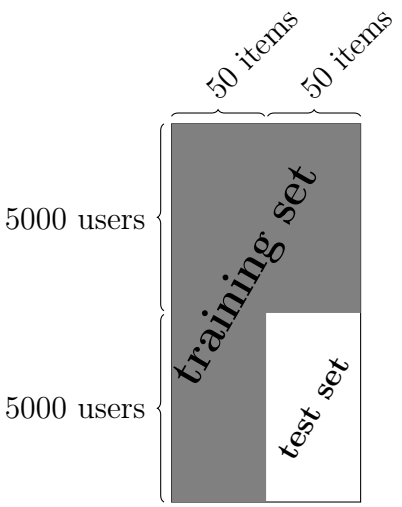


Figure 4.5: Sketch of the separation of the data set into training and test sets.

4.4 Results

We evaluated the methods presented in the previous sections on a data set provided courtesy of the *MyPersonality.org* application[1], which allows users of the Facebook social networking system to complete personality tests. The entire dataset comprises approximately 4 million users, but to limit computational cost, was limited to a subset from $10,000$ users. The test taken is a version of the "Big Five" personality test introduced in Section 4.2. The version of the test used here has 100 items, answered on a one-out-of-five Likert scale. As discussed before, each item is associated, in theory, with one and only one of five traits called "openness", "conscientiousness", "extraversion", "agreeableness" and "neuroticism". Because all algorithms require training data for both users and items, the data set was split into training and test set as follows (Figure 4.5): A first set of $5,000$ users was used to train item characteristics ($\boldsymbol{b}_i$). The second half ($5,000$ users) was split in half by removing every other question, with one half providing another training set for user parameters ($\boldsymbol{b}_u$), and the second half being the test set. For the Rasch and CPGM models, the latent trait is the only variable conveying information about the user; for our approximate scheme we used the users' inferred traits as well as the inferred user boundaries.

4.4.1 Computational cost

The GPCM and Rasch models were trained with the publicly available *ltm* package[2] for the R programming language. This package returns a maximum likelihood fit for the $I(R+1)$ parameters of the GPCM model (Equation (4.2)), and the IR

[1] `http://mypersonality.org/research/interested-in-collaborating/`
[2] `http://rwiki.sciviews.org/doku.php?id=packages:cran:ltm`

parameters of the Rasch model (Equation (4.2) with $\beta = 1$). It also returns maximum a-posteriori Gaussian approximations for the latent traits x_{uc} of users. The EP algorithm was implemented in the .NET language F#. Due to the incomparable platforms used, a precise timing comparison is not meaningful, but the overall computation times were roughly comparable (the GPCModels took 5 hours to conclude, the EP implementation about 2 hours), even though the probabilistic algorithm infers about 100 times more latent parameters.

4.4.2 Approximate Bayesian Estimates

Figure 4.6 shows beliefs over the values of $\boldsymbol{b}_u$ and $\boldsymbol{b}_i$ for the same users and items as shown in Figure 4.1. Note that the scale for user 8, who rarely uses response $r = 3$, is "steeper", with the boundaries $b^2_{u=3}$ and $b^3_{u=3}$ moved closer together. The opposite effect can be seen for user 2, who prefers response 3. User 15 barely uses the lowest response $r = 1$, leading to the boundary $b^1_{u=15}$ moving further down, and the uncertainty on its value to increase. Similar effects can be seen for the items, shown on the right. Since every item takes part in a much larger number of user-item pairs (5000 pairs for items in the test set) than the individual users (50 pairs for each user in the test set), the beliefs on the item thresholds are much more precise than those on the users.
The bottom half of Figure 4.6 shows the resulting belief on the threshold $\tilde{\boldsymbol{h}}_{ui}$ for one specific item $i = 2$ and the three same users as above. Note that, despite the relatively high precision of the user and item marginals, the noise term β still produces a relatively broad distribution. It is important to note that, while all the figures show the beliefs on threshold values as four individual distributions, these are marginals of a joint distribution. One can imagine samples from this distribution to be threshold values taking small or large values in co-ordination (Figure 4.7).

4.4.3 Predictive Performance

Table 4.1 reports the log probability assigned by the variant methods to the response chosen by the user. The probabilistic algorithm can make use of its more expressive model to predict users' answers with higher accuracy. The table also contains predictive probabilities for the approximate inference model with $\boldsymbol{b}_i$ and $\boldsymbol{b}_u$ fixed to $\mathbf{0}$, respectively. In the former setting, which is very similar to the GPCM setup, the probabilistic algorithm produces approximately the same predictive performance as GPCM, while using only user-specific thresholds yields a less expressive model. This is not surprising, as this setup effectively creates independent datasets for every user, each consisting of 50 data points used to predict 50 other data points.

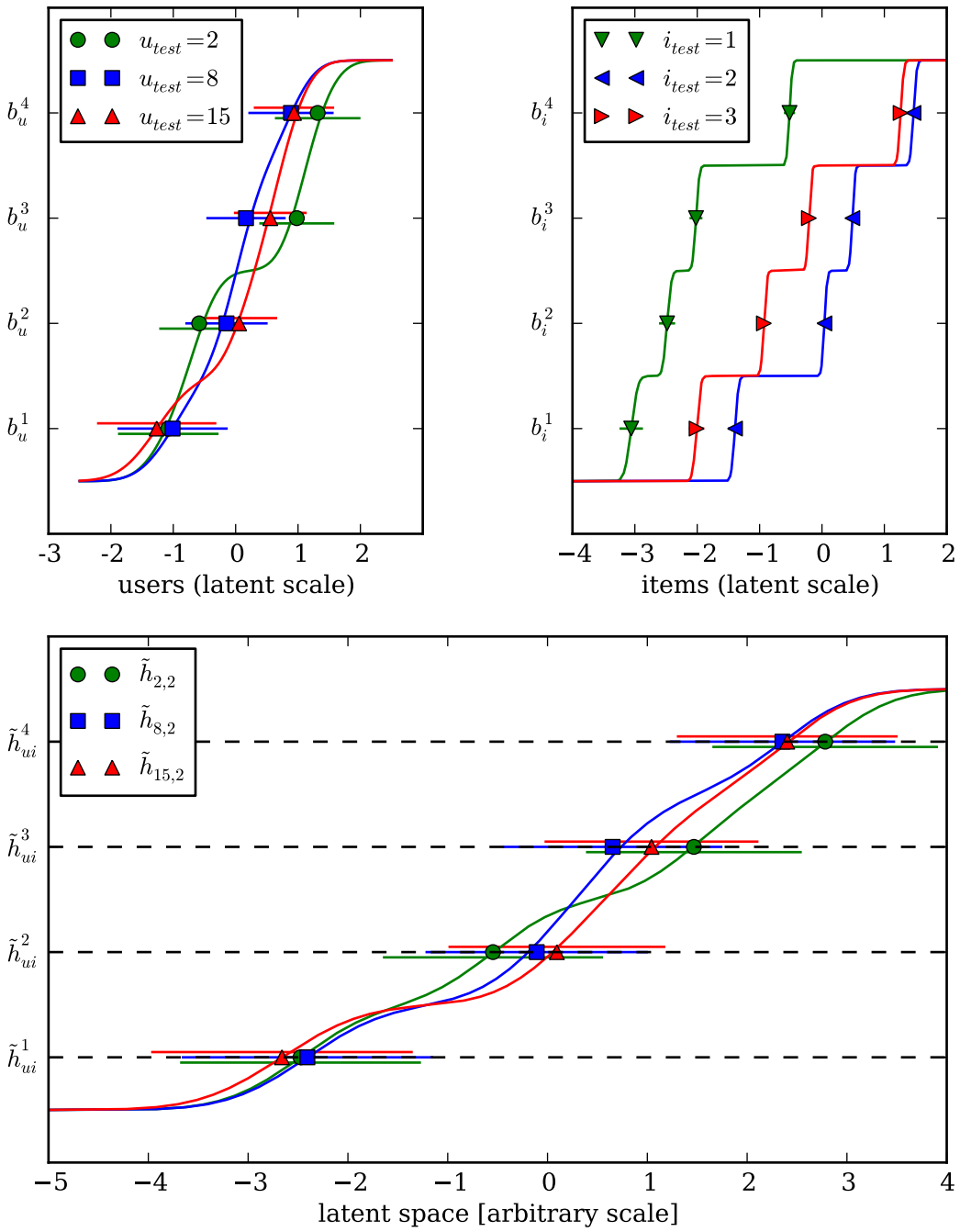


Figure 4.6: Thresholds learned by the approximate inference algorithm, for the users and items introduced in Figure 4.1. **Top left:** Marginal beliefs for the first three users. Means denoted by a marker, "error bars" of two standard deviations width to each side in the same colour as the marker (the error bars have been slightly off-set from the means to avoid overlap). The vertical position of each marginal is indicative of its ordinal position, from threshold 1 at the bottom to threshold 4 at the top. The connecting lines are for visual aid only. To create a smooth connection, they were generated by evaluating the function $f(x) = \sum_\ell \Phi(x; \mu_{b_{ui}^\ell}, \sigma^2_{b_{ui}^\ell})$, where $\Phi(x, \mu, \sigma^2)$ is the cumulative density function of the one-dimensional Gaussian with mean μ and variance σ^2. Nonlinear functions of analogous form are sometimes used to represent thresholded models in the literature on GPCM; note, however, that this distribution is not connected in an obvious way to the actual predictive probability of replies r_{ui}. **Top right:** Similar plot for the three items shown in Figure 4.1. **Bottom:** A similar plot for the threshold $\tilde{\boldsymbol{h}}_{ui}$ resulting for the combination of user $u = 8$ and item $i = 2$.

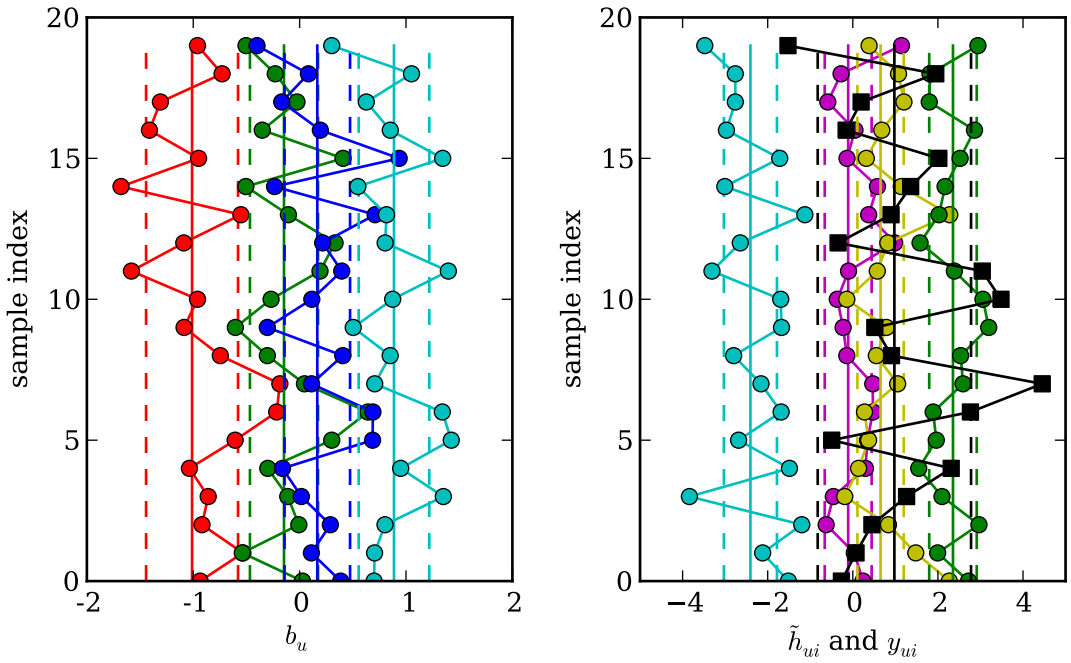


Figure 4.7: 20 sampled threshold values, for the item-respondent combination $u = 8$, $i = 2$ (see also Figure 4.6). **Left:** sampled respondent-specific thresholds $b_{u=8}$. Samples as circles, connecting lines for visual aid only (rows of samples were generated independently, their order is arbitrary). Marginal means and standard deviations as thick and dashed lines, respectively. The order of the threshold elements is from 1 (left) to 4 (right). Note the ordering among the thresholds, and that the approximate character of the inference means that minor variations of the ordering are not entirely impossible. **Right:** Similar plot for the combined thresholds $\tilde{\boldsymbol{h}}_{u=8,i=2}$ (coloured circles) and the predicted opinion y_{ui} (black squares). Due to the independent regression noise τ, these thresholds do mix sometimes, which is an explicit part of the model, not an artifact of approximation.

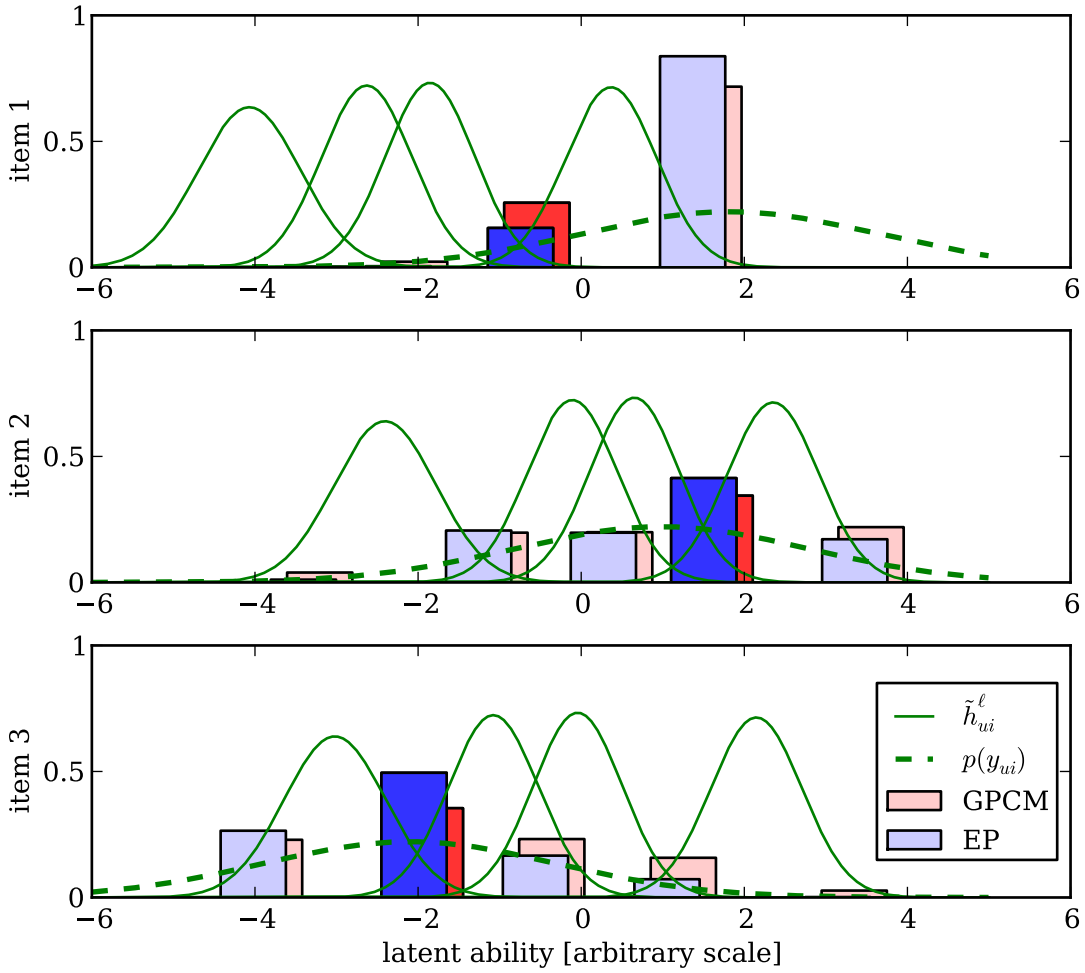


Figure 4.8: Predictions for the responses of user 8 (see Figure 4.1) to items 1 (**top**), 2 (**middle**), and 3 (**bottom**). All these items are in the test set, so the algorithms have not observed this combination of item and user before. Solid green distributions: Marginals for the locations of the four latent thresholds $\tilde{\boldsymbol{h}}_{ui}$ (note that these are marginals of the elements of a highly correlated joint distribution). Dashed green distributions: Marginals on this user's latent ability y_{ui}. Blue bars (in front): Predictions of the approximate inference scheme on user-item pair regression for the five possible answers. Red bars (in background): Predictions of the GPCModel. The answer actually chosen by the user is highlighted in a stronger colour. The location of the bars is essentially arbitrary as the replies are elements of a different space as the thresholds and abilities, but have been chosen suggestively: The left-most bar, to the left of the first threshold, corresponds to reply 1, and so on up to reply 5, to the right of the highest threshold.

Method $\mathcal{M}$	$1/N_{\text{test}} \cdot \log p(D_{\text{test}}\|\mathcal{M})$	$p^{1/N_{\text{test}}}(D_{\text{test}}\|\mathcal{M})$
Chance (for comparison)	-1.609	0.200
Rasch Model (ML)	-1.362	0.256
GPCM (ML)	-1.299	0.273
Ordinal Model, items only	-1.306	0.271
Ordinal Model, users only	-1.385	0.250
Ordinal Model, users & items	$\mathbf{-1.257}$	**0.285**

Table 4.1: Average log and direct probabilities assigned, by the different models, to the item actually chosen by the user, on the test set of $5,000$ users, on 50 items (see Figure 4.5). Larger values are better. "Items only" denotes a model only modeling item-specific thresholds; analogously for "users only".

Figure 4.8 gives an intuition of the prediction process. The marginals of the thresholds are shown as Gaussian distributions, the marginal on the latent ability y_{ui} as a broader, dashed Gaussian distribution. The predictions of both the GPCM and the probabilistic algorithm are shown as bars, with the actual response (which was not observed by the algorithms) highlighted. Note that these predictions are a nontrivial function of the depicted Gaussian beliefs and can not simply be read off from the plot. The figure shows the beliefs for the combination of user 8 with the first three items in the test set, as also shown in Figure 4.6. The GPCM tends to make broader, less certain predictions than the more expressive probabilistic model. In case of item 1, the GPCM happens to put larger mass on the correct answer, in the other two cases, the probabilistic algorithm could predict the right answer with higher probability. As Table 4.1 shows, overall the probabilistic algorithm is more successful in predicting the users responses.

4.5 Discussion

The model and accompanying inference algorithms presented in this paper are intended mainly as technical demonstrations. They obviously do not constitute psychological research. We have presented and evaluated them using the framework and evaluation criteria of the machine learning community. It should be pointed out that the psychometric literature differs from this point of view in some important aspects. Most importantly, in psychological contexts inferring and then studying the latent dimensions of the data is the main aim of the research process. Many psychologists would thus find it not acceptable to retain a finite-width Bayesian belief over possible values of these values, even though, from a statistician's perspective, this would be a good course of action if the ultimate goal of the research was to predict human behavior. The need for scientific as well as public communication of the research results has so far discouraged reporting a set of beliefs as final research outcomes. This also means that testing a particular pro-

babilistic model solely on its predictive power on held-out test data might not be sufficient for applications in psychology, and that interpretability of the inferred latent structure is just as important. On the other hand, it should be pointed out that, on a fundamental level, neither the models in general use today nor the probabilistic algorithm presented here can claim to be a faithful representation of the complicated generative process for item responses. Such a model, if it exists, would have to take into account both the semantic structure of the items and the complicated thought process the human respondent uses to arrive at an answer. In contrast to the GPCM, the algorithm presented here at least provides some crude treatment of the latter aspect.

If modern inference methods are to make a significant contribution to the development of psychological analysis, it will be necessary to find solutions to these conceptual issues. The aim of this work is, more humbly, to communicate to the machine learning community that psychometric datasets provide interesting and challenging applications for machine learning methods. Thanks to social networking, such datasets are reaching previously unseen scales, making efficient data modeling techniques crucial.

4.6 Conclusion

Psychometry is a well-established discipline within psychology, with considerable mathematical underpinnings in classical statistics, and developed through a large body of literature. The recent advent of social networking has created a surge in test results that have the potential to raise fidelity and reliability of psychometric tests. At the same time, the ongoing fast pace of machine learning research has generated new tools for efficient and robust approximate inference, which allow inference in larger models of more complex structure. As a demonstration of the usefulness of contemporary machine learning methods for psychological modeling, we have presented an approximate inference scheme based on a combination and extension of several recently developed methods and models, which allows fully probabilistic inference from ordinal discrete item responses on latent structural descriptions of human behavior.

Chapter 5

Fast, Online Inference for Conditional Topic Models

The work presented in this chapter is the result of a collaboration with **Thore Graepel**, **Ralf Herbrich** and **David Stern**, all of Microsoft Research Ltd.. Except for where the work of others is explicitly cited, all mathematical derivations, algorithmic implementations and experimental evaluations were performed by the author of this thesis. The author thanks Thomas Borchert for providing access to a different method as a benchmark during development, as well as David Knowles and Simon Lacoste-Julien for helpful comments.
A shorter version of this chapter is under review by the International World Wide Web Conference 2011.

Abstract

Topic models provide a means of describing semantic similarity between documents in a multidimensional latent space, based on similarity of word frequencies. Unfortunately, virtually all contemporary inference algorithms for topic models require multiple passes over the document corpus, making them ill-suited for use on the large scale corpora typical of the web. Moreover, modern online document repositories regularly provide document features other than the words themselves, which can provide helpful semantic information but are not part of standard topic models. These issues have so far confined the use of topic models to small and specific corpora. Using a series of approximations that have proven helpful in other areas of machine learning, we construct an efficient approximate inference scheme for topic models conditional on arbitrary features of the document. We also study the viability of *online* (single pass) inference in such models, and show experimental

results from large online document corpora.

5.1 Introduction

Topic models are probabilistic descriptions of corpora of text documents as "bags of words" (i.e. ignoring word order), generated from a mixture of discrete distributions over words. Each document in a collection is considered to be generated by a mixture of topics such that each word in the document is generated by first drawing a particular topic, then drawing the word from that topic. One way to interpret these models is as a form of dimensionality reduction: Instead of having to explain every word in every document individually, the model describes the documents' words in terms of a lower-dimensional space of topics (see sketch in Figure 5.1, top). Such discrete mixture models are of course more generally applicable than only to texts. But texts are currently the most important application, and the metaphors of documents, topics and texts are useful for visualization. This chapter will thus stick to this nomenclature, and all experiments will be carried out on collections of actual text documents.

A wide array of generative models for text — with a view towards indexing — have been studied in the past, starting from ad-hoc methods such as the *tf-idf* heuristic still popular in information retrieval [Salton and McGill, 1983], to methods such as *latent semantic indexing* [Deerwester et al., 1990], which is based on a maximum likelihood estimate of a singular value decomposition. Seminal steps towards models retaining probabilistic beliefs for this task were the development of a probabilistic model for latent semantic indexing [Hofmann, 1999], and the introduction of latent Dirichlet allocation [Blei et al., 2003] (in the compression and language modelling communities, whose goals differ from the indexing setting discussed here; probabilistic language models based on latent Dirichlet variables had been studied much earlier than that, see e.g. MacKay and Bauman-Peto [1995]). Since then, a considerable amount of further work on this subject has been produced. An overview can be found in [Blei and Lafferty, 2009]. Of particular relevance for the work presented here are the development of collapsed inference algorithms [Griffiths and Steyvers, 2004, Teh et al., 2007], and of topic models conditioned on features of the document [Mimno and McCallum, 2008, Zhu and Xing, 2010].

In most published applications, topic models are applied to collections of "conventional" documents, such as scholarly articles, or newswire reports. In such corpora, the overall number of documents is limited, and usually on the order of a few thousand to tens of thousands, making several iterations over the corpus computationally feasible with standard hardware. Because the corpus is either of fixed size or growing slowly, computation time (typically on the order of hours or days) is short compared to the pace at which the corpus develops. Document collections

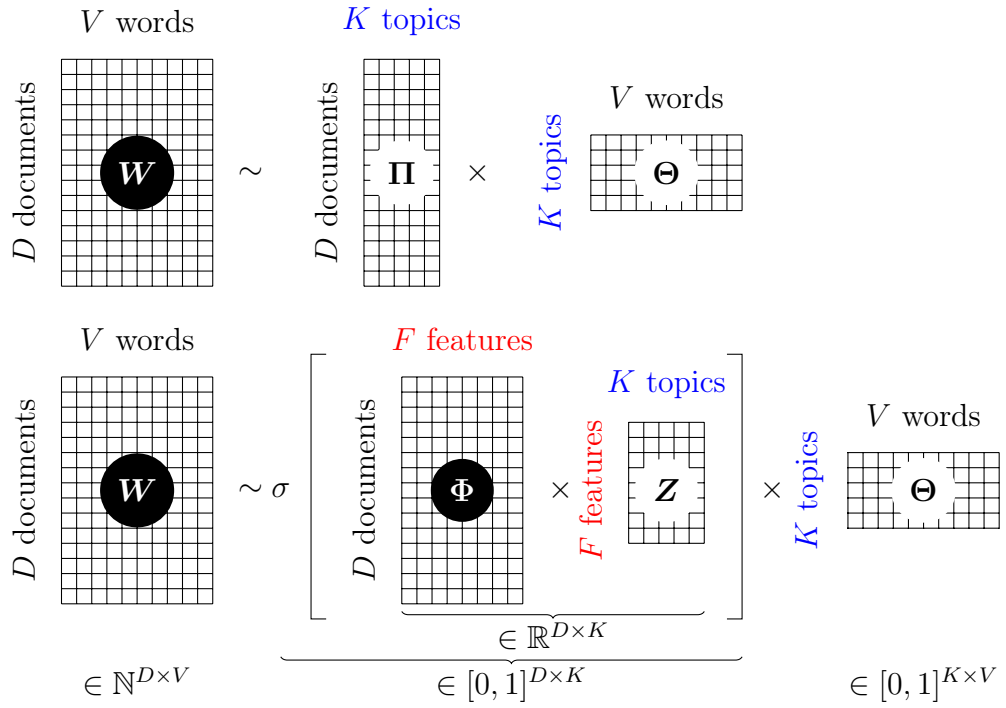


Figure 5.1: Conceptual sketches. **Top:** Classic topic model. Words from a vocabulary of size V observed with varying frequencies over a corpus of D documents. The generative process for the words is described in terms of a K-dimensional latent space of topics (with $K \ll D$ and $K \ll V$). **Bottom:** Topic model conditioned on observable features of the document. Note projection through the softmax function σ. The symbols in the matrices refer to the nomenclature used throughout the chapter, where the elements of the matrices are denoted by the corresponding lower-case symbols. Black circles denote observed variables, white circles latent ones. In many applications, $\mathbf{\Phi}$ is a sparse matrix and F is large, though usually smaller than D. The colour coding for features and topics will be re-used in Section 5.2.3 and Section 5.2.3 to clarify two different sources of conditional dependence.

on the web differ considerably from this setup: these corpora are constantly growing, and topics must be available (almost) in real time, so an inference algorithm has to be able to run at speeds comparable to the growth rate of the corpus. An extreme example for such a setting is the microblogging service Twitter.com, where users publish short "status updates" of ≤ 140 characters in length. At the time of writing, the English subset of the Twitter corpus grows at $> 10^6$ status updates per hour, and this rate is increasing. This makes single pass, or *online* algorithms the only option for inference on the structure of this and similar corpora (such as Wikipedia.org, Facebook.com, etc., each of which contain well over 10^7 documents and are growing constantly).

A second, potentially beneficial aspect of documents on the web is the wide availability of explanatory metadata, such as the identity of the author, the time of writing, annotation tags by other users, geographic location of the author or her/his membership in certain groups. Such features can provide strong information about the topic distribution of a document. For example, being in a region currently struck by natural disaster may almost completely determine the topic of a Twitter status update. A more mundane example is the high frequency of "Good morning world" status updates on Twitter at certain times of day, or the focus of individual Wikipedia authors on particular subjects.

The following sections present an extension of the latent Dirichlet allocation (LDA) model, with a focus on computational efficiency, and the use of (sparse) features for regression on topics. Such conditional topic models have recently been introduced by other authors [Mimno and McCallum, 2008, Zhu and Xing, 2010]; but those implementations have relatively high computational cost (see Section 5.2), making these algorithms difficult to apply to large corpora. Instead, we propose a semi-collapsed variational inference scheme adapted from work by Teh et al. [2007], and combine it with sparse linear Gaussian regression based on assumed density filtering in Gaussian message passing (see e.g. Opper [1996]). The necessary link function between these two schemes is provided by a Laplace approximation for Dirichlet distributions in the softmax basis first introduced by MacKay [1998]. The result is a fast implementation of conditional topic models that can deal with document collections of the size of Twitter and other online document collections, and allows prediction of topics from features, inference of a document's topic distribution from words and features, as well as inference on each topic's word distribution.

5.2 Model

Consider a corpus of D documents. Document d contains I_d words $w_{di} \in \{1, \ldots, V\}$, $d \in \{1, \ldots, D\}$, $i \in \{1, \ldots, I_d\}$ from a vocabulary of size V. Some description of d is available in the form of a feature vector $\boldsymbol{\phi}_d \in \mathbb{R}^F$ (in many applications, this

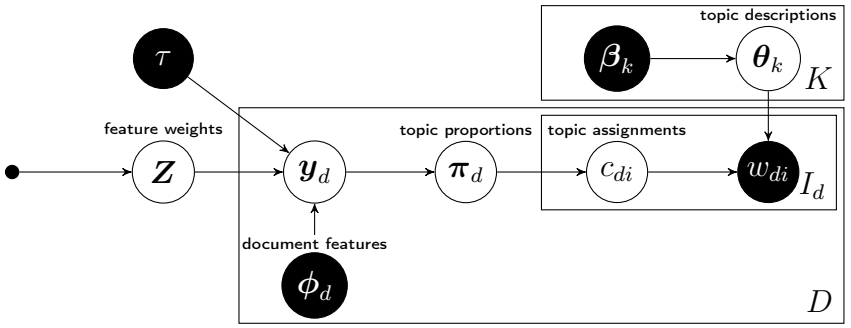


Figure 5.2: Directed graphical model of the conditional topic model. Some of the latent parameters have been labeled with descriptions for clarity.

vector will be sparse). We construct a topic model conditional on observable features of the documents, using the following generative process for all vectors $\boldsymbol{w}_d$, $d \in [1, \ldots, D]$, from K topics:

▷ For each topic $k \in \{1, \ldots, K\}$, generate a discrete probability distribution with parameters $\boldsymbol{\theta}_k \in [0,1]^V$ over the vocabulary of size V by sampling from a Dirichlet distribution with parameter vector $\boldsymbol{\beta}_k$:

$$p(\boldsymbol{\theta}_k \,|\, \boldsymbol{\beta}_k) = \mathcal{D}(\boldsymbol{\theta}_k; \boldsymbol{\beta}_k) = \frac{\Gamma\left(\sum_v^V \beta_{kv}\right)}{\prod_v^V \Gamma(\beta_{kv})} \prod_k \theta_{kv}^{\beta_{kv}-1} \tag{5.1}$$

where Γ is the Gamma function.

▷ Generate a matrix $\boldsymbol{Z} \in \mathbb{R}^{F \times K}$ of feature-topic weights, by sampling each weight independently from a Gaussian distribution with mean μ_{fk} and variance σ_{fk}^2:

$$p(z_{fk} \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(z_{fk}; \mu_{fk}, \sigma_{fk}^2) \tag{5.2}$$

(note that this notation allows a nonzero mean for every individual topic, assuming the existence of a "bias" feature with fixed value 1.)

▷ For each document d with features $\boldsymbol{\phi}_d \in \mathbb{R}^F$,

– Draw a latent variable $\boldsymbol{y}_d$ from the noisy linear model with a single noise parameter τ:

$$p(\boldsymbol{y}_d \,|\, \boldsymbol{Z}, \tau, \boldsymbol{\phi}_d) = \mathcal{N}(\boldsymbol{y}_d; \boldsymbol{Z}^\mathsf{T}\boldsymbol{\phi}_d, \mathrm{diag}(\tau^2)) \tag{5.3}$$

– Define the topic proportions $\boldsymbol{\pi}_d = \boldsymbol{\sigma}(\boldsymbol{y}) \in [0,1]^K$ where $\boldsymbol{\sigma}$ is the vector softmax function

$$\sigma_k(\boldsymbol{y}) = \frac{\exp(y_k)}{\sum_\ell^K \exp(y_\ell)} \tag{5.4}$$

– For the i-th of I_d words

∗ draw a topic c_{di} from the discrete distribution defined by $\boldsymbol{\pi}_d$:

$$p(c_{di} = k \,|\, \boldsymbol{\pi}_d) = \pi_{dk} \tag{5.5}$$

∗ draw word w_{di} from the discrete distribution of topic c_{di}:

$$p(w_{di} = v \,|\, c_{di}, \boldsymbol{\Theta}) = \theta_{c_{di}v} \tag{5.6}$$

Figure 5.2 shows a directed graphical model representing this generative model. If we replace everything to the left of $\boldsymbol{\pi}_d$ in that figure by a single Dirichlet parameter vector $\boldsymbol{\alpha}$ (identical for all d), then the parts shown to the right of and including the node $\boldsymbol{\pi}$ correspond to the traditional LDA model [Blei et al., 2003]. The extension to a conditional topic model as defined above has been used before by Mimno and McCallum [2008], who developed an inference scheme based on expectation maximization and Gibbs sampling for it. Such an algorithm does not lend itself to large datasets. There is also a related probabilistic model, known as the *correlated topic model* [Blei and Lafferty, 2007], which shares the softmax relationship introduced above and everything to its right in Figure 5.2, but not the regression element to its left. While it is straightforward to extend the correlated topic model with the regression part as introduced above, the published inference method for this model uses an overall variational bound that can be optimized only numerically and involves frequent matrix multiplications of cost $\mathcal{O}(K^2)$; this scheme thus comes at considerable computational cost, too. Our contribution will be the derivation of a fast inference scheme. To construct this algorithm, we will integrate out the distribution over $\boldsymbol{\Theta}$ (i.e. we will treat this part of the model exactly), and derive a variational bound on $\boldsymbol{\Pi}$ and $\boldsymbol{c}$ (Section 5.2.1), using approximate discrete beliefs on the c_{di} and Dirichlet beliefs on the $\boldsymbol{\pi}_d$. This requires a Dirichlet prior for $\boldsymbol{\pi}_d$. To construct this prior, we will use factorized Gaussian message passing in the regression algorithm (Section 5.2.3), which establishes a Gaussian belief on $\boldsymbol{y}_d = \boldsymbol{\sigma}^{-1}(\boldsymbol{\pi}_d)$. We will use a Laplace approximation to establish a match between Gaussians and Dirichlets in the softmax basis (Sections 5.2.2 and Appendix C).

5.2.1 Semi-Collapsed Variational Inference

This section constructs a variational inference scheme for LDA which we will call *semi-collapsed* for the following reason: the necessary derivations are an extension of a fully collapsed variational bound derived in an excellent paper by Teh et al. [2007]. The parts of our algorithm formulated in this section differ from Teh's work in only two details, which cause minor changes in the derivation but will be crucial for their use in our overall model: We will retain an explicit variational approximation to the posterior beliefs on the $\boldsymbol{\pi}_d$, instead of integrating them out

(i.e. this part of the inference is *not* collapsed). We will also make the more general assumption of non-uniform Dirichlet parameters. To keep the notation uncluttered, it will be assumed that a Dirichlet prior $p(\boldsymbol{\pi_d} \,|\, \boldsymbol{\alpha}_d) = \mathcal{D}(\boldsymbol{\pi}_d; \boldsymbol{\alpha}_d)$ on $\boldsymbol{\pi}_d$ is available at the time of inference. This prior will be constructed in Section 5.2.2.
From the definitions in Section 5.2, the joint probability of all latent and observed variables in the LDA model, as a function of the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, is a large product of Dirichlet distributions. To see this, note that for every observed document $\boldsymbol{w}_d$, the likelihood for $\boldsymbol{\pi}_d$ and $\boldsymbol{\theta}$ is multinomial in both parameters:

$$p(\boldsymbol{w}_d, \boldsymbol{c}_d \,|\, \boldsymbol{\pi}_d, \boldsymbol{\theta}_d) \propto \prod_i^{I_d} \pi_{dc_{di}} \theta_{c_{di} w_i}. \tag{5.7}$$

Because the two Dirichlet distributions in the prior (per-document over topics and per-topic over words) are conjugate to this double multinomial, the posterior is Dirichlet. A compact representation can be achieved using counts

$$n_{dkv} \equiv |\{i \text{ s.t. } c_{di} = k; w_{di} = v\}|. \tag{5.8}$$

We will use a dot to denote that a dimension has been summed out; for example $n_{\cdot kv} = \sum_d n_{dkv}$, etc. With this, we get the joint (note that the left-hand-side variable $\boldsymbol{C}$ is represented implicitly on the right through the counts n_{dkv})

$$p(\boldsymbol{W}, \boldsymbol{C}, \boldsymbol{\Theta}, \boldsymbol{\Pi} \,|\, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_d \frac{\Gamma\left(\sum_k \alpha_{dk}\right)}{\prod_k \Gamma\left(\alpha_{dk}\right)} \prod_k \pi_{dk}^{\alpha_{dk} - 1 + n_{dk\cdot}} \prod_k \frac{\Gamma(\sum_v \beta_{kv})}{\prod_v \Gamma(\beta_{kv})} \prod_v \theta_{kv}^{\beta_{kv} - 1 + n_{\cdot kv}}. \tag{5.9}$$

Standard inference in LDA [Blei et al., 2003] uses a fully factorized approximate distribution

$$\tilde{q}_{\text{LDA}}(\boldsymbol{\Pi}, \boldsymbol{C}, \boldsymbol{\Theta}) = \prod_d \tilde{q}(\boldsymbol{\pi}_d) \prod_i \tilde{q}(c_{di}) \prod_k \tilde{q}(\boldsymbol{\theta}_k) \tag{5.10}$$

on all latent parameters. This raises two problems.

- ▷ Due to the mixing terms $n_{dk\cdot}$, $n_{\cdot k\cdot}$ and $n_{\cdot kv}$, the posterior beliefs over the c_{dn} are highly correlated. The fully factorized bound thu may be loose.

- ▷ Tracking approximate beliefs for $\boldsymbol{\Theta}$ and $\boldsymbol{\Pi}$ is computationally inefficient. This is less a problem of memory or nominal computation cost than of algorithmic design, and related to the previous point: In a standard implementation, the algorithm passes through the corpus, updating the estimates on $\boldsymbol{\Pi}$ and $\boldsymbol{C}$. It then steps out and updates the estimate on $\boldsymbol{\Theta}$. Because the other estimates are highly correlated with $\boldsymbol{\Theta}$, they now have to be re-estimated, and this cycle has to be repeated many times: The algorithm converges slowly. Note that this problem is more pronounced for inference on $\boldsymbol{\Theta}$ than for inference on $\boldsymbol{\Pi}$ because the latter depends directly only on the words in one document,

while the former is a function of the entire corpus.

To address this problem, we adapt Equation (5.10) by introducing a dependence of $\boldsymbol{\Theta}$ on $\boldsymbol{C}$:

$$q(\boldsymbol{\Theta}, \boldsymbol{\Pi}, \boldsymbol{C}) = q(\boldsymbol{\Theta} \,|\, \boldsymbol{C}) \prod_{d}^{D} q(\boldsymbol{\pi}_d) \prod_{i}^{I_d} q(c_{di}) \tag{5.11}$$

The variational bound (see Section 2.3.3) is

$$\begin{aligned}
&\mathcal{L}[q(\boldsymbol{C})q(\boldsymbol{\Theta} \,|\, \boldsymbol{C})q(\boldsymbol{\Pi})] \\
&\quad = \mathsf{E}_{q(\boldsymbol{Z})q(\boldsymbol{\Theta} \,|\, \boldsymbol{Z})q(\boldsymbol{\Pi})} \left[\log p(\boldsymbol{W}, \boldsymbol{C}, \boldsymbol{\Pi}, \boldsymbol{\Theta} \,|\, \boldsymbol{\beta}, \boldsymbol{\alpha})\right] + \mathsf{H}\left[q(\boldsymbol{C})q(\boldsymbol{\Theta} \,|\, \boldsymbol{C})q(\boldsymbol{\Pi})\right] \\
&\quad = \mathsf{E}_{q(\boldsymbol{C})q(\boldsymbol{\Pi})} \left[\mathsf{E}_{q(\boldsymbol{\Theta} \,|\, \boldsymbol{Z})} \left[\log p(\boldsymbol{W}, \boldsymbol{C}, \boldsymbol{\Pi}, \boldsymbol{\Theta} \,|\, \boldsymbol{\beta}, \boldsymbol{\alpha})\right] + \mathsf{H}[q(\boldsymbol{\Theta} \,|\, \boldsymbol{C})]\right] + \mathsf{H}\left(q(\boldsymbol{C})q(\boldsymbol{\Pi})\right)
\end{aligned} \tag{5.12}$$

Since we are not restricting the functional form, optimizing for $q(\boldsymbol{\Theta} \,|\, \boldsymbol{C})$ leads to a unique optimum at the *exact* posterior $p(\boldsymbol{\Theta} \,|\, \boldsymbol{C}, \boldsymbol{W}, \boldsymbol{\beta})$.

$$\begin{aligned}
\mathcal{L}(q(\boldsymbol{C})q(\boldsymbol{\Pi})) &= \max_{q(\boldsymbol{\Theta} \,|\, \boldsymbol{C})} \mathcal{L}[q(\boldsymbol{C})q(\boldsymbol{\Theta} \,|\, \boldsymbol{C})q(\boldsymbol{\Pi})] \\
&= \mathsf{E}_{q(\boldsymbol{C})q(\boldsymbol{\Pi})} \left[\log p(\boldsymbol{W}, \boldsymbol{C}, \boldsymbol{\Pi} | \boldsymbol{\beta}, \boldsymbol{\alpha})\right] + \mathsf{H}\left[q(\boldsymbol{C})q(\boldsymbol{\Pi})\right]
\end{aligned} \tag{5.13}$$

An obvious advantage of this treatment over the classic factorized bound is that, because Equation (5.11) is a strictly weaker assumption on the structure than Equation (5.10), the resulting bound on the true posterior under the model is strictly tighter. A second advantage will become apparent when we derive an explicit scheme for the optimization of remaining the approximate distributions. To do so, we first integrate out all $\boldsymbol{\theta}_k$ from Equation (5.9), arriving at

$$\begin{aligned}
p(\boldsymbol{W}, \boldsymbol{C}, \boldsymbol{\Pi} \,|\, \boldsymbol{\beta}, \boldsymbol{\alpha}) = &\prod_{d} \frac{\Gamma\left(\sum_k \alpha_{dk}\right)}{\prod_k \Gamma\left(\alpha_{dk}\right)} \prod_{k} \pi_{dk}^{\alpha_{dk} - 1 + n_{dk\cdot}} \\
&\prod_{k} \frac{\Gamma\left(\sum_v \beta_{kv}\right)}{\Gamma\left(n_{\cdot k \cdot} + \sum_v \beta_{kv}\right)} \prod_{v} \frac{\beta_{kv} + n_{\cdot kv}}{\Gamma(\beta_{kv})}
\end{aligned} \tag{5.14}$$

we now *choose* a particular parametric form for the approximating distributions, namely independent discrete distributions on the c_{di} with parameters $q(c_{di} = k) = \gamma_{dik}$ and independent Dirichlet distributions on the $\boldsymbol{\pi}_d$, with parameter vectors $\boldsymbol{\nu}_d \in \mathbb{R}_+^K$:

$$q(\boldsymbol{C}) = \prod_{d,i} \prod_{k} \gamma_{dik} \qquad \text{and} \qquad q(\boldsymbol{\Pi}) = \prod_{d} \mathcal{D}(\boldsymbol{\pi_d} \,|\, \boldsymbol{\nu}_d) \tag{5.15}$$

We insert this and Equation (5.14) into Equation (5.13), only retain the terms containing π, and use that the entropy of the Dirichlet is [e.g. Bishop, 2006]

$$\mathsf{H}[D(x \,|\, \omega)] = \log \left(\frac{\prod_k \Gamma(\omega_k)}{\Gamma\left(\hat{\omega}\right)} \right) + \left(\hat{\omega} - K\right) \digamma\left(\hat{\omega}\right) - \sum_k (\omega_k - 1) \digamma\left(\omega_k\right) \tag{5.16}$$

with $\hat{\omega} \equiv \sum_k \omega_k$ and the Digamma function $\digamma$, which is the first derivative of the log of the Gamma function. This gives an explicit expression for the bound on each document's topic proportion as a function of ν_d, up to additive constants:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\nu}_d) = &\sum_k [\alpha_{dk} - 1 + \mathsf{E}_q(n_{dk\cdot})]\mathsf{E}_q[\log(\pi_{dk})] \\ &+ (\hat{\nu}_d - K)\digamma(\hat{\nu}_k) - \sum_k (\nu_{dk} - 1)\digamma(\nu_{dk}) \\ &+ \sum_k \log\Gamma(\nu_{dk}) - \log\Gamma(\hat{\nu}_k)\end{aligned} \tag{5.17}$$

Using that the expected logarithm under Dirichlet beliefs is

$$\mathsf{E}_{\mathcal{D}(\omega)}[\log x] = \digamma(\omega_k) - \digamma(\hat{\omega}) \tag{5.18}$$

we get

$$\begin{aligned}\mathcal{L}(\boldsymbol{\nu}_d) = &\sum_k (\alpha_{dk} + \mathsf{E}_q(n_{dk\cdot}) - \nu_{dk}) \left[\digamma(\nu_{dk}) - \digamma\left(\sum_\ell^K \nu_{d\ell}\right)\right] \\ &+ \sum_k \log\Gamma(\nu_{dk}) - \log\Gamma(\hat{\nu}_k).\end{aligned} \tag{5.19}$$

Differentiating this expression with respect to ν_{dk} and setting to zero establishes an optimum at

$$\nu_{dk} = \alpha_{dk} + \mathsf{E}_q(n_{dk\cdot}). \tag{5.20}$$

We will deal with the expectation in this equation in the next section. First, we consider the optimization with respect to γ_{dik}. To do so, we introduce the notation $n_{dkv}^{\backslash di}$ for n_{dkv} counting all words except the one at location (d, i). The entropy of the discrete distribution with parameters γ_v is $\mathsf{H}_\gamma[q(c)] = -\sum_v \gamma_v \log \gamma_v$. Further, we can expand

$$\log \frac{\Gamma(x+a)}{\Gamma(x)} = \sum_{\ell=0}^{a-1} \log(x + \ell) \qquad \forall x \in \mathbb{R}_+ \,;\, a \in \mathbb{N}. \tag{5.21}$$

Using all this, again Equation (5.14) and some of the results in the previous equations, canceling terms appearing in both nominator and denominator, we arrive at the update rule

$$\gamma_{dik} \propto \exp\left\{\digamma(\nu_{dk}) + \mathsf{E}_q\left[\log(\beta_{kw_{di}} + n_{\cdot kw_{di}}^{\backslash di})\right] - \mathsf{E}_q\left[\sum_v \beta_{kv} + n_{\cdot k\cdot}^{\backslash di}\right]\right\}. \tag{5.22}$$

A Gaussian approximation to the expected values of counts

The final problem in this section is the evaluation of the expectation terms in Equations (5.19) and (5.22). Computing these terms exactly is computationally expensive. However, Teh et al. [2007] point out that these terms are sums of usually large numbers of independent Bernoulli variables. For example,

$$n_{\cdot kv}^{di} = \sum_{d' \neq d; i' \neq i} \mathbb{I}(c_{d'i'} = k) \tag{5.23}$$

(and analogously for the other counts). Sums of independent Bernoulli variables are approximately Gaussian distributed, with mean and variance given by the sums of the means and variances of the individual Bernoulli variables

$$\begin{aligned} \mathsf{E}_q(n_{\cdot k\cdot}) &= \sum_{d,i} \gamma_{dik} \\ \text{and} \qquad \mathsf{Var}_q(n_{\cdot k\cdot}) &= \sum_{d,i} \gamma_{dik}(1 - \gamma_{dik}) \end{aligned} \tag{5.24}$$

We can thus approximate by expanding

$$\mathsf{E}_q[\log(\beta_{kw_{di}} + n_{\cdot kw_{di}}^{\backslash di})] \approx \log(\beta_{kw_{di}} + \mathsf{E}_q[n_{\cdot kw_{di}}^{\backslash di}]) - \frac{\mathsf{Var}_q(n_{\cdot kw_{di}}^{\backslash di})}{2(\beta_{kw_{di}} + \mathsf{E}_q[n_{\cdot kw_{di}}^{\backslash di}])^2} \tag{5.25}$$

Teh et al. [2007] studied this approximation and concluded that, even though the count is usually larger than β, this Taylor expansion gives good results in this application. In our model, this gives the update equation

$$\begin{aligned} \gamma_{dik} \propto \exp[F(\nu_{dk})] & \frac{\beta_{kw_{di}} + \mathsf{E}_q n_{\cdot kw_{di}}^{\backslash di}}{\sum_v \beta_{kv} + \mathsf{E}_q(n_{\cdot k\cdot}^{\backslash di})} \\ & \cdot \exp\left\{ - \frac{\mathsf{Var}_q\, n_{\cdot kw_{di}}^{\backslash di}}{2(\beta_{kw_{di}} + \mathsf{E}_q[n_{\cdot kw_{di}}^{\backslash di}])^2} + \frac{\mathsf{Var}_q\, n_{\cdot k\cdot}^{\backslash di}}{2(\sum_v \beta_{kv} + \mathsf{E}_q[n_{\cdot k\cdot}^{\backslash di}])^2} \right\}. \end{aligned} \tag{5.26}$$

Comparison to fully collapsed model

It is instructive to take a short diversion here and study the differences between this *semi*-collapsed algorithm, where only $\boldsymbol{\Theta}$ is integrated out, and the fully collapsed algorithm derived in Teh et al. [2007]. In the latter scheme, there is no

approximation on $\mathbf{\Pi}$ and the update rule for γ_{dik} is

$$\gamma_{dik} \propto (\alpha_{dk} + \mathsf{E}_q(n_{dk\cdot}^{\backslash di}) \frac{\beta_{kw_{di}} + \mathsf{E}_q n_{\cdot kw_{di}}^{\backslash di}}{\sum_v \beta_{kv} + \mathsf{E}_q(n_{\cdot k\cdot}^{\backslash di})} \cdot \exp\left\{ - \frac{\mathsf{Var}_q\, n_{dk\cdot}^{\backslash di}}{2(\alpha_{dk} + \mathsf{E}_q[n_{dk\cdot}^{\backslash di}])^2} - \frac{\mathsf{Var}_q\, n_{\cdot kw_{di}}^{\backslash di}}{2(\beta_{kw_{di}} + \mathsf{E}_q[n_{\cdot kw_{di}}^{\backslash di}])^2} + \frac{\mathsf{Var}_q\, n_{\cdot k\cdot}^{\backslash di}}{2(\sum_v \beta_{kv} + \mathsf{E}_q[n_{\cdot k\cdot}^{\backslash di}])^2} \right\} \tag{5.27}$$

We observe further that the Taylor expansion of the Digamma function is given by ([Abramowitz and Stegun, 1972, §6.4.12])

$$\digamma(x) = \log(x) - \frac{1}{2x} + \mathcal{O}(x^{-2}). \tag{5.28}$$

So if ν_{dk} is updated after every update of γ (which is not a computationally efficient scheme, but possible in principle), the two update rules are identical, up to second order corrections, up to a factor

$$\begin{aligned}
\Delta &= \frac{\gamma_{dik}^{\text{semi-collapsed}}}{\gamma_{dik}^{\text{collapsed}}} \\
&\approx \exp\left(-\frac{1}{2(\alpha + \mathsf{E}_q[n_{dk\cdot}^{\backslash di}])} + \frac{\mathsf{Var}_q\,[n_{dk\cdot}^{\backslash di}]}{2(\alpha + \mathsf{E}_q[n_{dk\cdot}^{\backslash di}])^2} \right) \\
&= \exp\left(\frac{\mathsf{Var}_q\,[n_{dk\cdot}^{\backslash di}] - \alpha - \mathsf{E}_q[n_{dk\cdot}^{\backslash di}]}{2(\alpha + \mathsf{E}_q[n_{dk\cdot}^{\backslash di}])^2} \right) \\
&= \exp\left(-\frac{\alpha + \sum_{i' \neq i} \gamma_{di'k}^2}{2\left(\alpha + \sum_{i' \neq i} \gamma_{di'k}\right)^2} \right) \qquad \text{(using Eq. 5.24)}
\end{aligned} \tag{5.29}$$

Because $\gamma_{dik} < 1\ \forall d, i, k$, we have $\Delta \sim 1$, and the update rules are approximately equal up to second order corrections. From this point of view, the main difference between the two schemes is thus one of scheduling. The fully collapsed scheme can utilize a slightly more efficient representation of the approximate beliefs, which can aid mixing. However, for large corpora where the mean field is dominated by effects from other documents, the differences between the two algorithms can be expected to be small.

5.2.2 Laplace Approximation for Dirichlets

The previous section introduced a variational inference scheme for the subgraph to the right of and including $\boldsymbol{\pi}_d$ in Figure 5.2. The next section will construct a fast approximate message passing scheme for regression from Gaussian beliefs on

$\boldsymbol{y}_d$, to the left of that variable in the graphical model. To link these two parts of the inference, we require some way of connecting the beliefs on $\boldsymbol{\pi}_d$ and $\boldsymbol{y}_d$. The deterministic functional relationship $\boldsymbol{\sigma}(\boldsymbol{y}_d) = \boldsymbol{\pi}_d$ identifies this task as a probabilistic form of logistic regression (probabilistic in the sense that discrete samples c_{dn} from the distribution defined by $\boldsymbol{\pi}_d$ are replaced by probabilistic beliefs over c_{dn}).

As pointed out above, solutions to this problem found by previous authors suffer from comparatively high computational cost. Here, we will retain the Dirichlet beliefs on $\boldsymbol{\pi}_d$ returned by the semi-collapsed variational scheme, and explicitly construct approximate Gaussian beliefs on $\boldsymbol{y}_d$. For this, we utilize a Laplace approximation in the softmax basis, which was derived by MacKay [1998], and can be extended such that it provides an invertible map from the set of parameters of K-dimensional Dirichlet distributions to a subset of the parameters of K-dimensional Gaussians, which can be approximately represented by the means and variances of K independent Gaussians. The detailed derivation of this approximation is lengthy, and has thus been moved to Appendix C. The map between a Dirichlet with pseudocounts $\boldsymbol{\alpha}$ and a multivariate Gaussian with mean and covariance matrix $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is (Equations C.32 and C.21)

$$\mu_k = \log \alpha_k - \frac{1}{K} \sum_{\ell=1}^{K} \log \alpha_\ell \tag{5.30}$$

$$\Sigma_{k\ell} = \delta_{k\ell} \frac{1}{\alpha_k} - \frac{1}{K} \left[\frac{1}{\alpha_k} + \frac{1}{\alpha_\ell} - \frac{1}{K} \left(\frac{1}{\tau} + \sum_{u}^{K} \frac{1}{\alpha_u} \right) \right] \tag{5.31}$$

$$\text{and } \alpha_k = \frac{1}{\Sigma_{kk}} \left(1 - \frac{2}{K} + \frac{e^{\mu_k}}{K^2} \sum_{\ell}^{K} e^{-\mu_\ell} \right) \qquad \text{for } k = 1, \ldots, K \tag{5.32}$$

In the following Section 5.2.3, we will use this approximation in a special form of generalized regression, to link a set of approximately independent Gaussians to a Dirichlet distribution. A few interesting characteristics to note are:

- The correlation between components of the Gaussian approximation is small for $K \gg 1$ and will thus be ignored here, giving K independent Gaussians with means as above and variances (see also Equation C.22)

  $$\sigma_k = \Sigma_{kk} = \frac{1}{\alpha_k} \left(1 - \frac{2}{K} \right) + \frac{1}{K^2} \sum_{\ell}^{K} \frac{1}{\alpha_\ell} \tag{5.33}$$

  In the other direction, from Gaussians to Dirichlets, the approximation discards any correlation structure between the components. Since the factorized approximation to be introduced in Section 5.2.3 does not capture such correlations anyway, this does not cause any additional issues.

▷ The Gaussian resulting from this map has weaker tails than the Dirichlet distribution in the softmax basis. The resulting weights are thus slightly overconfident. The factorized regression introduced in the next section will have a similar defect, and both of these problems need to be addressed by ad-hoc solutions.

▷ Inspecting the Equations following (5.30), it is clear that this approximation is well defined for the entire parameter space of Dirichlets, including values of $0 < \alpha_k < 1$. For such sparse cases, the resulting Gaussian approximation can have considerable uncertainty, but it does not lead to ill-defined parameter settings.

See Figure C.3 in Appendix C for an intuition on this approximation.

5.2.3 Gaussian Regression

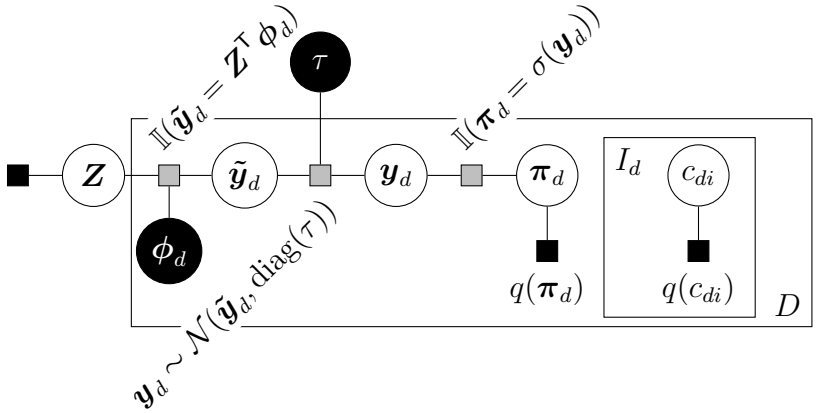


Figure 5.3: Factor graph representation of approximate inference in the conditional topic model. Note that the semi-collapsed variational algorithm provides independent beliefs $q(\boldsymbol{\pi}_d)$ on the per-document topic distributions, allowing inference by message passing on the weights $\boldsymbol{Z}$ in the Gaussian regression part. The regression is here represented in its exact joint form. Figure 5.4 and Section 5.2.3 introduce a further factorizing approximation allowing $\mathcal{O}(EK)$ inference, where E is the average number of non-zero features per document.

Under the exact model given by Eq. (5.9), the beliefs on the per-document topic distributions $\boldsymbol{\pi}_d$ are correlated indirectly through the correlation of the word topic labels c_{dn}, and would only become independent if these labels were observed. The semi-collapsed variational inference scheme introduced in Section 5.2.1 explicitly constructed *independent* beliefs on the $\boldsymbol{\pi}_d$ (see factor graph representation in Figure 5.3). In this section, we will make use of these independent beliefs to learn relations between observable features of the document and the topic mixture of documents, using the linear model with weights z_{fk} as defined in Eq. (5.3). Thanks to the Laplace map introduced in Section 5.2.2, we have access to an approximate

Gaussian belief on the inverse softmax of $\boldsymbol{\pi}_d$, denoted by $\boldsymbol{y}_d = \sigma^{-1}(\boldsymbol{\pi}_d)$. This reduces the problem to straightforward Gaussian regression, which has been studied very extensively in the past [e.g. Bishop, 2006, Section 3.3], and will be only outlined in this section.

Fully Connected Regression

Noting that the Laplace approximation returns a correlated belief over the elements of $\boldsymbol{y}_d$, and because "explaining away" causes correlations between the weights z_{fk} for different features f, an exact treatment of the regression problem calls for a fully connected (i.e. joint) posterior on the weights $\boldsymbol{Z}$. Because this exact solution requires the inversion of a $K \times K$ matrix, of cost $\mathcal{O}(K^3)$, it is usually too costly for real applications, and the next Section 5.2.3 will introduce a factorized scheme which is not exact, but much faster. For completeness, this section first provides the exact answer to the regression problem. It is the multivariate version of the Gaussian linear regression introduced in Section 2.3.6.

Because we will be interested in the posterior on the z_{fk}, to simplify the algebra, we introduce the stacked vector $\boldsymbol{z} = \mathrm{vec}(\boldsymbol{Z}) \in \mathbb{R}^{KF}$, i.e. a re-arrangement of the matrix $\boldsymbol{Z}$ into a vector, and the matrix $\boldsymbol{F}_d \in \mathbb{R}^{K \times FK}$, a re-arrangement of the vector $\boldsymbol{\phi}_d$, such that

$$\boldsymbol{Z}^\mathsf{T}\boldsymbol{\phi}_d \equiv \boldsymbol{F}_d\boldsymbol{z} = \tilde{\boldsymbol{y}}_d \tag{5.34}$$

The concrete realization of the projection is irrelevant for the analysis, as long as it is consistent. The likelihood for $\boldsymbol{z}$ under one document "data point" $\boldsymbol{y}_d$ is then

$$p(\boldsymbol{y}_d \,|\, \boldsymbol{z}, \boldsymbol{F}) = \mathcal{N}(\boldsymbol{y}_d; \boldsymbol{F}_d\boldsymbol{z}, \mathrm{diag}(\tau^2)) \tag{5.35}$$

To get the full posterior, assuming a Gaussian prior $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{m}, \boldsymbol{S})$, we use Bayes' rule, and "complete the square" (for a step-by-step derivation, see for example Bishop [2006], Section 2.3.6). The posterior is also Gaussian, and takes the form

$$\begin{aligned} p(\boldsymbol{z} \,|\, \boldsymbol{y}_d, \boldsymbol{F}_d) &= \mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\Psi}\left[\boldsymbol{F}^\mathsf{T}\,\mathrm{diag}(\tau^{-2})\boldsymbol{y}_d + \boldsymbol{S}^{-1}\boldsymbol{m}\right], \boldsymbol{\Psi}\right) \\ \text{where} \qquad \boldsymbol{\Psi} &= \left[\boldsymbol{S}^{-2} + \boldsymbol{F}^\mathsf{T}\,\mathrm{diag}(\tau^{-2})\boldsymbol{F}\right]^{-1} \\ &= \boldsymbol{\Sigma} - \boldsymbol{S}\boldsymbol{F}^\mathsf{T}(\mathrm{diag}(\tau^2) + \boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^\mathsf{T})^{-1}\boldsymbol{F}\boldsymbol{S} \end{aligned} \tag{5.36}$$

where we have applied the matrix inversion lemma (C.17) in the last conversion, which reduces the size of the matrix to be inverted from $KF \times KF$ to the size $K \times K$ of the Schur complement. Since the variational inference provides only a belief $p(\boldsymbol{y}_d) = \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d)$, rather than an exact value, we have to marginalize

over $\boldsymbol{y}$:

$$\begin{aligned} p(\boldsymbol{z} \mid \boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d, \boldsymbol{F}_d) &= \int p(\boldsymbol{z} \mid \boldsymbol{y}_d, \boldsymbol{F}_d) p(\boldsymbol{y}_d)\, \mathrm{d}\boldsymbol{y}_d \\ &= \mathcal{N}\big(\boldsymbol{z}; \boldsymbol{\Psi}\left[\boldsymbol{F}_d{}^{\mathsf{T}} \operatorname{diag}(\tau^{-2})\boldsymbol{\mu}_d + \boldsymbol{S}^{-1}\boldsymbol{m}\right], \\ &\qquad \boldsymbol{\Psi} + \boldsymbol{\Psi}\boldsymbol{F}_d{}^{\mathsf{T}} \operatorname{diag}(\tau^{-2})\boldsymbol{\Sigma}_d \operatorname{diag}(\tau^{-2})\boldsymbol{F}_d\boldsymbol{\Psi}\big). \end{aligned} \tag{5.37}$$

This scheme corresponds to applying the Sum-Product algorithm (Section 2.2) to the factor graph shown in Figure 5.3. Note the neat separation of the two different types of correlation present in the model: Equation (5.36) introduces the "explaining away" correlations between features, caused by the sum factor (see Section 2.3.6 and Figure 2.8); Equation (5.37) propagates the correlations between the elements (topics) of $\boldsymbol{y}_d$, created by the softmax factor and encoded in the matrix $\boldsymbol{\Sigma}_d$. This separation is indicated by the colours in the covariance matrix in Equation (5.37), which should be compared with the colour coding in Figures 5.1 and 5.4.

Factorized Regression

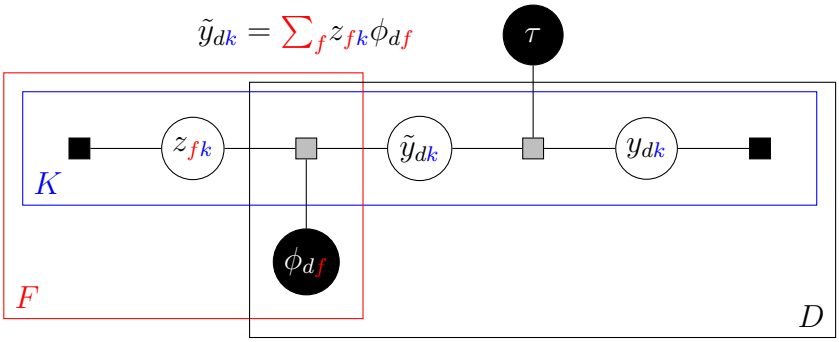


Figure 5.4: Factor graph representation of factorized Gaussian regression using the sum factor.

In large-scale applications of topic models, the dimensionalities of the topic model can be considerable. For a concrete example, consider once again the Twitter messaging service. An obvious feature of interest in Twitter posts is the user-ID, which currently means that $F > 5 \cdot 10^7$ using a 1-in-n encoding. With so many features, even small values of K lead to prohibitively high memory requirements if Equation (5.36) is used for regression. As mentioned in Section 2.3.6, implementations of linear cost in D and K that can also deal with sparse features efficiently are then the only possible solutions. This, of course, comes at the cost of accuracy, as we will have to make further approximating assumptions.

Figure 5.4 shows a factor graph for a fully factorized approximation. We retain only the diagonal elements σ_k^2 of $\boldsymbol{\Sigma}$ and we will store only the diagonal elements

m_{fk}, s^2_{fk} of the elements of $\boldsymbol{m}$ and $\boldsymbol{S}$. With the likelihood

$$p(z_{fk} \mid z_{f' \neq f,k}, y_{dk}, \boldsymbol{\phi}_d, \tau) = \mathcal{N}\left[z_{fk}; \frac{1}{\phi_{df}}\left(y_{dk} - \sum_{f' \neq f} \phi_f z_{f'k}\right), \frac{\tau^2}{\phi^2_{df}}\right], \tag{5.38}$$

the sum-product message into z_{fk} is easily obtained by marginalizing over the beliefs on all other variables connected to the sum factor, giving

$$\mathrm{msg}(z_{fk}) = \mathcal{N}\left[z_{fk}; \frac{1}{\phi_{df}}\left(\mu_{dk} - \sum_{f' \neq f} \phi_{df'} m_{f'k}\right), \frac{1}{\phi^2_{df}}\left(\tau^2 + \sigma^2_{dk} + \sum_{f' \neq f} \phi^2_{df'} s^2_{f'k}\right)\right] \tag{5.39}$$

Such factorized approximations have been used widely in the machine learning literature to construct linear-cost algorithms. Since we had to discard several correlation effects to arrive at this result, this scheme will make overconfident predictions. Equations (C.7) and (C.10) give an intuition for the factor determining the quality of this approximation: We can expect it to work well if K is large and $\boldsymbol{\alpha}$ not extremely sparse, and if $\boldsymbol{\phi}_d$ is sparse. The first and last assumptions are usually well satisfied by topic models on big corpora. The effect of a sparse $\boldsymbol{\alpha}$ (which is required to a certain degree for a useful topic model) is of order $\mathcal{O}(1/K)$ and thus less problematic.

5.2.4 Inference from Data Streams

Since the topic-model estimates depend on the regression predictions and vice versa, inference in the overall model should be done iteratively in principle: Iterate over the corpus; infer the topics of each document, passing Gaussian messages to the regression weights z_{fk}. For each document d, make a prediction from the current beliefs on the weights z_{fk} and the features $\boldsymbol{\phi}_d$. This prediction provides a Gaussian message to $\boldsymbol{y}_d$ which, after application of the Laplace map, acts as a Dirichlet prior on $\boldsymbol{\pi}_d$. Semi-collapsed variational inference on the topics leads to new Dirichlet and Gaussians marginals on $\boldsymbol{\pi}_d$ and $\boldsymbol{y}_d$, respectively. Dividing out the Gaussian message to $\boldsymbol{y}_d$ provides a message into the sum factor, with which the beliefs on the weights are updated. In subsequent iterations, an updated prediction can be gathered by dividing the marginals on z_{fk} by the message sent by y_{dk} in the previous iteration.

For very large corpora which continue to grow during inference, such as our example of Twitter, such a scheme is not feasible, because the rate of growth of the corpus is comparable to the speed of inference. In such a setup, an online, or single-pass scheme can be constructed in the following way: For each new document d arriving in the corpus, predict the topic distributions based on $\boldsymbol{\phi}_d$ just as above. Semi-collapsed variational inference on d, to convergence, returns an update message to

the weights as above. Now, however, the messages to the weights received earlier during the inference are incorrect because they are the result from a less converged topic model. Situations like this are typical for online algorithms, and can be addressed by ad-hoc methods such as *exponential forgetting*: After each update to the beliefs $p(z_{fk}) \sim \mathcal{N}(m_{fk}, s_{fk}^2)$, decrease the precision of the belief to

$$s_{fk}^{-2} \leftarrow s_{fk0}^{-2} + \varepsilon(s_{fk}^{-2} - s_{fk0}^{-2}) \qquad \text{where } \varepsilon \lesssim 1. \tag{5.40}$$

The analogous operation on the side of the per-topic word distributions is to multiply the counts $n_{\cdot kv}$ with a constant $\xi \lesssim 1.0$ in regular intervals, which corresponds to raising the variance of the corresponding Dirichlet beliefs. Algorithm 5.2.4 contains high-level pseudocode describing the individual steps necessary for topic inference from a single document in a stream.

Algorithm 2 Single-Pass Conditional Topic Inference.

Require: $\mathsf{E}[n_{\cdot kv}], \mathsf{Var}[n_{\cdot kv}], \mathsf{E}[n_{\cdot k\cdot}], \mathsf{Var}[n_{\cdot k\cdot}]\ \forall k, v$ ▷ topic stats
Require: $\mathcal{N}(m_{fk}, s_{fk}) \forall f, k$ ▷ regression stats
1: **procedure** INFER($\boldsymbol{\phi}_d$,$\boldsymbol{w}_d$)
2: add noise to $p(\boldsymbol{z}_f)\ \forall f$ with $\phi_{df} \neq 0$ ▷ Eq. (5.40)
3: $m_{\rightarrow}(y_{dk}) \leftarrow \mathcal{N}(\boldsymbol{\phi}_d^{\mathsf{T}} \boldsymbol{m}_k, \tau^2 + \sum_f \phi_{fk}^2 s_{fk}^2)$ ▷ incoming message
4: $\mathcal{D}(\boldsymbol{\pi}_d; \boldsymbol{\alpha}_d) \leftarrow \text{LAPLACE}(\mathcal{N}(\boldsymbol{y}_d; \boldsymbol{\mu}_d, \boldsymbol{\sigma^2}_d))$ ▷ Eq. (5.30)
5: $\boldsymbol{\nu}_d \leftarrow \boldsymbol{\alpha}_d$; $\boldsymbol{\gamma}_{di} \leftarrow \boldsymbol{\alpha}_d$+rnd() ▷ initialise variational parameters
6: **repeat** ▷ variational inference
7: **for** $i \in 1, \ldots, I_D$ **do**
8: $\boldsymbol{\gamma}_{di} \leftarrow$ Equation (5.26)
9: update $\mathsf{E}[n_{\cdot kv}], \mathsf{Var}[n_{\cdot kv}], \mathsf{E}[n_{\cdot k\cdot}], \mathsf{Var}[n_{\cdot k\cdot}]$ ▷ $\mathcal{O}(1)$ update
10: **end for**
11: $\boldsymbol{\nu}_d \leftarrow$ Equation (5.20)
12: **until** converged
13: $\mathcal{N}(\boldsymbol{y}_d; \boldsymbol{\mu}', \boldsymbol{\sigma^2}') \leftarrow \text{LAPLACE}(\mathcal{D}(\boldsymbol{\pi}_d; \boldsymbol{\alpha}_d + \boldsymbol{\nu}_d))$ ▷ Eq. (5.30)
14: $m_{\leftarrow}(y_{dk}) \leftarrow \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\sigma^2}')/\mathcal{N}(\boldsymbol{\mu}_d, \boldsymbol{\sigma^2}_d)$ ▷ message to regression
15: $\text{msg}(z_{fk}) \leftarrow$ Equation (5.39) $\forall f, k$
16: $p(z_{fk}) \leftarrow p(z_{fk}) \cdot \text{msg}(z_{fk})$ ▷ assumed density filtering
17: **end procedure**

5.2.5 Queries to the Model

From the point of view of the user, there are several different queries that can be posed to a conditional topic model, after it has converged on a sufficiently large corpus:

▷ To infer a topic distribution for a given document from $(\boldsymbol{\phi}_d, \boldsymbol{w}_d)$, use Algorithm 5.2.4 up to line 11, return

$$q(\boldsymbol{\pi}_d \,|\, \boldsymbol{\phi}_d, \boldsymbol{w}_d) = \mathcal{D}(\boldsymbol{\pi}_d; \boldsymbol{\alpha}_d + \boldsymbol{\nu}_d)$$

▷ To predict the topic distribution given certain features $\boldsymbol{\phi}_d$, use Algorithm 5.2.4 up to line 4

$$q(\boldsymbol{\pi}_d \mid \boldsymbol{\phi}_d) = \mathcal{D}(\boldsymbol{\pi}_d; \boldsymbol{\alpha}_d)$$

▷ An approximate posterior belief over the topics themselves is given by

$$\bar{q}(v \mid k) = \mathcal{D}(v; \boldsymbol{\beta}_k + \mathsf{E}[n_{\cdot kv}])$$

5.3 Experiments

There are three distinct approximations used in our inference algorithm:

1. a variational bound on $\boldsymbol{\Pi}$ and $\boldsymbol{C}$ (Section 5.2.1)
2. the "Laplace bridge" between the regression and mixture model parts of the generative process (Section 5.2.2)
3. the factorization assumptions in the sparse-feature regression (Section 5.2.3)

The quality of the variational bound has been studied in Teh et al. [2007]. Factorization assumptions in linear regression are well understood and used extensively throughout machine learning and thus will not be re-evaluated here. What remains is the use of the Laplace approximation.

5.3.1 Quality of the Laplace Bridge

Although the quality of the Laplace approximation to the Dirichlet in the softmax basis in itself has previously been investigated in MacKay [1998], the use of this approximation here differs considerably from the setting studied in the cited paper (which dealt with evidence estimation in neural networks). The setup in which the approximation is used here effectively amounts to the following:

▷ Some unobserved process with known parameters $\boldsymbol{\mu}, \boldsymbol{\zeta}$ generates data according to the following process:

- Sample $\boldsymbol{x} \in \mathbb{R}^K \sim \prod_k \mathcal{N}(\mu_k, \zeta_k^2)$
- Map $\boldsymbol{\pi} = \boldsymbol{\sigma}(\boldsymbol{x})$
- Sample data points c according to $p(c = k \mid \boldsymbol{\pi}) = \boldsymbol{\pi}$

▷ The inference method tries to infer $\boldsymbol{x}$ in the following way:

- Use the Laplace map to gain a Dirichlet belief on $\boldsymbol{\pi}$ from the prior $\prod_k \mathcal{N}(\mu_k, \zeta_k)$

- Update this belief using the data (which is trivial, due to the Dirichlet's conjugacy to the Multinomial distribution)
- map the Dirichlet belief back to $\mathbb{R}^k$ using the Laplace map in the opposite direction; claim the resulting belief to be an approximate posterior on $\boldsymbol{x}$

Figure 5.5 compares this approximate scheme to an asymptotically exact Markov Chain Monte Carlo scheme (the particular MCMC method chosen for this task is elliptical slice sampling [Murray et al., 2010], which has the advantages of fast convergence and having no free parameters). More specifically, the Figure shows — averaged over 10 independent experiments — the 2-norm error of a point estimate for $\boldsymbol{x}$ returned by the two methods (solid lines) and error estimates constructed from the algorithms' results. For the MCMC sampler, these two estimates are the sample mean and (unbiased) sample covariance. For the Laplace approximations, the two estimates are the mean and standard deviation of the approximate Gaussian belief. Note that the Laplace bridge does not show any discernible bias or over-convergence, except that the error estimate is slightly too big.

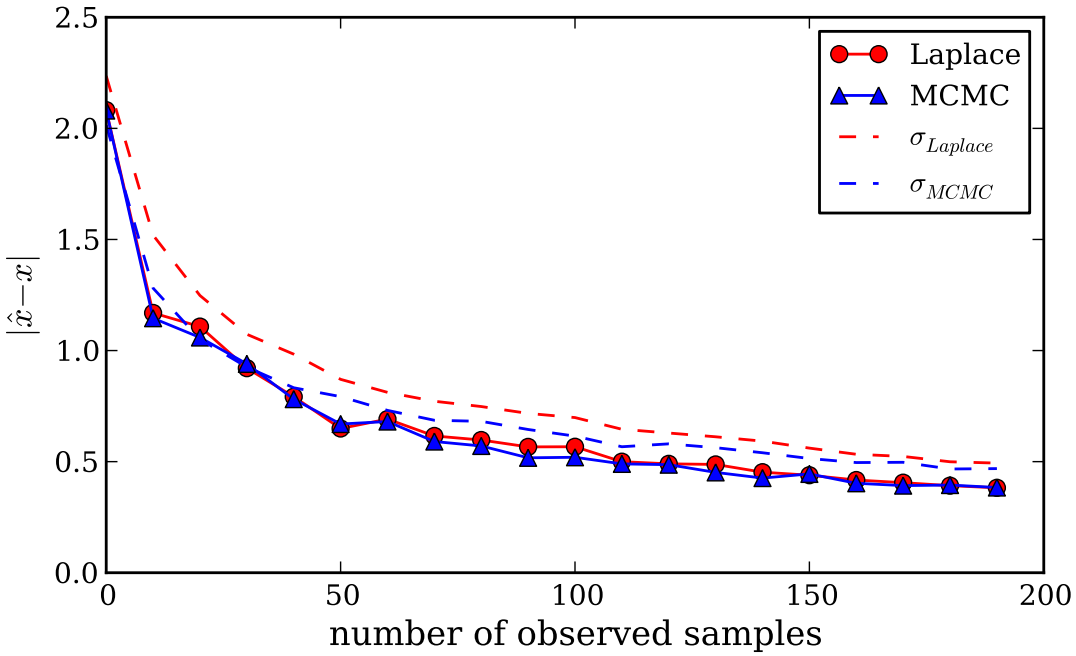


Figure 5.5: Convergence behaviour of approximate inference using the Laplace bridge compared to exact inference using a Markov Chain Monte Carlo scheme. The solid lines and data points represent the deviation of the mean estimate (sample mean for MCMC) from ground truth, the dashed lines represent the error estimate of the inference algorithm (one standard deviation). The results shown are averages over 10 independent experiments. Both methods were initialised with a prior of $\mu = 0, \zeta = 1$.

5.3.2 Experiments on the Twitter Corpus

Twitter provides a challenging dataset: Because the individual documents are very short (< 140 characters, but regularly as short as 3–5 words), the overall number of co-occurences, which are central to topic models, is low even if the number of documents is large. We generated a dataset of Twitter status updates by selecting posts from 19 different "Twitter lists" — collections of posts authored by users with a stated topical preference, which yields a certain "denseness" of authors. Some posts by random authors from the unbiased Twitter stream were inserted as well. A set of $6,635$ authors was selected by choosing those authors with more than 5 documents in the corpus. The parameters of the topic model were set to $K = 40$, with a fixed value $\beta = 10^{-4}$ for all words in all topics (these parameter settings were found through a rough grid-search based on log evidence on the training set). Experiments without conditioning on features were performed with a fixed $\alpha = 10^{-2}$. When the model was conditioned on features, the individual weights were given Gaussian priors such that the average document had an initial prior (i.e. before learning the regression weights) producing a Dirichlet prior on $\boldsymbol{\pi}_d$ with approximately the same parameter value $\alpha = 10^{-2}$. The noise in the regression term was set to $\tau = 10^{-4}$. The feature set was chosen with $F = 6,636$, with one always present bias feature $\phi_0 = 1$, and 6,635 binary features for every author in the aforementioned set of prolific users. The model was run for 100 iterations in the iterative setup.

Only minimal preprocessing was performed: The text of each document was converted to lower case, split on spaces, punctuation was removed, and members of a conservative list of 180 stop words were removed. Of the remaining set of words, all terms with frequencies < 10 in the corpus were discarded.

5.3.3 Learning Sparse Topics

After inference, the learned regression weights for each author can be used to generate predictions of the topics used by this author. If a broad Gaussian prior is used for the regression, and τ is chosen relatively large (e.g. $1 \lesssim \tau \lesssim 10$), then the learned features will be sparse. Figure 5.6 gives an impression of the sparsity of the topic distributions learned by our model. Most authors can be represented almost completely with only a few topics. Also note the kink in tails of the empirical cumulative density functions shown in Figure 5.6, indicating the presence of a "heavy tail", assigning finite probability to all topics. Such a tail is expected, because the fixed regression noise τ enforces a non-zero pseudo-count for all dimensions of the Dirichlet prior on documents, even in the limit of infinite data. Since Figure 5.6 is rather technical and might be difficult to interpret, Figure 5.7 shows concrete topic distributions for four specific authors (the same authors

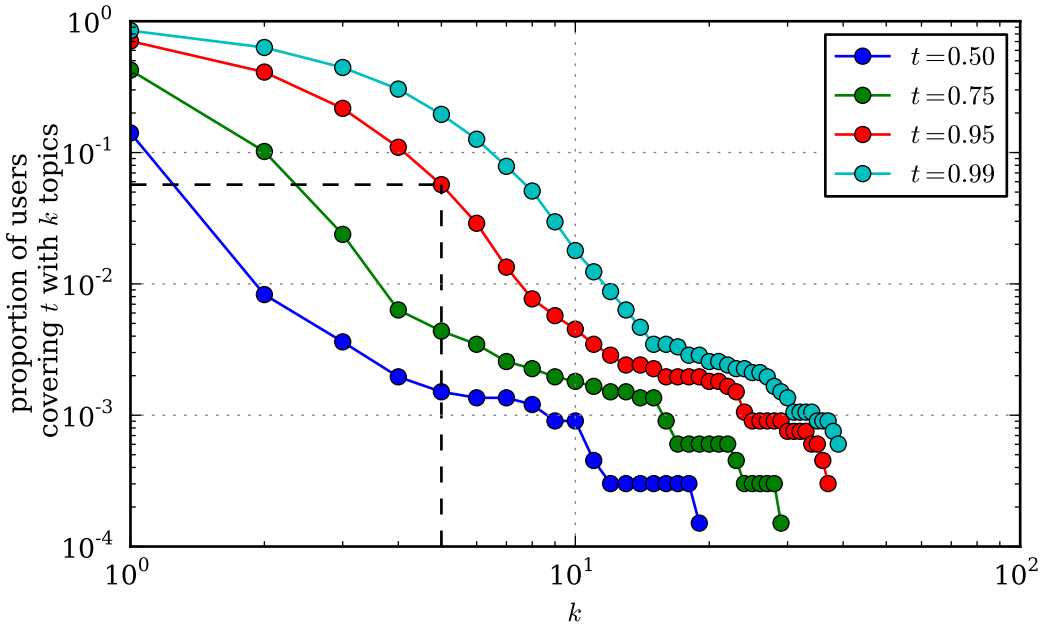


Figure 5.6: Sparsity of learned author topic predictions. Shown are, for different values of $0 < t < 1$, the proportion of authors in the dataset for which the top k topics cover less than t of the probability mass of the topic predictions. For example (dashed black lines), for $\sim (1 - 0.05) = 95\%$ of all authors, 5 topics cover at least 95% of the prediction mass. Note the double logarithmic scale. Lines for visual aid only.

will be used again at the end of Section 5.3.4).

5.3.4 Comparing Conditioned and Unconditioned Models

There is no generally accepted way of evaluating the performance of a topic model. Since inference in topic models is an unsupervised learning problem, there is no ground truth to compare to, and Chang et al. [2009] showed that evaluations based on predictive strength (i.e. perplexity; log probability of the dataset) can differ from subjective evaluations by human subjects. We will thus strike a compromise by using human evaluations in this section, and predictive strength in Section 5.3.5.

Figures 5.8 and 5.9 show the topics learned by conditional topic models with and without access to the author's ID, respectively. Following Blei and Lafferty [2009], we do not represent topics by words ranked by raw frequency, but weighted according to the following score inspired by the tf-idf heuristic [Salton and McGill, 1983]:

$$\text{score}_{kv} = \hat{\beta}_{kv} \left[\log \hat{\beta}_{kv} - \frac{1}{K} \sum_{\ell=1}^{K} \log \hat{\beta}_{\ell v} \right] \tag{5.41}$$

where

$$\hat{\beta}_{kv} = \frac{\beta_{kv} + \mathsf{E}_q n_{\cdot kv}}{\sum_v \beta_{kv} + \mathsf{E}_q n_{\cdot k \cdot}} \tag{5.42}$$

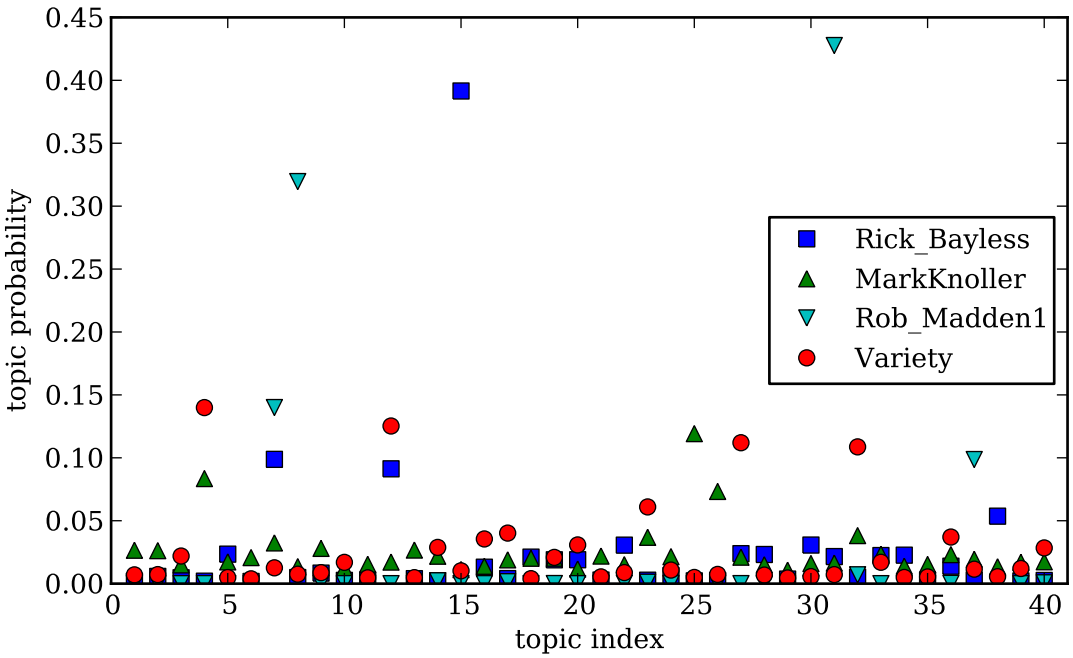


Figure 5.7: Predictive topic distributions conditioned on individual authors, for four authors in the dataset. This Figure provides an intuition for the distributions characterized more abstractly for the whole data set in Figure 5.6. See Figure 5.10 for more details on these authors, Figure 5.8 for a characterisation of the learned topics.

For words with the same marginal probability, this weights words of uneven distributions over the topics higher than those with more even distributions, thereby providing a more informative representation of each topic (note that $\hat{\beta}_{kv}$ is the model's estimate of the probability of word v in topic k, the second factor in Equation (5.41) is a measure of how similar this word's probability in topic k is to its probabilities in all other models).

Simply inspecting the learned topics, one might find it evident that the model conditioned on author features was able to learn much more meaningful topics. To quantify this initial intuition, we performed a simple experiment to collect unbiased human evaluations of the two models against each other: A small group of volunteers, which had not seen either of the topic models before, were shown, in sequence, pairs of the topics shown in Figures 5.8 and 5.9, and asked to choose one over the other. We will make the widely accepted, albeit arguable, assumption that the participants are reasonably representative of the wider population for this task. To avoid bias, the topics were shown in randomized positions on the screen; i.e. the unconditioned and conditioned models' topics were shown in first and second place on the screen randomly, with equal probability. The order of topics was kept as shown in the figures (since the topics are already generated randomly by the inference algorithm itself, there is no need to further randomize their order). The precise task description given to the participants was

> The screen will now repeatedly show two collections of words, called "topics". These are supposed to represent groupings of semantically similar words. You have to decide which of the two topics is a better overall match of words together to a topic. There are 40 such topic pairs overall.
> Please note: The pairs of topics DO NOT necessarily describe the same topic. In fact, they have been randomly matched. You should not attempt to find similarities between the topics. Just decide which of the word collections makes for a neater topic, in your opinion.

The participants performed the experiments alone without supervision, so any psychological influence of the author on the participants' decisions can be ruled out.
The result of each experiments is a set of 40 binary variables $D = \{\omega_i\}$, indicating whether the topic inferred by the model conditioned on author features "won" over the topic inferred by the model without author features. In such a setup, a simple Bayesian hypothesis test is possible, using a Beta prior

$$p(\xi|\mathcal{H}) = \mathcal{B}(\xi; a, b) \equiv \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\xi^{a-1}(1-\xi)^{b-1} \tag{5.43}$$

for the probability ξ of choosing the author-based model over the no-author model. The null hypothesis is $H_0 = (\xi < 0.5)$, i.e. "The no-author model has a larger chance of being chosen". We choose a uniform prior by setting $a = 1$, $b = 1$ (this is a conservative choice. A (0,0) prior would lead to stronger p-values in the following analysis). The Beta distribution is the exponential family conjugate to the Bernoulli probability distribution. The sufficient statistics are the number n_D of positive answers (i.e. $n_D \equiv |\{\omega_i \in D \,|\, \omega_i = 1\}|$) and the total number $N = 40$ of comparisons. We can thus perform a Bayesian hypothesis test[1] by evaluating the posterior probability for $\xi < 0.5$ (the *evidence* for the null hypothesis), which is given by the normalized incomplete Beta function[2]:

$$p(H_0 \,|\, D, \mathcal{H}) = \frac{B_{0.5}(a, b)}{B(a, b)} \tag{5.44}$$

Table 5.1 shows the resulting evidences and n_d for all the participants in the experiment. As the table shows, all participants' responses lead to evidences well below the 1% level, confidently rejecting the null hypothesis. Since there is only one alternative hypothesis, we can thus conclude that each of the participants preferred the author-based model over its alternative. This hard categorization allows an easy hypothesis test on the next higher conceptual level: Observing 4 partic-

[1] From a Frequentist viewpoint, the evidence mass calculated in this hypothesis test may be intepreted as a one-tailed "p-value".

[2] $B_x(a, b) \equiv \int_0^x t^{a-1}(1-t)^{b-1}\,\mathrm{d}t$, and $B(a, b) \equiv B_1(a, b)$.

participant	n_D (out of 40)	evidence for $\mathcal{H}_0$
#1	31	$2.2 \cdot 10^{-3}$
#2	33	$1.3 \cdot 10^{-5}$
#3	36	$5.1 \cdot 10^{-8}$
#4	32	$5.6 \cdot 10^{-5}$

Table 5.1: Evidence ("p-values") for the null hypothesis of the no-author model providing equally good or better topics than the model conditioned on author features, for the individual (anonymized) participants of the experiment.

ipants preferring the author model and no participant preferring the alternative, the posterior belief that the probability of *the frequency of participants preferring the no-author model* is larger than 0.5 for the entire population, using again a uniform Beta prior, is 3%. Based on this evidence, one might be convinced to reject this hypothesis as well: We expect, with high (97%) confidence, that the majority of people will prefer the model conditioned on author features over the model without author features.

This shows that metadata can provide valuable additional information for topic modeling. Of course, the author feature is of particular importance in corpora with short documents and several documents per author, such as Twitter. Being able to describe authors, rather than documents, in terms of topic mixtures is a useful feature in itself. Figures 5.7 and 5.10 show the predictive topic distributions for four individual authors.

5.3.5 Single Pass versus Iterative Inference

As pointed out in the preceding sections, the standard implementation of variational inference in topic models involves repeated iterations over the bound on a fixed data set. The single pass setup introduced in Section 5.2.4 has a more ad-hoc character than this iterative optimization scheme. Nevertheless, there are two distinct issues which might make single-pass inference more attractive than the iterative scheme in certain situations:

Data Abundance: Iterative inference has higher computational cost, and also a considerably higher memory cost, due to the need to store the messages sent in the regression module and the pseudo-counts determining the variational bound for each document. Where essentially arbitrary amounts of training data are available, limited computation time might be better spent considering *more* data than *re-considering* approximate beliefs on old data in an iterative optimization scheme.

Topic Drift: Infinite data streams may change their topic structure over time. If the drift is slow, a simple heuristic dynamic model like the one described

in Section 5.2.4 may then be able to model such a drift, in contrast to the inherently time-free description of the standard implementation (there has been work on dynamical topic models [Blei and Lafferty, 2006, Wang and McCallum, 2006], but these solutions again require multiple passes over the dataset and are thus not suitable for data streams). This issue is somewhat linked to the first issue — if the corpus has inhomogenous structure, using streaming inference may or may not lead to better performance, depending on what exactly the task of the topic model should be (see more details below).

We study both these issues in two experiments on a second data set, consisting of articles from the English Wikipedia. While this dataset has finite size, it is very large, and comes from the provider in form of a database[3] with some accidental semantic structure (of course, since the corpus is finite, it would be easy to randomize, but this structure is an interesting real-world example of the effects expected in streaming datasets as well; it is thus retained on purpose). The parameters of the model were chosen based on an elaborate grid search for maximal evidence, performed by another researcher for standard latent Dirichlet allocation on this dataset[4] to be $K = 100$, $\beta = 0.1$, and regression priors implicitly defined by a desired Dirichlet prior on the document topic distributions with parameter $\alpha = 0.25$. All models used as features of the document the identity of the last author having edited a given article (note that multiple authorship would be easy to represent for the linear regression, but the database used does not provide this information). Robot authors (scripts, which are ubiquitous on Wikipedia) and authors with very few overall edits were lumped together under an "anonymous author" feature.
Figure 5.11 shows an experimental setup focusing on the first issue of data abundance. We compare an iterative algorithm running for 200 *iterations* (i.e. to convergence) on a fixed training set of 1000 articles, to streaming inference algorithms observing 200 *times* as much data, but inferring every document's topics only once, which leads to comparable computation cost. As described in Section 5.2.4, the streaming algorithm has two parameters controlling the discount of older data over new data. Because the author features are relatively sparse, it became clear during preliminary experiments that a discount on the regression weights is not necessary. Figure 5.11 thus shows a family of performances for varying values of only the topic distribution discount parameter ξ. The chosen performance measure is average predictive probability of unseen words on a test set of 1000 documents (both algorithms were allowed to observe the first half of all these documents, then predicted the second half of the words of these documents based on the inferred topics). Some interesting aspects to note are:

[3] see `http://en.wikipedia.org/wiki/Wikipedia:Database_download`
[4] Carl Scheffler, personal communication.

- ▷ For a good setting of the discount parameter, observing more data leads to better performance than re-visiting and optimizing on old data.

- ▷ The iterative algorithm actually goes through a phase of decreasing performance while optimizing its hyperparameters.

- ▷ As indicated above, inhomogenuous structure of the corpus can lead to varying performance for the single-pass algorithms. A good example of this behaviour are the spikes visible between documents number $50,000$ and $80,000$ in Figure 5.11 (and again around document $140,000$). These fluctuations are caused by a range of atypical documents at these locations in the database (closer inspections reveals these to be clusters of documents about small towns in the United States).

Figure 5.12 shows a second comparison between an iterative and a single-pass algorithm, focusing more on the second issue of adaptation to fluctuations in the dataset. The only single pass algorithm used here was chosen, based on the results of the previous experiment, with $\xi = 0.95$. In this setup, both algorithms first get to observe a small training set of 1000 documents (left part of Figure 5.12). The iterative algorithm (top half of Figure 5.12) runs to convergence on this set. It then cedes updating its internal model parameters while passing once through the remaining $190,000$ documents, for each document inferring topics from the first half of its words and then predicting the second half. The average log probability assigned to ground truth by this prediction is used as a performance measure. In contrast, the algorithm shown in the lower half of Figure 5.12 only iterates 10 times over the training set, then continues to update its model based on the first half of all the remaining documents observed during evaluation — this again leads to comparable computational cost (however, due to the need to store messages, the batch phase has a considerably higher memory cost. In fact, it cannot readily be scaled to training on the entire dataset). Two points to note are

- ▷ Again, overall predictive performance (straight lines in Figure 5.12 on the right, reproduced for easier comparison on the left) is (slightly) better for the streaming inference algorithm, which spends less time on each document, but instead incorporates more data.

- ▷ The semantic changes of the documents around index 50k to 80k, which looked disadvantageous in the previous experiment, now actually run in favour of the single-pass algorithm, which can adapt well to the new structure.

5.4 Conclusion

We constructed an efficient approximate inference algorithm for topic models conditioned on arbitrary features of the documents, of computational cost linear in the number and length of documents, the number of topics and the number of active features of each document, and independent of the size of the vocabulary. Reaching this low complexity required an array of different methods and approximations. The computational cost's (though not the memory cost's) dependence on vocabulary size was removed by analytical integration. A lower bound on the mixture model was found by means of a variational approximation and a fast Gaussian approximation. Regression on the feature weights was realized using the sum-product algorithm on a factorized approximation of the exact beliefs. Finally, a link between the regression in $\mathbb{R}^K$ and the mixture modeling on the $[0,1]^K$ simplex was constructed from a Laplace approximation to the Dirichlet distribution in the softmax basis.

This concoction of approximations might seem unsatisfying, and of course a more exact solution would be more appealing. But approximations are the price of fast inference. Other authors have previously demonstrated the abilities of topic models using more exact, more costly approximate inference schemes. Our experiments show that document features can provide crucial information about the content of a document, especially if the document itself is short. If topic models are to become useful for the semantic description of documents at large scale, such as those found on the web, then fast approximate algorithms will be necessary. The advantage of approximate Bayesian approaches like those presented here is that the exact objective is clearly spelled out in the form of the generative model, providing a guiding light for the construction of the algorithm, and a means to assess the quality of the approximations used. The result of our analysis is a fast algorithm retaining approximate probabilistic beliefs, rather than point estimates. Since the algorithm is linear in the number of documents, features and topics, and can be run in a single pass over the data, it constitutes the first practical topic model inference algorithm for very large web-scale corpora.

1. via, http, life, another, end, bit, august, people, wednesday, ask, ...
2. post, new, blog, page, now, money, start, around, time, better, ...
3. really, life, want, know, good, nba, draft, mean, moment, cream, ...
4. news, change, says, climate, russian, afghanistan, president, #climate, leaders, public, ...
5. iphone, twitter, like, use, design, great, know, things, think, miss, ...
6. please, art, latest, live, boy, #tcot, word, blog, museum, security, ...
7. run, care, update, says, hit, local, year, second, gets, calls, ...
8. world, cup, #worldcup, england, japan, match, goal, final, football, team, ...
9. dog, via, world, press, toronto, week, dogs, daily, story, die, ...
10. love, follow, lol, like, back, movie, thank, girl, justin, happy, ...
11. post, via, science, study, research, work, yesterday, important, death, training, ...
12. today, film, photos, birthday, day, check, hours, show, gallery, amazing, ...
13. right, good, said, #news, may, lose, finished, keep, believe, now, ...
14. june, market, recovery, washington, rise, low, set, trade, may, space, ...
15. food, summer, dinner, sale, eat, beautiful, sweet, recipes, healthy, lunch, ...
16. full, new, wine, best, show, photography, red, mind, winner, interesting, ...
17. thx, article, brain, storm, coast, breaking, area, mexico, near, car, ...
18. think, know, like, yeah, back, going, even, guys, good, might, ...
19. call, office, give, youtube, long, see, john, jackson, brown, words, ...
20. new, music, video, plus, week, dead, coming, challenge, july, mark, ...
21. good, tonight, fun, done, going, trying, everyone, pretty, thanks, much, ...
22. game, season, win, tonight, night, last, tomorrow, congrats, games, play, ...
23. record, news, talks, west, house, two, says, former, camp, report, ...
24. book, review, read, years, reading, books, budget, ice, year, david ...,
25. oil, obama, gulf, vote, spill, #oilspill, mcchrystal, power, energy, sen, ...
26. bill, court, #fb, kagan, financial, senate, bank, jobs, america, street, ...
27. time, cont, told, days, today, back, another, daily, now, right, ...
28. love, god, always, like, never, someone, see, people, heart, good, ...
29. com, video, twitter, www, blog, live, check, #ff, que, help, ...
30. hey, great, good, day, ever, favorite, sure, black, way, thanks, ...
31. home, new, garden, tips, water, gardening, #sports, blog, summer, city, ...
32. following, three, stop, chance, james, photo, pick, paul, top, man, ...
33. live, day, rugby, set, race, week, now, win, today, join, ...
34. new, times, women, post, photo, cool, ipad, sex, iphone, fun, ...
35. now, need, travel, great, website, project, thanks, store, phone, #jobs, ...
36. social, media, facebook, york, cloud, job, based, service, san, million, ...
37. deal, one, home, #etsy, time, enough, face, #wwes, league, actually, ...
38. help, recipe, easy, buy, part, success, #food, new, pls, save, ...
39. google, source, business, health, #economy, company, stock, marketing, sales, china, ...
40. london, open, #olympics, star, meet, olympic, visit, days, pic, sports, ...

Figure 5.8: Topics learned by the conditioned topic model from the cleaned twitter dataset. See text for details. Some of the learned topics reflect news items of the summer of 2010, such as the G20 summit in Toronto (topic 4), the football world cup (topic 8), the Great Recession (topics 14 and 26), the BP oil spill in the Gulf of Mexico (topic 25), the sacking of the US supreme commander in Afghanistan, Gen. McChrystal (also topic 25), and Congress's confirmation of Elena Kagan to the US supreme court (topic 26).

1. japan, took, rich, football, boy, sign, california, cover, mumbai, level, ...
2. official, canada, power, recipe, thought, usa, fall, winning, williams, place, ...
3. chris, web, serious, money, etc, street, dogs, reality, thursday, hall, ...
4. become, org, child, system, fair, human, continues, markets, weekly, download, ...
5. behind, sun, spain, special, gardening, amp, #nascar, despite, stuff, saw, ...
6. baseball, alex, #scotus, huge, spent, bloggers, english, early, fan, private, ...
7. solar, india, security, inside, far, stories, rescue, info, meeting, whether, ...
8. tony, petraeus, nothing, photography, jersey, hello, happened, improve, olympics, wedding,
9. mcchrystal, federal, friend, tip, joe, ipo, quote, analysis, buy, access, ...
10. enough, tune, finds, double, king, spot, strategy, sunday, mexico, try, ...
11. four, program, los, project, marketing, celebrate, #ff, calls, caught, room, ...
12. mail, playing, plans, fresh, hill, smart, collection, believe, bet, opening, ...
13. manager, answer, followers, goes, ago, consumer, true, anti, guardian, miles, ...
14. #economy, france, return, poor, blogs, martin, shooting, breakfast, jackson, hair, ...
15. air, baby, nearly, bring, winner, government, model, running, putting, pres, ...
16. research, months, drive, storm, session, fit, guide, owner, girls, forum, ...
17. beat, dead, talks, gift, wanted, without, #cloud, lunch, finally, actually, ...
18. reform, code, omg, agree, ocean, star, diabetes, problem, push, row, ...
19. fund, less, okay, #food, light, hear, vacation, paraguay, came, simple, ...
20. building, competition, george, #tedxoilspill, five, sox, #olympics, proud, thru, deficit, ...
21. america, heat, dont, looks, hard, seen, films, ball, town, date, ...
22. hours, sometimes, class, scientists, store, face, fantastic, european, crazy, finish, ...
23. weeks, image, sold, chance, expected, round, reading, interview, center, lower, ...
24. girl, details, hands, pretty, ppl, yay, aug, cheese, yahoo, wake, ...
25. #climate, posted, police, clean, australia, sense, pre, africa, pizza, comes, ...
26. using, training, site, haha, general, shows, bag, exclusive, forget, conference, ...
27. create, economic, #oilspill, yeah, song, price, tom, speak, often, challenge, ...
28. rate, hulu, following, writing, soccer, cuts, west, mobile, #cnn, capital, ...
29. tweets, enjoy, guy, modern, changes, publishing, signed, dreams, sea, record, ...
30. piece, couple, risk, breaking, cancer, search, including, play, link, justice, ...
31. gallery, congress, fifa, hearings, don, spies, features, wrote, extra, drug, ...
32. mark, though, york, wants, finished, won, head, dream, heart, budget, ...
33. lots, view, son, harry, given, important, russia, track, #tls, teams, ...
34. apps, added, heard, study, sweet, toy, instead, forget, across, comments, ...
35. hate, quick, earth, pro, bbc, #jobs, french, award, coast, experience, ...
36. question, #traveltuesday, area, value, wish, bed, moment, germany, dating, brand, ...
37. men, welcome, act, understand, camp, cute, wsj, former, eat, single, ...
38. album, based, cnn, advice, chinese, shot, yesterday, dies, super, ones, ...
39. links, link, photography, article, maybe, podcast, pet, rock, country, brian, ...
40. campaign, style, books, update, statement, wife, #oilspill, different, edition, tax, ...

Figure 5.9: Topics learned by a conditional topic model without access to the author features, from the same dataset as in Figure 5.8. Note the lower degree of topical separation, indicating the usefulness of author information. This model only learns weights for the bias features. Results from a fully unconditioned model with fixed α (not shown) look even less structured.

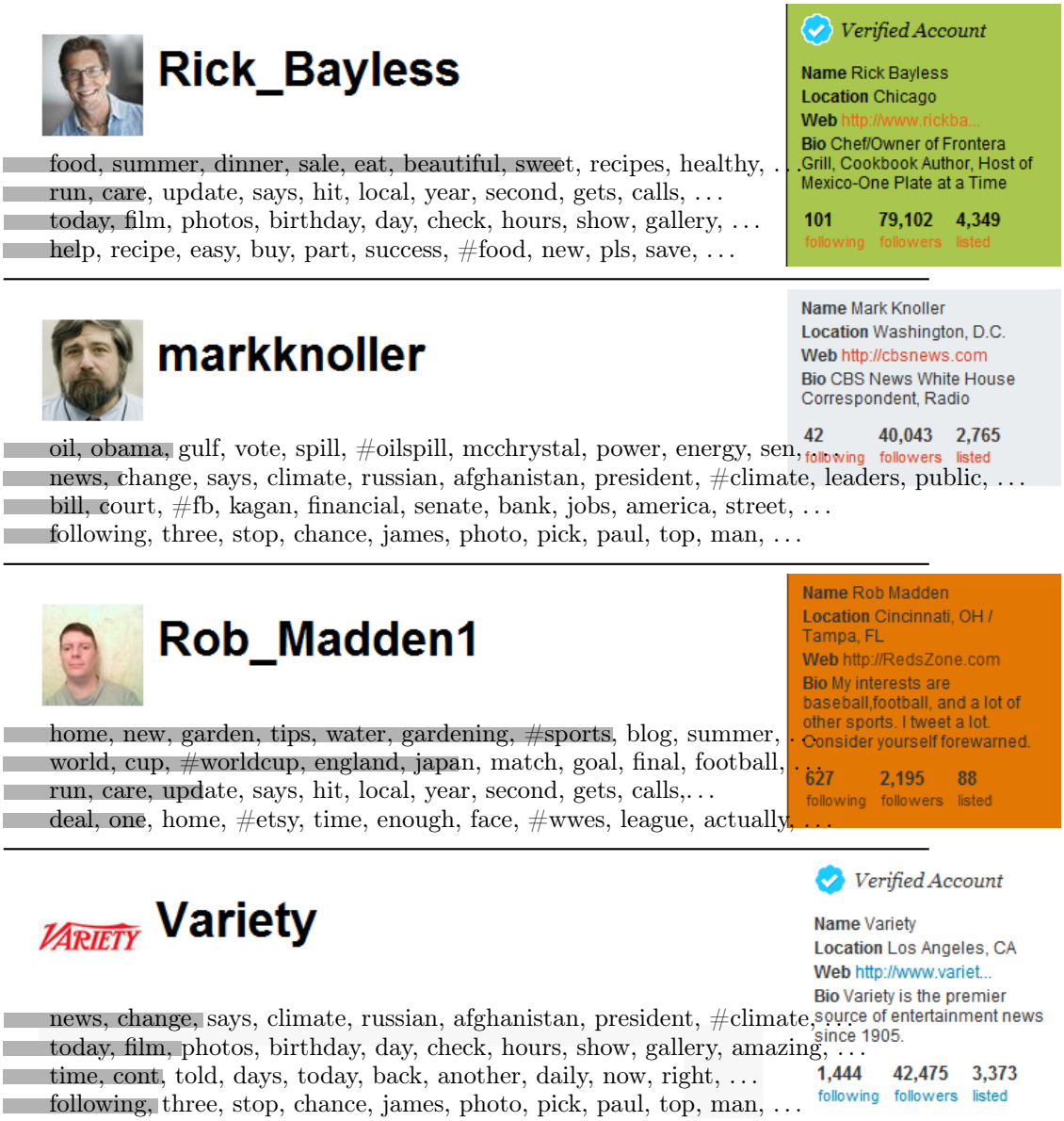


Figure 5.10: Four prolific authors in the dataset, with the top three topics for each author as identified by the algorithm. The bars in the background are indications of the relative weight of the corresponding topics (arbitrary scale, precise numbers can be found in Figure 5.7).

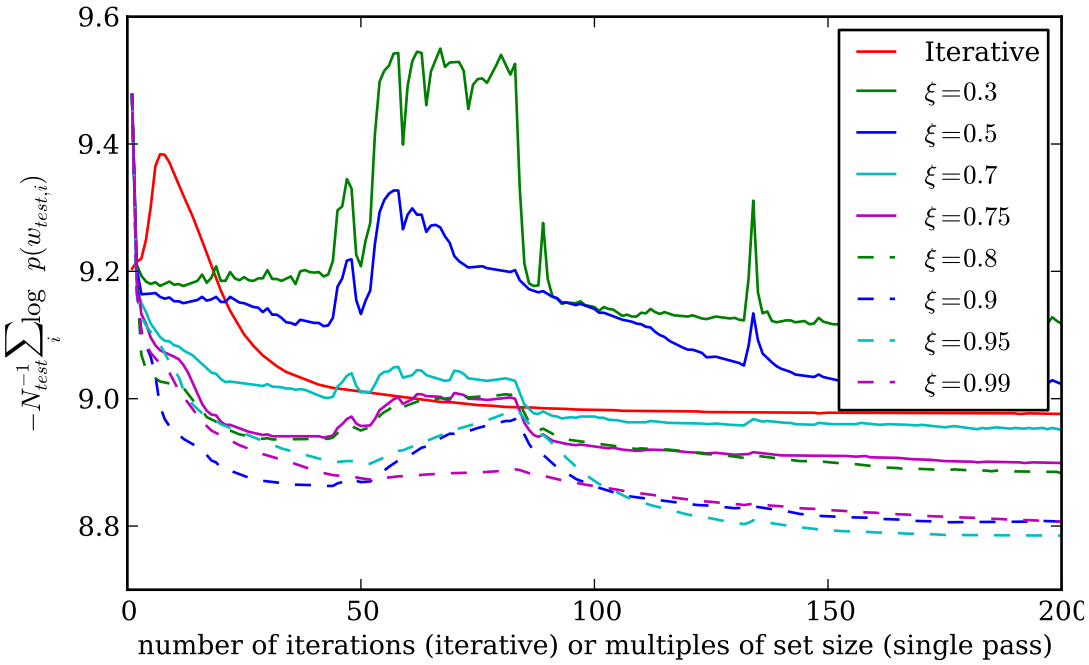


Figure 5.11: Comparison of convergence, as a function of computation time spent, between iterative inference and a series of online inference algorithms with varying settings of the discount parameter ξ. The convergence measure is negative average predictive log probability on a fixed test set (i.e. lower values are better). Note the spikes around 50k to 80k documents for the streaming inference algorithms, where the corpus has a quantitative change in structure.

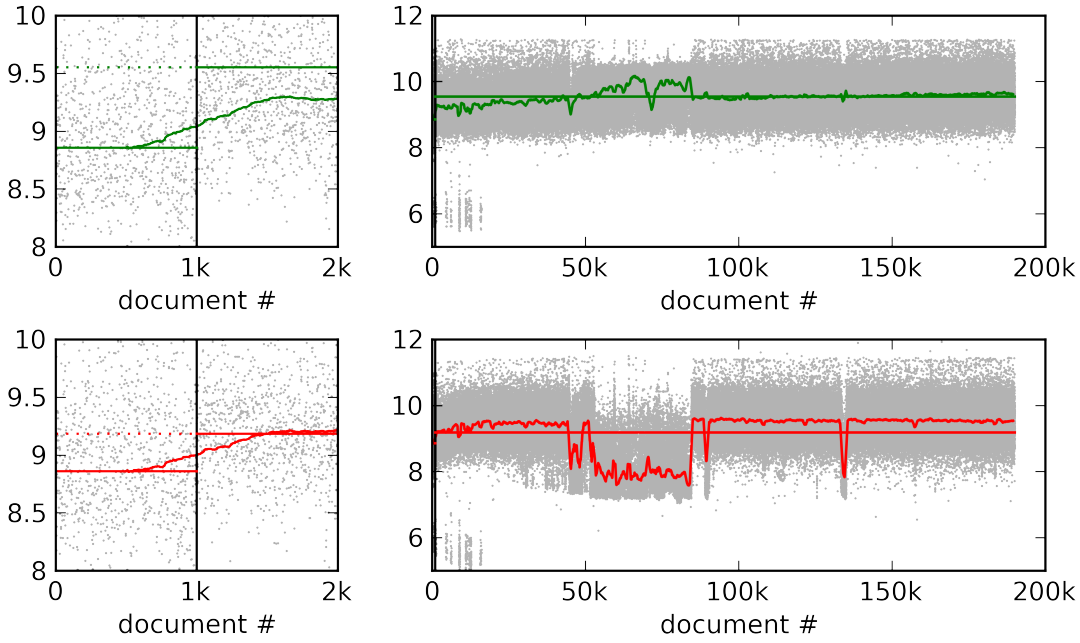


Figure 5.12: Predictive power (negative average log probability assigned to unobserved words, smaller numbers are better) for a batch (top) and an online (bottom) algorithm. Left: performance on the first 1000 articles (on which the batch algorithm is trained exclusively). Right: performance on remaining dataset. Straight lines: averages over these two entire data regions (right averages repeated in left plots for easier comparison). Data points in grey, overlaid with running averages over a 1000 document window.

Chapter 6

Conclusions

This thesis offered contributions on three conceptual levels.

Applications Chapters 3, 4 and 5 each presented independent solutions to applied problems from different research communities.

- ▷ Chapter 3 proposed a Bayesian tree search algorithm. While the metaphor of games was used there, it is easy to imagine that variations of this algorithm might be applicable to other tree search problems, for reasons similar to the motivation for the game models (i.e. unstructured tree search problems are in some sense easy, because large parts of the tree are irrelevant to the solution. Hence, hard tree search problems may benefit from using a generative model). Tree search is a widely used paradigm in computer science, so the results presented here might be of broader use.

- ▷ Chapter 4 presented an extended "noise" model for psychometric questionnaires. Psychometrics is an influential discipline, with applications from the management of human resources to criminal law, so any improvements to the mathematical methods used in this field have high potential for societal impact.

- ▷ Chapter 5 offered a computationally lightweight model for the topics of documents with metadata. The results from this chapter have already found real-world, commercial use within the Microsoft Corporation, and are the subject of a patent application submitted by Microsoft Research Ltd. on behalf of the author of this thesis.

Algorithms Some of the algorithmic constructs used in this thesis, such as the methods presented in the following Appendices, form self-contained solutions for specific approximate inference challenges, and may be re-usable in other settings than those presented here. For example, the "max-factor" detailed in Appendix A

is applicable to probabilistic shortest path problems, and the "Laplace bridge" of Appendix C addresses an issue common in Bayesian logistic regression.

The Approximate Inference Paradigm Despite the ostensible diversity of the applications addressed in chapters 3–5, the solutions presented in this thesis were all derived following the same three conceptual steps:

1. **Define an explicit generative probabilistic model for the data at hand.** Having done so, we immediately know, in principle, that the correct answer to any inference question involving variables of this model: It is given by Bayes' theorem. Since Bayesian inference is known to be both optimal and isomorphic to any other sensible framework of reasoning, this approach avoids having to invent a new ad-hoc inference method for every new problem. The probabilistic framework also forces modeling assumptions to be made explicit, which facilitates analysis and comparison to other models.

2. **Write down the graphical model corresponding to the algebraic probabilistic model.** The directed graph clarifies conditional independence and makes the generative model easer to understand. Rewriting it as a factor graph elucidates analytical as well as computational bottlenecks.

3. **Find specific approximations** for these bottlenecks, and evaluate them through unit tests. The graphical model framework often allows such separation of larger problems into conceptual sub-problems. This also means that good solutions, once found, can be re-used in other problems. Notice that this phase benefits crucially from both previous steps. Ad-hoc approximations may not work well and tend not to generalise well. But approximations addressing aspects of the algebraic structure of probability distributions, as exposed by graphical models, can be meaningfully tested and re-used.

The fact that this general approach works well for applications as wide-ranging as tree search, psychometrics and semantic language modelling indicates that approximate probabilistic inference, particularly using graphical models, is a powerful tool for the analytic scientist. It provides a language of uncertainty that removes problem-specific technicalities to reveal the mathematical structure of the underlying inference task. The solutions presented in this thesis are far from a complete compendium of these methods, but they showcase several families of approximate inference algorithms as part of a framework. Further extension of this framework may well lead to a general paradigm for tractable inference. All human behaviour is learning, predicting, and acting based on these predictions. Probability theory is the mathematical abstraction of these three abilities. Approximate inference is one approach making them tractable.

Appendix A

The Maximum of Correlated Gaussian Variables

The derivations presented in this chapter were published as the technical report [Hennig, 2009] *Expectation Propagation on the Maximum of Correlated Normal Variables*, P. Hennig, arXiv [stat.ML] 0910.0115, October 2009

Abstract

This appendix derives the first two moments of the distribution of the maximum of a pair of correlated Gaussian variables. In a second step, an extension to the maximum of finite sets is developed through an iterative approximation.

A.1 Introduction

The tree search problem introduced in Chapter 3 is not the only application requiring a probabilistic belief over the maximum of a set of variables. Other problems in this class include shortest path problems [Burton and Toint, 1992], Reinforcement Learning [Dearden et al., 1998], and scientific inference in Seismology [Neumann-Denzau and Behrens, 1984], to name but a few. Often, there is a corresponding inverse optimization problem [Ahuja and Orlin, 2001, Heuberger, 2004], where the optimal solution is known with some uncertainty and the question is about the quantities generating this optimum. Most contemporary algorithms for this case aim to provide a point estimate (typically the least-squares solution), but have trouble offering an error estimate on this estimate as well.

This chapter derives (Section A.2) mean and variance of the posterior of the maximum of two correlated Gaussian variables (for forward optimization problems), and the mean and variance on the posterior of the Gaussian variables generating the maximum (for inverse optimization problems). It will be shown how these

results can be used to build a heuristic approximation to the maximum of a finite set of normal variables (Section A.3). This provides the necessary results for Expectation Propagation with a black box "max"-factor on factor graphs. Because maximum and minimum are linked by $\max(\{x_i\}) = -\min(\{-x_i\})$, this also allows inference on the minimum where necessary. Limitations of this approximation are examined in Section A.4.

The moments of the normalized likelihood function of the maximum of two normal variables have previously been derived by Clark [1961]. To my best knowledge, this is the first publication deriving the full posterior, and the first to report the posterior for the inverse problem (see also Section A.2.4).

A.2 The Maximum of Two Gaussian Variables

A.2.1 Notation

We consider two normally distributed variables x_1 and x_2, forming the vector $\boldsymbol{x}$. Let there be some prior information $\mathcal{I}_\mathfrak{g}$ giving rise to the belief

$$\begin{aligned} p(x_1, x_2 \,|\, \mathcal{I}_\mathfrak{g}) &= \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_\mathfrak{g}, \boldsymbol{\Sigma}_\mathfrak{g}) \\ &= \frac{1}{2\pi\sigma_{\mathfrak{g}1}\sigma_{\mathfrak{g}2}(1-\varrho^2)^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_\mathfrak{g})^\mathsf{T}\boldsymbol{\Sigma}_\mathfrak{g}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_\mathfrak{g})\right) \end{aligned} \tag{A.1}$$

over their values. Here we have defined a mean vector $\boldsymbol{\mu}_\mathfrak{g} = (\mu_{\mathfrak{g}1}, \mu_{\mathfrak{g}2})^\mathsf{T}$ and a covariance matrix $\boldsymbol{\Sigma}_\mathfrak{g}$. The latter has the form

$$\begin{aligned} \boldsymbol{\Sigma}_\mathfrak{g} &= \begin{pmatrix} \sigma_{\mathfrak{g}1}^2 & \varrho\sigma_{\mathfrak{g}1}\sigma_{\mathfrak{g}2} \\ \varrho\sigma_{\mathfrak{g}1}\sigma_{\mathfrak{g}2} & \sigma_{\mathfrak{g}2}^2 \end{pmatrix} \qquad \text{and thus} \\ \boldsymbol{\Sigma}_\mathfrak{g}^{-1} &= \frac{1}{\sigma_{\mathfrak{g}1}^2\sigma_{\mathfrak{g}2}^2(1-\varrho^2)} \begin{pmatrix} \sigma_{\mathfrak{g}2}^2 & -\varrho\sigma_{\mathfrak{g}1}\sigma_{\mathfrak{g}2} \\ -\varrho\sigma_{\mathfrak{g}1}\sigma_{\mathfrak{g}2} & \sigma_{\mathfrak{g}1}^2 \end{pmatrix} \end{aligned} \tag{A.2}$$

with the *linear coefficient of correlation*

$$\varrho = \frac{\operatorname{cov}(x_1, x_2)}{\sigma_{\mathfrak{g}1}\sigma_{\mathfrak{g}2}} \tag{A.3}$$

(for notational convenience, the index $\mathfrak{g}$ is dropped from ϱ because there will be little risk of confusion). We further introduce the variable $m = \max(x_1, x_2)$, and assume that there is some outside prior information $\mathcal{I}_\mathfrak{m}$ on the value of m as well:

$$p(m \,|\, \mathcal{I}_\mathfrak{m}) = \mathcal{N}(m; \mu_\mathfrak{m}, \sigma_\mathfrak{m}^2) \tag{A.4}$$

The inference problems to be solved are

▷ The belief over m given both $\mathcal{I}_\mathfrak{m}$ and $\mathcal{I}_\mathfrak{g}$ (jointly called $\mathcal{I}_\mathfrak{c}$). This is not truly a "posterior" in the strict technical sense, but the normalised product of two different factors (independent sets of information):

$$\begin{aligned} p(m \,|\, \mathcal{I}_\mathfrak{c}) &= \frac{p(m \,|\, \mathcal{I}_\mathfrak{m}) \int p(\boldsymbol{x} \,|\, m) p(\boldsymbol{x} \,|\, \mathcal{I}_\mathfrak{g}) \,\mathrm{d}\boldsymbol{x}}{\int \left[p(m \,|\, \mathcal{I}_\mathfrak{m}) \int p(\boldsymbol{x} \,|\, m) p(\boldsymbol{x} \,|\, \mathcal{I}_\mathfrak{g}) \,\mathrm{d}\boldsymbol{x} \right] \mathrm{d}m} \\ &= Z^{-1} p(m \,|\, \mathcal{I}_\mathfrak{m}) \int p(\boldsymbol{x} \,|\, m) p(\boldsymbol{x} \,|\, \mathcal{I}_\mathfrak{g}) \,\mathrm{d}\boldsymbol{x} \end{aligned} \tag{A.5}$$

with the normalization constant $Z = \iint p(\boldsymbol{x}, m \,|\, \mathcal{I}_\mathfrak{c}) \,\mathrm{d}\boldsymbol{x} \,\mathrm{d}m$. This problem will be called the "forward" problem here.

▷ The posterior over $\boldsymbol{x}$ given $\mathcal{I}_\mathfrak{c}$,

$$\begin{aligned} p(\boldsymbol{x} \,|\, \mathcal{I}_\mathfrak{c}) &= \frac{p(\boldsymbol{x} \,|\, \mathcal{I}_\mathfrak{g}) \int p(m \,|\, \boldsymbol{x}) p(m \,|\, \mathcal{I}_\mathfrak{m}) \,\mathrm{d}m}{\int \left[p(\boldsymbol{x} \,|\, \mathcal{I}_\mathfrak{g}) \int p(\boldsymbol{x} \,|\, m) p(m \,|\, \mathcal{I}_\mathfrak{m}) \,\mathrm{d}m \right] \mathrm{d}\boldsymbol{x}} \\ &= Z^{-1} p(\boldsymbol{x} \,|\, \mathcal{I}_\mathfrak{g}) \int p(m \,|\, \boldsymbol{x}) p(m \,|\, \mathcal{I}_\mathfrak{m}) \,\mathrm{d}m. \end{aligned} \tag{A.6}$$

This problem will be called the "inverse" problem.

Throughout the derivations, the notation

$$\begin{aligned} \mathcal{N}(x; \mu, \sigma^2) &\equiv \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2 \right] \\ \phi(x) &\equiv \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2} \right) \\ \Phi(x) &\equiv \int_{-\infty}^{x} \phi(t) \,\mathrm{d}t = \frac{1}{2} \left[1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}} \right) \right] \end{aligned} \tag{A.7}$$

will be used to denote the general and standard normal probability density functions (PDF) and the standard normal cumulative distribution function (CDF), respectively.

A.2.2 Some Integrals

The following derivations will repeatedly feature certain integrals. The first two incomplete moments of the standard Gaussian are

$$\begin{aligned} \int_{-\infty}^{y} t\phi(t) \,\mathrm{d}t &= -\phi(y) \\ \int_{-\infty}^{y} t^2 \phi(t) \,\mathrm{d}t &= \Phi(y) - y\phi(y). \end{aligned} \tag{A.8}$$

This is obvious directly from differentiation. A simple substitution gives

$$\int_{-\infty}^{y} t\mathcal{N}(t;\alpha,\beta^2)\,\mathrm{d}t = \alpha\Phi\left(\frac{y-\alpha}{\beta}\right) - \beta\phi\left(\frac{y-\alpha}{\beta}\right) \tag{A.9}$$

$$\int_{-\infty}^{y} t^2\mathcal{N}(t;\alpha,\beta^2)\,\mathrm{d}t = (\alpha^2+\beta^2)\Phi\left(\frac{y-\alpha}{\beta}\right) - (\alpha+y)\beta\phi\left(\frac{y-\alpha}{\beta}\right) \tag{A.10}$$

Further, we will use

Lemma

$$\begin{aligned}
\int_{-\infty}^{\infty}\Phi\left(\frac{x-a}{b}\right)\mathcal{N}(x;\alpha,\beta^2)\,\mathrm{d}x &= \Phi(z)\\
\int_{-\infty}^{\infty}x\Phi\left(\frac{x-a}{b}\right)\mathcal{N}(x;\alpha,\beta^2)\,\mathrm{d}x &= \alpha\Phi(z) + \frac{\beta^2}{b\sqrt{1+\beta^2/b^2}}\phi(z)\\
\int_{-\infty}^{\infty}x^2\Phi\left(\frac{x-a}{b}\right)\mathcal{N}(x;\alpha,\beta^2)\,\mathrm{d}x &= (\alpha^2+\beta^2)\Phi(z) + \left[2\alpha\frac{\beta^2}{b\sqrt{1+\beta^2/b^2}} - z\frac{\beta^4}{b^2+\beta^2}\right]\phi(z)\\
&\text{where}\qquad z = \frac{\alpha-a}{b\sqrt{1+\beta^2/b^2}}
\end{aligned} \tag{A.11}$$

Proof: To prove the Lemma, we follow Rasmussen and Williams [2006, §3.9] and expand

$$\begin{aligned}
&\int_{-\infty}^{\infty}\Phi\left(\frac{x-a}{b}\right)\mathcal{N}(x;\alpha,\beta^2)\,\mathrm{d}x\\
&\qquad= \int_{-\infty}^{\infty}\int_{-\infty}^{x}\mathcal{N}(y;a,b^2)\mathcal{N}(x;\alpha,\beta^2)\,\mathrm{d}y\,\mathrm{d}x\\
&\qquad= \int_{-\infty}^{\infty}\int_{-\infty}^{x}\mathcal{N}\left[\begin{pmatrix}y\\x\end{pmatrix};\begin{pmatrix}a\\\alpha\end{pmatrix},\begin{pmatrix}b^2 & 0\\0 & \beta^2\end{pmatrix}\right]\mathrm{d}y\,\mathrm{d}x\\
&\qquad= \int_{-\infty}^{\infty}\int_{-\infty}^{\alpha-a}\mathcal{N}\left[\begin{pmatrix}w\\z\end{pmatrix};\begin{pmatrix}0\\0\end{pmatrix},\begin{pmatrix}b^2+\beta^2 & \beta^2\\\beta^2 & \beta^2\end{pmatrix}\right]\mathrm{d}w\,\mathrm{d}z
\end{aligned} \tag{A.12}$$

where we have introduced the auxiliary variables $w \equiv y - a - (x-\alpha)$ and $z \equiv x - \alpha$. Note that the integration limit is independent of z, so we can exchange the integrations. But, because Gaussians have the convenient marginalization property (see e.g. von Mises [1964], §9.3, Equations (A.11–A.13))

$$\int\mathcal{N}\left[\begin{pmatrix}X\\Y\end{pmatrix};\begin{pmatrix}\tilde{X}\\\tilde{Y}\end{pmatrix},\begin{pmatrix}A & B\\C & D\end{pmatrix}\right]\mathrm{d}X = \mathcal{N}(Y;\tilde{Y},D) \tag{A.13}$$

the now inner integral over z is trivial, and the Lemma follows directly. □

A.2.3 Analytic Forms

Forward Problem

Neither of the posterior distributions is normal itself. The forward posterior is

$$\begin{aligned} p(m|\mathcal{I}_\mathfrak{c}) &= Z^{-1} p(m\,|\,\mathcal{I}_\mathfrak{m}) \iint_{-\infty}^{\infty} p(\boldsymbol{x}\,|\,m) p(\boldsymbol{x}\,|\,\mathcal{I}_\mathfrak{g})\,\mathrm{d}\boldsymbol{x} \\ &= Z^{-1} p(m\,|\,\mathcal{I}_\mathfrak{m}) \int_{-\infty}^{\infty} \left[\int_{-\infty}^{x_1} \delta(x_1 - m) p(\boldsymbol{x}\,|\,\mathcal{I}_\mathfrak{g})\,\mathrm{d}x_2 + \int_{x_1}^{\infty} \delta(x_2 - m) p(\boldsymbol{x}\,|\,\mathcal{I}_\mathfrak{g})\,\mathrm{d}x_2 \right] \mathrm{d}x_1 \\ &= Z^{-1} \underbrace{p(m\,|\,\mathcal{I}_\mathfrak{m}) \int_{-\infty}^{\infty} \delta(x_1 - m) \int_{-\infty}^{x_1} p(\boldsymbol{x}\,|\,\mathcal{I}_\mathfrak{g})\,\mathrm{d}x_2\,\mathrm{d}x_1}_{\nu_1} + \\ &\quad Z^{-1} \underbrace{p(m\,|\,\mathcal{I}_\mathfrak{m}) \int_{-\infty}^{\infty} \delta(x_2 - m) \int_{-\infty}^{x_2} p(\boldsymbol{x}\,|\,\mathcal{I}_\mathfrak{g})\,\mathrm{d}x_1\,\mathrm{d}x_2}_{\nu_2}. \end{aligned} \tag{A.14}$$

For a motivation of the change in the integration ranges from the second to the third line in Equation (A.14), consider the sketch in Figure A.1. Since the two

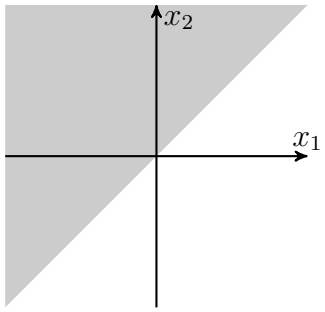


Figure A.1: Sketch of the integration range for ν_2 (shaded). The open set $(x_1, x_2) \in ((-\infty, \infty), (x_1, \infty))$ is identical to the open set $(x_1, x_2) \in ((-\infty, x_2), (-\infty, \infty))$.

summands are related to each other through the symmetry $x_1 \leftrightarrow x_2$, we consider only the first term, ν_1. To solve the integrals, note that the bi-variate Gaussian $p(\boldsymbol{x}\,|\,\mathcal{I}_\mathfrak{g})$ can be re-written as

$$\begin{aligned} p(x_1, x_2\,|\,\mathcal{I}_\mathfrak{g}) &= p(x_1\,|\,\mathcal{I}_\mathfrak{g}) p(x_2\,|\,x_1, \mathcal{I}_\mathfrak{g}) \\ &= \frac{1}{\sqrt{2\pi\sigma_{\mathfrak{g}1}^2}} \exp\left[-\frac{1}{2} \left(\frac{x_1 - \mu_{\mathfrak{g}1}}{\sigma_{\mathfrak{g}1}} \right)^2 \right] \\ &\quad \frac{1}{\sqrt{2\pi\sigma_{\mathfrak{g}2}^2(1-\varrho^2)}} \exp\left[-\frac{1}{2\sigma_{\mathfrak{g}2}^2(1-\varrho^2)} \left(x_2 - \left(\mu_{\mathfrak{g}2} + \varrho \frac{\sigma_{\mathfrak{g}2}}{\sigma_{\mathfrak{g}1}} (x_1 - \mu_{\mathfrak{g}1}) \right) \right)^2 \right]. \end{aligned} \tag{A.15}$$

So we can simplify ν_1 to

$$\begin{aligned}\nu_1 &= p(m \,|\, \mathcal{I}_\mathfrak{m})\mathcal{N}(m; \mu_{\mathfrak{g}1}, \sigma^2_{\mathfrak{g}1})\times \\ &\qquad \int_{-\infty}^{m} \frac{1}{\sqrt{2\pi\sigma^2_{\mathfrak{g}2}(1-\varrho^2)}} \exp\left[-\frac{1}{2(1-\varrho^2)}\left(\frac{x_2-\mu_{\mathfrak{g}2}}{\sigma_{\mathfrak{g}2}} - \varrho\frac{m-\mu_{\mathfrak{g}1}}{\sigma_{\mathfrak{g}1}}\right)^2\right] \mathrm{d}x_2 \\ &= p(m \,|\, \mathcal{I}_\mathfrak{m})\mathcal{N}(m; \mu_{\mathfrak{g}1}, \sigma^2_{\mathfrak{g}1})\times \\ &\qquad \int_{-\infty}^{m} \frac{1}{\sqrt{2\pi\sigma^2_{\mathfrak{g}2}(1-\varrho^2)}} \exp\left[-\frac{1}{2}\left(\frac{x_2-\mu_{\mathfrak{g}2}-\varrho\frac{\sigma_{\mathfrak{g}2}}{\sigma_{\mathfrak{g}1}}(m-\mu_{\mathfrak{g}1})}{\sigma_{\mathfrak{g}2}(1-\varrho^2)^{1/2}}\right)^2\right] \mathrm{d}x_2\end{aligned} \tag{A.16}$$

The substitution

$$t(x_2) \equiv \frac{x_2-\mu_{\mathfrak{g}2}-\varrho\frac{\sigma_{\mathfrak{g}2}}{\sigma_{\mathfrak{g}1}}(m-\mu_{\mathfrak{g}1})}{\sigma_{\mathfrak{g}2}(1-\varrho^2)^{1/2}} \quad \text{with Jacobian} \quad \frac{\mathrm{d}t}{\mathrm{d}x_2} = \frac{1}{\sigma_2(1-\varrho^2)^{1/2}} \tag{A.17}$$

allows us to solve the integral and find the posterior up to normalization

$$\begin{aligned}p(m \,|\, \mathcal{I}_\mathfrak{c}) = Z^{-1}&\mathcal{N}(\mu_\mathfrak{m}; \mu_{\mathfrak{g}1}, \sigma^2_\mathfrak{m} + \sigma^2_{\mathfrak{g}1})\mathcal{N}(m; \mu_{\mathfrak{c}1}, \sigma^2_{\mathfrak{c}1})\times \\ &\Phi\left(\frac{(\sigma_{\mathfrak{g}1}-\varrho\sigma_{\mathfrak{g}2})m - \sigma_{\mathfrak{g}1}\mu_{\mathfrak{g}2} + \varrho\sigma_{\mathfrak{g}2}\mu_{\mathfrak{g}1}}{\sigma_{\mathfrak{g}1}\sigma_{\mathfrak{g}2}(1-\varrho^2)^{1/2}}\right) + \\ Z^{-1}&\mathcal{N}(\mu_\mathfrak{m}; \mu_{\mathfrak{g}2}, \sigma^2_\mathfrak{m} + \sigma^2_{\mathfrak{g}2})\mathcal{N}(m; \mu_{\mathfrak{c}2}, \sigma^2_{\mathfrak{c}2})\times \\ &\Phi\left(\frac{(\sigma_{\mathfrak{g}2}-\varrho\sigma_{\mathfrak{g}1})m - \sigma_{\mathfrak{g}2}\mu_{\mathfrak{g}1} + \varrho\sigma_{\mathfrak{g}1}\mu_{\mathfrak{g}2}}{\sigma_{\mathfrak{g}2}\sigma_{\mathfrak{g}1}(1-\varrho^2)^{1/2}}\right),\end{aligned} \tag{A.18}$$

where we have used the abbreviations

$$\sigma^2_{\mathfrak{c}1} \equiv \frac{\sigma^2_{\mathfrak{g}1}\sigma^2_\mathfrak{m}}{\sigma^2_{\mathfrak{c}1}+\sigma^2_\mathfrak{m}} \qquad \text{and} \qquad \mu_{\mathfrak{c}1} \equiv \left(\frac{\mu_{\mathfrak{g}1}}{\sigma^2_{\mathfrak{g}1}} + \frac{\mu_\mathfrak{m}}{\sigma^2_\mathfrak{m}}\right)\sigma^2_{\mathfrak{c}1} \tag{A.19}$$

for the mean and variance of the product of two Gaussians. This is using the result, derived in Equation (2.18), that

$$\begin{aligned}&\mathcal{N}(x;a_1,b_1^2)\mathcal{N}(x;a_2,b_2^2) = \\ &\qquad \mathcal{N}(a_1;a_2,b_1^2+b_2^2)\mathcal{N}\left[x;\left(\frac{a_1}{b_1^2}+\frac{a_2}{b_2^2}\right)\left(\frac{1}{b_1^2}+\frac{1}{b_2^2}\right)^{-1}, \left(\frac{1}{b_1^2}+\frac{1}{b_2^2}\right)^{-1}\right],\end{aligned} \tag{A.20}$$

The forms of $\mu_{\mathfrak{c}2}$ and $\sigma_{\mathfrak{c}2}$ are entirely analogous. To find the normalization constant Z, we use the first identity in Equation (A.11) to get

$$Z = \mathcal{N}(\mu_\mathfrak{m}; \mu_{\mathfrak{g}1}, \sigma^2_\mathfrak{m} + \sigma^2_{\mathfrak{g}1})\Phi(k_1) + \mathcal{N}(\mu_\mathfrak{m}; \mu_{\mathfrak{g}2}, \sigma^2_\mathfrak{m} + \sigma^2_{\mathfrak{g}2})\Phi(k_2) \tag{A.21}$$

with

$$\begin{aligned} k_1 &\equiv \frac{(\sigma_{\mathfrak{g}1} - \varrho\sigma_{\mathfrak{g}2})\mu_{\mathfrak{c}1} - \sigma_{\mathfrak{g}1}\mu_{\mathfrak{g}2} + \varrho\sigma_{\mathfrak{g}2}\mu_{\mathfrak{g}1}}{\left[\sigma_{\mathfrak{g}1}^2\sigma_{\mathfrak{g}2}^2(1-\varrho^2) + (\sigma_{\mathfrak{g}1} - \varrho\sigma_{\mathfrak{g}2})^2\sigma_{\mathfrak{c}1}^2\right]^{1/2}} \quad \text{and} \\ k_2 &\equiv \frac{(\sigma_{\mathfrak{g}2} - \varrho\sigma_{\mathfrak{g}1})\mu_{\mathfrak{c}2} - \sigma_{\mathfrak{g}2}\mu_{\mathfrak{g}1} + \varrho\sigma_{\mathfrak{g}1}\mu_{\mathfrak{g}2}}{\left[\sigma_{\mathfrak{g}1}^2\sigma_{\mathfrak{g}2}^2(1-\varrho^2) + (\sigma_{\mathfrak{g}2} - \varrho\sigma_{\mathfrak{g}1})^2\sigma_{\mathfrak{c}2}^2\right]^{1/2}}. \end{aligned} \tag{A.22}$$

Inverse Problem

The conditional probability of $\boldsymbol{x}$ on m is

$$p(x_1, x_2 \,|\, m) = \theta(x_1 - x_2)\delta(x_1 - m) + \theta(x_2 - x_1)\delta(x_2 - m) \tag{A.23}$$

where $\theta(y)$ is Heaviside's step function. Therefore, the conditional of $\boldsymbol{x}$ on $\mathcal{I}_{\mathfrak{m}}$ (the likelihood of $[x_1, x_2]$) is

$$\begin{aligned} f(x_1, x_2 \,|\, \mathcal{I}_{\mathfrak{m}}) &= \int_{-\infty}^{\infty} p(m \,|\, x_1, x_2) p(m|\mathcal{I}_{\mathfrak{m}}) \,\mathrm{d}m \\ &= \theta(x_1 - x_2)\mathcal{N}(x_1; \mu_{\mathfrak{m}}, \sigma_{\mathfrak{m}}^2) + \theta(x_2 - x_1)\mathcal{N}(x_2; \mu_{\mathfrak{m}}, \sigma_{\mathfrak{m}}^2) \end{aligned} \tag{A.24}$$

which, as a likelihood, is not a proper (i.e. normalizable) distribution, but becomes normalizable after multiplication with the prior:

$$\begin{aligned} p(\boldsymbol{x} \,|\, \mathcal{I}_{\mathfrak{c}}) = Z^{-1} \underbrace{\Theta(x_1 - x_2)\mathcal{N}(x_1; \mu_{\mathfrak{m}}, \sigma_{\mathfrak{m}}^2)\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{\mathfrak{g}}, \boldsymbol{\Sigma}_{\mathfrak{g}})}_{\xi_1} + \\ Z^{-1} \underbrace{\Theta(x_2 - x_1)\mathcal{N}(x_2; \mu_{\mathfrak{m}}, \sigma_{\mathfrak{m}}^2)\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{\mathfrak{g}}, \boldsymbol{\Sigma}_{\mathfrak{g}})}_{\xi_2} \end{aligned} \tag{A.25}$$

Figure A.2 illustrates the shape of these functions by way of some concrete examples.

A.2.4 Moment Matching

The analytical forms derived in the preceding sections are clearly not members of the normal exponential family. If $\boldsymbol{x}$ has more than two elements, they also quickly take on complicated forms that are expensive to evaluate. If the application in question allows, it might thus be desirable to find Gaussian approximations to the posteriors. The next sections derive the moments of these distributions for use with the EP approximation.

Forward Problem

We will denote the mean and variance of the posterior of the max as $\mu_{m(12)}$ and $\sigma_{m(12)}^2$ for reasons that will become clear in Section A.3. The corresponding integrals

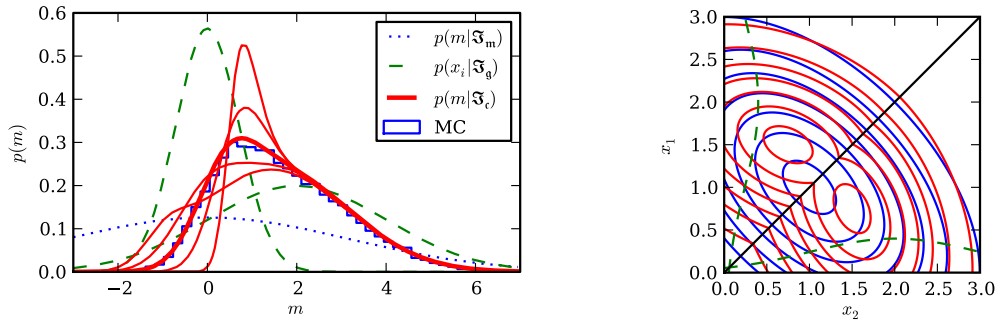


Figure A.2: Illustrative plots for the analytical form of the forward and inverse posteriors. **Left:** Inference on m. Prior distribution and marginals on $\boldsymbol{x_i}$. Posteriors for five different values of ϱ: -0.9 (most peaked), -0.5, 0.0 (thick line), 0.5 and 0.9 (broadest). As an experimental verification, a histogram of 20,000 samples from the posterior (generated by rejection sampling, with $\varrho = 0$) is shown in blue. **Right:** Inference on the inverse problem: Prior with $\mu_\mathfrak{g} = (1, 1)^\mathsf{T}$, $\sigma_{\mathfrak{g}1} = \sigma_{\mathfrak{g}2} = 1$ and $\varrho = -0.5$. Data on m with $\mu_\mathfrak{m} = 1$, $\sigma_\mathfrak{m} = 1$ gives the posterior in red. Note the bimodality arising in this particular case.

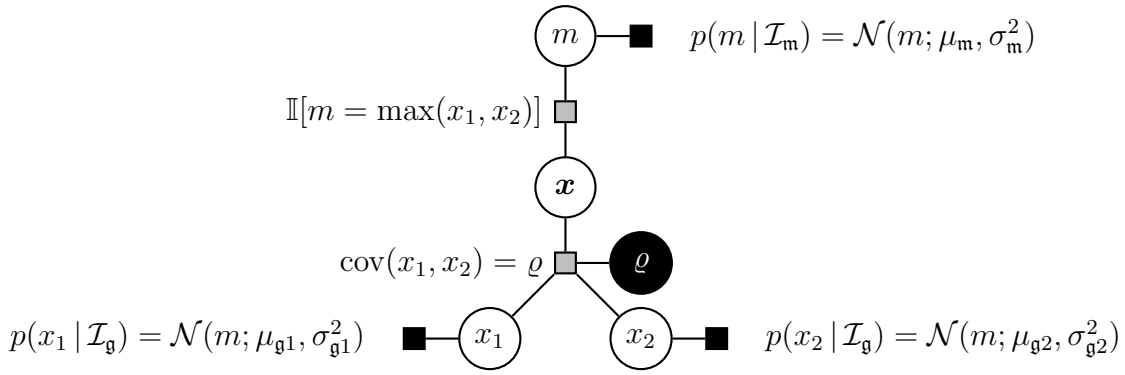


Figure A.3: Minimal factor graph using the max factor.

to solve are

$$\begin{aligned} \langle m \rangle \equiv \mu_{m(12)} &= \int_{-\infty}^{\infty} m p(m \mid \mathcal{I}_\mathfrak{c}) \, \mathrm{d}m = Z^{-1} \int m(\nu_1 + \nu_2) \, \mathrm{d}m \\ \langle m^2 \rangle - \langle m \rangle^2 \equiv \sigma^2_{m(12)} &= \int_{-\infty}^{\infty} m^2 p(m \mid \mathcal{I}_\mathfrak{c}) \, \mathrm{d}m - \mu_{m(12)} \end{aligned} \tag{A.26}$$

Comparison with Equation (A.18) shows that these two integrals are solved by Equation (A.11). The solutions are thus, after some algebra,

$$\begin{aligned} \mu_{m(12)} &= w_1 \left[\mu_{\mathfrak{c}1} + \sigma_{\mathfrak{c}1} \frac{b_1}{a_1} \frac{\phi(k_1)}{\Phi(k_1)} \right] + w_2 \left[\mu_{\mathfrak{c}2} + \sigma_{\mathfrak{c}2} \frac{b_2}{a_2} \frac{\phi(k_2)}{\Phi(k_2)} \right] \\ \sigma^2_{m(12)} &= w_1 \left\{ \left[\mu^2_{\mathfrak{c}1} + \sigma^2_{\mathfrak{c}1} \right] + \left[2\mu_{\mathfrak{c}1}\sigma_{\mathfrak{c}1} \frac{b_1}{a_1} - k_1 \sigma^2_{\mathfrak{c}1} \frac{b_1^2}{a_1^2} \right] \frac{\phi(k_1)}{\Phi(k_1)} \right\} + \\ &\quad w_2 \left\{ \left[\mu^2_{\mathfrak{c}2} + \sigma^2_{\mathfrak{c}2} \right] + \left[2\mu_{\mathfrak{c}2}\sigma_{\mathfrak{c}2} \frac{b_2}{a_2} - k_2 \sigma^2_{\mathfrak{c}2} \frac{b_2^2}{a_2^2} \right] \frac{\phi(k_2)}{\Phi(k_2)} \right\} - \mu^2_{m(12)} \end{aligned} \tag{A.27}$$

where

$$w_1 = Z^{-1}\mathcal{N}(\mu_{\mathfrak{m}}; \mu_{\mathfrak{g}1}, \sigma_{\mathfrak{m}}^2 + \sigma_1^2)\Phi(k_1) \qquad w_2 = Z^{-1}\mathcal{N}(\mu_{\mathfrak{m}}; \mu_{\mathfrak{g}2}, \sigma_{\mathfrak{m}}^2 + \sigma_2^2)\Phi(k_2) \tag{A.28}$$

$$a_1 = \left[\sigma_{\mathfrak{g}1}^2\sigma_{\mathfrak{g}2}^2(1-\varrho^2) + (\sigma_{\mathfrak{g}1} - \varrho\sigma_{\mathfrak{g}2})^2\sigma_{\mathfrak{c}1}^2\right]^{1/2} \quad a_2 = \left[\sigma_{\mathfrak{g}1}^2\sigma_{\mathfrak{g}2}^2(1-\varrho^2) + (\sigma_{\mathfrak{g}2} - \varrho\sigma_{\mathfrak{g}1})^2\sigma_{\mathfrak{c}2}^2\right]^{1/2} \tag{A.29}$$

$$b_1 = \sigma_{\mathfrak{c}1}(\sigma_{\mathfrak{g}1} - \varrho\sigma_{\mathfrak{g}2}) \qquad b_2 = \sigma_{\mathfrak{c}2}(\sigma_{\mathfrak{g}2} - \varrho\sigma_{\mathfrak{g}1}) \tag{A.30}$$

Inverse Problem

The derivation for the inverse problem is just slightly more involved. We are interested in the moments of the marginals $p(x_1 \,|\, \mathcal{I}_{\mathfrak{c}})$ and $p(x_2 \,|\, \mathcal{I}_{\mathfrak{c}})$, and will denote these means and variances with $\mu_{1(m2)}$, $\sigma_{1(m2)}^2$, et cetera. From Equation (A.25), we get

$$\begin{aligned}\mu_{1(m2)} = \langle x_1 \rangle_{\mathcal{I}_{\mathfrak{c}}} = &\int_{-\infty}^{\infty} x_1 \int_{-\infty}^{x_1} \mathcal{N}(x_1; \mu_{\mathfrak{m}}, \sigma_{\mathfrak{m}}^2)\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{\mathfrak{g}}, \boldsymbol{\Sigma}_{\mathfrak{g}})\,\mathrm{d}x_2\,\mathrm{d}x_1 + \\ &\int_{-\infty}^{\infty}\int_{-\infty}^{x_2} x_1\mathcal{N}(x_2; \mu_{\mathfrak{m}}, \sigma_{\mathfrak{m}}^2)\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{\mathfrak{g}}, \boldsymbol{\Sigma}_{\mathfrak{g}})\,\mathrm{d}x_1\,\mathrm{d}x_2.\end{aligned} \tag{A.31}$$

The first integral is in fact identical to the first term of $\mu_{m(12)}$. The second term, however, involves the first *incomplete* moment:

$$\begin{aligned}&\int_{-\infty}^{\infty}\int_{-\infty}^{x_2} x_1\mathcal{N}(x_2; \mu_{\mathfrak{m}}, \sigma_{\mathfrak{m}}^2)\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{\mathfrak{g}}, \boldsymbol{\Sigma}_{\mathfrak{g}})\,\mathrm{d}x_1\,\mathrm{d}x_2 \\ &= \int_{-\infty}^{\infty} \mathcal{N}(x_2; \mu_{\mathfrak{m}}, \sigma_{\mathfrak{m}}^2)\mathcal{N}(x_2; \mu_{\mathfrak{g}2}, \sigma_{\mathfrak{g}2}^2)\times \\ &\qquad \int_{-\infty}^{x_2} x_1\mathcal{N}\left[x_1; \mu_{\mathfrak{g}1} + \varrho\frac{\sigma_{\mathfrak{g}1}}{\sigma_{\mathfrak{g}2}}(x_2 - \mu_{\mathfrak{g}2}), \sigma_{\mathfrak{g}1}^2(1-\varrho^2)\right]\mathrm{d}x_1\,\mathrm{d}x_2\end{aligned} \tag{A.32}$$

The inner integral can be solved using the result given in Equation (A.10), leading to an expression solved by Equation (A.11). After a bit of algebra, we arrive at the final result

$$\begin{aligned}\mu_{1(m2)} &= w_1\left[\mu_{\mathfrak{c}1} + \sigma_{\mathfrak{c}1}\frac{b_1}{a_1}\frac{\phi(k_1)}{\Phi(k_1)}\right] + w_2\left[\left(\mu_{\mathfrak{g}1} + \varrho\frac{\sigma_{\mathfrak{g}1}}{\sigma_{\mathfrak{g}2}}(\mu_{\mathfrak{c}2} - \mu_{\mathfrak{g}2})\right) + \frac{A}{a_2}\frac{\phi(k_2)}{\Phi(k_2)}\right] \\ \sigma_{1(m2)}^2 &= w_1\left\{\left[\mu_{\mathfrak{c}1}^2 + \sigma_{\mathfrak{c}1}^2\right] + \left[2\mu_{\mathfrak{c}1}\sigma_{\mathfrak{c}1}\frac{b_1}{a_1} - k_1\sigma_{\mathfrak{c}1}^2\frac{b_1^2}{a_1^2}\right]\frac{\phi(k_1)}{\Phi(k_1)}\right\} + \\ &\qquad w_2\left\{\sigma_{\mathfrak{g}1}^2\left[\left(\frac{\mu_{\mathfrak{g}1}}{\sigma_{\mathfrak{g}1}} + \varrho\frac{(\mu_{\mathfrak{c}2} - \mu_{\mathfrak{g}2})}{\sigma_{\mathfrak{g}2}}\right)^2 + (1-\varrho^2) + \varrho^2\frac{\sigma_{\mathfrak{c}2}^2}{\sigma_{\mathfrak{g}2}^2}\right] + \right. \\ &\qquad \left.\left[\frac{B}{h(1+\sigma_{\mathfrak{c}2}^2/h^2)^{1/2}} - \frac{C}{h^3(1+\sigma_{\mathfrak{c}2}^2/h^2)^{3/2}}\right]\frac{\phi(k_2)}{\Phi(k_2)}\right\} - \mu_1^2\end{aligned} \tag{A.33}$$

where

$$
\begin{aligned}
A &= \varrho\sigma_{\mathfrak{c}2}^2\sigma_{\mathfrak{g}1}\left(1-\varrho\frac{\sigma_{\mathfrak{g}1}}{\sigma_{\mathfrak{g}2}}\right)-\sigma_{\mathfrak{g}1}^2\sigma_{\mathfrak{g}2}(1-\varrho^2)\\
B &= 2\varrho^2\frac{\sigma_{\mathfrak{g}1}^2}{\sigma_{\mathfrak{g}2}^2}\sigma_{\mathfrak{c}2}^2(\mu_{\mathfrak{c}2}-\mu_{\mathfrak{g}2})+\varrho\frac{\sigma_{\mathfrak{g}1}}{\sigma_{\mathfrak{g}2}}\left(2\sigma_{\mathfrak{c}2}^2\mu_{\mathfrak{g}1}+\mu_{\mathfrak{g}2}\frac{\sigma_{\mathfrak{g}1}^2\sigma_{\mathfrak{g}2}(1-\varrho^2)}{\sigma_{\mathfrak{g}2}-\varrho\sigma_{\mathfrak{g}1}}\right)\\
&\quad-\mu_{\mathfrak{g}1}\sigma_{\mathfrak{g}1}^2(1-\varrho^2)\frac{\sigma_{\mathfrak{g}2}}{\sigma_{\mathfrak{g}2}-\varrho\sigma_{\mathfrak{g}1}}\\
C &= \varrho^2\frac{\sigma_{\mathfrak{g}1}^2}{\sigma_{\mathfrak{g}2}^2}\sigma_{\mathfrak{c}2}^4(\mu_{\mathfrak{c}2}-f)+\sigma_{\mathfrak{g}1}^2(1-\varrho^2)\left(1+\varrho\frac{\sigma_{\mathfrak{g}1}}{\sigma_{\mathfrak{g}2}}\right)\frac{\sigma_{\mathfrak{g}2}}{\sigma_{\mathfrak{g}2}-\varrho\sigma_{\mathfrak{g}1}}(\mu_{\mathfrak{c}2}h^2+f\sigma_{\mathfrak{c}2}^2)
\end{aligned}
\tag{A.34}
$$

with

$$
f=\frac{\sigma_{\mathfrak{g}2}\mu_{\mathfrak{g}1}-\varrho\sigma_{\mathfrak{g}1}\mu_{\mathfrak{g}2}}{\sigma_{\mathfrak{g}2}-\varrho\sigma_{\mathfrak{g}1}}\qquad\text{and}\qquad h=\frac{\sigma_{\mathfrak{g}1}\sigma_{\mathfrak{g}2}(1-\varrho^2)^{1/2}}{\sigma_{\mathfrak{g}2}-\varrho\sigma_{\mathfrak{g}1}}\tag{A.35}
$$

The corresponding result for the posterior marginal on x_2 can be derived trivially from these results by exchanging the indices 1 and 2. Note that, as mentioned above, the first terms of these mixtures are shared with the posterior for m. Intuitively, this can be interpreted as follows: For the posterior on m, the first term (ν_1) corresponds to the statement that "if $x_1 > x_2$" (the probability of this is encoded by the cumulative density term in Equation (A.18)) "then m is distributed like x_1" (represented by the product of the probability density functions in (A.18)). This part of the relationship features in the inverse problem as well: If $x_1 > x_2$, then x_1 is distributed like m. The second term in the posterior marginal on x_1 corresponds to the statement that "if $x_1 < x_2$, then x_2 is distributed like m, and x_1 is distributed such that its distribution fits with the updated marginal of x_2 given the correlation between x_1 and x_2 and the prior marginal on x_1.

Related Work

The moments of the likelihood of the max have been derived before by Clark [1961]. That is, for $\sigma_{\mathfrak{m}} \to \infty$, the posterior $p(m \,|\, \mathcal{I}_{\mathfrak{c}})$ reported here simplifies to a result reported by Clark:

$$
\begin{aligned}
\mu_{m(12)} &\to \Phi(k)\left[\mu_{\mathfrak{g}1}+\sigma_{\mathfrak{g}1}\frac{(\sigma_{\mathfrak{g}1}-\varrho\sigma_{\mathfrak{g}2})}{a}\frac{\phi(k)}{\Phi(k)}\right]+\Phi(-k)\left[\mu_{\mathfrak{g}2}+\sigma_{\mathfrak{g}2}\frac{(\sigma_{\mathfrak{g}2}-\varrho\sigma_{\mathfrak{g}1})}{a}\frac{\phi(-k)}{\Phi(-k)}\right]\\
\sigma_{m(12)}^2 &\to \Phi(k)\left\{\left[\mu_{\mathfrak{g}1}^2+\sigma_{\mathfrak{g}1}^2\right]+\left[2\mu_{\mathfrak{g}1}\sigma_{\mathfrak{g}1}\frac{(\sigma_{\mathfrak{g}1}-\varrho\sigma_{\mathfrak{g}2})}{a}-k\sigma_{\mathfrak{g}1}^2\frac{(\sigma_{\mathfrak{g}1}-\varrho\sigma_{\mathfrak{g}2})^2}{a^2}\right]\frac{\phi(k)}{\Phi(k)}\right\}\\
&\quad+\Phi(-k)\left\{\left[\mu_{\mathfrak{g}2}^2+\sigma_{\mathfrak{g}2}^2\right]+\left[2\mu_{\mathfrak{g}2}\sigma_{\mathfrak{g}2}\frac{(\sigma_{\mathfrak{g}2}-\varrho\sigma_{\mathfrak{g}1})}{a}+k\sigma_{\mathfrak{g}2}^2\frac{(\sigma_{\mathfrak{g}2}-\varrho\sigma_{\mathfrak{g}1})^2}{a^2}\right]\frac{\phi(-k)}{\Phi(-k)}\right\}\\
&\quad-\mu_{m(12)}^2\\
\text{where}\quad a &= \sqrt{\sigma_{\mathfrak{g}1}^2+\sigma_{\mathfrak{g}2}^2-2\varrho\sigma_{\mathfrak{g}1}\sigma_{\mathfrak{g}2}}\qquad\text{and}\qquad k=\frac{\mu_{\mathfrak{g}1}-\mu_{\mathfrak{g}2}}{a}
\end{aligned}
\tag{A.36}
$$

As expected, the posterior of the inverse problem simply becomes equal to the prior in this case. From Equation (A.33) we find

$$\begin{aligned}\mu_{1(m2)} &\to \Phi(k)\mu_1 + \sigma_1\frac{\sigma_1 - \varrho\sigma_2}{a}\phi(k) + \Phi(-k)\mu_1 - \sigma_1\frac{\sigma_1 - \varrho\sigma_2}{a}\phi(-k) \\ &= \Phi(k)\mu_1 + \sigma_1\frac{\sigma_1 - \varrho\sigma_2}{a}\phi(k) + (1 - \Phi(k))\mu_1 - \sigma_1\frac{\sigma_1 - \varrho\sigma_2}{a}\phi(k) \\ &= \mu_1\end{aligned} \tag{A.37}$$

and similarly for the variance.

The max-factor is also part of the Infer.NET software package [Minka and Winn, 2008] (to my knowledge, the derivations for this code have not been published yet). However, their implementation can only handle two independent Gaussian inputs (Section A.3 introduces the max over a finite set of correlated variables). So their implementation corresponds to the case of $\varrho = 0$, which leads to the following simplifications, presented here for reference:

$$k_1 = \frac{\mu_{\mathfrak{c}1} - \mu_{\mathfrak{g}2}}{(\sigma_{\mathfrak{g}1} + \sigma_{\mathfrak{c}2})^{1/2}} \qquad a_1 = \sigma_{\mathfrak{g}1}(\sigma_{\mathfrak{g}1} + \sigma_{\mathfrak{c}2})^{1/2} \qquad b_1 = \sigma_{\mathfrak{c}1}\sigma_{\mathfrak{g}1} \tag{A.38}$$

$$A = \sigma_{\mathfrak{g}1}^2\sigma_{\mathfrak{g}2} \qquad B = -\mu_{\mathfrak{g}1}\sigma_{\mathfrak{g}1}^2 \qquad C = \sigma_{\mathfrak{g}1}^2(\mu_{\mathfrak{c}2}\sigma_{\mathfrak{g}1}^2 + \mu_{\mathfrak{g}1}\sigma_{\mathfrak{c}2}^2) \tag{A.39}$$

$$f = \mu_{\mathfrak{g}1} \qquad h = \sigma_{\mathfrak{g}1} \tag{A.40}$$

Figure A.4 shows some of these approximations. The parameter settings used in this figure represent a worst case (e.g., the posterior over $\boldsymbol{x}$ is rarely so strongly bimodal.)

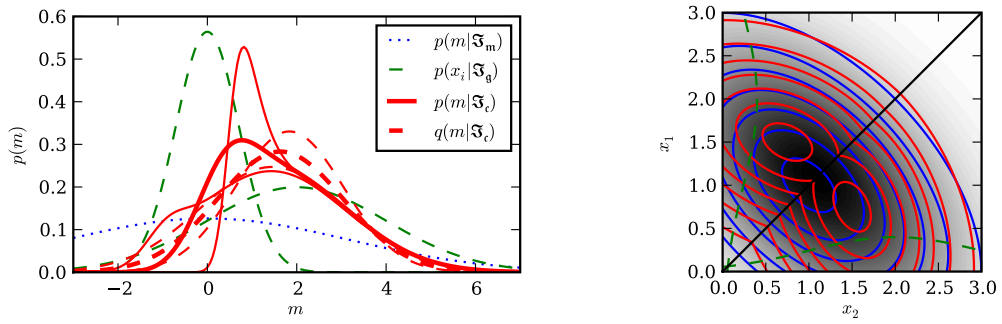


Figure A.4: Illustrative plots for the Gaussian approximations to the posteriors. Same beliefs in $\mathcal{I}_\mathfrak{c}$ as in Figure A.2. **Left:** For the sake of readability, only the cases $\varrho = -0.9$ (broadest), $\varrho = 0$ and $\varrho = 0.9$ are plotted here. In red dashed lines, the corresponding three Gaussian approximations. Note the varying quality of fit. **Right:** Gaussian approximation (with $\boldsymbol{\mu}_{1(m2)} = 1.06$ and $\sigma^2_{1(m2)} = 0.94$) indicated by shaded area.

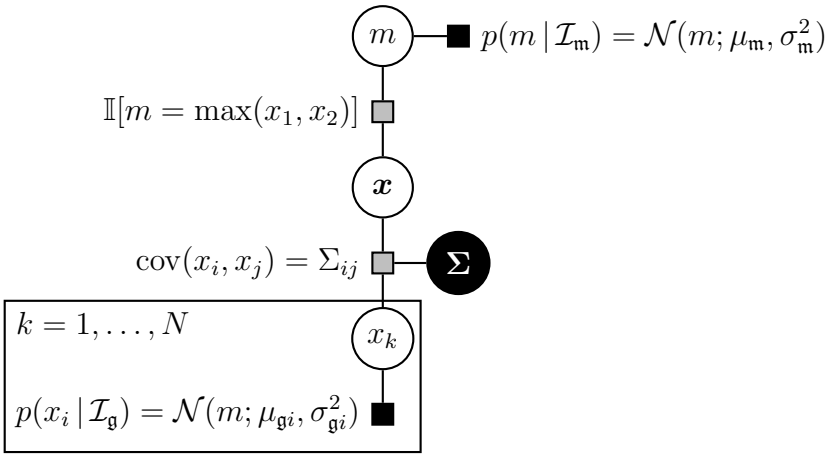


Figure A.5: Factor graph representation of the inference problem on a finite set. Note the plate representin N copies of generating variable nodes.

A.3 The Maximum of a Finite Set

A.3.1 Analytic Form

Extending the analysis of Section A.2.3, we can write the posterior over the max m of a finite set $\{x_i\}_{i=1,\ldots,N}$ of variables, distributed according to an N-dimensional version of Equation (A.1), with a new normalization constant Z_N, as

$$\begin{aligned} p(m \mid \mathcal{I}_\mathfrak{c}) &= Z_N p(m \mid \mathcal{I}_\mathfrak{m}) \int p(\boldsymbol{x} \mid m) p(\boldsymbol{x} \mid \mathcal{I}_\mathfrak{g}) \, \mathrm{d}\boldsymbol{x} \\ &= Z_N \mathcal{N}(m; \mu_\mathfrak{m}, \sigma_\mathfrak{m}) \times \\ &\qquad \left[\sum_{i=1}^{N} \int_{-\infty}^{\infty} \delta(m - x_i) p(x_i \mid \mathcal{I}_\mathfrak{g}) \int \cdots \int_{-\infty}^{x_i} p(\{x_j\}_{j \neq i} \mid x_i, \mathcal{I}_\mathfrak{g}) \prod_{j \neq i} \mathrm{d}x_j \, \mathrm{d}x_i \right] \\ &= Z_N \sum_i \left[\mathcal{N}(\mu_\mathfrak{m}; \mu_{\mathfrak{g}i}, \sigma^2_\mathfrak{m} + \sigma^2_{\mathfrak{g}i}) \mathcal{N}(m; \mu_{\mathfrak{c}i}, \sigma^2_{\mathfrak{c}i}) \times \right. \\ &\qquad \left. \int \cdots \int_{-\infty}^{x_i} \mathcal{N}(\boldsymbol{x}_{\setminus i}; \boldsymbol{\mu}_{\mathfrak{g}\setminus i}(x_i), \boldsymbol{\Sigma}_{\mathfrak{g}\setminus i}) \, \mathrm{d}\boldsymbol{x}_{\setminus i} \right] \end{aligned} \tag{A.41}$$

where $\boldsymbol{x}_{\setminus i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N)$. The conditional mean is [see e.g. Bishop, 2006, Section 2.3.2]

$$\left(\boldsymbol{\mu}_{\setminus i}(x_i)\right)_j = \mu_{\mathfrak{g}j} + \Sigma_{\mathfrak{g}ji} \Sigma^{-1}_{\mathfrak{g}ii} (x_i - \mu_{\mathfrak{g}i}) = \mu_{\mathfrak{g}j} + \varrho_{ij} \frac{\sigma_{\mathfrak{g}j}}{\sigma_{\mathfrak{g}i}} (x_i - \mu_{\mathfrak{g}i}) \tag{A.42}$$

with the linear coefficient of correlation $\varrho_{ij} = \Sigma_{\mathfrak{g}ij} / (\sigma_{\mathfrak{g}i} \sigma_{\mathfrak{g}j})$. The conditional covariance matrix is the Schur complement of $\Sigma_{\mathfrak{g}ii} = \sigma^2_{\mathfrak{g}i}$ in $\boldsymbol{\Sigma}_\mathfrak{g}$:

$$\Sigma_{\mathfrak{g}\setminus i, kj} = \Sigma_{\mathfrak{g}kj} - \Sigma_{\mathfrak{g}ki} \sigma^{-2}_{\mathfrak{g}i} \Sigma_{\mathfrak{g}ij} \tag{A.43}$$

In principle, it would be possible to follow the path laid out in the previous sections to calculate the first two moments of this distribution. However, while the univariate Gaussian CDF (essentially an evaluation of the error function) has computational cost comparable to evaluating an exponential function, computationally efficient ways of calculating a multivariate Gaussian CDF are not generally available. See, however, Appendix B for a numerical scheme which does in fact offer a good approximation to the required integrals. Unfortunately, even though it is a highly optimized numerical scheme, its computational cost is still considerable.

A.3.2 A Heuristic Approximation

Another, cheaper option is to use an iterative procedure initially proposed by Clark [1961]. The idea is to start out with the approximation for only two of the generating variables. W.l.o.g., let these be x_1 and x_2, resulting in $m_{(12)} = \max(x_1, x_2)$. Next, estimate $m_{(123)} = \max(x_3, m_{(12)})$ and so on up to $m_{(1\dots N)}$. For the intermediate maxima, the likelihoods presented in Equation (A.36) suffice, and the prior is included in the last step (using Equation (A.27)) to gain an approximate posterior over the maximum of the whole set. Of course, this necessitates an analytic expression for the correlation coefficient $\varrho_{i(1\dots i-1)}$ between the i-th variable and the max over the preceding variables. This was derived by Clark. Adopted to the notation used here and made more explicit, his result is

$$\varrho_{3(12)} = \sigma_{(12)}^{-1} \left(\sigma_1 \varrho_{31} \Phi(k_{(12)}) + \sigma_2 \varrho_{32} \Phi(-k_{(12)})\right), \tag{A.44}$$

where $\varrho_{ij} = \Sigma_{ij}/\sigma_i\sigma_j$, the index $\mathfrak{g}$ has been dropped for simplicity and $k_{(12)} = (\mu_1 - \mu_2)/\sqrt{\sigma_1^2 + \sigma_2^2 - 2\varrho_{12}\sigma_1\sigma_2}$ is the simplified version of k_1 arising from Equation (A.22) under $\sigma_{\mathfrak{m}} \to \infty$. Using this, we can build a recursive algorithm to calculate $\varrho_{i(1\dots j)}$ with $j < i$ as

$$\varrho_{i(1\dots j)} = \sigma_{(1\dots j)}^{-1} \cdot \begin{cases} \sigma_j \varrho_{ij} & \text{if } j = 1 \\ \Phi(-k_{(1\dots j)})\sigma_j \varrho_{ij} + \Phi(k_{(1\dots j)})\varrho_{i(1\dots j-1)} & \text{otherwise} \end{cases} \tag{A.45}$$

this necessitates a list $\boldsymbol{k}_{[j]} = (k_{(12)}, \dots k_{(1\dots i-1)})$ which is available at the necessary point in time from the calculation of previous maxima over the preceding parts of the set. Note that calculating $\varrho_{i(1\dots i-1)}$ involves $i-1$ recursive function calls, so building the full approximation over the max of N variables is of complexity $\mathcal{O}(N^2)$, as might be expected (although there are only $(N-1)$ uses of the results in Equation (A.27)). If all correlation coefficients are the same, $\varrho_{ij} = \varrho \,\forall ij$, then the recursive evaluations can be re-used in consecutive evaluations and the complexity drops to $\mathcal{O}(N)$.

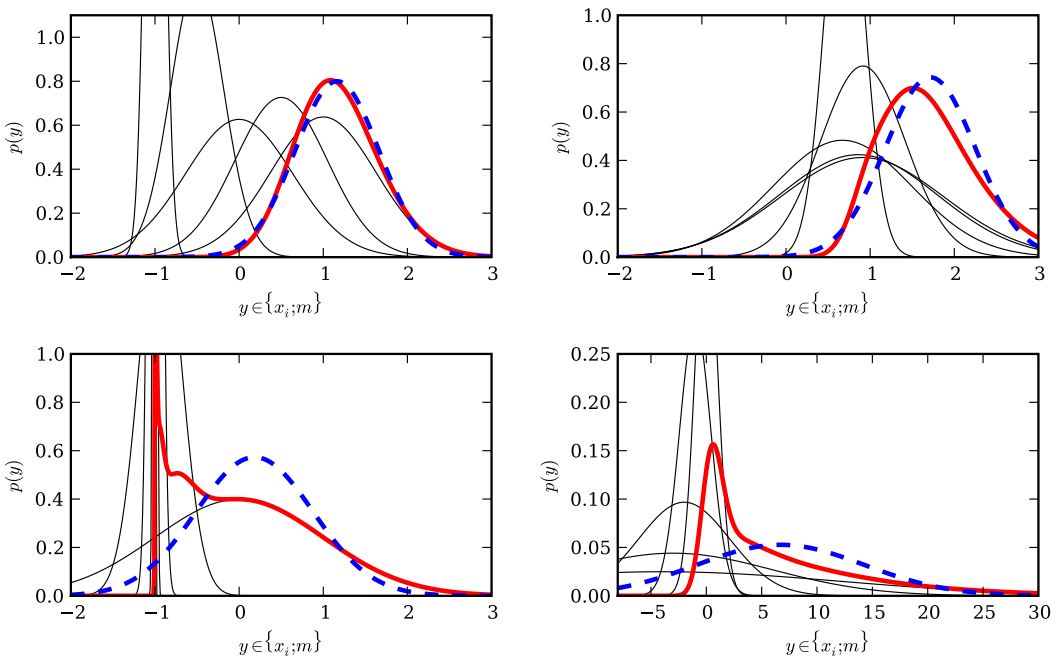


Figure A.6: Quality and failure modes of the EP approximation. Max of five uncorrelated Gaussians. **Top row:** examples of good fits. **Left:** well separated beliefs. **Right:** similar beliefs. **Bottom row:** worst case examples. **Left:** high certainty contributions within the center. **Right:** high uncertainty in one tail. In all plots, beliefs over the x_i as slim black lines. True posterior over m in thick red, approximation in thick dashed blue. For simplicity, $p(m \,|\, \mathcal{I}_\mathfrak{m})$ was set to an uninformative value. See text for details.

Inverse Problem

The same iterative scheme can be used to provide an approximation for the inverse problem's posterior. First, the list $\boldsymbol{k}_{[j]}$ is build as in the preceding section. Then, approximations to the posterior marginals are build iteratively, starting with $q(x_N \,|\, \mathcal{I}_\mathfrak{c})$, ending with $q(x_2 \,|\, \mathcal{I}_\mathfrak{c})$ and $q(x_1 \,|\, \mathcal{I}_\mathfrak{c})$. At each intermediate step, we use the EP approximation: To get $q(x_i \,|\, \mathcal{I}_\mathfrak{c})$, use $q(m_{(1\ldots i)} \,|\, \mathcal{I}_\mathfrak{m}) = q(m_{(1\ldots i)} \,|\, \mathcal{I}_\mathfrak{c}) / q(m_{(1\ldots i)} \,|\, \mathcal{I}_\mathfrak{g})$ as an approximation to the prior over the subset max, and $q(m_{(1\ldots i-1)} \,|\, \mathcal{I}_\mathfrak{g})$ as the approximation on the max over the subset up to x_{i-1}.

A.4 Discussion of the Approximation's Quality

Figure A.6 gives some intuition on the quality of the approximation. For the purpose of this comparison, uncorrelated Gaussians were used because this allows the analytic evaluation of the true posterior (the CDF factorizes into individual one-dimensional CDFs). The fit is reasonably good if the beliefs over the x_i are either very similar (Figure A.6 top right), or if the beliefs are "separated", in the sense that one of the x_i provides a dominant contribution to the overall mixture (top left). The fit becomes bad when the mixture has many modes (bottom left) or a strong

asymmetry (bottom right). The corresponding worst case distributions shown here were generated by setting $\mu_{\mathfrak{g}i} = a + b^{-i}$ and $\sigma^2_{\mathfrak{g}i} = b^{-i}$ (left, $a = -1, b = 16$) or $\mu_{\mathfrak{g}i} = ci$ and $\sigma^2_{\mathfrak{g}i} = i^d + 1$ (right, $c = -1, d = 16$). More quantitatively, consider Equation (A.36) or Equation (A.18), the case of the max of only two Gaussians. The two cases of good fit described above correspond to

1. one mixture component dominating the mixture, i.e.

$$|k_{12}| = \frac{|\mu_{\mathfrak{g}1} - \mu_{\mathfrak{g}2}|}{\sqrt{\sigma^2_{\mathfrak{g}1} + \sigma^2_{\mathfrak{g}2} - 2\varrho\sigma_{\mathfrak{g}1}.\sigma_{\mathfrak{g}2}}} \gg 0 \tag{A.46}$$

   The likelihood then has one clearly dominating Gaussian component and the fit is good. In this case, the inverse problem is also a good fit, as each of the generating variables x_1, x_2 has one dominating component in its posterior.

2. the two mixture components being almost identical:

$$\mu_{\mathfrak{g}1} \approx \mu_{\mathfrak{g}2} \qquad \text{and} \qquad \sigma_{\mathfrak{g}1} \approx \sigma_{\mathfrak{g}2}. \tag{A.47}$$

   The likelihood then consists of two roughly identical Gaussian components with roughly the same weights, and is therefore roughly Gaussian. However, the approximation is *bad* for the inverse problem here, as the true posterior marginals become bimodal (c.f. Figure A.2, right). This effect is particularly pronounced if the mean of the prior and the likelihood differ significantly.

These observations suggest a potential increase in the quality of the approximation to be gained from calculating all $N(N-1)$ weight-generators k_{ij} as defined in Equation (A.46) and iteratively choosing the pair ij with maximal k_{ij}. However, this re-ordering has to be updated after each incremental two-component max operation, involving a re-calculation of up to N correlation coefficients. It thus raises the complexity of calculating the approximation for the overall max from $\mathcal{O}(N^2)$ to $\mathcal{O}(N^3)$. Initial experiments suggest that the potential gain in fit is almost always negligible.

A.5 Conclusion

This chapter derived the first two moments of the posterior over the maximum of a pair of Gaussian variables, and over the posterior over the two generating variables. These moments can be used for approximate inference on their own, or as part of a larger graphical model using Expectation Propagation. I have also shown how to extend the usefulness of these approximations to finite sets of Gaussian variables using a heuristic iterative approximation. The quality of the approximation

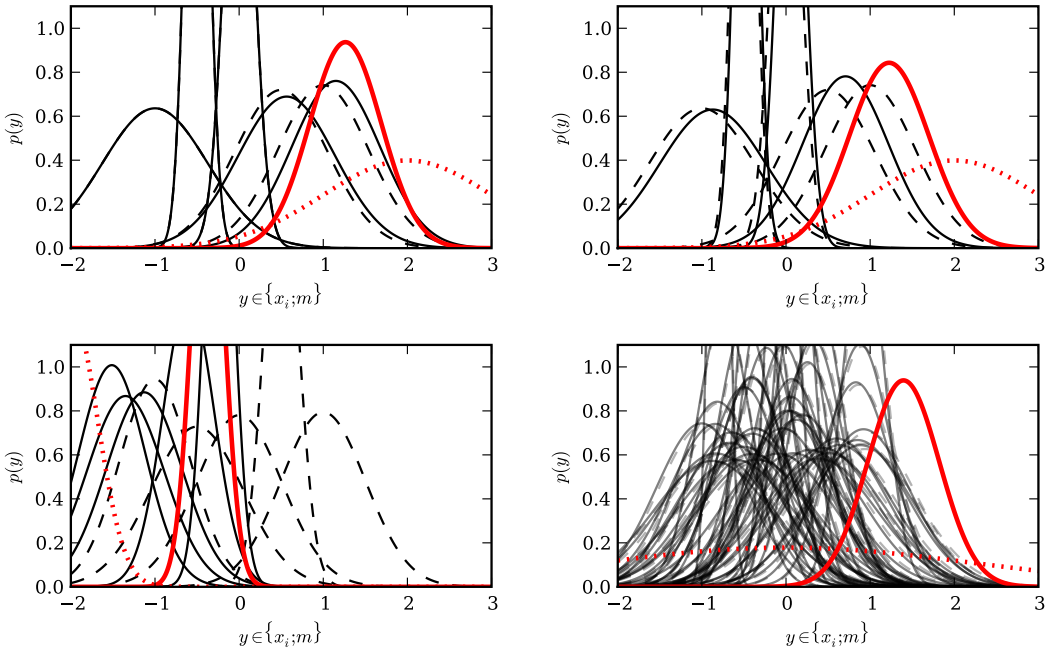


Figure A.7: Illustrative examples for uses of approximate max factor. $p(x_i \mid \mathcal{I}_\mathfrak{g})$ in black dashed lines. $p(m \mid \mathcal{I}_\mathfrak{m})$ in red dotted. Marginals after EP message passing as corresponding solid lines. **Top left:** Max over 5 uncorrelated variables. Only the two variables contributing significantly to the max change their beliefs. **Top right:** same as previous, but with $\varrho_{ij} = 0.9$ for all ij. The change in belief over the dominating x_i now also affects the other beliefs, as expected. **Bottom left:** The approximation is well-behaved under inconsistent beliefs. $p(m \mid \mathcal{I}_\mathfrak{m})$ was set inconsistently low relative to the beliefs on the x_i (all $\varrho_{ij} = 0.2$). Note that the belief over the largest x_i extends beyond the belief over m as a result of the moment-matching. **Bottom right:** The approximation is stable for large values of N. Maximum over 50 correlated normals, all ϱ_{ij} were set to 0.5.

depends on the location and precision of the belief over the generating variables relative to each other, but is always good enough to provide a meaningful point estimate and error measure. It is sufficiently robust to deal with inconsistent belief assignments and large numbers of generating variables (see Figure A.7).

Appendix B

Efficient Rank 1 EP Updates

Chapter 4 mentions a particularly efficient way of updating multivariate Gaussian beliefs from factors with rank 1 derivatives. This update rule was derived by Ralf Herbrich and communicated in a technical report by Minka [2008]. The report is very concise and has not been published in a permanently accessible outlet. The entire derivation is thus reproduced here for completeness, with some notational adaptations and expansions to improve accessibility. We start by deriving the *general* update rules for a Gaussian approximation under the Expectation Propagation scheme: We want to approximate a factor $f_i(\boldsymbol{x})$ by a Gaussian term of the form

$$\tilde{f}_i(\boldsymbol{x}) = s_i \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^\mathsf{T}\boldsymbol{V}_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)\right) \tag{B.1}$$

The approximate unnormalized Gaussian marginal on $\boldsymbol{x}$ is

$$q(\boldsymbol{x}) = s\mathcal{N}(\boldsymbol{x};\boldsymbol{m},\boldsymbol{V}) \tag{B.2}$$

where the normalization constant s can be used for model comparison or as a predictive probability, as in Chapter 4. We will require the cavity distribution for the derivation of the updates, which is defined as

$$\begin{aligned} q^{\backslash i}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x};\boldsymbol{m}^{\backslash i},\boldsymbol{V}^{\backslash i}) &\propto \frac{q(\boldsymbol{x})}{\tilde{f}_i(\boldsymbol{x})} \\ \text{with} \quad \boldsymbol{V}^{\backslash i} &= (\boldsymbol{V}^{-1}-\boldsymbol{V}_i^{-1})^{-1} \\ \text{and} \quad \boldsymbol{m}^{\backslash i} &= \boldsymbol{V}^{\backslash i}(\boldsymbol{V}^{-1}\boldsymbol{m}-\boldsymbol{V}_i^{\backslash i}\boldsymbol{m}_i) \end{aligned} \tag{B.3}$$

An important quantity is the normalization constant of the EP update,

$$Z_i(\boldsymbol{m}^{\backslash i},\boldsymbol{V}^{\backslash i}) = \int f_i(\boldsymbol{x}) q^{\backslash i}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \tag{B.4}$$

Its derivative with respect to $\boldsymbol{m}^{\setminus i}$ can be rewritten to reveal an expression for the mean of the EP message, in the following way.

$$\begin{aligned} \nabla_m \log Z_i &= \frac{1}{Z_i} \int f_i(\boldsymbol{x}) q^{\setminus i}(\boldsymbol{x}) \boldsymbol{V}_i^{-1} (\boldsymbol{x} - \boldsymbol{m}^{\setminus i}) \, \mathrm{d}\boldsymbol{x} \\ \Rightarrow \qquad \boldsymbol{m} &= \boldsymbol{m}^{\setminus i} + \boldsymbol{V}_i \nabla_m \log Z_i \end{aligned} \tag{B.5}$$

and analogously, after a few more lines of algebra

$$\boldsymbol{V} = \boldsymbol{V}^{\setminus i} - \boldsymbol{V}^{\setminus i} \left[\left(\nabla_m \nabla_m^\mathsf{T} - 2\nabla_v \right) \log Z_i \right] \boldsymbol{V}^{\setminus i} \tag{B.6}$$

Using Equation (B.6), Equation (B.3), and the matrix inversion lemma (Equation C.17), we get an expression for the covariance matrix of the EP message

$$\boldsymbol{V}_i = \left[\left(\nabla_m \nabla_m^\mathsf{T} - 2\nabla_v \right) \log Z_i \right]^{-1} - \boldsymbol{V}^{\setminus i} \tag{B.7}$$

using further Equations (B.5) and (B.3) reveals an expression for the message mean:

$$\begin{aligned} \boldsymbol{m}_i &= \boldsymbol{V}_i (\boldsymbol{V}^{-1} \boldsymbol{m} - (\boldsymbol{V}^{\setminus i})^{-1} \boldsymbol{m}^{\setminus i}) \\ &= \boldsymbol{m}^{\setminus i} + (\boldsymbol{V}_i + \boldsymbol{V}^{\setminus i})(\boldsymbol{V}^{\setminus i})^{-1} (\boldsymbol{m} - \boldsymbol{m}^{\setminus i}) \\ &= \boldsymbol{m}^{\setminus i} + (\boldsymbol{V}_i + \boldsymbol{V}^{\setminus i}) \nabla_m \log Z_i \\ &= \boldsymbol{m}^{\setminus i} + \left[\left(\nabla_m \nabla_m^\mathsf{T} - 2\nabla_v \right) \log Z_i \right]^{-1} \nabla_m \log Z_i \end{aligned} \tag{B.8}$$

Even the contribution to the normalization constant can be expressed with these two derivatives. Using the result (A.20) again, we get

$$\begin{aligned} s_i &= Z_i \frac{|\boldsymbol{V}_i + \boldsymbol{V}^{\setminus i}|^{1/2}}{|\boldsymbol{V}_i|^{1/2}} \exp\left(\frac{1}{2} (\boldsymbol{m}_i - \boldsymbol{m}^{\setminus i})^\mathsf{T} (\boldsymbol{V}_i + \boldsymbol{V}^{\setminus i})^{-1} (\boldsymbol{m}_i - \boldsymbol{m}^{\setminus i}) \right) \\ &= Z_i \left| \mathbf{I} + \boldsymbol{V}^{\setminus i} \boldsymbol{V}_i^{-1} \right|^{1/2} \exp\left(\frac{1}{2} \nabla_m^\mathsf{T} (\nabla_m \nabla_m^\mathsf{T} - 2\nabla_v)^{-1} \nabla_m \log Z_i \right) \end{aligned} \tag{B.9}$$

where the differential operators in the last line should be understood to all act on $\log Z_i$, the necessary brackets have been left out for readability. This special form for the updates can be leveraged if the derivatives have low rank. In particular, if the derivatives have rank 1:

$$\begin{aligned} \nabla_m \log Z_i &= \alpha_i \boldsymbol{\xi}_i \\ \nabla_m \nabla_m^\mathsf{T} - 2\nabla_v \log Z_i &= \beta_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i^\mathsf{T} \\ \boldsymbol{\xi}_i^\mathsf{T} (\nabla_m \nabla_m^\mathsf{T} - 2\nabla_v \log Z_i)^{-1} \boldsymbol{\xi}_i &= \beta_i^{-1} \end{aligned} \tag{B.10}$$

where α_i and β_i are some scalars and $\boldsymbol{\xi}_i$ is a vector, then the messages can be represented by one-dimensional Gaussians with mean m_i and variance v_i, defined

through

$$\begin{aligned} \boldsymbol{V}_i^{-1} &= v_i^{-1}\boldsymbol{\xi}_i\boldsymbol{\xi}_i^\mathsf{T} \\ \boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{V}_i\boldsymbol{\xi}_i &= v_i \\ m_i &= \boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{m}_i \end{aligned} \tag{B.11}$$

Updating such a one-dimensional message also only involves one-dimensional projections, namely $\boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{V}^{\setminus i}\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{m}^{\setminus i}$, which can be found through

$$\begin{aligned} \boldsymbol{V}^{\setminus i} &= (\boldsymbol{V}^{-1} - v_i^{-1}\boldsymbol{\xi}_i\boldsymbol{\xi}_i^\mathsf{T})^{-1} \\ &= \boldsymbol{V} + (\boldsymbol{V}\boldsymbol{\xi}_i)(v_i - \boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{V}\boldsymbol{\xi}_i)^{-1}(\boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{V}) \\ \boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{V}^{\setminus i}\boldsymbol{\xi}_i &= \boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{V}\boldsymbol{\xi}_i(1 - v_i^{-1}\boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{V}\boldsymbol{\xi}_i)^{-1} \qquad \text{and} \\ \boldsymbol{m}^{\setminus i} &= \boldsymbol{m} + \boldsymbol{V}^{\setminus i}\boldsymbol{V}_i^{-1}(\boldsymbol{m} - \boldsymbol{m}_i) \\ &= \boldsymbol{m} + \boldsymbol{V}^{\setminus i}\boldsymbol{\xi}_i v_i^{-1}(\boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{m} - m_i) \\ &= \boldsymbol{m} + (\boldsymbol{V}\boldsymbol{\xi}_i)(1 - v_i^{-1}\boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{V}\boldsymbol{\xi}_i)^{-1}v_i^{-1}(\boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{m} - m_i) \\ \boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{m}^{\setminus i} &= \boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{m} + (\boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{V}\boldsymbol{\xi}_i)(1 - v_i^{-1}\boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{V}\boldsymbol{\xi}_i)^{-1}v_i^{-1}(\boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{m} - m_i) \end{aligned} \tag{B.12}$$

Note that these terms can be evaluated in order $\mathcal{O}(D^2)$ if $V \in \mathbb{R}^D$, instead of the general $\mathcal{O}(D^3)$ of Equation (B.7). Often, the vector $\boldsymbol{\xi}_i$ will be sparse, reducing cost further. Equations (B.7), (B.8) and (B.9) now become one-dimensional updates:

$$\begin{aligned} m_i &= \boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{m}^{\setminus i} + \frac{\alpha_i}{\beta_i} \\ v_i &= \beta_i^{-1} - \boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{V}^{\setminus i}\boldsymbol{\xi}_i \\ s_i &= Z_i(\beta_i v_i)^{-1/2}\exp\left(\frac{\alpha_i^2}{2\beta_i}\right) \end{aligned} \tag{B.13}$$

The natural parameters of the Gaussian are v_i^{-1} and $v_i^{-1}m_i$ (see Equation 2.32). In those parameters, the updates to the messages are,

$$\begin{aligned} \text{using} \quad r &\equiv \frac{\beta_i}{(\boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{V}^{\setminus i}\boldsymbol{\xi}_i)^{-1} - \beta_i}, \\ v_i^{-1} &= r(\boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{V}^{\setminus i}\boldsymbol{\xi}_i)^{-1} \\ v_i^{-1}m_i &= r\left(\alpha_i + \frac{\boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{m}^{\setminus i}}{\boldsymbol{\xi}_i^\mathsf{T}\boldsymbol{V}^{\setminus i}\boldsymbol{\xi}_i}\right) + \alpha_i \end{aligned} \tag{B.14}$$

The updates to the marginal can be derived from Equation (B.5) and Equation (B.6), and are also of cost $\mathcal{O}(D^2)$, or less if $\boldsymbol{\xi}$ is sparse:

$$\begin{aligned} \text{using} \quad & \Delta(v_i^{-1}) \equiv (v_i^{\text{new}})^{-1} - (v_i^{\text{old}})^{-1} \\ \text{and} \quad & \Delta(v_i^{-1} m_i) \equiv (v_i^{\text{new}})^{-1} m_i^{\text{new}} - (v_i^{\text{old}})^{-1} m_i^{\text{old}}, \\ \text{we get} \quad & \boldsymbol{V}^{\text{new}} = \boldsymbol{V} - \frac{\Delta(v_i^{-1})}{1 + \boldsymbol{\xi}_i \boldsymbol{V} \boldsymbol{\xi}_i \Delta(v_i^{-1})} \boldsymbol{V} \boldsymbol{\xi}_i \boldsymbol{\xi}_i^\mathsf{T} \boldsymbol{V} \\ \text{and} \quad & \boldsymbol{m}^{\text{new}} = \boldsymbol{m} + \frac{\Delta(v_i^{-1} m_i) - \boldsymbol{\xi}_i^\mathsf{T} \boldsymbol{m} \Delta(v_i^{-1})}{1 + \boldsymbol{\xi}_i^\mathsf{T} \boldsymbol{V} \boldsymbol{\xi}_i \Delta(v_i^{-1})} \boldsymbol{V} \boldsymbol{\xi}_i. \end{aligned} \tag{B.15}$$

The normalization constant (evidence term) of the marginal can be evaluated after the algorithm has converged. Its logarithm is

$$\log s = \frac{1}{2} \left[\boldsymbol{m}^\mathsf{T} \boldsymbol{V}^{-1} \boldsymbol{m} - \boldsymbol{m}_0^\mathsf{T} \boldsymbol{V}_0^{-1} \boldsymbol{m}_0 + \log |\boldsymbol{V}| - \log |\boldsymbol{V}_0| - \sum_i \frac{m_i^2}{v_i} \right] + \sum_i \log s_i, \tag{B.16}$$

where the quantities with subscript 0 denote the values of the "prior", i.e. the initial normalized marginal before the EP messages are incorporated. Because precision matrix and precision-adjusted mean are the Gaussian's natural parameters (Equation (2.32)), we can utilize Equation (2.16) write the marginal's parameters at any point during the iterative updates compactly in terms of the contributions from prior and messages as

$$\begin{aligned} \boldsymbol{V}^{-1} &= \boldsymbol{V}_0^{-1} + \sum_i v_i^{-1} \boldsymbol{\xi}_i \boldsymbol{\xi}_i^\mathsf{T} \\ \boldsymbol{m} &= \boldsymbol{V} \left(\boldsymbol{V}_0^{-1} \boldsymbol{m}_0 + \sum_i v_i^{-1} m_i \boldsymbol{\xi}_i \right) \end{aligned} \tag{B.17}$$

B.1 The Step Factor

The derivations in Chapter 4 require the EP updates for the step function factor $f_i(\boldsymbol{x}) = \theta[\boldsymbol{\xi}_i^\mathsf{T}(\boldsymbol{x} - \boldsymbol{\omega}_i)]$ with arbitrary vectors $\boldsymbol{\xi}_i$ and $\boldsymbol{\omega}_i$ (this provides an approximation to the multivariate Gaussian integral over polyhedral regions). For this particular factor, the normalization constant is

$$\begin{aligned} Z_i(\boldsymbol{m}^{\backslash i}, \boldsymbol{V}^{\backslash i}) &= \int \theta[\boldsymbol{\xi}_i^\mathsf{T}(\boldsymbol{x} - \boldsymbol{\omega}_i)] \mathcal{N}(\boldsymbol{x}; \boldsymbol{m}^{\backslash i}, \boldsymbol{V}^{\backslash i}) \, \mathrm{d}\boldsymbol{x} \\ &= \int_{-\infty}^{z} \mathcal{N}(z'; 0, 1) \, \mathrm{d}z' = \Phi(z) \\ \text{with} \quad z &\equiv \frac{\boldsymbol{\xi}_i^\mathsf{T}(\boldsymbol{m}^{\backslash i} - \boldsymbol{\omega}_i)}{\sqrt{\boldsymbol{\xi}_i^\mathsf{T} \boldsymbol{V}^{\backslash i} \boldsymbol{\xi}_i}} \end{aligned} \tag{B.18}$$

and the scalars required in Equation (B.10) are

$$\begin{aligned} \text{and} \quad \alpha_i &= \frac{1}{\sqrt{\boldsymbol{\xi}_i^\mathsf{T} \boldsymbol{V}^{\setminus i} \boldsymbol{\xi}_i}} \frac{\mathcal{N}(z;0,1)}{\Phi(z)} \\ \text{and} \quad \beta_i &= \alpha_i \left(\alpha_i + \frac{\boldsymbol{\xi}_i^\mathsf{T} (\boldsymbol{m}^{\setminus i} - \boldsymbol{\omega}_i)}{\boldsymbol{\xi}_i \boldsymbol{V}^{\setminus i} \boldsymbol{\xi}_i} \right). \end{aligned} \tag{B.19}$$

Deriving these specific results is tedious, so only the most important steps will be reproduced here. Unnecessary subscripts will be left out from here on to reduce clutter. Because $\mathcal{N}(\boldsymbol{x};\boldsymbol{m},\boldsymbol{V}) = \mathcal{N}(\boldsymbol{x}-\boldsymbol{\omega};\boldsymbol{m}-\boldsymbol{\omega},\boldsymbol{V})$, we will limit the derivations to the case $\boldsymbol{\omega} = \boldsymbol{0}$, the general case can then be found simply by replacing $\boldsymbol{m} \to \boldsymbol{m} - \boldsymbol{\omega}$.

To get from the first line of Equation (B.18) to the second, notice that, for any vector $\boldsymbol{\xi}$, there exists an orthonormal basis $\{\boldsymbol{\zeta}, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_D\}$ of $\mathbb{R}^D$ containing a normalized form $\boldsymbol{\zeta} = \boldsymbol{\xi}/\sqrt{\boldsymbol{\xi}^\mathsf{T}\boldsymbol{\xi}}$. In other words, there exists an orthonormal matrix

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{\zeta} & \boldsymbol{w}_2 & \ldots & \boldsymbol{w}_D \end{pmatrix} \qquad \text{with} \qquad \boldsymbol{A}^\mathsf{T}\boldsymbol{A} = \mathbf{I}_D \tag{B.20}$$

which we can use to perform a change of basis to new co-ordinates $\boldsymbol{y} \equiv \boldsymbol{A}^\mathsf{T}\boldsymbol{x}$. Because $\boldsymbol{A}$ is orthonormal, its determinant is 1 and the measure is conserved. We can re-write the quadratic form of the Gaussian distribution as

$$\begin{aligned} (\boldsymbol{x}-\boldsymbol{m})^\mathsf{T}\boldsymbol{V}^{-1}(\boldsymbol{x}-\boldsymbol{m}) &= (\boldsymbol{x}-\boldsymbol{m})^\mathsf{T}\boldsymbol{A}\boldsymbol{A}^\mathsf{T}\boldsymbol{V}^{-1}\boldsymbol{A}\boldsymbol{A}^\mathsf{T}(\boldsymbol{x}-\boldsymbol{m}) \\ &= (\boldsymbol{x}-\boldsymbol{m})^\mathsf{T}\boldsymbol{A}\left(\boldsymbol{A}^\mathsf{T}\boldsymbol{V}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^\mathsf{T}(\boldsymbol{x}-\boldsymbol{m}) \end{aligned} \tag{B.21}$$

so the Gaussian can be rewritten in the new co-ordinate system as

$$\mathcal{N}(\boldsymbol{x};\boldsymbol{m},\boldsymbol{V}) = \mathcal{N}(\boldsymbol{A}^\mathsf{T}\boldsymbol{x};\boldsymbol{A}^\mathsf{T}\boldsymbol{m},\boldsymbol{A}^\mathsf{T}\boldsymbol{V}\boldsymbol{A}) \tag{B.22}$$

and we can now integrate out all the dimensions orthogonal to $\boldsymbol{\zeta}$, where the integration region spans the entire dimension, using Equation (A.13), leaving only a one-dimensional incomplete Gaussian integral

$$Z = \int \theta(\boldsymbol{\xi}^\mathsf{T}\boldsymbol{x})\mathcal{N}(\boldsymbol{\zeta}^\mathsf{T}\boldsymbol{x};\boldsymbol{\zeta}^\mathsf{T}\boldsymbol{m},\boldsymbol{\zeta}^\mathsf{T}\boldsymbol{V}\boldsymbol{\zeta})\,\mathrm{d}\boldsymbol{\zeta}^\mathsf{T}\boldsymbol{x} \tag{B.23}$$

The normalization terms $\sqrt{\boldsymbol{\xi}^\mathsf{T}\boldsymbol{\xi}}$ in the Gaussian all cancel, leading to the result of Equation (B.18). The forms of α and β can be found by differentiating Z with respect to $\boldsymbol{m}^{\setminus i}$ and the elements of $\boldsymbol{V}^{\setminus i}$, analogous to Equations (B.10).

Appendix C

A Laplace Map Linking Gaussians and Dirichlets

The Dirichlet distribution $\mathcal{D}$ is a measure over discrete probability distributions; the Gaussian $\mathcal{N}$ a measure over real values. The two are designed for quite different purposes. Nevertheless, sometimes the need to match them to each other arises. For example, consider a situation where discrete samples $\{c_n\} \in \mathbb{N}$ are available in several different "locations" x_n, and we would like to perform regression on the discrete probability $p(c_n = k \,|\, x_n)$ distributions in those and other locations. This situation is a case of *generalized regression*, and there are many different ways of dealing with it, using *link functions* mapping discrete probabilities to real values (the inverse of the link function is also known as the *activation function* in machine learning, due to historical links to neural systems). Two particularly well-known methods involve

- ▷ the evaluation of the *probit* link function, which provides a direct link to the Gaussian domain but is only easy to evaluate in the binary case (see, however, Section B.1 for a computationally involved yet very expressive approach to the multinomial case)

- ▷ using the softmax activation function $\boldsymbol{\sigma}$ with elements

$$\sigma_k(\boldsymbol{y}) = \frac{\exp(y_k)}{\sum_{k'}^{K} \exp(y_{k'})} \tag{C.1}$$

  to map a probability measure (typically a Gaussian) on $\mathbb{R}^K$ to normalized probabilities $\boldsymbol{\pi} \in [0,1]^K$. A disadvantage of this approach is that the likelihood $p(c_n = k|\boldsymbol{y}_n)$ is not Gaussian, and the posterior is thus not of Gaussian form either, and has to be approximated somehow to keep computations tractable.

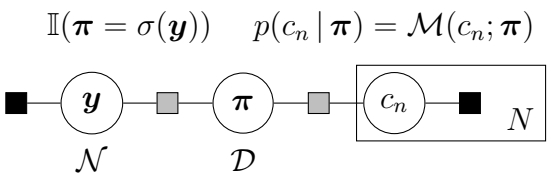


Figure C.1: Factor graph representation of the softmax factor. The parametric approximate beliefs used in the Laplace map on the variables $\boldsymbol{y}$ and $\boldsymbol{\pi}$ are indicated as labels underneath the graph.

A problem with both these approaches is that they presume the data c_n be available as actual observations. During inference in graphical models, we often have access to only a probabilistic belief over the values of c_i. This chapter introduces a computationally lightweight approximation for this case. The main ideas are as follows.

- ▷ Retain a Dirichlet belief $\mathcal{D}(\boldsymbol{\pi}; \boldsymbol{\alpha})$ over the probabilities $\boldsymbol{\pi}$. Because the Dirichlet is conjugate to the multinomial, this allows exact, fast evaluations of the posterior given (probabilistic) data.

- ▷ To perform generalized regression, approximate the Dirichlet with a Gaussian, using a Laplace approximation. The crucial idea here is that the right basis to do this approximation in is the softmax basis, in which the Dirichlet has full support on $\mathbb{R}^K$ and is much closer to a Gaussian in shape than in the standard basis with support on $[0, 1]^K$.

The idea to use a basis transformation to improve Laplace approximations was first raised by MacKay [1998]. That paper also pointed out the softmax basis as convenient for Gaussian approximations on the Dirichlet, but it does not contain explicit forms for the resulting Gaussian. Since the corresponding derivation is not entirely trivial, it is developed here in some depth.
The remaining parts of this chapter proceed by first introducing the softmax basis form of the Dirichlet (C.1). This basis has a technical issue involving identifiability, which is addressed in Section C.1.1. The actual Laplace approximation is performed in Section C.2, in which we arrive at a one-to-one map between Dirichlets and a strict subset of the parameter space of multivariate Gaussians.
As with all approximations, the results presented in this chapter have their shortcomings, and should not be used without some preliminary thought about the application in question. The point of this particular approximation is that it is computationally lightweight, and that it can be used as a black box providing a numerically stable map between the parameters of a multivariate Gaussian and those of a Dirichlet. None of this changes the fact, though, that Gaussians and Dirichlets are not the same thing. There are two big shortcomings of the Laplace map: The Gaussian has weaker tails than the Dirichlet in the softmax basis (i.e.

the Gaussian resulting from mapping a Dirichlet underestimates the probability of extreme sparse values of $\boldsymbol{\pi}$). In the other direction of the map, the Dirichlet has less parameters than the Gaussian, so almost all of the covariance structure of a multivariate Gaussian is lost in translation (see Section C.1.1 for more details). It would be ill-advised to use this map in cases of high correlation. In such cases, computational cost is probably of lesser concern, and more elaborate inference schemes can be used instead.

C.1 The Dirichlet in the Softmax Basis

[The derivations in this Section are reproduced from [MacKay, 1998] with only minor changes, and are provided here solely for completeness.]

To Dirichlet in its standard basis on the space $[0,1]^K$ has the form

$$\mathcal{D}_\pi(\boldsymbol{\pi};\boldsymbol{\alpha}) = \frac{\Gamma\left(\sum_k^K \alpha_k\right)}{\prod_k^K \Gamma(\alpha_k)} \prod_k^K \pi_k^{\alpha_k - 1} \delta(\mathbf{1}^\mathsf{T}\boldsymbol{\pi} - 1) \equiv Z_\mathcal{D}^{-1} \prod_k^K \pi_k^{\alpha_k - 1} \delta(\mathbf{1}^\mathsf{T}\boldsymbol{\pi} - 1) \quad \text{(C.2)}$$

where Γ denotes the Gamma function (the normalization constant $Z_\mathcal{D}$ is also known as the multinomial Beta function), and the subscript π on the distribution name denotes the basis used. The Dirac δ-distribution ensures that $\boldsymbol{\pi}$ is in fact a discrete probability measure (i.e. that its elements sum to 1). The vector $\mathbf{1}$ is the one-vector $\mathbf{1} = [1, 1, 1, \dots]$.

C.1.1 Ensuring Identifiability

We can remove the notational complication of the Dirac distribution in Equation (C.2) by defining the function on a $K-1$ subspace ensuring the requirement $\mathbf{1}^\mathsf{T}\boldsymbol{\pi} = 1$. In the subspace, we can define a new parameter vector $\boldsymbol{\varrho}$ with

$$\pi_k = \begin{cases} \varrho_k & \text{if } k = 1, 2, \dots, K-1 \\ 1 - \sum_{k=1}^{K-1} \varrho_k & \text{if } k = K \end{cases} \quad \text{(C.3)}$$

The situation in the softmax space is similar: $\boldsymbol{\sigma}^{-1}$ only fixes $\boldsymbol{y}$ up to additive constants: For a given $\boldsymbol{y}$, all $\boldsymbol{y}'$ satisfying $\boldsymbol{y}' = \boldsymbol{y} + \xi\mathbf{1}$ share the same value of $\boldsymbol{\sigma}(\boldsymbol{y}')$ for any $\xi \in \mathbb{R}$. Since we are attempting to match the Dirichlet distribution in $\mathbb{R}^K$ with another distribution (a Gaussian), we need to ensure that this distribution is proper, which requires us to introduce a further restriction to solve this ambiguity. For this purpose, we are free to choose any restriction r of the functional form

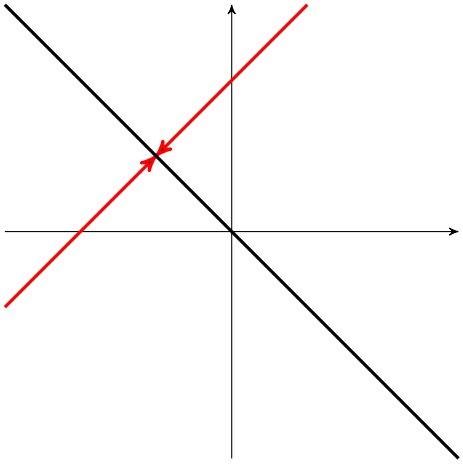


Figure C.2: conceptual sketch of the effect of the restriction r. All points on the red line share the same values of $\boldsymbol{\sigma}(\boldsymbol{y})$ and are projected "softly" (i.e. with precision τ) onto the black hyperplane by r.

$r(\mathbf{1}^\mathsf{T}\boldsymbol{y})$. Following MacKay once more, we use the squared exponential

$$r = \exp\left[-\frac{\tau}{2}(\mathbf{1}^\mathsf{T}\boldsymbol{y})^2\right] \tag{C.4}$$

One way to get an intuition for what this amounts to is to interpret this restriction as an unnormalized Gaussian "message" of precision τ on the deviation of the sum of the elements of $\boldsymbol{y}$ from 0. Another is to see r as a soft projection of the subspaces forming lines parallel to $\mathbf{1}$ onto their intersection with the hyperplane defined by $\mathbf{1}^\mathsf{T}\boldsymbol{y} = 0$. Figure C.1.1 contains a sketch of this operation.

In the special case of $\tau \to \infty$, where the constraint becomes a Dirac distribution, we can use an analogous re-formulation of the parameter space, using $K-1$ parameters $\boldsymbol{a}$ defined through

$$y_k = \begin{cases} a_k & \text{if } k = 1, 2, \ldots, K-1 \\ -\sum_{k=1}^{K-1} a_k & \text{if } k = K \end{cases} \tag{C.5}$$

C.1.2 Transforming to the Softmax Basis

We first consider the special case of the hard constraint $\tau \to \infty$. For probability functions in general, changes of basis correspond to changes of the measure, mediated by the Jacobian matrix $\boldsymbol{J}$:

$$p_u(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u} = p_g(\boldsymbol{g}(\boldsymbol{u}))\,\mathrm{d}\boldsymbol{g} \quad \Rightarrow \quad p_u(\boldsymbol{u}) = p_g(\boldsymbol{g}(\boldsymbol{u}))\,|\det \boldsymbol{J}| \quad \text{with} \quad J_{k\ell} = \frac{\partial g_k}{\partial u_\ell} \tag{C.6}$$

[see e.g. Bishop, 2006, Section 1.2.1]. For the softmax map in particular, we can write the Jacobian for the density over $\boldsymbol{\varrho}$ as a function of the density over $\boldsymbol{a}$ in

terms of the original parameters $\boldsymbol{\pi}$ and $\boldsymbol{y}$:

$$J_{k\ell} = \frac{\partial \varrho_k}{\partial a_\ell} = \sum_{h=1}^{K} \frac{\partial \pi_k}{\partial y_h} \frac{\partial y_h}{\partial a_\ell} = \delta_{k\ell}\pi_k - \pi_\ell \pi_k + \pi_k \pi_K = \pi_k(\delta_{k\ell} - (\pi_k - \pi_K)) \quad \text{(C.7)}$$

Now we define the $K-1$ dimensional vector $\boldsymbol{\pi}_k^+ \equiv \pi_k - \pi_K$, which gives

$$\det \boldsymbol{J} = \det \left[\mathbf{I} - \mathbf{1}\boldsymbol{\pi}^{+\mathsf{T}} \right] \prod_{k=1}^{K-1} \pi_k \quad \text{(C.8)}$$

We can use the matrix determinant lemma $\det[\boldsymbol{A} - \boldsymbol{x}\boldsymbol{y}^\mathsf{T}] = (1 - \boldsymbol{x}\boldsymbol{A}^{-1}\boldsymbol{y}) \det(\boldsymbol{A})$ (see e.g. [Roweis, 1999]) to get

$$\det \boldsymbol{J} = (1 - \mathbf{1}\boldsymbol{\pi}^+) \prod_{k=1}^{K-1} \pi_k = \pi_K \prod_{k=1}^{K-1} \pi_k = \prod_{k=1}^{K} \pi_k \quad \text{(C.9)}$$

Hence, the Dirichlet in the softmax basis takes the form

$$\mathcal{D}_y(\boldsymbol{\pi}(\boldsymbol{y}); \boldsymbol{\alpha}) = \frac{\Gamma\left(\sum_k^K \alpha_k\right)}{\prod_k^K \Gamma(\alpha_k)} \prod_k^K \pi_k^{\alpha_k} \delta(1 - \mathbf{1}\boldsymbol{\pi}) \quad \text{(C.10)}$$

This form is in fact also correct if we relax the hard Dirac constraint back to the soft constraint of Equation (C.4), because integrating over that distribution does not change the value of $\boldsymbol{\pi}$.

David MacKay [1998] points out several interesting aspects of this representation. For example, the mean of this distribution, at $\boldsymbol{\pi}(\boldsymbol{y}) = \boldsymbol{\alpha}/\|\boldsymbol{\alpha}\|$, now falls together with its mode, which is not the case in the standard basis. As we got rid of the -1 terms in the exponent, the distribution now also does not diverge any more for $\alpha_k < 1$, but is much more well behaved as a function of $\boldsymbol{y}$, approaching zero for all large values of $\boldsymbol{y}$.

C.2 The Laplace Map

The idea of the Laplace approximation (see also Section 2.3.4) is to approximate a distribution $p(\boldsymbol{x})$ with a Gaussian distribution $q(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ by setting the mean $\boldsymbol{\mu}$ of q to the location of the mode of p, and the covariance matrix $\boldsymbol{\Sigma}$ of q to the inverse of the Hessian of $\log p$ at its mode (because the logarithm is a monotonic function, a mode of p is also a mode of $\log p$). This is often considered a weak approximation, mainly for the following three reasons

- ▷ Modes are local features and do not necessarily represent the overall "location" of p well at all.

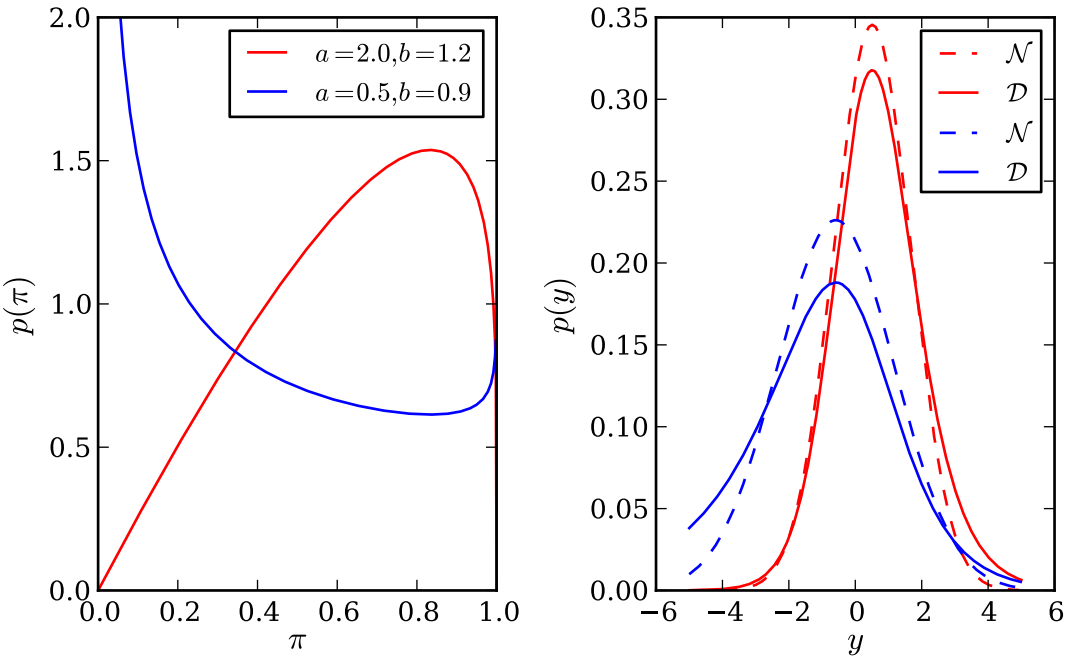


Figure C.3: Effect of the softmax basis change on the Dirichlet distribution. **Left:** Two Beta distributions (i.e. 2-dimensional Dirichlet distributions) for parameter settings of a and b. Note that a naïve Laplace approximation to these distributions with a Gaussian in this basis would give a bad match (The second distribution does not even have a proper mode). **Right:** solid lines: the same distributions in the softmax basis, with Gaussian approximations from the Laplace map as dashed lines. Note that there is now a well defined mode. While the distributions are not symmetric, the Gaussian approximations do provide meaningful approximations. Also note that the Gaussian approximations have weaker tails than the Dirichlets.

- ▷ This approximation evidently misses any multimodality that might be present in p.

- ▷ since the Gaussian has relatively weak tails, the Laplace approximation often underestimates the spread of p.

In the case in question here, however, the first two points are moot, because we have just found a representation in which mean and (unique) mode fall together. The softmax basis representation also makes the last point less severe, as we will see soon.

C.2.1 Mode and Hessian in the Softmax Basis

We have already established that the mode of the Dirichlet in the softmax basis lies at a value of y satisfying $\boldsymbol{\sigma}(\boldsymbol{y}) = \boldsymbol{\alpha}/\|\boldsymbol{\alpha}\|$. The remaining degree of freedom is fixed by the restriction introduced in Section C.1.1, which requires the elements of $\boldsymbol{y}$ to sum to zero. So we set the mean $\boldsymbol{\mu}$ of the Gaussian approximation to the

Dirichlet to

$$\mu_k = \log \alpha_k - \frac{1}{K} \sum_{\ell}^{K} \log \alpha_\ell, \tag{C.11}$$

which is evidently the only setting fulfilling both requirements. The logarithm of the Dirichlet with the soft identifiability constraint is, up to additive constants,

$$\log p_y(\boldsymbol{y} \,|\, \alpha) \triangleq \sum_k \alpha_k \pi_k - \frac{\tau}{2} \mathbf{1}^\mathsf{T} \boldsymbol{y}. \tag{C.12}$$

Plugging in the definition $\pi_k = \sigma_k(\boldsymbol{y})$ we find, after some simple algebra, the elements of the Hessian $\boldsymbol{L}$

$$L_{k\ell} = \hat{\alpha} \left(\delta_{k\ell} \hat{\pi}_k - \hat{\pi}_k \hat{\pi}_\ell \right) + \tau (\mathbf{1}\mathbf{1}^\mathsf{T})_{k\ell}, \tag{C.13}$$

where we have introduced the shorthands $\hat{\alpha} \equiv \sum_k \alpha_k$ and $\hat{\pi}_k = \alpha_k / \hat{\alpha}$ for the value of π at the mode. The term $(\mathbf{1}\mathbf{1}^\mathsf{T})_{k\ell}$ is a convoluted way of writing a 1, which will make the subsequent algebra easier to parse.

C.2.2 A Sparse Representation

As mentioned in the introduction to this chapter, the previous sections of this chapter were largely based on the work by MacKay [1998]. The following section contains a novel but straightforward extension.

In principle, the previous section provided everything necessary to construct a Gaussian approximation to the Dirichlet. However, in most cases where we might be tempted to use this approximation, we will be hard pressed to save computation time. Often, we will want to discard the correlation present in $\boldsymbol{L}$ and treat the resulting Gaussian as a direct product of independent univariate Gaussians. Noting that the elements of the Dirichlet are "almost" independent (independent up to normalization) to begin with, this is not even too bad an approximation. But it would not be a good idea to just throw away the off-diagonal elements of $\boldsymbol{L}$ and use the diagonal elements as the precisions of independent Gaussians, because $\boldsymbol{L}$'s diagonal elements each only depend on one single element of $\boldsymbol{\alpha}$ and do not capture the correlation introduced by normalization.

Instead, this section will derive the analytic inverse of $\boldsymbol{L}$. The diagonal elements of the resulting covariance matrix $\boldsymbol{\Sigma}$ will provide a better starting point for a sparse approximation. To construct this inverse, we introduce the rectangular matrix

$\boldsymbol{X} \in \mathbb{R}^{K\times 2}$ with elements

$$X_{ku} = \hat{\pi}_k \delta_{1u} + \mathbf{1}_k \delta_{2u} = \begin{pmatrix} \hat{\pi}_1 & 1 \\ \hat{\pi}_2 & 1 \\ \vdots & \vdots \\ \hat{\pi}_K & 1 \end{pmatrix} \tag{C.14}$$

and the square matrices $\boldsymbol{A} \in \mathbb{R}^{K\times K}$ and $\boldsymbol{B} \in \mathbb{R}^{2\times 2}$ with

$$\boldsymbol{A} = \operatorname{diag}(\boldsymbol{\alpha}) \qquad \text{and} \qquad \boldsymbol{B} = \begin{pmatrix} -\hat{\alpha} & 0 \\ 0 & \tau \end{pmatrix} \tag{C.15}$$

which allows us to write

$$\boldsymbol{L} = \boldsymbol{A} + \boldsymbol{X}\boldsymbol{B}\boldsymbol{X}^\mathsf{T} \tag{C.16}$$

Both $\boldsymbol{A}$ and $\boldsymbol{B}$ are diagonal with strictly positive diagonal elements, and thus invertible. Hence we can use the well-known matrix inversion lemma[1], which states

$$\left(\boldsymbol{A} + \boldsymbol{X}\boldsymbol{B}\boldsymbol{X}^\mathsf{T}\right)^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{X}\left(\boldsymbol{B}^{-1} + \boldsymbol{X}^\mathsf{T}\boldsymbol{A}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\mathsf{T}\boldsymbol{A}^{-1} \tag{C.17}$$

The 2×2 expression in brackets, known as the *Schur complement*, is, using the summation convention [Einstein, 1916]

$$\begin{aligned} \left(\boldsymbol{B}^{-1} + \boldsymbol{X}^\mathsf{T}\boldsymbol{A}^{-1}\boldsymbol{X}\right)_{ij} &= \boldsymbol{B}^{-1}_{ij} + (\frac{\alpha_k}{\hat{\alpha}}\delta_{i1} + n_k\delta_{i2})\frac{1}{\alpha_k}\delta_{k\ell}(\frac{\alpha_\ell}{\hat{\alpha}}\delta_{j1} + n_\ell\delta_{j2}) \\ &= \boldsymbol{B}^{-1}_{ij} + \frac{1}{\hat{\alpha}}\delta_{i1}\delta_{j1} + \frac{D}{\hat{\alpha}}(\delta_{i1}\delta_{j2} + \delta_{i2}\delta_{j1}) + \delta_{i2}\delta_{j2}\sum_k \frac{1}{\alpha_k} \\ \boldsymbol{B}^{-1} + \boldsymbol{X}^\mathsf{T}\boldsymbol{A}^{-1}\boldsymbol{X} &= \begin{pmatrix} 0 & K/\hat{\alpha} \\ K/\hat{\alpha} & \tau^{-1} + \sum_k \alpha_k^{-1} \end{pmatrix} \end{aligned} \tag{C.18}$$

The inverse of a 2×2 matrix is

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \tag{C.19}$$

so we get for the inverse of the Schur complement (which exists in particular for

[1]This lemma is also known as the Woodbury identity, based on a technical report by that author [Woodbury, 1950]. Hager [1989] points out that the formula itself showed up in several earlier papers, but it appears Woodbury was the first to study it in detail. The special case where the Schur complement is one-dimensional is also known as the Sherman-Morrison formula, although it seems it was actually introduced by Bartlett [1951].

all α, τ with $\alpha_k > 0 \; \forall k$ and $\tau > 0$)

$$\left(\boldsymbol{B}^{-1} + \boldsymbol{X}^\mathsf{T}\boldsymbol{A}^{-1}\boldsymbol{X}\right)^{-1} = \begin{pmatrix} -\frac{\hat{\alpha}^2}{K}\left(\frac{1}{\tau} + \sum_k \frac{1}{\alpha_k}\right) & \frac{\hat{\alpha}}{K} \\ \frac{\hat{\alpha}}{K} & 0 \end{pmatrix} \tag{C.20}$$

With some simple algebra similar to Equation (C.18), we project back to $\mathbb{R}^{K\times K}$ and get the final value for the inverse of the Hessian

$$L^{-1}_{k\ell} = \delta_{k\ell}\frac{1}{\alpha_k} - \frac{1}{K}\left[\frac{1}{\alpha_k} + \frac{1}{\alpha_\ell} - \frac{1}{K}\left(\frac{1}{\tau} + \sum_u^K \frac{1}{\alpha_u}\right)\right] \tag{C.21}$$

because the inverse is defined for all positive values of τ, we can now safely take the limit of $\tau \to \infty$, which hardens the constraint on the subspace $\mathbf{1}^\mathsf{T}\boldsymbol{y}$. Then, the diagonal elements of $\boldsymbol{\Sigma} = \boldsymbol{L}^{-1}$, which we are interested in, are

$$\Sigma_{kk} = \frac{1}{\alpha_k}\left(1 - \frac{2}{K}\right) + \frac{1}{K^2}\sum_\ell^K \frac{1}{\alpha_\ell}. \tag{C.22}$$

Looking at this form, we firstly note that each diagonal element of $\boldsymbol{\Sigma}$ now depends on all the elements of α, which is reassuring. The term $1-2/K$ might look worrying at first, but it turns out that the formula is also meaningful for $K = 2$ (and in fact even the trivial $K = 1$). For this bivariate situation, the variance on both dimensions becomes identical, which is expected as this case corresponds to a Laplace approximation on a Beta distribution. For more on this special case, see Section C.3.

C.2.3 Inverse Map

Having established a map from the parameter space of Dirichlets into a subset of the parameters of the Gaussian exponential family, we would now like to invert this map, to arrive at a rule mapping sets of K independent scalar Gaussians to the parameters of a K-dimensional Dirichlet. To do so, we first transform Equation (C.11) to

$$\alpha_k = e^{\mu_k}\prod_\ell^K \alpha_\ell^{1/K} \tag{C.23}$$

inserting this form for α_k into Equation (C.22) and re-arranging gives

$$\prod_\ell^K \alpha_\ell^{1/K} = \frac{1}{\Sigma_{kk}}\left[e^{-\mu_k}\left(1 - \frac{2}{K}\right) + \frac{1}{K^2}\sum_u^K e^{-\mu_u}\right] \qquad \forall k \in 1, \ldots, K \tag{C.24}$$

which we can re-insert into Equation (C.23) to get

$$\alpha_k = \frac{1}{\Sigma_{kk}} \left(1 - \frac{2}{K} + \frac{e^{-\mu_k}}{K^2} \sum_{\ell}^{K} e^{-\mu_\ell} \right) \qquad \text{for } k = 1, \ldots, K \tag{C.25}$$

The results of the preceding sections will be summed up in section C.4. Before that, though, it is instructive to take a closer look at the special case of $K = 2$.

C.3 The Two-Dimensional Case

The bivariate version of the Dirichlet distribution is the Beta distribution. Because the probability π_1 of label 1 completely determines $\pi_2 = 1 - \pi_1$, it is possible to write it as a function of only one variable π. Its two parameters are often denoted a and b, rather than α_1 and α_2:

$$\mathcal{D}_{\pi}^{(2)}(\pi; a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \pi^{a-1}(1-\pi)^{b-1} \tag{C.26}$$

The same fact also simplifies the mapping to a subset of $\mathbb{R}^2$. Because the subspace of $\boldsymbol{y} \in \mathbb{R}^2$ which satisfies $\mathbf{1}^\mathsf{T}\boldsymbol{y} = 0$ is isomorphic to $\mathbb{R}$ itself, we can save the regularization work of Section C.1.1 and use the one-dimensional version of the softmax as the link function. This function is known as the logistic:

$$\sigma_1(y) = \frac{\exp(y)}{1 + \exp(y)} \tag{C.27}$$

As the one-dimensional case of Equation (C.1), the logistic has the Jacobian (see Equation (C.7))

$$\frac{\mathrm{d}\sigma_1(y)}{\mathrm{d}y} = \sigma_1(y)(1 - \sigma_1(y)) \tag{C.28}$$

so the analysis of Equations (C.10) and following carries through. It is then not difficult to see that this distribution, in the logistic basis, has its mode at

$$\mu = \log\left(\frac{a}{b}\right) \tag{C.29}$$

and the Hessian (i.e. the second derivative) of its logarithm at this point is

$$\sigma^2 = \frac{a+b}{ab} \tag{C.30}$$

This map can be easily inverted to give

$$a = \frac{\exp(\mu) + 1}{\sigma^2} \qquad \text{and} \qquad b = \frac{\exp(-\mu) + 1}{\sigma^2} \tag{C.31}$$

C.4 Summary

This appendix developed a Laplace approximation to the Dirichlet distribution using the softmax basis. Introducing a regularizing distribution and inverting the Hessian analytically, we arrived at a K-variate Gaussian approximation to the Dirichlet with a simple covariance structure, which can be described by K parameters, and is fully determined through the diagonal elements of the covariance matrix. The maps between the Dirichlet parameters $\boldsymbol{\alpha}$ and the Gaussian parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is

$$\mu_k = \log \alpha_k - \frac{1}{K} \sum_{\ell=1}^{K} \log \alpha_\ell \tag{C.32}$$

$$\Sigma_{k\ell} = \delta_{k\ell} \frac{1}{\alpha_k} - \frac{1}{K} \left[\frac{1}{\alpha_k} + \frac{1}{\alpha_\ell} - \frac{1}{K} \sum_{u=1}^{K} \frac{1}{\alpha_u} \right] \tag{C.33}$$

$$\text{and } \alpha_k = \frac{1}{\Sigma_{kk}} \left(1 - \frac{2}{K} + \frac{e^{\mu_k}}{K^2} \sum_{\ell}^{K} e^{-\mu_\ell} \right) \qquad \text{for } k = 1, \ldots, K \tag{C.34}$$

Note that, if $K \gg 1$ and $\alpha_k \gg 0 \; \forall k$, the covariance matrix approaches a diagonal $\boldsymbol{\Sigma} = \text{diag}\left[(\alpha_k)^{-1}\right]$. In situations like this, where the dimensionality is high and the Dirichlet is not sparse, the output of the Laplace map can thus be treated as a set of K approximately independent Gaussians.

Computationally lightweight approximate regression algorithms often return independent Gaussian beliefs over covariates. Using the inverse of the map, these can be used to construct approximate Dirichlet beliefs.

Some experimental evaluations of the quality of such approximations for evidence estimation can be found in the previously cited paper by MacKay [1998]. Chapter 5 contains experimental evaluations of the quality of this approximation when used in combination with a factorized regression algorithm.

Bibliography

M. Abramowitz and I.A. Stegun, editors. *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*. Dover, 1972. 89

R.K. Ahuja and J.B. Orlin. Inverse optimization. *Operations Research*, pages 771–783, 2001. 113

D.G. Altman. Statistics in medical journals. *Statistics in Medicine*, 1(1):59–71, 1982. 4

D.G. Altman. The scandal of poor medical research. *British Medical Journal*, 308 (6924):283, 1994. 4

S. Amari. *Differential-geometrical methods in statistics*. Springer, 1985. 24

S. Arima. Peabody picture vocabulary test-revised data: a Bayesian approach to item response theory. Master's thesis, Universiteit Hasselt, Belgium, 2006. 61

J.R. Ashford and R.R. Sowden. Multi-variate probit analysis. *Biometrics*, pages 535–546, 1970. 66

M.R. Barrick and M.K. Mount. The Big Five personality dimensions and job performance: a meta-analysis. *Personnel Psychology*, 44, 1991. 62

M.S. Bartlett. An inverse matrix adjustment arising in discriminant analysis. *The Annals of Mathematical Statistics*, 22(1):107–111, 1951. 142

E.B. Baum and W.D. Smith. A Bayesian approach to relevance in game playing. *Artificial Intelligence*, 97(1-2):195 – 242, 1997. 39

T. Bayes. On a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society*, 1763. 2

M.J. Beal, Z. Ghahramani, and C.E. Rasmussen. The infinite hidden Markov model. *Advances in neural information processing systems*, 1:577–584, 2002. 4

R.E. Bellman. *Dynamic Programming*. Princeton University Press, 1957. 7

J. Bennett and S. Lanning. The Netflix prize. In *Proceedings of KDD Cup and Workshop*, 2007. 31, 62

C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 3, 9, 10, 31, 33, 34, 86, 92, 124, 138

D. Blei, T.L. Griffiths, M.I. Jordan, and J.B. Tenenbaum. Hierarchical topic models and the nested Chinese restaurant process. *Advances in neural information processing systems*, 16:106, 2004. 4

D.M. Blei and J.D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 113–120, 2006. 103

D.M. Blei and J.D. Lafferty. A correlated topic model of Science. *Annals of Statistics*, 1(1):17–35, 2007. 84

D.M. Blei and J.D. Lafferty. Topic models. In A. Srivastava and M. Sahami, editors, *Text Mining: Theory and Application*. Taylor & Francis, 2009. 80, 99

D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. 80, 84, 85

D. Burton and P.L. Toint. On an instance of the inverse shortest paths problem. *Mathematical Programming*, 53(1):45–61, 1992. 113

M. Campbell, A.J. Hoane, and F.H. Hsu. Deep Blue. *Artificial Intelligence*, 134 (1-2):57–83, 2002. 38, 42

R.B. Cattell. The description of personality: basic traits resolved into clusters. *Journal of Abnormal and Social Psychology*, 38(4):476–506, 1943. 62

J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, and D.M. Blei. Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems*, 2009. 99

C.E. Clark. The greatest of a finite set of random variables. *Operations Research*, 9(2):145–162, 1961. 114, 122, 125

D.D. Cox. An analysis of bayesian inference for nonparametric regression. *Annals of Statistics*, 21(2):903–923, 1993. 3

R.T. Cox. Probability, frequency and reasonable expectation. *American Journal of Physics*, 14(1):1–13, 1946. 1, 2

R. Dearden, N. Friedman, and S. Russell. Bayesian Q-Learning. In *Proceedings of the Thirteenth AAAI Conference on Artificial Intelligence*, pages 761–768, 1998. 49, 113

S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990. 80

J.M. Digman. Personality structure: Emergence of the five-factor model. *Annual Review of Psychology*, 41(1):417–440, 1990. 62

M.O.G. Duff. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. PhD thesis, U of Massachusetts, Amherst, 2002. 39

A. Einstein. Die Grundlage der allgemeinen Relativitätstheorie. *Annalen der Physik*, 354(7):770–822, 1916. 21, 142

B.J. Frey. *Graphical Models for Machine Learning and Digital Communication*. MIT Press, 1998. 5, 9

B.J. Frey and D.J.C. MacKay. A revolution: Belief propagation in graphs with cycles. In *Advances in Neural Information Processing Systems*, 1998. 5, 9, 67

B.J. Frey, F.R. Kschischang, H.-A. Loeliger, and N. Wiberg. Factor graphs and algorithms. In *Proc. 35th Allerton Conf. on Communications, Control, and Computing*, pages 666–680, 1997. 9

S. Gelly and D. Silver. Achieving master level play in 9 x 9 computer Go. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 1537–1540, 2008. 38

S. Gelly, Y. Wang, R. Munos, and O. Teytaud. Modification of UCT with patterns in Monte-Carlo Go. Research Report RR-6062, INRIA, 2006. 51

S. Ghosal, J.K. Ghosh, and A.W. van der Vaart. Convergence rates of posterior distributions. *Ann. Statistics*, 28(2):500–531, 2000. 3

W.R. Gilks and P. Wild. Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, 41(2):337–348, 1992. 34

G.H. Golub and C.F. Van Loan. *Matrix computations*. Johns Hopkins Univ Pr, 1996. 14

T.L. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. *Advances in neural information processing systems*, 18:475, 2006. 4

T.L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl 1):5228, 2004. 80

W.W. Hager. Updating the inverse of a matrix. *SIAM Review*, 31(2):221–239, 1989. 142

R.K. Hambleton and H. Swaminathan. *Item Response Theory*. Kluwer-Nijhoff, 1990. 60

P. Hennig. Expectation propagation on the maximum of correlated normal variables. `arXiv:0910.0115v1 [stat:ML]`, October 2009. URL `http://arxiv.org/abs/0910.0115`. 113

P. Hennig, D. Stern, and T. Graepel. Coherent inference on optimal play in game trees. *Journal of Machine Learning Research, W&CP*, 9, 2010. 37

R. Herbrich, T.P. Minka, and T. Graepel. TrueSkill™: A Bayesian skill rating system. *Advances in Neural Information Processing Systems*, 20:569–576, 2007. 44, 68

C. Heuberger. Inverse combinatorial optimization: A survey on problems, methods, and results. *Journal of Combinatorial Optimization*, 8(3):329–361, 2004. 113

T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, 1999. 80

J. Huguenin, F. Pelgrin, and A. Holly. Estimation of multivariate probit models by exact maximum likelihood. Technical report, University of Lausanne, Institute of Health Economics and Management (IEMS), 2009. 66

D. Hume. *An Enquiry Concerning Human Understanding*. original publisher uknown. Currently under publication, for example, by Oxford Univ. Press, 1739. 5

I.A. Ibragimov and R.Z. Has'minskii. *Statistical Estimation: Asymptotic Theory*. Springer, New York, 1981. 3

A.J. Jaffe and H.F. Spirer. *Misused Statistics: Straight Talk for Twisted Numbers*. Marcel Dekker, New York, 1987. 4

E.T. Jaynes and G.L. Bretthorst. *Probability Theory: the Logic of Science*. Cambridge University Press, 2003. 2, 3

J.L.W.V. Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1):175–193, 1906. 26

R. Kleinberg, A. Slivkins, and E. Upfal. Multi-armed bandits in metric spaces. In *Proc. ACM Symposium on Theory of Computing*, pages 681–690, 2008. 39

L. Kocsis and C. Szepesvári. Bandit based Monte-Carlo planning. In *Proceedings of the European Conference on Machine Learning*, pages 282–293. Springer, 2006. 38, 49, 53

A.N. Kolmogorov. Grundbegriffe der Wahrscheinlichkeitsrechnung. *Ergebnisse der Mathematik und ihrer Grenzgebiete*, 2, 1933. 2

J.Z. Kolter and A.Y. Ng. Near-Bayesian exploration in polynomial time. In *Proceedings of the 26th International Conference on Machine Learning*. Morgan Kaufmann, 2009. 49

F.R. Kschischang, B.J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, February 2001. 10

S. Kullback and R.A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951. 20

P.S. Laplace. Mémoire sur la probabilité des causes par les évènemens. *Mémoires de mathématique et de physique presentés à l'Académie royale des sciences, par divers savans, et lûs dans ses assemblées*, 6:621–656, 1774. 2

S.L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *J. R. Stat. Soc.*, 50:157–224, 1988. 5, 9, 13

L.M. Le Cam. Convergence of estimates under dimensionality restrictions. *Ann. Statistics*, 1:38–53, 1973. 3

R. Likert. A technique for the measurement of attitudes. *Archives of Psychology. Vol*, 22(140):55, 1932. 60

F. Lord. Statistical inferences about true scores. *Psychometrika*, 24:1–17, 1959. 63

F.M. Lord. Maximum likelihood and Bayesian parameter estimation in item response theory. *Journal of Educational Measurement*, 23(2):157–162, 1986. 61

D.J.C. MacKay. Choice of basis for Laplace approximation. *Machine Learning*, 33 (1):77–86, 1998. 82, 90, 96, 136, 137, 139, 141, 145

D.J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge Univ Press, 2003. 3, 26, 30, 34, 46

D.J.C. MacKay and L.C. Bauman-Peto. A hierarchical Dirichlet language model. *Natural Language Engineering*, 1(03):289–308, 1995. 80

P.S. Maybeck. *Stochastic Models, Estimation and Control*, chapter 12.7. Academic Press, 1982. 32

W. McDougall. Of the word character and personality. *Character personality*, 1: 3–16, 1932. 62

D. Mimno and A. McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *Uncertainty in Artificial Intelligence*, 2008. 80, 82, 84

T. Minka. Divergence measures and message passing. Technical Report MSR-TR-2005-173, Microsoft Research Technical Report, 2005. 24

T.P. Minka. Expectation Propagation for approximate Bayesian inference. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 362–369, San Francisco, CA, USA, 2001. Morgan Kaufmann. ISBN 1-55860-800-1. 5, 9, 20, 21, 66, 67

T.P. Minka. Power EP. Technical report, Dep. Statistics, Carnegie Mellon University, Pittsburgh, PA, 2004. URL `http://research.microsoft.com/en-us/um/people/minka/papers/`. 23, 24

T.P. Minka. EP: A quick reference. Technical report, Microsoft Research, 2008. URL `http://research.microsoft.com/en-us/um/people/minka/papers/`. 68, 129

T.P. Minka and J. Winn. *infer.NET*. Microsoft Research Cambridge, 2008. URL `http://research.microsoft.com/en-us/um/cambridge/projects/infernet/`. 8, 123

E. Muraki. A generalized partial credit model. *Applied Psychological Measurement*, 2, 1992. 64

I. Murray. *Advances in Markov Chain Monte Carlo Methods*. PhD thesis, University College London, 2007. 30

I. Murray, R.P. Adams, and D.J.C. MacKay. Elliptical slice sampling. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 2010. 97

R.M. Neal. *Bayesian learning for neural networks*. Springer Verlag, 1996. 4

G. Neumann-Denzau and J. Behrens. Inversion of seismic data using tomographical reconstruction techniques for investigations of laterally inhomogeneous media. *Geophysical Journal International*, 79(1):305–315, 1984. 113

N.J. Nilsson. *Problem-Solving Methods in Artificial Intelligence.* McGraw-Hill Pub. Co., 1971. 38

M.R. Novick. The axioms and principal results of classical test theory. *Journal of Mathematical Psychology*, 3(1):1–18, 1966. 60, 63

M. Opper. On-line versus off-line learning from random examples: General results. *Phys. Rev. Lett.*, 77(22):4671–4674, Nov 1996. 32, 82

P. Orbanz. Construction of nonparametric Bayesian models from parametric Bayes equations. In *Adv. in Neural Information Processing Systems*, 2009. 19

A.J. Palay. *Searching with probabilities.* Pitman Publishing, Inc., Marshfield, MA, USA, 1985. 39

R.J. Patz and B.W. Junker. A straightforward approach to Markov chain Monte Carlo methods for item response models. *Journal of Educational and Behavioral Statistics*, 24(2):146, 1999. 61

J. Pearl. *Heuristics. Intelligent search strategies for computer problem solving.* Addison-Wesley, 1985. 38

J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, 1988. 5, 9, 32

K. Popper. *Logik der Forschung.* Mohr Siebeck, 1934. 5

G. Rasch. Probabilistic Models for Some Intelligence and Attainment Tests, 1960. 64

C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, 2006. 116

T. Richardson and R.L. Urbanke. *Modern coding theory.* Cambridge University Press, 2008. 46

J.M. Robson. The complexity of Go. *Proc. Information Processing*, 83:413–417, 1983. 38

S. Roweis. Matrix identities. Technical report, University of Toronto, 1999. 139

D.M. Roy and Y.W. Teh. The Mondrian process. In *Adv. in neural information processing systems*, volume 21, 2009. 4

S. J. Russell and E. Wefald. *Do the Right Thing*. MIT Press, 1991. 39

G. Salton and M.J. McGill. *Introduction to modern information retrieval*. McGraw-Hill New York, 1983. 80, 99

C.E. Shannon. Programming a computer for playing chess. *Philosophical Magazine (Series 7)*, 41(314):256–275, 1950. 38

D.H. Stern, R. Herbrich, and T. Graepel. Learning to solve game trees. In *Proceedings of the 24th International Conference on Machine Learning*, pages 839–846, New York, NY, USA, 2007. ACM. 39

D.H. Stern, R. Herbrich, and T. Graepel. Matchbox: large scale online bayesian recommendations. In *18th International Conference on the World Wide Web*, pages 111–120. ACM New York, NY, USA, 2009. 68

Y.W. Teh, D. Newman, and M. Welling. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. *Advances in neural information processing systems*, 19:1353, 2007. 80, 82, 84, 88, 96

G. Tesauro, V.T. Rajan, and R. Segal. Bayesian inference in Monte Carlo tree search. In *Uncertainty in Artificial Intelligence*, July 2010. 56

W.R. Thompson. On the likelihood that one unknown probability exceeds another in view of two samples. *Biometrika*, 25:275–294, 1933. 49

R. von Mises. *Mathematical Theory of Probability and Statistics*. Academic Press New York, 1964. 116

S. Walker. New approaches to Bayesian consistency. *The Annals of Statistics*, 32 (5):2028–2043, 2004. 2

X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 424–433, 2006. 103

C.J.C.H. Watkins and P. Dayan. Technical note: Q-learning. *Machine Learning*, 8:279 – 292, 1992. 39

N. Wiener. *Cybernetics or Control and Communication in the Animal and the Machine*. MIT Press, 1948. 38

P. Wild and WR Gilks. Algorithm AS 287: Adaptive rejection sampling from log-concave density functions. *Applied Statistics*, 42(4):701–709, 1993. 34

C.K.I Williams and C.E. Rasmussen. Gaussian processes for regression. In *Advances in neural information processing systems*, 1996. 4

J. Winn and C.M. Bishop. Variational message passing. *Journal of Machine Learning Research*, 6(1):661, 2006. 5, 9, 28

M.A. Woodbury. Inverting modified matrices. Technical Report no. 42, Statistical Research Group, Princeton University, Princeton, N.J., 1950. 142

J. Zhu and E.P. Xing. Conditional topic random fields. In *International Conference on Machine Learning*, 2010. 80, 82

Index