New probabilistic inference algorithms that harness the strengths of variational and Monte Carlo methods

by

Peter Carbonetto

B.Sc., McGill University, 2001
M.Sc., University of British Columbia, 2003

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate Studies

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

August 2009

© Peter Carbonetto 2009

Abstract

The central objective of this thesis is to develop new algorithms for inference in probabilistic graphical models that improve upon the state-of-the-art and lend new insight into the computational nature of probabilistic inference. The four main technical contributions of this thesis, described separately in four chapters, are: 1) a new framework for inference in probabilistic models based on stochastic approximation, variational methods and sequential Monte Carlo is proposed that achieves significant improvements in accuracy and reductions in variance over existing Monte Carlo and variational methods, and at a comparable computational expense, 2) for many instances of the proposed approach to probabilistic inference, constraints must be imposed on the parameters, so I describe a new stochastic approximation algorithm that adopts the methodology of primal-dual interior-point methods and handles constrained optimization problems much more robustly than existing approaches, 3) a new class of *conditionally-specified* variational approximations based on mean field theory is described, which, when combined with sequential Monte Carlo, overcome some of the limitations imposed by conventional variational mean field approximations, and 4) I show how recent advances in variational inference can be used to implement inference and learning in a novel contingently acyclic probabilistic relational model, a model developed for the purpose of making predictions about relationships in a social network.

In addition to these technical contributions, also contained within this thesis are several important *ideas:* 1) interior-point methods can be leveraged for solving canonical problems in machine learning, 2) meaningful connections are drawn between two divergent philosophies on approximate inference, Monte Carlo simulation and variational methods, 3) and through these connections, approximate inference is framed as a trade-off between achieving an estimator with low bias and one with low variance, 4) one can develop new inference algorithms by framing the variational objective with respect to the conditionals of a target distribution, and 5) it is possible to model interdependent relationships (such as friendship) using a directed graphical model just as well as with an undirected graphical model by introducing latent variables that explicitly guarantee acyclicity of the underlying directed graph. Finally, and not least significantly, this thesis is a broad synthesis of probabilistic inference that spans many different scientific and mathematical disciplines.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

> Were it not for number and its nature, nothing that exists would be clear to anybody either in itself or in its relation to other things... You can observe the power of number exercising itself not only in the affairs of demons and gods but in all the acts and the thoughts of men in all handicrafts and music. *Philolaus, 5th century B.C.*

This thesis is about probabilistic inference. More precisely, it is about approximate probabilistic inference. The aims of this thesis are to explore the challenges in making accurate inferences, to further our understanding of these challenges, and to advance the state-of-the-art in probabilistic inference. Achieving these objectives will involve answering the following questions: What is probabilistic inference? Of what value is it? What makes it so difficult? What problems can we solve effectively and, for that matter, what does it mean to have an "effective" solution? Conversely, where are improvements in inference needed? What mathematical tools do we have at our disposal in order to understand the underlying challenges of probabilistic inference? And finally: how can we build upon existing mathematical tools to conceive new and more effective approaches to probabilistic inference?

Let me start with a formula that will be familiar to many readers:

$$p(x \mid e, \theta) = \frac{p(e \mid x, \theta)\, p(x \mid \theta)}{\int p(e \mid x, \theta)\, p(x \mid \theta)\, dx}. \tag{1.1}$$

This is Bayes' rule.[1] Bayes' rule tells us that the *posterior* $p(x \mid e, \theta)$—the distribution over the unknowns x that take into account the evidence e—is proportional to $p(e \mid x, \theta)$, the *likelihood* of observing the evidence e given x and θ, multiplied by $p(x \mid \theta)$, the *prior* probability of x occurring. The denominator in (1.1) ensures that the product represents a probability. It is often called the *marginal density* or the *marginal likelihood*. The posterior captures the model's predictions about the world when provided with measurements collected either through experiment, or through observations made by a human or a robot. For instance, a court judge might be interested in creating a model of a crime scene and predicting whether a particular suspect should be sentenced given witness accounts and evidence gathered by the lawyers.

In the past few decades, Bayes' rule has played an increasingly prominent role in the design and use of scientific models in a diverse array of scientific disciplines: ecology (Clark, 2005; Punt & Hilborn, 1997), information retrieval (Blei et al., 2003; Buntine & Jakulin, 2004; Griffiths & Steyvers, 2004), evolution and genetics (Huelsenbeck et al., 2001; Pritchard et al., 2000a), physics and astronomy (Gregory, 2005), theory of human cognition, perception and sensorimotor control (Körding & Wolpert, 2006; Tenenbaum et al., 2006), political science (King et al., 2001; Park et al., 2004),

[1] Also known less correctly as Bayes' theorem.

robotic navigation (Murphy, 2002), oncogenomics (Shah et al., 2007), decoding parity check codes (Frey & MacKay, 1997; Moon, 2005), economics (Greenberg, 2007; Lancaster, 2004) and public health (Samet et al., 2000), to name but a few. The majority of these scientific developments wouldn't have been possible without advances in methods for probabilistic inference.

Let me offer a couple observations concerning Bayes' rule. Consider a random variable X representing the roll of a die, and $P(X = x)$ is the probability that the die roll X comes up with the number x. From our knowledge of basic probability, the probability that a die roll X comes up with a 5 or a 6 is equal to $P(X=5)+P(X=6)$, while the probability that two dice thrown consecutively show a 5 and then a 6 is $P(X=5) \times P(X=6)$. What Reverend Bayes tells us is that the posterior probability of x is the likelihood that x generated the evidence e *and* the probability that $p(x \mid \theta)$ produced a x, not "or". In other words, Bayes' rule says that if we have information about x coming from two different sources—the likelihood and the prior—we should combine these sources by taking their product. There is no reason why we can't extend this logic to multiple sources of information.

The second observation: the fact that $p(e \mid x, \theta)$ and $p(x \mid \theta)$ represent probabilities is incidental; we could replace them with non-negative functions $f(e \mid x, \theta)$ and $f(x \mid \theta)$ without harming the interpretation of the posterior $p(x \mid e, \theta)$ as a probability. The non-negative functions are often called *potentials*, terminology borrowed from the physics literature. It can be extremely convenient from a computational point of view to treat all terms in the product as potentials, and indeed this is often done by representing the distribution of interest by a product of real-valued functions, functions alternately called *potentials* or *factors*. The theorem of Hammersley and Clifford (Besag, 1974) provides us with the conditions upon which a product of factors defines a valid probability density. Such representations are often called *Markov random fields* (Kindermann & Snell, 1980) or *conditional random fields* (Lafferty et al., 2001). They are used widely in many disciplines, notably computer vision (Carbonetto et al., 2004; Marroquin, 1985; Wang et al., 2006; Winn & Shotton, 2006).

The computational challenges of probabilistic inference stem from the integral in the denominator that appears in the application of Bayes' rule; in order to be able to evaluate the posterior probability of an assignment x, we need to be able to compute the integral. In very special cases, this integral has a closed-form solution and the solution is easy to compute, so we can infer expectations with respect to the posterior exactly. This only happens for simple and uninteresting models when the prior is *conjugate* to the likelihood. In many cases, it is conceivable to compute the integral, but it is deemed *intractable* because the amount of computation required to come up with the answer grows exponentially with the number of variables.[2] And there are many cases in which no one has discovered a closed form solution, meaning that *it is impossible to compute the integral exactly.* When the integral is intractable or has no solution, we need to appeal to approximate methods for probabilistic inference. There

[2]For example, inference in a multiply connected Bayesian network with discrete random variables can be reduced to the problem of counting the number of satisfying assignments of a propositional logic formula, so it belongs to the class of #P-hard problems.

are two traditions of approximate inference: the *variational* approach to inference based on optimization, and the *Monte Carlo* approach based on simulation.[3]

It is precisely this problem—the computation of an integral with no known analytic solution—that Metropolis and Ulam (1949) confronted at Los Alamos in the 1940s when they were trying to simulate the behaviour of nuclear events following the mechanics of the Fokker-Planck equation. Since their statistical-mechanical systems involved a great number of individual interactions, they argued that one should only count those that have a large probability of occurring. Such insights led to the development of the Monte Carlo method.

Today, the two most popular Monte Carlo methods are Markov Chain Monte Carlo (MCMC) and importance sampling (Robert & Casella, 2004). Importance sampling is perhaps the most easily understood of the two. The idea is to come up with an artificial distribution over the unknowns, which I'll denote by $q(x)$, such that it is possible to draw samples from this distribution. The samples are then weighted to account for any discrepancies between the sampling distribution and the target posterior. For instance, suppose we are designing a decoder for a parity check code, and we want to calculate the expectation that the ith bit of the transmitted codeword x is zero given our prior knowledge of codewords, and given the received (noisy) signal e. In other words, we want to estimate the posterior probability $p(x_i = 0 \,|\, e, \theta)$. A Monte Carlo estimate of this probability would be the weighted count

$$p(x_i = 0 \,|\, e, \theta) \approx \frac{1}{n} \sum_{s=1}^{n} w(x^{(s)})\, \delta_0(x_i^{(s)}) \tag{1.2}$$

where n is the number of samples drawn, $w(x)$ is the importance weight of sample x, and $\delta_y(x)$ is a delta-mass function, which is equal to 1 when $x = y$, and 0 otherwise. The accuracy of this estimate will of course depend on the number of samples drawn. It will also depend on the extent to which the sampling distribution $q(x)$ agrees with the posterior distribution; if the shape of the proposal distribution $q(x)$ bears little semblance to $p(x \,|\, e, \theta)$, the importance sampling method will deteriorate in high dimensions, making it practically useless for all but the smallest problems. Likelihood weighting (Fung & Chang, 1989) is an important case in point. Thus, an effective proposal distribution must be tuned to the target posterior. Since its inception, much of the work on importance sampling has focused on devising better sampling distributions by teasing out analytic structure from posteriors via, for instance, dynamic programming and Rao-Blackwellization (Andrieu & Doucet, 2002; Doucet et al., 2000a; Martinez-Cantin et al., 2007).

The other main branch of Monte Carlo methods, MCMC, is founded on some of

[3]There is actually a third approach to probabilistic inference that isn't captured by variational or Monte Carlo methodology—it is based on algorithms for heuristic search such as branch-and-bound (Pearl, 1984). This approach has primarily lead to a formal characterization of the space and time complexity tradeoffs in exact inference for Bayesian networks (Darwiche, 2001; Dechter & Mateescu, 2007; Park & Darwiche, 2003; Peot & Shachter, 1991). This approach should not be confused with the application of Bayesian reasoning to search (Hansson & Mayer, 1989).

the most elegant (but not necessarily easily grasped) theory in all of mathematics. The most important incarnation of MCMC is the method first devised by Metropolis and then later generalized by the Canadian mathematician Hastings. The Metropolis-Hastings method is relatively simple to understand and use, and it subsumes many other popular methods, including Gibbs sampling. Unlike importance sampling, the samples are not drawn independently of one another; the samples form a *Markov chain*. The theory tells us that as long as the chain is *ergodic* and its *invariant distribution* is the posterior in question—in other words, it meets the *detailed balance condition* (Chib & Greenberg, 1995)—then simulating the Markov chain will (eventually) give us the right answer (Meyn & Tweedie, 1993). The beauty of the Metropolis-Hastings method is that it automatically guarantees that the Markov chain will have the proper invariant distribution, and it is ergodic under easily-satisfied conditions. Therefore, you don't have to be an expert in probability theory to use the Metropolis-Hastings algorithm to simulate a Markov chain for your statistical model. Due to their broad applicability and ease of use, nearly every survey of Bayesian methods includes an introduction to MCMC methods.

Now the bad news. We can think of the Markov chain as exploring the space of possible configurations, visiting the regions of high probability most frequently. The fundamental challenge lies in designing a Markov chain that explores the space efficiently; that is, it does not take a lot of iterations to move from one region of high probability to another. The Gibbs sampler, for instance, may be extremely slow to converge if the random variables exhibit strong correlations (Liu et al., 1994; Liu & Wu, 1999). There are also many well-studied problems for which the Gibbs sampler or Metropolis-Hastings method fail miserably on this account, to the point that they often get "stuck" in a region of high probability; see Celeux et al. (2000) and Newman and Barkema (1999). This happens because the low-probability regions act as a barrier that prevents the Markov chain from escaping. This is particularly troubling since it means that a finite Markov chain can get stuck in a local mode, thus failing to produce a representative prediction. Hastings (1970) himself anticipated these problems when he wrote, "even the simplest of numerical methods may yield spurious results if insufficient care is taken in their use, and how difficult it often is to assess the magnitude of the errors." There has been much work on coming up with ways to overcome Hastings' criticism, with mixed success. Larger moves often overcome low-probability barriers and explore the state space in a more effective manner. For example, Hamze and de Freitas (2004) show how to exploit the structure of a sparse, discrete Markov random field to produce a better Gibbs sampler. A well-engineered Metropolis-Hastings algorithm can overcome some the stated drawbacks, but it is not immune to the curse of dimensionality: large moves can have poor acceptance rates, yielding as before, slow convergence of the Markov chain.

A very different class of methods for probabilistic inference called *variational methods*—also with strong ties to physics, most notably in the work of Bethe and Kikuchi—transform the integration problem to an optimization problem. The key idea is to come up with a class of approximating distributions $q(x)$, then optimize some criterion to find the $q(x)$ that most closely matches the posterior $p(x \mid e, \theta)$.

In information theory, the criterion used to measure the distance between $q(x)$ and $p(x \mid e, \theta)$ is called the *Kullback-Leibler divergence*. In statistical physics, they use a very different-looking but equivalent distance measure called the *Gibbs free energy.* The variational approach to probabilistic inference has been thoroughly explored in the past couple of decades. It has been the subject of many recent Ph.D. dissertations, notably Beal (2003), Lawrence (2000), Jaakkola (1997), Minka (2001b), Ravikumar (2007) and Wainwright (2002).

In order to devise a practical variational inference procedure, it is necessary to choose a class of approximating distributions $q(x)$ that have "nice" analytic properties. For instance, the *mean field* class of approximating distributions factorize in an analytically convenient fashion. Mean field remains a popular tool for statistical inference because it applies to a wide range of problems. Bethe-Kikuchi approximations overcome some of the severe restrictions on factorizability by decomposing the entropy according to a *junction graph* (Aji & McEliece, 2001) or, more generally, according to a *region graph* (Yedidia et al., 2005). And expectation propagation (Minka, 2001a) produces a practical inference procedure by projecting the resulting beliefs onto a tractable family of distributions. Once we've chosen an approximating class, we can use our techniques from numerical optimization (e.g. Newton's method, coordinate ascent) to compute a $q(x)$ that minimizes the distance measure. The nice analytical properties of $q(x)$ allow us to easily answer queries regarding x.

The big problem with variational methodology I've just described is that the class of "nice" distributions may be so limiting that the closest $q(x)$ will still be far away from the true posterior, meaning that our estimates risk being excessively biased. As remarked by Yedidia in Opper and Saad (2001), mean field approximations often impose unrealistic or questionable factorizations, leading to biased solutions. With some ingenuity you may be able to design a less limiting class of approximating distributions by exploiting special properties of the posterior; see, for instance, Saul and Jordan (1996), Teh et al. (2007a), Wiegerinck (2000) and Xing et al. (2003).

Notice the common thread throughout all approaches to approximate inference: the most effective procedures are those that take advantage of the structure inherent within the model. But what is "structure" exactly? I propose the following definition: It is the relationship between model semantics and the implementation of a probabilistic inference algorithm.[4] This definition deliberately leaves room for interpretation, in the way that the word *gene* avoids definition (Keller, 2000). The two principal research challenges of probabilistic inference are to discover structure, and to invent new ways to describe structure in a clear and cohesive manner. The latter challenge motivates, at least in part, research in probabilistic logic languages (Getoor & Taskar, 2007; Milch, 2006). The challenges of discovering and exploiting structure surface in other machine learning problems, such as learning and planning in partially observable Markov decision processes, or POMDPs (Poupart, 2005). As new mathematical or analytical tools come to our disposal, or as we uncover tools from other

[4] A shortcoming of this definition is that structure is related to the choice of inference algorithm. Arguably, one should be able to talk about model structure before deciding on the inference strategy.

scientific disciplines, we will undoubtedly find new ways to reason about structure in probabilistic models. A practitioner must always design a model cognizant of whether it is possible to make accurate and efficient inferences—or whether it is possible to conduct inference at all—so as we discover new kinds of structure in probabilistic models, scientists will benefit by being able to design models with greater freedom.

Many notions of structure only come to light under certain algorithmic frameworks for inference, and depending on how we choose to describe the model. To date, the best-understood notion of structure is, arguably, conditional independence. The conditional independencies are most clearly and concisely described using an undirected graphical model (Jordan, 2004). The conditional independencies are less obvious in a directed graphical representation. However, the directed model can tell us whether a node is irrelevant to a query and, if so, it can be ignored during variable elimination (Shachter, 1998; Zhang & Poole, 1994). The extent to which structure is revealed very much depends on the representation we choose for our model. Thus, model representation and model structure are intertwined.

One of the difficulties in forming a cohesive picture of the challenges of probabilistic inference is the great variety of model structure that a practitioner might encounter. Examples of types of structure that can be exploited for more efficient and accurate inference include:

- Conjugacy (Gelman et al., 2003).
- Junction tree width (Paskin, 2004).
- Causal independence (Zhang & Poole, 1996).
- Context-specific independence (Boutilier et al., 1996).
- First-order information for lifted inference (Poole, 2003; Milch et al., 2008).
- Exchangeability (Aldous, 1985; Ghosh & Ramamoorthi, 2003).
- Structure in the query itself; for instance, the second-order terms in the variance may have a negligible impact on the final answer, hence can be ignored (Gillespie, 2004; Hudson, 1991).
- Stationary increments in continuous-time Markov processes (Ross, 2007).
- Structure in the kernel matrix (Shen et al., 2006).
- Many inference algorithms exploit the special properties of the exponential family (Dobson, 2002; Wainwright & Jordan, 2003a).

Some examples of structure exist only in very specific cases, such as an Ising ferromagnet with a uniform magnetic field, and the maximum a posteriori solution to a Markov random field with binary random variables (Boykov et al., 2001; Kolmogorov & Zabih, 2004).

Another obstacle to cultivating a global understanding of probabilistic inference is the great variety of needs. For instance, in some cases it is perfectly acceptable to obtain a single point estimate, such as the most likely outcome (e.g. for binary classification). In some cases, we would prefer computing an expected value, but this may not be a realistic prospect given that we need to take into account a very large amount of data, say, a large cohort of genotype sequences, or perhaps the preferences

of hundreds of thousands of users of a movie website. In other cases, it is extremely important to obtain estimates with high accuracy—say, for determining whether a patient should undergo treatment for cancer, or whether a satellite will collide with another large object in orbit—as incorrect estimates may bear a high cost. We may also have severe time constraints on inference when, for example, we want to know whether there is a pedestrian crossing the road at the upcoming intersection, or we want to display movie recommendations to a user in a timely fashion. Decisions and maximizing one's utility may guide us in these matters, as we should spend more computational effort on inferring quantities that matter the most (Russell, 1997).

1.1 Contributions

My thesis is about making inferences in large probabilistic models when we need accurate answers, and when there is relatively little structure to bring to the table. I develop new, more accurate approximate inference algorithms for such challenging inference problems by exploiting the strengths of Monte Carlo and variational methods. These algorithms improve upon the large bias of variational methods, and they improve upon the high variance importance sampling methods. They are covered in Chapters 3 and 4.

The first attempt to develop my thesis—which I called *conditional mean field*—hinges upon a new class of conditionally-specified variational approximations, and uses a sequence of successively less biased variational approximations together with the sequential Monte Carlo framework to implement probabilistic inference. My first attempt, however, suffers the limitations of variational mean field approximations. The conditional mean field algorithm is covered in Chapter 4. Most of the material in this chapter was originally published in Carbonetto and de Freitas (2007).

My second attempt more successfully captures the strengths of Monte Carlo and variational methods because it allows for a natural trade-off between estimator bias (due to variational approximation) and variance (incurred from importance sampling). My new approach is described in Chapter 3. It can be interpreted as a variational method, a sequential Monte Carlo method, and as a stochastic approximation method. All the material contained in Chapter 3 is original work. Matthew King, a post-doctoral researcher in the Department of Botany at UBC, provided a great deal of assistance in the application of my work to population genetics.

Development of the algorithmic framework in Chapter 3 hinges upon a stochastic approximation method that can reliably handle constraints. Since no method fitting that description exists, I take a slight detour in my thesis in Chapter 2 and develop a constrained stochastic approximation algorithm. The algorithm is based on primal-dual interior-point methods which were originally developed for linear programming. Much of the material presented in Chapter 2 was originally published in Carbonetto et al. (2009). My co-author Mark Schmidt helped me apply the proposed interior-point stochastic approximation method to on-line learning with L_1 regularization.

Finally, in Chapter 5, I explore the issues in representing interdependencies in social network models. What is proposed is an alternative representation of social

networks using conditional probabilities, the main innovation being the introduction of latent variables that control for the direction of influence within the social network. I show how the context-specific independence structure of the proposed directed graphical model can be exploited to implement inference and learning using existing variational techniques. The material in Chapter 5 was written in collaboration with Jacek Kisyński, Michael Chiang and David Poole. The conception of the contingently acyclic model and the social network domain is credited to David Poole. I am primarily responsible for implementation of inference and learning in the model, as well as the empirical aspects of this research, with a great deal of assistance from colleague Jacek Kisyński.

Chapter 2

Stochastic approximation subject to constraints

The original paper on stochastic approximation by Robbins and Monro (1951) describes a simple algorithm for finding the solution to a nonlinear system of equations $F(x) = 0$ when we only have available noisy, unbiased measurements of $F(x)$. Of particular interest is the analysis of convergence for this algorithm, since this analysis lays the theoretical foundation for understanding many important, actively-studied problems in machine learning: policy gradient and temporal differences for reinforcement learning (Jaakkola et al., 1994; Sutton et al., 2000; Williams, 1992), regret minimization in repeated games (Hazan et al., 2007; Zinkevich, 2003), inference for tracking and filtering (George & Powell, 2006; Mathews & Xie, 1993), parameter estimation in probabilistic graphical models (Titterington, 1984; Vishwanathan et al., 2006; Younes, 1991) including the contrastive divergences algorithm (Hinton, 2002; Sun et al., 2008), and on-line learning (Bottou, 1998; Bottou, 2004; Delyon et al., 1999; Kivinen et al., 2004; Sato, 2000; Shalev-Shwartz et al., 2007). In this chapter, I highlight the last one, on-line learning. It is the problem of learning a model by making adjustments that take into account new observations without having to review all the previous observations.

The Robbins-Monro method is a simple algorithm with profound implications. It is simple because it is only a slight modification to the most basic method for optimization, steepest descent (Gill et al., 1986). It is profound because it suggests a fundamentally different way of optimizing a problem—instead of insisting on making progress toward the solution at every iteration, it only requires that progress be achieved *on average*.

Constrained optimization also plays a key role in machine learning; there is a vast array of problems formulated as constrained optimization. Researchers make use of the accompanying mathematical theory to develop efficient solutions to their problems. It is rather strange, then, that there is relatively little work on applying stochastic approximation to learning problems with constraints. Constrained optimization problems are pervasive, yet constrained optimization in the stochastic domain is relatively unexplored. The reason for this, we hypothesize, is that no robust, widely-applicable stochastic approximation method exists for handling such problems.

There is a sizable body of work on treating constraints by extending established optimization techniques to the stochastic setting. However, existing methods are either unreliable or suited only to specific types of constraints. The two major existing approaches involve either projecting the iterates onto the feasible set (Bertsekas, 1999; Poljak, 1978), or converting the problem to an unconstrained one by introducing a penalty term in the objective (Hiriart-Urruty, 1977; Kushner & Clark, 1978). There are several problems with both approaches. Projection may be a bad idea because

it may severely impede progress toward the solution when the iterates are near the boundary of the feasible set. Furthermore, it is expensive to compute the projection for all but the simplest constraints. Most penalty methods have been shown in the numerical optimization community to have severe drawbacks, and they cannot possibly be better in the stochastic setting. The augmented Lagrangian method (Wang & Spall, 2003) is considered to be the most promising penalty method, but suffers from serious deficits, namely sensitivity to the choice of penalty parameter, and lack of a strong guarantee on convergence. Sub-gradient methods have also been used in the stochastic approximation literature (Hazan et al., 2007; Nedic & Bertsekas, 2001; Sadegh, 1997; Shalev-Shwartz et al., 2007; Zheng, 2005), but they are only suited to a limited range of constraints and, as I show in experiments, they can be unreliable. I argue that a reliable stochastic approximation method that handles constraints is needed because constraints routinely arise in the mathematical formulation of learning problems, and the alternative approach—penalization—is often unsatisfactory.

In this chapter, I propose that *interior-point methods* are a natural solution. The main contribution is to present a new stochastic approximation method in which each step is the solution to the primal-dual system that arises in interior-point methods (Forsgren et al., 2002). I show that interior-point methods are remarkably well-suited to stochastic approximation, a conclusion that is far from trivial when one considers that stochastic algorithms do not behave like their deterministic counterparts. For instance, Wolfe conditions for line search (Nocedal & Wright, 2006) do not apply. The method is easy to implement and provides a satisfactory solution to constrained learning problems.

The original motivation behind the work in this chapter was the implementation of a stochastic approximation method for inference in probabilistic models, because constraints commonly arise in inference. (This will be presented in the next chapter.) For instance, a conditional probability table must lie within the probability simplex, while a covariance matrix must always be positive definite. Coming up with a reliable solution to this problem has turned into an interesting endeavor in its own right, and may be of interest to the machine learning and optimization communities at large.

I also investigate the application of the proposed stochastic approximation method to an on-line learning problem. In Sec. 2.5, I derive a variant of Widrow and Hoff's classic "delta rule" for on-line learning (Mitchell, 1997). It achieves feature selection via L_1 regularization, so it is well-suited for learning problems with lots of data in high dimensions, such as the problem of filtering spam from your email account (Sec. 2.5.8). To my knowledge, there is only one existing method that reliably achieves L_1 regularization in large-scale problems when data is processed on-line or on-demand. The on-line method of Garrigues and Ghaoui (2009), based on the homotopy algorithm for the Lasso, was published concurrently with my work (Carbonetto et al., 2009).

I begin this chapter by reviewing the basics behind stochastic approximation, describing the problem, and outlining the proposed solution (Sec. 2.1). To describe and analyze the proposed method in greater detail, I review the rich and extensive body of work on interior-point methods (Sec. 2.2). It is crucial that we establish convergence and numerical stability guarantees for our method (Sec. 2.3). To do so, I

rely on mathematical developments from both the worlds of stochastic approximation and numerical optimization.

2.1 Overview of algorithm

In their 1951 research paper, Robbins and Monro examined the problem of tuning a control variable x (for instance, the amount of alkaline solution) so that the expected outcome of the experiment $F(x)$ (the pH of the soil) attains a desired level α (so your *Hydrangea* have pink blossoms). When the distribution of the experimental outcomes is unknown to the statistician or gardener, it may be still possible to take observations at x. In such case, Robbins and Monro showed that a particularly effective way to achieve the desired response level is to take a (hopefully unbiased) noisy measurement $g_k \approx F(x_k)$, adjust the control variable x_k according to

$$x_{k+1} = x_k - a_k g_k \tag{2.1}$$

for some step size $a_k > 0$, then repeat. (I've assumed here that the derived level α is 0. This assumption can be made without loss of generality.) Provided the sequence of step sizes is chosen to behave like the harmonic series (Cormen et al., 2001), this algorithm converges to the correct solution $F(x^\star) = 0$. The Robbins-Monro procedure can be considerably more efficient than averaging over several observations g_k before adjusting the control variable; see Spall (2000).

Since the original publication, mathematicians have extended, generalized, and further weakened the convergence conditions; see Kushner and Clark (1978), Kushner and Yin (2003) and Scheber (1973) for some of these developments. Most of the effort in the mathematics community has been directed toward formulating general and verifiable conditions that ensure convergence of the stochastic approximation method, usually by assuming the noise in the gradient observation is a Martingale difference; see, for instance, Chapter 5 of Kushner and Yin (2003). In contrast to standard optimization methods such as steepest descent, there is no guarantee that progress toward the solution is made at each step. This means that the analysis of convergence must follow a different strategy.

Kiefer and Wolfowitz (1952) re-interpreted the stochastic process as one of solving an optimization problem; the root-finding problem relates to an optimization problem when $F(x)$ is the gradient $\nabla f(x)$ of some objective of interest $f(x)$. Later, Dvoretsky (1956) pointed out that each measurement g_k is actually the gradient $\nabla f(x_k) = F(x_k)$ plus some noise $\xi(x_k)$. Hence, the *stochastic gradient* algorithm. The Kiefer-Wolfowitz stochastic approximation serves as a generalization to Robbins-Monro, because it allows us to treat the case when only noisy measurements of the objective are available. More recent work has focused on the gradient-free case (Spall, 2000), or on improvements to the gradient descent step by adapting the step size or reducing the variance through scaling or second-order information; see, for instance, Lawrence et al. (2003) and Ruppert (1985). More recently, Kivinen (2003); Kivinen and Warmuth (1997) and others have re-interpreted stochastic approximation as a trade-off between two

objectives, and in certain cases, this new perspective may lead to better algorithms. In this chapter, I introduce a convergent sequence of nonlinear systems $F_\mu(x) = 0$ and interpret the Robbins-Monro process $\{x_k\}$ as solving a *constrained* optimization problem.

Many problems in the machine learning literature can be cast as the problem of optimizing some objective function $f(x)$ in which the optimizer only has access to noisy, unbiased estimates of the gradient $\nabla f(x)$. That is, the noisy gradient measurement g_k is generated some random process. (I discuss the conditions imposed on the behaviour of this random process in Sec. 2.3.) The stochasticity of the gradient can arise for various reasons. For instance, in policy gradient for reinforcement learning (Baxter & Bartlett, 2001; Lawrence et al., 2003) the exact form of the gradient may be unknown, but it is still may be possible to simulate a sample g_k that approximates the true gradient $\nabla f(x)$. Stochasticity may alternatively arise due to the fact that the gradient is composed of a linear combination of responses g_k—*i.e.* the gradient is an *expectation* of the noisy responses—and only a subset of the responses are accessible at any one point in time. Such is the case in *on-line learning*, in which we only have access to a small portion of the data at any one point in time. This is a scenario we investigate in detail in Sec. 2.5.

I extend the original stochastic approximation problem by introducing inequality constraints on x. I will focus on convex optimization problems (Boyd & Vandenberghe, 2004) of the form

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & c(x) \leq 0, \end{array} \tag{2.2}$$

where $c(x)$ is a vector of inequality constraints, and measurements g_k of the gradient at x_k are assumed to be noisy. The feasible set, by contrast, should be known exactly.[1] In Sec. 2.5, we will cast an important problem in machine learning as a constrained optimization problem of the form (2.2) in which we only have access to noisy estimates of the gradient.

The presence of noise in the gradient changes the constrained optimization problem (2.2) in a fundamental way because we can no longer rely on a *merit function* (Nocedal & Wright, 2006) to ensure progress toward the solution at every step. As a result, the convergence behaviour of stochastic approximation is very much unlike the behaviour of algorithms from the numerical optimization literature. Despite the stochastic nature of the constrained optimization problems explored in this chapter, the solutions to these problems always remain well-defined.

To simplify the exposition, I do not consider equality constraints; techniques for handling them are firmly established in the literature (Gould, 1985; Gould et al., 2005). One conventionally assumes convexity to simplify analysis of stochastic approximation. Besides, convergence in non-convex, constrained optimization is a far

[1] There are cases, such as in the literature on constrained Markov decision processes (Altman, 1999), in which the constraints themselves might be difficult to compute exactly. I do not explore this scenario.

- Let x_0, z_0 be given.
- **for** $k = 1, 2, 3, \ldots$
  1. Set maximum step size $\hat{a}_k$ and centering parameter σ_k.
  2. Set barrier parameter μ_k according to (2.18).
  3. Run simulation to obtain gradient observation g_k.
  4. Compute primal-dual search direction $(\Delta x_k, \Delta z_k)$ by solving (2.16) with $\nabla f(x) = g_k$.
  5. Run backtracking line search to the find largest step size $a_k \leq \min\{\hat{a}_k, 0.995 \min_i(-z_{k,i}/\Delta z_{k,i})\}$ such that $c(x_{k-1} + a_k \Delta x_k) < 0$, and $\min_i(\cdot)$ is over all i such that $\Delta z_{k,i} < 0$.
  6. Set $x_k = x_{k-1} + a_k \Delta x_k$.
  7. Set $z_k = z_{k-1} + a_k \Delta z_k$.

Figure 2.1: The interior-point stochastic approximation algorithm.

from settled topic. I also assume that the strictly feasible set is nonempty; that is, there is at least one point x such that $c(x) < 0$ holds. This is otherwise known in the convex analysis literature as *Slater's condition* (Boyd & Vandenberghe, 2004). Further conditions that are needed to guarantee convergence are discussed in Sec. 2.3.

Following the standard barrier approach (Forsgren et al., 2002), we frame the constrained optimization problem as a sequence of unconstrained objectives. This in turn is cast as a sequence of root-finding problems $F_\mu(x) = 0$, where $\mu > 0$ controls for the accuracy of the approximate objective and should tend toward zero. As I explain in Sec. 2.2, a dramatically more effective strategy is to solve for the root of the *primal-dual equations* $F_\mu(x, z)$, where z represents the collection of Lagrange multipliers or dual variables. This is the basic formula of the proposed interior-point stochastic approximation method.

Fig. 2.1 outlines the main contribution of this chapter. Each iteration of the main loop consists of choosing a suitable value for the barrier parameter μ, solving the primal-dual system with a noisy estimate of the gradient to obtain the search direction $(\Delta x, \Delta z)$, computing a step length that ensures the new iterate will remain within the boundary of the feasible set, then updating the primal and dual variables.[2] I will elaborate on all these in subsequent sections. The backtracking line search may or may not be necessary, depending on the form of the inequality constraints $c(x)$.

Provided x_0 is feasible and $z_0 > 0$, every subsequent iterate (x_k, z_k) will be an "interior" point as well. Notice the absence of a sufficient decrease condition on $\|F_\mu(x, z)\|$ or suitable merit function; this is not needed in the stochastic setting. The stochastic approximation algorithm requires a slightly non-standard treatment because the target $F_\mu(x, z)$ moves as μ changes over time. Fortunately, convergence

[2]Having separate step lengths for the primal and dual variables is often recommended in the numerical optimization literature, as it can lead to faster convergence. For the sake of simplicity, I assume that the primal and dual variables are updated with the same step length a_k.

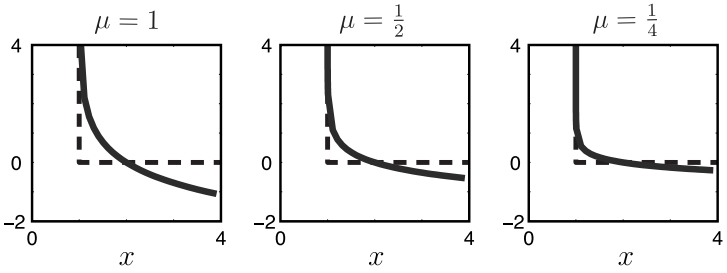

Figure 2.2: The log-barrier function (solid line) applied to the constraint $x \geq 1$ for different values of μ. The dashed line is the exact penalty function.

under non-stationarity has been studied in the literature on tracking and adaptive filtering (Benveniste et al., 1990; Kushner & Yang, 1995; Ljung, 1977; Métivier & Priouret, 1984). Much of this line of work has focused on adaptively selecting step sizes that promote faster convergence (George & Powell, 2006; Mathews & Xie, 1993). While this work is potentially helpful, I do not explore adaptive step sizes here.

In the next section, I derive the primal-dual system which is used to solve for the search direction $(\Delta x, \Delta z)$ in the interior-point stochastic approximation method. In Sec. 2.2.4, I outline the various standard approaches for computing the primal-dual search direction. And in Sec. 2.3, I discuss conditions upon which the interior-point stochastic approximation algorithm will eventually converge to the optimal solution.

2.2 Background on interior-point methods

In this section, I motivate and derive the primal-dual interior-point method starting from the logarithmic barrier method. Despite the incredible successes of interior-point methods for solving large-scale constrained optimization problems, these methods are not well known within the machine learning community. This section serves as an introduction to interior-point methods. For a more extensive overview of the mathematics behind interior-point methods, I refer the reader to Forsgren et al. (2002).

Barrier methods date back to the work of Fiacco and McCormick (1968), but they lost favour due to their unreliable nature. Ill-conditioning was long considered to be their undoing. However, careful analysis (Forsgren et al., 2002; Wright, 1995) has shown that poor conditioning is in fact not the problem—rather, it is a deficiency in the search direction. In the next section, I will exploit this very analysis to show that every iteration of our algorithm produces a stable iterate in the face of: 1) a highly ill-conditioned linear system, 2) noisy observations of the gradient.

The basic idea behind the interior-point method is to formulate an unconstrained approximation to the problem (2.2) via a barrier function. This is achieved by inserting into the objective a barrier or penalty term that mimics the behaviour of the inequality constraints. There is ample room for what qualifies as a barrier function, but I shall restrict my attention to the logarithmic barrier function since it is by far

the most studied and used. The log-barrier leads to the unconstrained problem

$$\text{minimize} \quad f_\mu(x) \equiv f(x) - \mu \sum_{i=1}^{m} \log(-c_i(x)), \tag{2.3}$$

where μ is a positive scalar called the *barrier parameter*, and m is the number of inequality constraints. The behaviour of the barrier function for a simple bound constraint $x \geq 1$ is illustrated in Fig. 2.2. From this example, we see that the quality of the log-barrier approximation grows as μ approaches zero.

The word "barrier" is apt: the logarithmic barrier grows without bound as a constraint approaches zero, thereby preventing iterates from ever traversing the boundary of the feasible set, and obviating the need for the constraints. Notice from the definition that $f_\mu(x)$ resolves to infinity for all points x that are not strictly feasible. Therefore, barrier methods only apply to inequality constraints for which strictly feasible points exist. The philosophy behind barrier methods differs fundamentally from "exterior" penalty methods that penalize points violating the constraints; see Chapter 17 of Nocedal and Wright (2006).

From standard results on unconstrained optimization, the solution to (2.3) is obtained when the gradient vanishes:

$$\nabla f_\mu(x) = \nabla f(x) - \mu \sum_{i=1}^{m} \nabla c_i(x)/c_i(x) = 0. \tag{2.4}$$

This defines the set of optimality conditions for the log-barrier unconstrained approximation, conditions that are very much reminiscent of the Karush-Kuhn-Tucker conditions for optimality in constrained optimization (Nocedal & Wright, 2006). We've transformed the problem of finding a constrained minimum into the problem of finding the root of $F_\mu(x)$, where $F_\mu(x)$ is the gradient of $f_\mu(x)$.

The central thrust of the primal interior-point method is to progressively push the barrier parameter μ to zero at a rate which allows the iterates to converge to the constrained optimum $x^\star$. I've illustrated this process with a small example borrowed from Fiacco and McCormick (1968). A small convex optimization problem is depicted in Fig. 2.3, in which the shaded region depicts the feasible set. The optimization problem is to minimize the linear function $f(x) = x_1 + x_2$ subject to the constraint that $x_1 \geq 0$, and that x_1 and x_2 lie above the parabola $x_1^2 = x_2$. The solution to this problem lies at $(0, 0)$. The solutions to four unconstrained approximations to this problem are also shown in the figure. Witness that as we decrease the barrier parameter μ, the solution to the barrier subproblem approaches the target solution. The line that connects all these solutions is called the *central path*—roughly speaking, it consists of solutions to all barrier subproblems. This concept will soon be important.

Writing out a first-order Taylor-series expansion to the optimality conditions $\nabla f_\mu(x) = 0$ about x, the Newton step Δx is the solution to the linear equations

$$\nabla^2 f_\mu(x)\, \Delta x = -\nabla f_\mu(x), \tag{2.5}$$

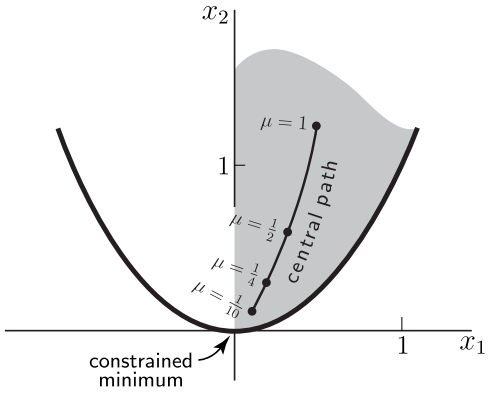

Figure 2.3: Illustration of the primal interior-point method for a simple constrained optimization problem with two variables. The shaded region is the feasible set.

where the Hessian of the log-barrier is derived to be

$$\nabla^2 f_\mu(x) = \nabla^2 f(x) - \mu \textstyle\sum_{i=1}^m \nabla^2 c_i(x)/c_i(x) + \mu J^T C^{-2} J. \tag{2.6}$$

In the above expression, C is the $m \times m$ diagonal matrix with the responses of the constraints along its diagonal, $J \equiv \nabla c(x)$ is the $m \times n$ Jacobian of the constraints evaluated at x, and n is the number of (primal) optimization variables. For conciseness, I've suppressed the dependence of x on the matrices C and J in the notation.

The barrier Hessian $\nabla^2 f_\mu(x)$ has long been known to be incredibly ill-conditioned. However, an analysis by Wright (1994) shows that the ill-conditioning is not harmful under the right conditions. The "right conditions" are that x be within a small distance[3] from the *central path* or *barrier trajectory* (see Fig. 2.3), which is defined to be the sequence of isolated minimizers $x^\star_\mu$ satisfying

$$\nabla f_\mu(x^\star_\mu) = 0 \quad \text{and} \quad c(x^\star_\mu) < 0. \tag{2.7}$$

Second-order sufficient conditions, a suitable constraint qualification, and strict complementarity should also hold (Forsgren et al., 2002; Wright, 1998). The bad news is that the barrier method is ineffectual at remaining on the barrier trajectory, as it pushes iterates too close to the boundary where they are no longer well-behaved (Forsgren et al., 2002; Wright, 1995). And, to make matters worse: 1) it is difficult to minimize the logarithmic barrier function when μ is small, as its surface can vary rapidly near the boundary of the feasible set, and 2) the barrier trajectory gets narrower as μ approaches zero. In practice, one circumvents these problems by measuring closeness to the solution at each value of μ (Gay et al., 1996). In the stochastic setting, however, there is no reliable way to do this.

The primal-dual method takes Newton steps along both the primal variables and

[3]See Sec. 4.3.1 of (Forsgren et al., 2002) for the precise meaning of a "small distance". Since x must be close to the central path but far from the boundary, the favourable neighbourhood shrinks as μ nears 0.

the Lagrange multipliers. Like classical barrier methods, they fail catastrophically outside the central path. But their virtue is that they happen to be extremely good at remaining on the central path (even in the stochastic setting; see Sec. 2.3.2). Primal-dual methods are also blessed with strong results regarding superlinear and quadratic rates of convergence (El-Bakry et al., 1996).

The principal innovation is to introduce variables $z_i \equiv -\mu/c_i(x)$, variables that look very much like Lagrange multipliers. By convention, they are called the *dual variables.* From the condition $\nabla f_\mu(x) = 0$, we recover a "perturbed" version of the first-order KKT conditions for optimality in constrained optimization:

$$F_\mu(x, z) \equiv \begin{bmatrix} \nabla f(x) + J^T Z \mathbf{1} \\ CZ\mathbf{1} + \mu\mathbf{1} \end{bmatrix} = 0, \tag{2.8}$$

where Z is the matrix with z along its diagonal, and $\mathbf{1}$ is a vector of ones. The top row of (2.8) is obtained by substituting z into the log-barrier function $f_\mu(x)$ then taking its gradient with respect to x, and the bottom row of (2.8) follows directly from the definition of z. This is precisely the moving target $F_\mu(x, z)$ mentioned in Sec. 2.1: in the interior-point stochastic approximation method, we decrease the barrier parameter μ over time, and as μ approaches zero the target approaches the solution of the constrained optimization problem.

The KKT system (2.8) consists of two parts: the upper half is the gradient of the Lagrangian function, and the lower half is the complementarity condition that arises in constrained optimization. The *complementarity slackness* conditions (Strang, 1980)—the bottom row of (2.8)—arise directly from the definition of z_i. These complementarity conditions are a source of great aggravation, for even when the objective and constraints are linear, these conditions will always be nonlinear.

It may seem rather remarkable that we've managed to derive a perturbed version of the KKT conditions starting from the log-barrier function. This connection is far from accidental. To cultivate a better understanding of the behaviour of primal-dual interior-point methods, I explore this connection in greater detail in the next section. Note that the material contained in the next two sections (Sec. 2.2.1 and Sec. 2.2.2) is aimed at those readers who would like to get a firm grasp on the mathematical basis for interior-point methods, and it this material is not essential for understanding the main contributions in this chapter.[4] After that, in Sec. 2.2.3 I derive the primal-dual search direction.

2.2.1 Connections to duality

The Lagrangian of the constrained optimization problem (2.2) is defined to be

$$L(x, z) \equiv f(x) + \textstyle\sum_{i=1}^m z_i c_i(x), \tag{2.9}$$

[4]Much of the analysis presented in the next section is based on Monteiro and Adler (1989a; 1989b).

where z is the collection of multipliers corresponding to the inequality constraints $c(x) \leq 0$. The Lagrange dual function is defined to be

$$q(z) \equiv \inf_x L(x, z), \tag{2.10}$$

where the infimum is over all x that satisfy the inequality constraints $c(x) \leq 0$. The infimum is a generalization of the minimum: the infimum of a collection of points is defined to be the largest number that acts as a lower bound on this collection. The Lagrangian function might not have a minimum on the feasible set, in which case the infimum is defined to be negative infinity (Boyd & Vandenberghe, 2004). Under the assumption that the objective is convex, the minimum of the Lagrangian function is achieved when the slope of the Lagrangian function vanishes; *i.e.* $\nabla_x L(x, z) = 0$.

In order to distinguish $f(x)$ from the dual $q(z)$, we refer to $f(x)$ as the *primal* objective. A crucial property of the Lagrange dual is that when $z \geq 0$, $q(z)$ is always a lower bound on the value of the primal objective at its solution $x^\star$. Proving this property isn't hard to do; see Sec. 5.1.3 of Boyd and Vandenberghe (2004). This property is important enough, however, that it is given a special name: *weak duality*. Since the Lagrange dual is always a lower bound on the solution to the original problem, it would make sense to try and find a point z that makes $q(z)$ as large as possible, hence offers the best lower bound. This realization leads to the Lagrange dual problem:

$$\begin{array}{ll} \text{maximize} & q(z) \\ \text{subject to} & z \geq 0. \end{array} \tag{2.11}$$

Implicitly, there is an additional constraint: the infimum in $q(z)$ should not be equal to negative infinity. Just as we said x is feasible if inequality constraints are satisfied, we say z is *dual feasible* if $z \geq 0$ and $q(z) > -\infty$. The dual problem (2.11) is always concave—even if the objective is not a convex function and the feasible domain is not a convex set—so it always has a unique maximum $z^\star$. This is easy to show: the Lagrangian is an affine function of z, hence concave, and the pointwise infimum over a set of concave functions is also concave.[5]

These thoughts suggest that we could solve the Lagrange dual problem instead of the primal (2.2). There are a couple of reasons why this is unlikely work: one, the dual function might not be available in closed form; two, the solution $q(z^\star)$ to (2.11) is only a lower bound on $f(x^\star)$. Under certain conditions, the lower bound is the tightest possible, meaning $f(x^\star) = q(z^\star)$, and when this happens, we achieve *strong duality*. Strong duality is guaranteed when we have a convex optimization problem, and an appropriate constraint qualification, such as the linear independence constraint qualification or Slater's condition. The proof of strong duality under constraint qualification is not terribly straightforward. See Sec. 5.3.2 of Boyd and Vandenberghe (2004) for an argument that uses the hyperplane theorem to prove strong duality un-

[5] See Sec. 3.2.3 of Boyd and Vandenberghe (2004) for an analogous result showing that the supremum preserves convexity.

der Slater's constraint qualification, and see Proposition 3.3.9 in Bertsekas (1999) for a more succinct proof that applies the Mangasarian-Fromovitz constraint qualification.

Since the Lagrange dual at any z is a lower bound on the solution, the difference

$$\eta(x, z) \equiv f(x) - q(z) \tag{2.12}$$

always provides an upper bound on the difference between the value of the objective at the current point x, and the value of the objective at the solution $x^\star$. This difference is called the *duality gap*. When x and z are both feasible, it is easy to show that the duality gap reduces to the simple form

$$\eta(x, z) = -c(x)^T z. \tag{2.13}$$

This expression provides a useful stopping criterion, as the current estimate at (x, z) is no less accurate than the value of the duality gap. See Sec. 5.5.1 of Boyd and Vandenberghe (2004) to see how to use the duality gap to compute an upper bound on the relative accuracy of the estimate.

We now have all the necessary ingredients to write down the necessary and sufficient conditions for (x, z) to be the primal and dual solution to the constrained optimization problem: the point must be primal feasible, it must be dual feasible, the duality gap must vanish, and the gradient of the Lagrangian must also vanish. (This last condition is needed to ensure that the duality gap represents the difference between the primal and dual objectives). Putting everything together, we have

$$\begin{aligned} \nabla_x L(x, z) = 0 \quad & \text{(infimum condition)} \\ c(x)^T z = 0 \quad & \text{(vanishing duality gap)} \\ c(x) \leq 0,\ z \geq 0 \quad & \text{(primal and dual feasibility).} \end{aligned} \tag{2.14}$$

What we have managed to do for a second time is derive the KKT conditions. Since the variables z must be positive and responses of the constraint functions $c(x)$ must be negative, the condition that the duality gap is zero is actually equivalent to requiring that $c_i(x)z_i = 0$, for all $i = 1, \ldots, n$, which, in matrix notation, was $CZ\mathbf{1} = 0$. These nonlinear equations are called the *complementary slackness conditions*. The first condition is often erroneously called the dual feasibility condition, but that would be incorrect as it is often not needed to ensure that the value of $q(z)$ is finite. I have chosen to call it the "infimum condition" instead.

The only real difference between the conditions for optimality (2.14) and the primal-dual system (2.8) is the presence of a perturbation μ in the duality gap. From this result, we see that the primal-dual interior-point method has the effect of relaxing the requirement of a zero gap between the primal and dual objectives.

2.2.2 A note on constraint qualifications

Above, I stated that one of the advantages of the proposed interior-point method for stochastic approximation is that it can handle a wide range of constraints. In this

section, I state in precise terms what I mean by "wide range."

The KKT conditions are necessary for optimality only when the constraints are linear. Thus, regularity conditions, or "constraint qualifications," are needed to establish convergence when the constraints are nonlinear. Such conditions ensure that a linearized approximation captures the essential geometric features of the feasible set in the neighbourhood of a point, so we can determine a feasible direction solely by examining the first derivatives of the objective and the constraints. Stated more formally, constraint qualifications guarantee that the set of Lagrange multipliers that satisfy the KKT conditions is both non-empty and bounded. In practice, however, it is common to encounter degenerate problems for which the set of Lagrange multipliers is unbounded or, worse, empty. Typically, these problems arise from "over-modeling" (Gould et al., 2005).

Standard implementations of interior-point methods assume the linear independence constraint qualification (LICQ); see, for instance, Wächter (2002). LICQ requires that when x is not strictly feasible, the Jacobian of the active inequality constraints must have full row rank (Forsgren et al., 2002). In other words, the gradients of the active constraints must be linearly independent. This implies that the set of Lagrange multipliers satisfying the KKT conditions is a single point, so it is easy to state first-order conditions for optimality. But LICQ is limiting: there are many examples where Lagrange multipliers exist but LICQ fails to hold.[6] The big problem is that if the Jacobian does not have linearly independent rows, then we cannot expect to isolate a solution using standard Newton steps. In such cases, a standard interior-point method implementation may converge to a non-stationary point. Mathematical programs with complementary constraints (Leyffer et al., 2006) are an extensively studied class of problems that violate LICQ. An important case from the machine learning literature is the constrained formulation of the group Lasso problem (Meier et al., 2008; Yuan & Lin, 2006). The alternative Mangasarian-Fromovitz constraint qualifications (Forsgren et al., 2002) are weaker, hence more general, than LICQ, but the disadvantage is that they are more difficult to verify.

There are several strategies for coping with constrained optimization problems that do not satisfy standard constraint qualifications. One approach is to find a well-behaved system that acts as a suitable approximation to the target system, in that it has Lagrange multipliers that are close to the ones of interest (Izmailov & Solodov, 2004). This approach is not practical for large problems because it involves computing a singular value decomposition. According to Leyffer et al. (2006), the most promising strategies are based on exact penalty reformulations of the constraints. Penalty approaches are the subject of ongoing research within the numerical optimization community (Anitescu et al., 2007; Chen & Goldfarb, 2006; Gould et al., 2003).

[6] See Fletcher et al. (2006), Forsgren et al. (2002), and Izmailov and Solodov (2004) for examples where LICQ fails to hold but degenerate solutions exist.

2.2.3 The primal-dual search direction

Just as we did for the primal interior-point method, to obtain the search direction $(\Delta x, \Delta z)$ we form a first-order Taylor series expansion of the nonlinear system (2.8) about point (x, z). The primal-dual Newton step is then the solution to

$$\nabla F_\mu(x, z) \begin{bmatrix} \Delta x \\ \Delta z \end{bmatrix} = -F_\mu(x, z). \tag{2.15}$$

Expanding the terms above, we obtain

$$\begin{bmatrix} W & J^T \\ ZJ & C \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta z \end{bmatrix} = - \begin{bmatrix} \nabla f(x) + J^T Z\mathbf{1} \\ CZ\mathbf{1} + \mu\mathbf{1} \end{bmatrix}, \tag{2.16}$$

where W the Hessian of the Lagrangian,

$$W = H + \textstyle\sum_{i=1}^m z_i \nabla^2 c_i(x), \tag{2.17}$$

and H is the Hessian of the objective. Typically, the Hessian is not available in the stochastic approximation setting, so we replace H by some other symmetric positive definite matrix such as the identity matrix. Note that the exact value of the Hessian is *not* needed to for asymptotic convergence of the primal-dual interior-point method; refer to Sec. 2.3. For further discussion on replacing the Hessian of the objective by a suitable approximation in the stochastic setting, see Sec. 2.4.

The primal-dual Newton step (2.16) provides further insight into the role of the barrier parameter μ. By setting μ to zero, for example, the search direction $(\Delta x, \Delta z)$ attempts to completely eliminate the duality gap. Normally, this won't be a particularly good idea because it regularly happens that only a small step can be taken along this so-called *affine search direction* before the constraints are violated. A less aggressive approach is needed: rather than attempt to eliminate the duality gap entirely, it is better to set a less ambitious goal and attempt to reduce the duality gap by some factor σ. From these considerations, a reasonable choice for the barrier parameter is

$$\mu = \sigma\eta(x, z)/m. \tag{2.18}$$

I divide the target duality gap by m because, as you might recall, the original condition $c(x)^T z = 0$ is separated into m complementary slackness conditions $CZ\mathbf{1} = 0$. When the iterates are not primal and dual feasible, setting μ according to the duality gap is heuristic, as the quantity $\eta(x, z)$ is no longer the difference between the primal and dual objectives.

One important issue we haven't touched upon is the choice of centering parameter σ. The choice for σ in Mehrotra's predictor-corrector algorithm (Gould et al., 2005; Lustig et al., 1992) is given by

$$\sigma = \left(\frac{\eta_{\text{aff}}(x, z)}{\eta(x, z)} \right)^e, \tag{2.19}$$

where $\eta(x, z)$ is the duality gap at the current point (x, z), and $\eta_{\text{aff}}(x, z)$ is the duality gap that would be achieved if we were to follow the largest feasible step along the affine scaling direction; *i.e.* the solution to (2.8) with $\mu = 0$. The exponent e is usually set to 3, but there is no strong reason for this choice. In the stochastic setting, I found that a smaller number around 2 worked better.

Let me give a brief rationale behind this choice of centering parameter. If a step along the affine scaling direction is able to make a large reduction in the duality gap, then we might as well follow it closely and make σ small. When $\sigma = 0$, we get the pure, unperturbed Newton or "affine scaling" step. On the other hand, if the affine scaling direction makes very little progress, we should emphasize a centering step, since it will set the stage for a larger reduction in the next iteration. At the $\sigma = 1$ extreme, the Newton direction defines a step that makes no attempt to reduce the duality gap, and solely tries to center the iterate so that the pairwise products $x_i z_i$ are identical to the average of the current duality gap.

In the stochastic setting, it is unclear whether the predictor-corrector update is a useful heuristic, as we cannot perform line search to assess the amount of progress made by the affine search direction. Regardless of how one updates the centering parameter σ, it is important to guarantee that it gradually approaches zero.

2.2.4 Solving the primal-dual system

Usually, most of the computational effort in primal-dual methods is directed at computation of the Newton search direction (2.16). Whenever possible, one should exploit the structure of the primal-dual system to reduce the effort.

Through block elimination (subtracting $J^T C^{-1}$ times the bottom row from the top row to obtain a lower triangular system) the Newton step Δx is the solution to the symmetric "augmented" system

$$(W - J^T \Sigma J)\Delta x = -\nabla f_\mu(x), \tag{2.20}$$

where $\Sigma \equiv C^{-1} Z$. Recall, W is the Hessian of the Lagrangian; see Eq. 2.17. The dual search direction is then recovered according to

$$\Delta z = -(z + \mu / c(x) + \Sigma J \Delta x). \tag{2.21}$$

Under the provision that our optimization problem is convex, the matrix appearing in the augmented system (2.20) is symmetric positive-definite.

The matrix appearing in the augmented system (2.20) is likely to be highly ill-conditioned, but due to the stability analysis above we can still solve for Δx using a standard factorization such as the Cholesky decomposition $A = R^T R$, followed by forward and backward substitutions (Strang, 1980). In the application to on-line L_1 regularization (Sec. 2.5), we arrive at the solution cheaply because the matrix is extremely sparse, and it has regular structure that is easily exploited by sparse matrix factorization algorithms. For less regular sparsity patterns, it may be useful to try heuristics that rearrange the equations, say, by minimizing the bandwidth of

the matrix to prevent excessive "fill-in" of the Cholesky factors. This is the strategy adopted by the Reverse Cuthill-McKee method (Duff et al., 1986).

Computing Δx in this manner could, regrettably, require factorizing a dense, symmetric matrix, even when the original primal-dual system is sparse. One commonly encounters Jacobian matrices that look like

$$J = \begin{bmatrix} \times & \times & \times & & & & & & \\ & & & \times & \times & \times & & & \\ & & & & & & \times & \times & \times \end{bmatrix},$$

where the $\times$'s correspond to nonzero entries in the matrix. This Jacobian will produce dense sub-blocks in the augmented system (2.20). In such case, it can be worthwhile to factorize the full primal-dual system (2.16) instead. By multiplying the bottom row of (2.16) by Z^{-1}, we can work with a symmetric version of the primal-dual system:

$$\begin{bmatrix} W & J^T \\ J & \Sigma^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta z \end{bmatrix} = - \begin{bmatrix} \nabla f(x) + J^T Z \mathbf{1} \\ c(x) + \mu / z \end{bmatrix}. \tag{2.22}$$

It is easy to see how this system becomes ill-conditioned: the diagonal entries of Σ^{-1} corresponding to the active constraints grow without bound as μ approaches zero. The symmetrized primal-dual system (2.22) is not positive-definite, but it does correspond to a saddle point problem; see p. 4 of Benzi et al. (2005). The matrix in (2.22) is guaranteed to be non-singular (this is easy to show), and when the optimization problem is convex, the matrix is *symmetric quasidefinite* (Vanderbei, 1995). Any symmetric quasidefinite matrix A yields a symmetric factorization $P^T A P = L D L^T$, where L is lower triangular, D is diagonal, and P is a permutation matrix.

When the matrix in (2.22) is indefinite, forming a diagonal matrix of pivots D is unstable, or impossible. Nevertheless, it is still important to take advantage of symmetry. The simple suggestion of Bunch and Parlett (1971) is to extend the notion of pivots to 2×2 blocks. Methods based on block-pivoting preserve the stability and symmetry of symmetric positive-definite factorizations, and require no more storage than a Cholesky factorization (Duff et al., 1986). Mature implementations of symmetric indefinite solvers use supernodal techniques to identify pivots that preserve sparsity, and they behave well in the face of the types of ill-conditioned matrices found in interior-point methods (Amestoy et al., 2001; Schenk & Gärtner, 2006).

Because of the ill-conditioned nature of the systems (2.16) and (2.20), direct factorization techniques are considered to be the most reliable. However, sparse matrix implementations may still scale poorly with problem size. The advantage of iterative Krylov subspace methods (Saad, 1996) is that they rely only on matrix-vector products, so they are ideally suited for large, sparse systems. To implement inexact or "truncated" Newton steps correctly, a suitable termination criterion is needed to ensure that progress is made toward the solution at each iteration of the interior-point solver. In primal interior-point methods, this is relatively straightforward to achieve through existing work on truncated Newton methods for unconstrained optimization (Eisenstat & Walker, 1994); for an example of this approach, see Kim et al. (2007).

However, inexact steps are more difficult to implement in primal-dual interior-point solvers, in part due to changing accuracy requirements (Cafieri et al., 2006). And in the stochastic setting, it is not clear what these accuracy requirements should be. Another critical issue is the choice of a suitable preconditioner to handle the ill-conditioned nature of the system. Preconditioners for interior-point methods have received a great deal of attention; see Rees and Greif (2007) for a recent review.

2.3 Analysis of convergence

So far, I've advocated the primal-dual search direction from interior-point methods for stochastic approximation, but I have yet to explain why this is a sound strategy. The explanation has two parts. First, I discuss conditions upon which the sequence of iterates $\{x_k\}$ generated by the algorithm converges almost surely to a solution $x^\star$ of the constrained optimization problem (2.2) as the amount of data or iteration count goes to infinity. This isn't an especially practical result because it requires us to wait for a very large (*i.e.* infinite) amount of time, but it is nonetheless important to establish. Second, I demonstrate dependable behaviour of the iterates x_k under finite-precision arithmetic.

2.3.1 Asymptotic convergence

In this section, I discuss conditions upon which our method will eventually converge to a solution of the constrained optimization problem (2.2). First, I state an informal proposition for asymptotic convergence and give the conditions that must be satisfied in order for the proposition to apply, then I give an incomplete proof of this proposition that builds on the theory of stochastic approximation and interior-point methods.

I should also state upfront that there is a small gap in the proof, hence the admission that it is "incomplete." The problem stems from the fact that the time-varying system $\nabla f_{\mu_k}(x) = F(x, k)$ must converge to a limiting value $\nabla f(x) = F(x)$ for any x as k goes to infinity. This holds trivially for any point x that is strictly feasible, and it has also been established for any point lying on the boundary that is also a constrained minimizer $x^\star$. But it remains then to show whether this result holds for boundary points that are not also constrained minimizers. It is quite possible that this gap can be resolved by generalizing some of the mathematical results of Wright (1992), but this is only a conjecture at this point in time. Otherwise, it may be possible to reason that the iterates will never reach a boundary point that is not a constrained minimizer, hence we do not have to worry about these points. In any case, resolving this "gap" is an interesting open question in its own right.

Assumptions. I establish convergence under the following fairly standard conditions. The conditions I outline here are intended to be easy for the reader to verify at the expense of being overly stringent.[7] The proposition does, however, require one condition regarding the sequence of barrier parameters that may be difficult to

[7]By contrast, some of the results from the theory on two-time scale approximation (Borkar, 2008) are simpler to state and apply, but the conditions are much harder to verify, hence are not as useful.

verify directly. Further analysis of the barrier function may yield more transparently applicable conditions. This is left to future work.

Some of the conditions may be weakened slightly by applying more general results from the stochastic approximation and constrained optimization literature. For a more detailed discussion of alternative conditions, including motivation for the conditions listed here, see the proof below.

1. **Unbiased observations:** the stochastic gradient g_k at iteration k is a discrete-time Martingale difference with respect to the true gradient $\nabla f(x_k)$; that is,

$$E(g_k \,|\, x_k, \text{history up to time } k) = \nabla f(x_k). \tag{2.23}$$

   This is a very commonly stated condition in the stochastic approximation literature (Berteskas & Tsitsiklis, 1996; Bottou, 1998; Spall, 2003). This condition is more stringent than necessary, and less restrictive conditions require only that the random variables g_k form a Markov chain (Ljung, 1977), or that they follow a stationary ergodic process (Métivier & Priouret, 1984). Andrieu et al. (2005) also states a convergence result under very general conditions. The applications to on-line learning presented in Sec. 2.5 both satisfy the Martingale independence condition (2.23) assuming that the training examples are derived independently.
2. **Sequence of step sizes:** The step sizes a_k must approach zero, but they cannot approach zero too quickly. Precisely,

$$\lim_{k \to \infty} a_k = 0, \qquad \sum_{k=0}^{\infty} a_k = \infty, \quad \text{and} \quad \sum_{k=0}^{\infty} a_k^2 < \infty. \tag{2.24}$$

   These are standard conditions used throughout the stochastic approximation literature. In practice, imposing these conditions on the maximum step sizes $\hat{a}_k$ instead (see Fig. 2.1) will guarantee that the conditions are satisfied for all $a_k \leq \hat{a}_k$.
3. **Bounded variance:** the variance of the gradient estimates g_k is bounded.
4. **Convexity:** The objective $f(x)$ and constraints $c(x)$ are convex.
5. **Smoothness:** The objective $f(x)$ and constraints $c(x)$ are twice continuously differentiable (Gill et al., 1986). This assumption is probably much stronger than necessary, and is imposed mainly to simplify the proof.
6. **Strict feasibility:** There must exist a point x that is strictly feasible; that is, $c(x) < 0$. This is otherwise known as *Slater's condition* in the convex analysis literature (Boyd & Vandenberghe, 2004).
7. **Bounded minimizers:** The set of minimizers $\{x^\star\}$ of the constrained optimization problem (2.2) is nonempty and bounded.
8. **Sequence of barrier parameters:** The barrier parameters μ_k must be positive, and must approach zero in the limit: $\lim_{k \to \infty} \mu_k = 0$. In the asymptotic regime considered here, we do not need to be concerned that the barrier parameters might converge "too quickly" to zero, as was the case for the step sizes a_k. Only in practice (under a finite number of iterations) is this a factor, out of concern for deviations from the central path. There is a further condition (2.40) that

is jointly imposed on the sequence of barrier parameters and step sizes. This condition is imposed in order to ensure that the iterates x_k remain bounded.

Proposition (conjectured). Suppose Assumptions 1 through 8 hold. Then the primal iterates x_k of the interior-point stochastic approximation method (Fig. 2.1) converge almost surely to a global minimizer $x^\star$ of the constrained optimization problem (2.2); that is, as k approaches the limit, $||x_k - x^\star|| = 0$ with probability 1.

Proof (incomplete). In effect, to prove convergence we need to show three things: that the time-varying nonlinear system (which is the gradient of the log-barrier function) converges to some limit point as k goes to infinity; that the iterates x_k converge to a stationary point of the time-varying system; and that any stationary point of the time-varying system is indeed the solution $x^\star$ of the constrained optimization problem (2.2) as k goes to infinity. Below I explain why this proof procedure is not completely sound.

The proof sketch consists of four parts. In Part I, I show that the primal iterates x_k of the primal-dual interior-point method eventually converge to a solution of the unconstrained approximation (2.3) for given $\mu > 0$. For this first part, I assume the deterministic case when there is no noise; that is, $g_k = \nabla f(x_k)$. A principal aim of this first part is to introduce the proper technical terms needed to investigate convergence of the primal-dual interior-point method in the remainder of the proof. In Part II, I generalize the claims of Part I to the case when the gradient estimates are stochastic. For this, I will need to define precisely what I mean by the stochastic process, and I will need to place some additional assumptions on the behaviour of the stochastic process. Part III examines the limiting behaviour of the interior-point method as the barrier parameter μ approaches zero. The key result is that the limit of the sequence of unconstrained approximations recovers a solution to the constrained optimization problem (2.2). The final part, Part IV, assembled Parts I, II and III to give us the desired result (with of course a caveat, as I detail below).

Part I of proof sketch: unconstrained optimization and the log-barrier method. First consider the basic unconstrained optimization problem, in which the objective is to minimize some function $f(x)$, where $x \in \mathbb{R}^n$.

The most ambitious goal is to find a *global minimizer*, which is a point $x^\star$ such that $f(x^\star) \leq f(x)$ for all $x \in \mathbb{R}^n$ (Nocedal & Wright, 2006). Usually we can only hope to find a local solution to the optimization problem. A point $x^\star$ is a *local minimizer* if there exists a neighbourhood of $x^\star$ such that $f(x^\star) \leq f(x)$ for all points x within the neighbourhood. A *neighbourhood* of a point x is the set of all points y within some specified distance δ, $\|x - y\| \leq \delta$, the distance here being defined according to some vector norm $\|u\|$ such as the L_1 or L_2 (Euclidean) norm; see Gill et al. (1986).

When there are constraints on the choice of x (see Part III), we will need to guarantee that the set of minimizers $x^\star$ is bounded. A set S is *bounded* if there is a real number δ such that the norm of all the points x within S is contained by this real number; *i.e.* $\|x\| < \delta$ holds for all points $x \in S$ (Derrick, 1984). It is easy to guarantee that the set of minimizers is bounded in the unconstrained case, as we only need to require that $f(x)$ is bounded from below.

Now suppose that the objective is convex. That is, the line segment joining any two points x and y lies above the graph of $f(x)$ (Berger, 1990). A key result is that if $f(x)$ is convex, then any locally optimal point $x^\star$ is also a globally optimal point. The proof consists in showing that if we suppose that $x^\star$ is not globally optimal—*i.e.* there is a point y such that $f(y) < f(x)$—then this leads to a contradiction because this will imply that we can always find a point x within the neighbourhood of $x^\star$ such that $f(x) < f(x^\star)$; see Boyd and Vandenberghe (2004). Thus, under convexity we can use "local" and "global" interchangeably. This result is easily extended to the case when we have a convex feasible set, the case we tackle in Part III.

Next, consider a gradient descent algorithm with decreasing step sizes for solving the unconstrained optimization problem, in which a_k is a decreasing sequence of step sizes. The algorithm consists of iterately updating the iterates according to

$$x_{k+1} = x_k - a_k \nabla f(x_k), \tag{2.25}$$

and the step sizes a_k are chosen so that they approach zero, but that they do not approach zero too quickly. Mathematically speaking, $\alpha_k \to 0$ as $k \to \infty$, and $\sum_{k=0}^{\infty} \alpha_k = \infty$. Under the additional assumption that the gradients of $f(x)$ are *Lipschitz-continuous*, Proposition 1.2.4 of Bertsekas (1999) shows that the sequence of iterates $\{x_k\}$ will converge to the a minimizer $x^\star$ of the unconstrained optimization problem with convex objective $f(x)$.[8] Since the gradient descent algorithm does not monitor the progress of the iterates x_k with some suitable merit function, we cannot guarantee descent at every iteration. However, we can legitimately guarantee descent if the step sizes are sufficiently small.

Proposition 1.2.4 of Bertsekas (1999) actually applies to a much wider range of algorithms than the one I just described. It applies to all algorithms with iterative updates of the form

$$x_{k+1} = x_k + a_k \Delta x_k, \tag{2.26}$$

in which the search direction makes an angle with the gradient greater than 90 degrees:

$$\nabla f(x_k)^T \Delta x_k < 0. \tag{2.27}$$

The search direction Δx_k is guaranteed to produce a decrease in the objective $f(x)$ provided the step length is sufficiently small (Nocedal & Wright, 2006). The rule (2.27) plainly applies to any search direction of the form

$$\Delta x_k = -B_k^{-1} \nabla f(x_k), \tag{2.28}$$

where B_k is a symmetric positive definite matrix. This corresponds to the Newton direction when B_k is the Hessian of the objective (Gill et al., 1986).

[8] To guarantee Lipschitz continuity, it is sufficient to guarantee that $f(x)$ is twice continuously differentiable (in other words, $f(x)$ is *smooth*; see Gill et al., 1986) and that the second-order partial derivatives are bounded.

The primal (2.5) and primal-dual (2.20) interior-point search directions are also of the form (2.28), so they satisfy the descent criterion as well. Furthermore, under the assumption that the objective $f(x)$ and constraints $c_i(x)$ are convex, the log-barrier function $f_\mu(x)$ is convex as well. This is so because $-\log(-u)$ is convex and monotonically increasing in u (Boyd & Vandenberghe, 2004). By similar logic, the gradient of the barrier function is Lipschitz-continuous provided the objective and constraints are Lipschitz-continuous. Therefore, we can directly apply the results of unconstrained optimization above to the constrained problem formulated using the log-barrier $f_\mu(x)$ for a given $\mu > 0$. This also means that we can ignore the dual variables z, hence the dual variables do not appear in the construction of the proof below (they only resurface in the second part of the convergence analysis, Sec. 2.3.2). We just need to make sure that when we take a step (2.26), the step size a_k is small enough that the new iterate remains strictly feasible. This is the only necessary modification. Because the opposite direction of the gradient will always point us away from the boundary of the feasible region, and because any minimizer of the log-barrier function is strictly feasible, we will always be able to find a positive step size a_k that keeps us in the feasible region; see Theorem 4 of Wright (1992) for details. Note that I will return to Theorem 4 of Wright (1992) in Part III of the proof sketch.

Part II of proof sketch: stochastic approximation. Now I turn to the second part of the proof, where I establish asymptotic convergence of the primal-dual interior-point method with stochastic estimates of the gradient.

To be as clear as possible, I introduce the following definitions. I treat each stochastic gradient measurement as a function $g_k(x, u)$ of the current iterate x and the random variable U with possible values u distributed according to some known density function $p(u \,|\, x)$. This definition follows the description of Berteskas and Tsitsiklis (1996). I then occasionally use the shorthand $g_k \equiv g_k(x_k, u_k)$. (In the more general scenario, each u_k could also depend on the previous samples $u_{k'}$ for $k' < k$.) In the basic Robbins-Monro formulation, when the nonlinear system does not change with time, $g_k(x, u) = g(x, u)$.

First consider the basic stochastic approximation method outlined in Sec. 2.1. The convergence question is to establish conditions upon which the Robbins-Monro recursion (2.1) will converge to a point $x^\star$ such that $F(x^\star) = 0$. This is of course equivalent to convergence to a stationary point $x^\star$ of the unconstrained optimization problem when $F(x) = \nabla f(x)$. There are two equivalent approaches to understanding the Robbins-Monro recursion. The first is to identify a mapping between the nonlinear system $F(x)$ and the expectation of the stochastic gradient estimates:

$$E[\, g(x, U)] = \int p(u \,|\, x)\, g(x, u)\, du = F(x), \tag{2.29}$$

The second approach is to treat the stochastic gradient measurement as the true response of the system plus some random noise:

$$g(x) = F(x) + \xi, \tag{2.30}$$

where ξ is drawn from some specified probability density $p(\xi \mid x)$. This is the approach taken in the analysis of the classic Kiefer-Wolfowitz stochastic gradient algorithm. I will follow the first approach.

Let me now briefly provide some motivation for the conditions (2.24) on the sequence of step sizes. Suppose for a moment that $\sum_{k=0}^{\infty} a_k < \infty$. Then the stochastic approximation iterations will be confined to lie within a finite radius from the starting point x_0, and if the desired solution $x^\star$ happens to be outside that radius, the algorithm will never reach the solution. This motivates the second identity in (2.24). Usually, the third identity in (2.24) is imposed to ensure that the sequence $\{x_k\}$ is bounded. Weaker conditions than (2.24) have been used to prove asymptotic convergence. For further discussion on this matter, see Berteskas and Tsitsiklis (1996).

There are two common approaches for establishing asymptotic convergence of stochastic approximation. One approach is to introduce a *Lyaunov function* (Luenberger, 1979) that provides a measure of the distance from the solution. In the stochastic setting, the Lyapunov function is taken to be the *expected* update direction, and then Martingale theory is used to study convergence of this function (Bottou, 1998; Métivier, 1982). Proposition 4.1 of Berteskas and Tsitsiklis (1996) is a clear example of the Lyapunov approach. It proves that the limit point of the sequence of iterates $\{x_k\}$ converges to a stationary point $x^\star$ under the conditions outlined in the next paragraph.

The first condition is a smoothness condition on the nonlinear system, namely that $F(x)$ must be Lipschitz-continuous. Second, the sequence of step sizes must satisfy the conditions (2.24) outlined earlier. The third condition is that there exist a positive constant c such that

$$c \, \|F(x)\|^2 \leq -F(x)^T E[\, g(x, U)], \tag{2.31}$$

where $\|u\|$ is the Euclidean norm of u. When $F(x)$ is the gradient of the objective, this condition plays an analogous role to the descent condition (2.27) in Part I. The fourth and final condition is that

$$E\|g(x, U)\|^2 \leq b + b'\|F(x)\|^2, \tag{2.32}$$

for positive constants b and b'. This ensures that the gradient estimates have a bounded second moment, which is very similar to the "bounded variance" condition I imposed at the beginning of this section.

The another widely adopted approach to analyzing the convergence of stochastic approximation was developed independently by Ljung (1977) and Kushner and Clark (1978). The basic idea of this approach is to relate the Robbins-Monro recursion (2.1) to an ordinary differential equation (ODE), and identify limit points as equilibrium points of the differential equation (or points belonging to the *invariant set* of the differential equation). So long as the initial point x_0 lies in the *domain of attraction* (Ljung & Söderström, 1983) of the differential equation, the sequence $\{x_k\}$ will reach the invariant set for large k. This is the approach I will follow for

analyzing the interior-point stochastic approximation method, mainly because it has been frequently applied to a broader context when the nonlinear system varies over time, and this is precisely the kind of system that procedes from the interior-point stochastic approximation method.

A caveat of the ODE approach is that it does not, strictly speaking, give us a convergence result. Intuitively, the ODE analysis only applies to points within the domain of attraction, so we must make sure that sequence $\{x_k\}$ touches the domain of attraction infinitely often. Thus, a bounded sequence is needed to guarantee convergence. (One way to accomplish this is to project the iterates into some compact set, as suggested by Ljung and Söderström (1983).) My convergence analysis follows from Métivier and Priouret (1984) because that article provides a clear and unified presentation of the ODE analysis of Kushner and Clark (1978) and Ljung (1977), and uses Martingale arguments to prove the required convergence result. First I state a slightly simplified version of the theorem of Kushner and Clark (1978) as it is presented in Métivier and Priouret (1984), and use this theorem to establish convergence of the Robbins-Monro recursion when the nonlinear system $F(x)$ is time-invariant. Next, I generalize this convergence result to the case when the system changes over time.

Theorem (Kushner & Clark, 1978). Suppose that the sequence of iterates $\{x_k\}$ is defined by the recursion

$$x_{k+1} = x_k - a_k(F(x_k) - \beta_k - \xi_k), \tag{2.33}$$

where ξ_k is the stochastic process and β_k is a deterministic term converging to zero almost surely (*i.e.* $\sup_k \|x_k\| < \infty$), and $\{a_k\}$ is a sequence of step sizes satisfying (2.24). Provided that the iterates x_k are bounded, then there exists a compact set belonging to a domain of attraction of locally stable point $x^\star$ that contains infinitely many iterates x_k for which $\lim_{k\to\infty} x_k = x^\star$.

The theorem of Kushner and Clark (1978) is easily applied to the Robbins-Monro algorithm (2.1) by setting $g_k = F(x_k) - \xi_k$ and $\beta_k = 0$. If we could prove that the sequence of iterates $\{x_k\}$ is bounded with probability one, then we would have convergence to a stationary point $x^\star$. The boundedness of the sequence can be guaranteed under the following conditions: $E\|g(x,U)\|^2$ is bounded from above, and

$$F(x)^T(x - x^\star) > 0. \tag{2.34}$$

See Métivier and Priouret (1984) for a proof of this result that uses a quasi-martingale argument under slightly looser restrictions.

The condition (2.34) appears to be rather limiting, but it actually holds whenever $F(x) = \nabla f(x)$ and $f(x)$ is convex and differentiable. It follows from the fact that

$$f(y) \geq f(x) + \nabla f(x)^T(y - x), \tag{2.35}$$

holds for all x and y. By replacing y with the stationary point $x^\star$, and by using the fact that $f(x^\star) \leq f(x)$, we recover the condition (2.34).

Next I consider the case when the nonlinear system changes over time, so that the

expectation of the stochastic gradient estimates follows

$$E[\, g_k(x, U)] = \int p(u \,|\, x)\, g_k(x, u)\, du = F(x, k). \tag{2.36}$$

This variation of Robbins-Monro fits within the Kushner-Clark framework by setting $\beta_k = F(x) - F(x, k)$, on the condition that there exists a $F(x)$ such that

$$\lim_k \|F(x) - F(x, k)\| = 0. \tag{2.37}$$

This condition is needed to ensure that $\{\beta_k\}$ converges to zero, as required above. It is precisely this condition that reveals a gap in the proof, as I explain in Part IV.

For convergence to hold in the time-varying case, we need an additional condition:

$$\textstyle\sum_{k=0}^{\infty} a_k \, \|F(x, k) - F(x)\| < \infty. \tag{2.38}$$

This condition will of course be satisfied if the system converges to $F(x)$ at a finite point in time, but this might cause a small hiccup for asymptotic convergence of the stochastic interior-point method. We reexamine this issue in Part IV.

Part III of proof sketch: convergence of the interior-point method. In this part, I look at the convergence properties of the log-barrier method as the barrier parameter μ is driven toward zero. Under the assumption that the initial point x_0 is strictly feasible, the convergence proof is to show that as we decrease the barrier parameter μ, the solution the barrier subproblem approaches the solution of the constrained optimization problem (2.2). The proof is considerably simpler for convex programs, so I will use the results of Wright (1992) rather than those of Forsgren et al. (2002). Wright (1992) assumes throughout that the objective and constraints are twice continuously differentiable. In Part IV, I will combine these results with the stochastic approximation convergence results from Part II.

First, I state some basic definitions for constrained optimization. A point x is *feasible* if it satisfies all the constraints, and it is *strictly feasible* if all the constraint functions are strictly positive. A point $x^\star$ is a local minimizer if it is feasible and there exists a neighbourhood of $x^\star$ such that $f(x^\star) \leq f(x)$ for all feasible points x within the neighbourhood. Note that the feasible region is convex because it is the intersection of a collection of m convex sets. This fact means we do not have to worry about topological inconsistencies that could arise when the interior of the feasible set is not the same as the strictly feasible region (Forsgren et al., 2002).

There are three possible constrained optimization situations to consider. In the first case, the minimum of the objective $f(x)$ lies in the interior of the feasible set. This case is not particularly interesting because the constraints have no effect on the solution. In the second case, the unconstrained minimizer of $f(x)$ is infeasible. In the third case, $f(x)$ is unbounded below when the constraints are removed. In these last two cases, the constrained solution lies on the boundary of the feasible set. As the barrier parameter μ decreases to zero, intuition suggests that minimizers of the unconstrained approximation will converge to a constrained solution that lies on the

boundary of the feasible set. However, there are two possible complications. One, the barrier method can never recover a minimizer lying on the boundary of the feasible set. Two, for smaller values of μ we observe an increasing steepness near stationary points of the log-barrier function. These two difficulties are resolved in Theorems 4 and 5 of Wright (1992).

Since local unconstrained minimizers are defined in terms of bounded sets, we first need to ensure existence of this bounded set for every log-barrier function $f_\mu(x)$ (Forsgren et al., 2002). Theorem 4 of Wright (1992) guarantees boundedness of barrier function level sets. This property is key because it establishes that the set of minimizers of the barrier function is bounded. There are only two conditions to this theorem: the strictly feasible region is nonempty, and the set of unconstrained minimizers is bounded.

The main result is Theorem 5 of Wright (1992). I restate parts of it here.

Theorem (Wright, 1992). Provided that $\{\mu_k\}$ is a decreasing sequence of positive barrier parameters such that $\lim_{k\to\infty} \mu_k = 0$, $f(x)$ and $c(x)$ are convex functions, the set of constrained minimizers is nonempty, and there is at least one point that is strictly feasible, then we have that:

(a) there exists a bounded and closed set S such that for all k, every minimizer x_k of the barrier subproblem (2.3) with barrier parameter μ_k is strictly feasible and lies in S;

(b) any sequence of minimizers $\{x_k\}$ to the unconstrained approximations with barrier parameters $\{\mu_k\}$ has at least one convergent subseqence, and every limit point of $\{x_k\}$ is a minimizer of the constrained optimization problem (2.2);

(c) for any sequence of minimizers $\{x_k\}$ to the unconstrained approximations with barrier parameters $\{\mu_k\}$, $\lim_{k\to\infty} f_{\mu_k}(x_k) = f(x^\star)$.

The proof of this theorem has two main parts. The first part verifies the existence of the bounded set S containing the minimizers of all the unconstrained approximations. In order to show this result, this first part of the proof uses the fact that as we decrease μ, the value of the objective $f(x)$ and the response of each term $\log(-c_i(x))$ at each unconstrained minimizer x must decrease. Since the sequence $\{x_k\}$ must be bounded, it contains at least one convergent subseqence with some limit point $\hat{x}$. The second task is to show that $\hat{x}$ will always be a minimizer $x^\star$ of the constrained optimization problem. The proof of this second part is treated in three separate cases: 1) when $x^\star$ is strictly feasible, 2) when $x^\star$ is not strictly feasible but $\hat{x}$ is strictly feasible, and 3) when neither $x^\star$ nor $\hat{x}$ are strictly feasible. Part (c) of the theorem evidently holds when $x^\star$ is strictly feasible, so the only challenge is to show that it holds when $x^\star$ lies on the boundary of the feasible set.

Note that this theorem only requires that the objective and constraints be continuous; see Forsgren et al. (2002). Thus, it may be possible to loosen the smoothness assumption stated at the beginning of this section.

Part IV of proof sketch: assembling the parts. In Part I, I explained how the barrier method retains feasibility of the iterates x_k, and how the primal-dual

search direction gives us a direction of descent under certain conditions. In Part II, I explained how the iterates x_k of the interior-point method with a primal-dual search direction and stochastic estimate of the gradient converge to a minimizer $\theta^\star$ under certain assumptions. In Part III, I showed that the unconstrained minimizers of the barrier subproblems (2.3) coincide with the minimizers of the constrained optimization problem as k approaches the limit.

The final task is to apply the results of Part III to the stochastic setting with a time-varying system treated in Part II. This is accomplished by setting the time-varying system to the gradient of the log-barrier function,

$$F(x, k) = \nabla f_{\mu_k}(x), \tag{2.39}$$

then g_k is the gradient of the log-barrier in which $\nabla f(x_k)$ replaced by a noisy estimate. Recall, Theorem 4 Wright (1992) as I stated in Part III guarantees that each set of unconstrained minimizers to the barrier subproblem (2.3) is bounded, which is of course necessary in the stochastic setting as well. The remainder of the convergence conditions have been stated at the beginning of this section.

For any point x that lies within the strictly feasible region (which coincides with the *interior* of the feasible region when we have a convex optimization problem), it is easily shown that $\lim_{k\to\infty} \nabla f_\mu(x) = f(x)$. Therefore, the required condition (2.37) holds for any strictly feasible point x.

The difficulty stems from the case when x lies on the boundary of the feasible set. It is clear that (2.37) does *not* hold for such an x, hence we need a weaker condition than (2.37) to apply the results of Métivier and Priouret (1984) to interior-point methods. From Part (c) of Theorem 5 in Wright (1992) as it is stated above, we know that it can be shown that $\lim_{k\to\infty} \nabla f_\mu(x_k) = f(x^\star)$ for any sequence $\{x_k\}$ of unconstrained minimizers, even when $x^\star$ lies on the boundary of the feasible set. Intuition dictates that we need an analogous result for a broader range of sequences—the result that stated above that applies solely to the sequence of unconstrained minimizers is clearly insufficient. The general results in the interior-point literature on *path-following methods* (Wright, 1997) suggest a strategy for resolving this problem.

I conclude this proof sketch with a couple additional points.

First, note that Theorem 5 of Wright (1992) requires that $\{\mu_k\}$ be a decreasing sequence. This condition is not necessary for convergence of the interior-point stochastic approximation method because we only need the barrier parameters to decrease within finite intervals.

Second, I mentioned a slight "hiccup" in Part III arising from the condition (2.38). For the interior-point stochastic approximation method, this condition would be

$$\textstyle\sum_{k=0}^{\infty} a_k \left\|\nabla f_{\mu_k}(x) - \nabla f(x)\right\| < \infty. \tag{2.40}$$

For peace of mind, it is certainly possible to satisfy (2.40) by designing the sequence of step sizes $\{a_k\}$ and the sequence of barrier parameters $\{\mu_k\}$ so that $\{a_k\}$ and the sequence $\{\|\nabla f_{\mu_k}(x) - \nabla f(x)\|\}$ are both bounded from above by the harmonic series

$\{\frac{1}{k}\}$ (Cormen et al., 2001). In practice it may be safe to ignore this matter.

2.3.2 Considerations regarding the central path

The object of this section is to establish that computing the stochastic primal-dual search direction is numerically stable. By "stable," I mean that small perturbations in the gradient estimate g will not lead to large perturbations in primal-dual search direction $\Delta\theta = (\Delta x, \Delta z)$. Mathematically speaking, the computation is stable when the ratio $\|\delta\Delta\theta\|/\|\delta g\|$ is never large, where δg is a perturbation of the gradient estimate and $\delta\Delta\theta$ is a perturbation of the primal-dual search direction $\Delta\theta$ (Trefethen & Bau, 1997). The concern is that noisy gradient measurements will lead to wildly perturbed search directions for x and z, hence a rigorous analysis of numerical stability is even more crucial here than in the standard, non-stochastic setting.

As I mentioned in Sec. 2.2, interior-point methods are surprisingly stable provided the iterates remain close to the central path, but the prospect of keeping close to the path seems particularly tenuous in the stochastic setting. The key observation is that the central path is itself perturbed by the stochastic gradient estimates. Following arguments similar to those presented in Sec. 5 of Forsgren et al. (2002), I show that the stochastic Newton step (2.16) stays on target.

For a given μ, suppose that (x, z) is the root of the nonlinear system of equations (2.8). Implicitly, x and z are functions of μ, and I'll write the relationship between μ and (x, z) as $\theta(\mu) = (x, z)$. This is a path in real space—in fact, it is the *central path*. Without knowing this function exactly, we can still compute the rate of change of (x, z) with respect to μ via implicit differentiation (Strang, 1991).

I define the *noisy central path* as $\theta(\mu, \varepsilon) = (x, z)$, where (x, z) is a solution to $F_\mu(x, z) = 0$, in which the gradient $\nabla f(x)$ is replaced by a noisy estimate $g = \nabla f(x) + \varepsilon$.[9] Suppose we are currently at point $\theta(\mu, \varepsilon) = (x, z)$ along the path, and the goal is to move closer to $\theta(\mu^\star, \varepsilon^\star) = (x^\star, z^\star)$ by solving (2.16). One way to assess the quality of the Newton step is to compare it to the tangent line of the noisy central path at (μ, ε). If the tangent line is a fairly reasonable approximation to the central path $\theta(\mu, \varepsilon)$, then a step along the tangent line will make good progress toward $(x^\star, z^\star)$. Taking implicit partial derivatives at (x, z), the tangent line is

$$\begin{aligned}\theta(\mu)(\mu^\star, \varepsilon^\star) &\approx \theta(\mu, \varepsilon) + (\mu^\star - \mu)\frac{\partial\theta(\mu, \varepsilon)}{\partial\mu} + (\varepsilon^\star - \varepsilon)\frac{\partial\theta(\mu, \varepsilon)}{\partial\varepsilon} \\ &= \begin{bmatrix} x + (\mu^\star - \mu)\frac{\partial x}{\partial\mu} + (g^\star - g)\frac{\partial x}{\partial\varepsilon} \\ z + (\mu^\star - \mu)\frac{\partial z}{\partial\mu} + (g^\star - g)\frac{\partial z}{\partial\varepsilon} \end{bmatrix}, \end{aligned} \tag{2.41}$$

[9] I've implicitly assumed here that all the entries of the gradient vector are perturbed by the same amount, but it is straightforward to extend my line of reasoning to the case when each coordinate has its own noise term.

such that

$$\begin{bmatrix} H & J^T \\ ZJ & C \end{bmatrix} \begin{bmatrix} (\mu^\star - \mu)\frac{\partial x}{\partial \mu} + (g^\star - g)\frac{\partial x}{\partial \varepsilon} \\ (\mu^\star - \mu)\frac{\partial z}{\partial \mu} + (g^\star - g)\frac{\partial z}{\partial \varepsilon} \end{bmatrix} = - \begin{bmatrix} g^\star - g \\ (\mu^\star - \mu)1 \end{bmatrix}, \tag{2.42}$$

with $g^\star \equiv \nabla f(x) + \varepsilon^\star$. Since (x, z) is the solution to $F_\mu(x, z) = 0$, the Newton step (2.16) at (x, z) with perturbation $\mu^\star$ and stochastic gradient estimate $g^\star$ is the solution to

$$\begin{bmatrix} H & J^T \\ ZJ & C \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta z \end{bmatrix} = - \begin{bmatrix} g^\star - g \\ (\mu^\star - \mu)1. \end{bmatrix}. \tag{2.43}$$

In conclusion, if the tangent line (2.41) is a fairly reasonable approximation to the central path, then the stochastic Newton step will make good progress toward $\theta(\mu^\star, \varepsilon^\star)$.

Having established that the stochastic algorithm closely follows the noisy central path, the analysis of Wright (1998) directly applies to computation of the search direction with noisy estimates of the gradient, in which roundoff error ($\epsilon_{\text{machine}}$) in the analysis is occasionally replaced by gradient noise (ε). I assume here that the reduced, symmetric positive-definite system (2.20) is being used to compute the primal search direction Δx. Similar results hold for computation via the indefinite primal-dual system (2.16).[10] Numerical stability under finite-precision arithmetic and stochasticity of $\nabla f(x)$ is of fundamental concern, particularly in computing the entries of the matrix $W - J^T \Sigma J$, computing the right-hand side of (2.20), and computing the solution to Δx and Δz using a backward stable algorithm. I take a few moments to step through Wright's line of reasoning that establishes some vital stability properties of primal-dual iterates that closely follow the central path.[11] The key points of Wright (1998) are as follows.

When solving for x in the linear system of equations $Ax = b$ using a backward stable algorithm such as the Cholesky factorization, the computed solution can be characterized as the exact solution to a system with perturbed matrix A and unperturbed vector b (Trefethen & Bau, 1997). In other words, a backward stable algorithm computes the correct answer to a slightly wrong problem. We need to consider three sources of error: error in forming the right-hand side b, error in forming A, and error when computing the solution to x with a backward stable algorithm.

Suppose the iterate (x, z) is within a small δ-neighbourhood of the central path without residing too close to the boundary of the feasible region (precisely speaking, $c(x)$ is asymptotically bounded from below by δ), and $\mu = O(\delta)$. Then the perturbation incurred when computing $A \equiv W - J^T \Sigma J$ is roughly proportional to $\epsilon_{\text{machine}}/\delta$, and furthermore the perturbation in the computed value of $b \equiv -\nabla f_\mu(x)$ is on the order of the gradient noise $\varepsilon \gg \epsilon_{\text{machine}}$.

[10] Symmetrization of the primal-dual system, as discussed in the next section, is benign in the sense that it does not affect the stability of the primal-dual search direction; see Wright (1998).

[11] Note that Wright (2001) presents an alternate derivation of stability that does not require the linear independence constraint qualification.

The relative condition numbers $\kappa(A)$ of the matrix-vector products Ax and $x = A^{-1}b$ are bounded from above by $\|A\|\|A^{-1}\|$, where $\|A\|$ is an induced matrix norm of A. This result is given in (Trefethen & Bau, 1997). The problem is that A—the matrix arising from the reduced system (2.20)—is notoriously ill-conditioned as μ approaches 0, so the bound on the relative condition number is very poor. Fortunately, we are able to derive a tighter bound by exploiting special structure present within the matrix.

The first observation is that the matrix $A = W - J^T \Sigma J$ is almost entirely dominated by a matrix that lies in the space spanned by the gradient vectors $\nabla c_i(x)$ of the active constraints i. The same observation holds for b. The perturbations of A and b are also restricted to this space. This is fortunate because this space corresponds precisely to a *well-conditioned* invariant subspace of A.

Let $A = U\Sigma V$ be the singular value decomposition of A (Strang, 1980). By dividing the matrix of singular values Σ into sub-blocks Σ_{large} and Σ_{small}, such that Σ_{large} contains a set of large singular values and Σ_{small} contains a collection of small singular values, it can be shown that the relative condition number of x projected onto the invariant subspace of Σ_{large} is bounded by $\|\Sigma_{\mathsf{large}}\|\|\Sigma_{\mathsf{large}}^{-1}\|$. When the separation of the singular values within the large set is much smaller than the separation of the full set, then the bound $\|\Sigma_{\mathsf{large}}\|\|\Sigma_{\mathsf{large}}^{-1}\|$ is much tighter than the bound $\|A\|\|A^{-1}\|$.

The matrix in (2.20) possesses precisely the properties we just stated. Without the previous considerations, the absolute perturbations of the solution Δx would be on the order of the magnitude of the noise ε, which would be a horrific amount of error when the search direction is close to zero, implying we are near the solution of the constrained optimization problem. From the results established so far, however, it can be shown that if we take into account perturbations in the computed right-hand side b, the computed solution Δx projected onto the range space of active constraints has a condition number on the order of $\delta\varepsilon$ instead, a factor of δ better than the previously stated bound. And perturbations from executing the backward stable algorithm only contribute an additional factor on the order of $\delta\epsilon_{\text{machine}}$ to the condition number. The error bounds for the remaining portion of Δx are much worse, but it is precisely this portion—the projection onto the the null space of the Jacobian of the active constraints—that has negligible impact on determination of the first-order KKT conditions near the solution (Nocedal & Wright, 2006, Theorem 12.1). Beware that these results give us bounds on the absolute error of the solution, so there is no guarantee that the relative error will be small when Δx approaches zero. Also note that these error bounds will likely be poor when J is ill-conditioned.

Finally, Wright (1998) applies analogous and rather lengthy analysis to the dual search direction to show that the computed value of Δz in (2.21) has a similar condition number. In conclusion, an off-the-shelf backward stable Cholesky factorization can be used to solve (2.20) so long as the iterates are feasible.

Of concern is the ill-effect of *cancellation* in the constraints near $c_i(x) = 0$; see Example 12.3 of Trefethen and Bau (1997). I defer the issue since cancellation did not arise in the experiments. Note that even though the full primal-dual system (2.16) is well-conditioned, the solution is not any more accurate because the right-hand side

incurs cancellation errors.

2.4 Damped quasi-Newton approximations

The Hessian of the objective is implicated in computation of the primal-dual search direction. In the stochastic setting, however, it may be difficult to obtain reliable estimates of the second-order derivatives. Furthermore, when the problem is large and the Hessian is dense, computing the exact Newton step may be computationally infeasible. The simplest—if not most effective—strategy is to simply ignore the scaling of the problem, and replace H with the identity. This is the primal-dual interior-point analog of steepest descent, and it is what I used for all the experiments in this chapter. This approach, however, will not work when the problem is severely ill-conditioned, as in the probabilistic inference problems I explore in the next chapter. The interior-point method has the virtue that it is easy to incorporate second-order information into the primal-dual search direction. By contrast, this is not so easily done for projection and sub-gradient strategies (Andrew & Gao, 2007; Gafni & Bertsekas, 1984). In this section, I propose two new, robust ways to obtain stochastic estimates of second-order partial derivatives based on quasi-Newton methods (Dennis & Moré, 1977; Nocedal & Wright, 2006).

2.4.1 Damped Barzilai-Borwein method

The first strategy is no more costly than the steepest descent direction. It is based on the simple quasi-Newton method described in Barzilai and Borwein (1988) and discussed in Fletcher (2005). The basic idea of the Barzilai-Borwein method is to find the scalar β that minimizes

$$\tfrac{1}{2}\|\beta\Delta x - \Delta y\|^2, \tag{2.44}$$

where Δx is not the search direction, but rather the difference between two iterates at some intermediate iteration of the optimization routine, and Δy is the difference of the gradients at those same two points. The above expression is the norm of the secant equation (Dennis & Moré, 1977), in which the full quasi-Newton approximation to the Hessian is replaced by the identity matrix times scalar β. So long as the dot product $\Delta x^T \Delta y$ is positive—which is guaranteed so long as the Wolfe conditions are satisfied (Nocedal & Wright, 2006)—the Barzilai-Borwein method provides us with a positive-definite Hessian approximation. In the stochastic setting, the differences Δy are noisy, so there is no way to ensure the Wolfe conditions hold. So I propose the following recourse.

Instead, I propose to directly apply the Robbins-Monro method to minimizing the secant equation (2.44). This leads to *damped* Barzilai-Borwein updates. Taking first and second derivatives of the secant equation, the Newton step leads to Robbins-

Monro updates of the form

$$\beta_{k+1} = \beta_k - a_k \left(\beta_k - \frac{\Delta x_k^T \Delta y_k}{\Delta x_k^T \Delta x_k} \right). \tag{2.45}$$

This update should be skipped whenever $\Delta x_k^T \Delta y_k$ makes β_{k+1} very small.

In practice, I've found that this stochastic version of the Barzilai-Borwein method finds an appropriate global scaling of the objective whenever the objective is convex. Based on my experience, I do not recommend the stochastic Barzilai-Borwein method for non-convex optimization problems.

2.4.2 Damped BFGS method

The second strategy employs a full quasi-Newton approximation to the Hessian. This will not be appropriate for larger optimization problems, as the complexity of each iteration is $O(n^2)$ or $O(n^3)$ when we have constraints, where n is the number of (primal) optimization variables. This strategy is based on the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update (Dennis & Moré, 1977). Since satisfaction of the Wolfe conditions cannot be guaranteed, I employ the damped updates proposed in a 1978 paper by Powell. The derivation of the damped BFGS update is straightforward.

First of all, the BFGS update of the Hessian aproximation B is

$$B^{(\text{new})} = B + \frac{\Delta y \Delta y^T}{\langle \Delta y, \Delta x \rangle} + \frac{B \Delta x \Delta x^T B}{\langle \Delta x, B \Delta x \rangle}, \tag{2.46}$$

where $\langle u, v \rangle$ is the dot product of u and v. It is a symmetric update of rank two. To retain a positive-definite quasi-Newton approximation, the strategy is to check its determinant. Applying the matrix determinant identity

$$\det(A + uv^T) = \det A(1 + v^T A^{-1} u), \tag{2.47}$$

the determinant of a rank-two update works out to be

$$\det(A + u_1 u_2^T + u_3 u_4^T) = \det A(1 + u_2^T A^{-1} u_1)(1 + u_4^T (X + u_1 u_2^T)^{-1} u_3).$$

Next, applying the Sherman-Morrison-Woodbury formula (Dennis & Schnabel, 1996), which states the the inverse of a rank-one update is given by

$$(A + uv^T)^{-1} = A^{-1} - (1 + v^T A^{-1} u)(A^{-1} u v A^{-1}), \tag{2.48}$$

we obtain the following formula for the rank-two update of the determinant:

$$\begin{aligned} &\det(A + u_1 u_2^T + u_3 u_4^T) \\ &\quad = \det A\{(1 + u_1^T A^{-1} u_2)(1 + u_3^T A^{-1} u_4) - (u_1^T A^{-1} u_4)(u_2^T A^{-1} u_3)\}. \end{aligned} \tag{2.49}$$

Applying this expression to the BFGS update (2.46), then rearranging and simplify-

ing, we find that

$$\det B^{(\text{new})} = \det B \times \frac{\langle \Delta y, \Delta x \rangle}{\langle \Delta y, B\Delta x \rangle}. \tag{2.50}$$

Now suppose that we want the determinant of $B^{(\text{new})}$ to be at least as large as the determinant of B times some factor $\gamma \in (0, 1)$. This is equivalent to asking that

$$\frac{\langle \Delta y, \Delta x \rangle}{\langle \Delta y, B\Delta x \rangle} \geq \gamma. \tag{2.51}$$

To satisfy this condition, Powell (1978) proposes to replace Δy by a damped version $t\Delta y + (1-t)B\Delta x$, and then find the largest step $t \in [0, 1]$ that satisfies (2.51). In effect, $B\Delta x$ represents the curvature information accumulated from previous iterations, and Δy is the new curvature. Plugging the damped update into (2.51), we arrive at

$$t \leq (1 - \gamma)\frac{\langle \Delta x, B\Delta x \rangle}{\langle B\Delta x - \Delta y, \Delta x \rangle}. \tag{2.52}$$

There are two cases to consider. If the largest t that satisfies the above condition is greater than 1 or, in other words, if

$$\langle \Delta y, \Delta x \rangle \geq \gamma \langle \Delta x, B\Delta x \rangle, \tag{2.53}$$

then set t to 1. This corresponds to an un-damped update. Otherwise, set t so that it satisfies (2.52) with equality. In the stochastic setting, the damped update should be modified so that t is bounded by some decreasing sequence of step sizes, such as $\{a_k\}$. For application to stochastic approximation, I found that setting the damping factor γ to a number between 0.75 an 0.9 worked well. For unconstrained problems, a computational complexity of $O(n^2)$ can be achieved at each iteration by keeping track of both a quasi-Newton approximation to the Hessian and its inverse. In practice, I found that damped BFGS updates proved to be much more stable than the Barzilai-Borwein approximation for non-convex optimization problems.

To conclude this section, I would like to briefly note that it is an open question how—or whether it is possible—to implement damped versions of limited-memory quasi-Newton methods for stochastic approximation. These methods are ideally suited to large problems, as computing the primal-dual search direction with a limited-memory quasi-Newton approximation imposes a relatively small additional cost over the steepest descent direction (Byrd et al., 1994; Waltz et al., 2006). It is not clear, however, how the notion of damped updates can be implemented in a principled manner in limited-memory representations. Limited-memory updates are used in Schraudolph et al. (2007a), but they do not address the question I pose here, nor is it clear whether their approach extends to other stochastic approximation problems, constrained or not.

2.5 On-line L1 regularization

In this section, I apply my previous findings to the problem of computing an L_1-regularized least squares estimator in an "on-line" manner; that is, by making adjustments to each new training example without having to review previous training instances. The results presented in this section extend quite naturally to other supervised and unsupervised learning problems. I start with some background behind linear regression (Sec. 2.5.1) and L_1 regularization (Sec. 2.5.2), motivate the on-line learning approach and connect it to stochastic approximation (Sec. 2.5.4), draw some experimental comparisons with existing methods (Sec. 2.5.7), then show that the proposed algorithm can also be used to filter spam (Sec. 2.5.8).

2.5.1 Linear regression

Consider the problem of learning a simple discriminative regression model under full supervision. Suppose we are provided with a collection of n data points a_i paired with scalar outputs y_i. It is a regression problem when the outputs are numbers on the real line. Each of the training examples is a vector of length m, and each vector element is an observed feature, so that a_{ij} is the jth feature induced by the ith training example.

Treating each of the pairs (a_i, y_i) as independent and identically distributed events, our goal is to come up with a conditional probability distribution that is able to accurately predict an output y_i given an input vector a_i. The simple model assumed here is a normal conditional probability density with mean $a_i^T w$ and fixed variance σ^2, where w is the vector of regression coefficients of length m. The coefficients are the parameters of our regression model.

The least squares estimate w minimizes the mean squared error,

$$\mathrm{MSE}(w) = \tfrac{1}{2n} \textstyle\sum_{i=1}^{n} \big(y_i - \sum_{j=1}^{m} a_{ij} w_j\big)^2. \tag{2.54}$$

For a simple regression model, minimizing the mean squared error (2.54) is equivalent to maximizing the likelihood of the training data. Linear regression based on the maximum likelihood estimator is one of the basic statistical tools of science and engineering and, while primitive, generalizes to many popular statistical estimators: logistic regression, linear discriminant analysis, ANOVA, probabilistic kernel methods, kernel methods and support vector machines, Gaussian processes, and boosting (Friedman et al., 2000; Hastie et al., 2001; Rasmussen & Williams, 2006). Adopting matrix-vector notation, the mean squared error measure can be written as

$$\mathrm{MSE}(w) = \tfrac{1}{2n} \|y - Aw\|^2, \tag{2.55}$$

where A is the $n \times m$ matrix of training examples in which each row corresponds to a single training example a_i, y is a vector of training outputs of length n, and $\|x\|$ is the Euclidean norm of vector x.

2.5.2 L1 regularization

Because the least squares estimator is unstable when m is large, it can generalize poorly to unseen examples. The standard cure is "regularization." The classical Tikhonov regularization technique, for example, favours estimators with a small L_2 or Euclidean norm (Hastie et al., 2001). An L_2 penalty corresponds to a zero-mean normal prior on the regression weights w. Regularization introduces bias, but typically produces estimators that are better at predicting the outputs of unseen examples. The mean squared error with an L_1-penalty instead,

$$\mathrm{MSE}^{(L_1)}(w) = \tfrac{1}{2n}\|y - Aw\|^2 + \tfrac{\lambda}{n}\|w\|_1, \tag{2.56}$$

not only prevents overfitting, but tends to produce estimators that shrink many of their components w_j to zero, resulting in sparse codes (Lee et al., 2008; Olshausen & Field, 1997; Ravikumar et al., 2008).[12] Here, $\|x\|_1$ is the L_1 norm of vector x, which is the sum of absolute values of x. The scalar $\lambda > 0$ controls for the level of regularization (Sardy et al., 2000). Regularization based on the L_1 has been independently studied in a variety of research contexts—by statisticians as the Lasso (Tibshirani, 1996), by signal processing engineers as basis pursuit denoising (Chen et al., 1999), and by mathematicians as total variation denoising (Rudin et al., 1992)—precisely because it is effective at choosing useful features for prediction. Regularization based on the L_1 norm also has important connections to the study of compressed sensing (Candès et al., 2006; Donoho, 2006). See Tropp (2006) for a survey of recent mathematical developments on learning with L_1 regularization.

When the loss function is formulated from the negative log-likelihood, the L_1 penalty has the direct interpretation as a hierarchical Bayesian prior on the regression coefficients w. The connection is made by introducing independent normal priors with zero mean and variances τ_j on the regression coefficients w_j, and in turn sampling each of the latent variances τ_j from an exponential prior with mean $1/\gamma$ (Figueiredo, 2003). Integrating out the latent variance τ_j leads to the Laplace density

$$p(w \,|\, \gamma) = \textstyle\int p(w \,|\, \tau)\, p(\tau \,|\, \gamma)\, d\tau \propto \exp(-\sqrt{\gamma/2}\,|w|), \tag{2.57}$$

which is equivalent to the L_1 penalty in (2.56) when the penalty strength is λ is equal to $\sqrt{\gamma/2}$. Analogous connections have been made between more sophisticated hierarchical priors and norms on groups of variables (van den Berg et al., 2008; Yuan & Lin, 2006). The equivalence to the Laplace prior can be used to derive an expectation maximization (EM) algorithm, in which the E step consists of computing expectations with respect to the latent variables τ_j. The EM formulation leads to a non-convex problem, and is subsumed by the sub-gradient method described below.

[12] Regularization may also be posed instead as a constraint on the L_1 norm.

2.5.3 Stochastic gradient and the Widrow-Hoff delta rule

The partial derivative of the mean squared error (2.54) with respect to a single regression weight is

$$\frac{\partial \text{MSE}}{\partial w_j} = -\frac{1}{n} \sum_i a_{ij}(y_i - a_i^T w).$$

The gradient can be written more compactly with matrix-vector notation:

$$\nabla \text{MSE} = -\frac{1}{n} A^T (y - Aw). \tag{2.58}$$

We can treat the gradient of the mean squared error as a sample expectation over responses of the form

$$g_i = -a_i(y_i - a_i^T w). \tag{2.59}$$

So the on-line or stochastic update

$$w^{(\text{new})} = w + t a_i(y_i - a_i^T w), \tag{2.60}$$

improves the linear regression with only a single data point ($t > 0$ is the step size). This is the famed "delta rule" of Widrow and Hoff (Mitchell, 1997).[13]

The on-line update (2.60) corresponds to the Robbins-Monro update (2.1) if we make the following connection: set the iterate x to be the vector of regression coefficients w, define the objective $f(x)$ to be the mean squared error $\text{MSE}(w)$, define the noisy gradient estimate to be (2.59), and the ith iteration of stochastic approximation corresponds to the ith data pair (a_i, y_i). As I showed above, the expectation of the the stochastic gradients g_i recovers the exact gradient ∇MSE.

Since standard "batch" learning requires a full pass through the data for each gradient evaluation, an on-line update of the form (2.60) may be the only viable option when faced with a genetic sequence (Xing et al., 2001), an overcomplete wavelet dictionary (Chen et al., 1999), or an image collection obtained from a web-based annotation tool (Russell et al., 2008). On-line learning for regression and classification—including L_2 regularization—is a well-researched topic, particularly for neural networks (Saad, 1998) and support vector machines (Kivinen et al., 2004; Shalev-Shwartz et al., 2007). On-line learning with L_1 regularization, despite its ascribed benefits, has strangely avoided study. To our knowledge, the only published work that has approached the problem is Zheng (2005) using sub-gradient methods.

In the following, I present an on-line L_1-regularized learning rule based on the previously proposed interior-point stochastic approximation method. Many specialized implementations of interior-point methods have been developed for L_1-regularized

[13] The descent direction given by the negative gradient can be a poor choice because it ignores the scaling of the problem. Much work has focused on improving the delta rule (Amari, 1998; Roux & Bengio, 2000; Schraudolph et al., 2007a), sometimes at an increased cost per iteration. In this study, I stick to learning based on the gradient direction.

least squares problems (Chen et al., 1999; Johnson et al., 2000; Kim et al., 2007). In the derivations below, I replace the mean squared error with a generic loss function $\ell(w)$, so that the connection is immediately made to other L_1-regularized learning problems such as logistic regression (which is used to implement the spam filter in Sec. 2.5.8), probit regression, restricted Boltzmann machines (Lee et al., 2008), and on-line learning for conditional random fields (Vishwanathan et al., 2006).

In principle, feature selection comes "for free" with L_1 regularization. But in reality, the penalized objective is difficult to optimize due to the non-differentiability of the absolute values around zero. There are two possible strategies for coping with this difficulty. The first is to modify existing unconstrained optimization methods to handle nonsmoothness of the objective. This is the strategy behind the sub-gradient method (Sec. 2.5.6). The second approach is to reformulate the nonsmooth, unconstrained problem as a smooth but constrained optimization problem. This is accomplished through introduction of auxiliary variables. I then describe interior-point (Sec. 2.5.4) and projection (Sec. 2.5.5) approaches to solving the constrained formulation for the case when we have on-line estimates of the gradient.

2.5.4 Primal-dual interior-point method

In this section, I describe two alternative constrained formulations of the L_1-regularized objective. The first formulation is described in Chen et al. (2001) and Tibshirani (1996). It consists of splitting the regression coefficients w into positive and negative components. Since the optimization variables are all constrained to be positive, the non-differentiable sum of absolute values can be replaced by a simple sum. The second idea is to introduce auxiliary variables that bound the magnitude of the regression coefficients (Kim et al., 2007). This approach also used in Sec. 11.8.2 of Boyd and Vandenberghe (2004) for the related problem of minimizing the L_1 norm of a linear system of equations. I will discuss the pros and cons of each of these two formulations.

There is actually a third, lesser-known constrained formulation that Andersen (1996) suggested for the related problem of minimizing the sum of Euclidean norms.[14] The basic idea is to replace the absolute values by squares of square roots, and then introduce an additional variable that prevents the square roots from ever reaching zero. While this approach was originally suggested for a primal interior-point method, it is easily extended to a primal-dual algorithm. In practice, however, I found that this approach was inferior to the first two, likely due to instability of the derivatives.

First constrained formulation. We arrive at the first constrained formulation by dividing the regression weights into positive and negative components like so: $w = w_+ - w_-$, where $w_+ \geq 0$ and $w_- \geq 0$. In doing so, we double the number of optimization variables. To simplify the presentation, I use x to represent the full set of $2m$ optimization variables (w_+, w_-). I can now rewrite the objective and state the

[14]The problem of minimizing the sum of Euclidean norms and the discovery of its dual has an interesting historical connection (Andersen et al., 2000).

constrained optimization problem:

$$\begin{array}{ll} \text{minimize} & f(x) = \ell(w) + \frac{\lambda}{n} \sum_{j=1}^{2m} x_j, \\ \text{subject to} & x \geq 0. \end{array} \tag{2.61}$$

where $\ell(w)$ is the specified loss function. For instance, for a standard linear regression model the loss function $\ell(w)$ is given by the mean squared error MSE(w). The loss function for a logistic regression model is derived in Sec. 2.5.8. What we have here is an optimization problem with bound constraints.

The main ingredient in the interior-point method is the primal-dual Newton step *i.e.* the solution to (2.16). From the augmented system (2.20), the primal Newton step for (2.61) is given by the simple expression

$$(\nabla^2 \ell(w) + \Sigma)\Delta x = -\nabla \ell(w) - \tfrac{\lambda}{n}\mathbf{1} + \mu/x, \tag{2.62}$$

where $\Sigma = X^{-1}Z$. There is one dual variable for each of the $2m$ bound constraints. The dual step Δz is recovered according to

$$\Delta z = \mu/x - z - \Sigma \Delta x. \tag{2.63}$$

It is now easy to implement the interior-point stochastic approximation method for on-line learning with L_1 regularization. The regularized on-line update is obtained by direct application of the primal-dual interior-point search direction (2.62,2.63), with a stochastic gradient estimate in place of the gradient $\nabla \ell(w)$, and identity in place of the Hessian $\nabla^2 \ell(w)$. Similarly, the regularized Widrow-Hoff delta rule is obtained by substituting $\nabla \ell(w)$ for the on-line gradient observation g_i as it is defined in (2.59).

Since a feasible primal-dual iterate must lie within the positive quadrant, it is easy to calculate the largest step length $a_k > 0$ that satisfies the constraints without having to execute a full backtracking line search (Fig. 2.1). Consider a single entry u of the vector (x, z). When the corresponding search direction Δu is positive, u moves away from the boundary, and so any step size is acceptable. When the search direction is negative, the largest acceptable step size is given by $-u/\Delta u$.

The principle drawback of this constrained formulation is that the Hessian of the loss function with respect to the transformed coefficients (w_+, w_-) has unbounded condition number. This was not an obstacle to implementation of the Widrow-Hoff delta rule because I replaced the Hessian by the identity matrix. However, if we were to include second-order information (say, using the damped quasi-Newton method described in Sec. 2.4), then we need to a different strategy must be considered.

Note when calculating the Lasso estimate (2.56) for several values of λ, say, for cross-validation (Tibshirani, 1996), a good initial guess can be obtained from a previous run (see Sec. 2.2).

Second constrained formulation. Another way to transform the nonsmooth optimization problem into a constrained, differentiable problem is to introduce an auxiliary variable u_j paired with every parameter w_j. The constrained optimization

problem is then given by

$$\begin{array}{ll} \text{minimize} & \ell(w) + \lambda \sum_j u_j \\ \text{subject to} & -u \leq w \leq u. \end{array} \tag{2.64}$$

To simplify the presentation, I set $x = (w, u)$ to be the full collection of optimization variables, and I rewrite the inequality constraints to be of the form $c(x) \leq 0$ by writing the lower bounds on w as $c_L(x) = -w - u$, and the upper bounds on w as $c_U(x) = w - u$. Expanding out the terms in the augmented system (2.20), we obtain

$$-\begin{bmatrix} D_1 - \nabla^2 \ell(w) & D_2 \\ D_2 & D_1 \end{bmatrix} \begin{bmatrix} \Delta w \\ \Delta u \end{bmatrix} = -\begin{bmatrix} \nabla \ell(w) + \mu/c_L(x) - \mu/c_U(x) \\ \frac{\lambda}{n}\mathbf{1} + \mu/c_L(x) + \mu/c_U(x) \end{bmatrix}, \tag{2.65}$$

in which I define the diagonal matrices $D_1 \equiv \Sigma_L + \Sigma_U$, $D_2 \equiv \Sigma_L - \Sigma_U$, $\Sigma_L \equiv C_L^{-1} Z_L$ and $\Sigma_U \equiv C_U^{-1} Z_U$. A solution $\Delta x = (\Delta w, \Delta u)$ to the above linear system gives us the primal Newton search direction. When the Hessian of the loss function is replaced by a diagonal matrix, the matrix above is block-diagonal and is easily decomposed into sparse Cholesky factors. The search direction for the dual variables is recovered according to

$$\Delta z_L = -z_L - \mu/c_L(x) + \Sigma_L(\Delta u + \Delta w) \tag{2.66}$$

$$\Delta z_U = -z_U - \mu/c_U(x) + \Sigma_U(\Delta u - \Delta w), \tag{2.67}$$

where z_L is the set of Lagrange multipliers associated with the lower bounds $c_L(x)$, and z_U is the set of Lagrange multipliers corresponding to the upper bounds $c_U(x)$. As before, the alternative formulation of the regularized delta rule consists of the solution to (2.65-2.67) with an on-line gradient estimate, and with the Hessian replaced by the identity matrix.

Now I ask, what is the largest step length $a > 0$ we can take such that the constraints on the primal and dual variables remain satisfied? The constraints $z \geq 0$ on the dual variables are handled just as I've described above. Next, consider the largest step length such that the single inequality constraint $-u^\star \leq w^\star$ is satisfied (here I omit the subscripts on w_j and u_j). First we need to check whether the search direction $(\Delta w, \Delta u)$ is moving toward the boundary. This occurs when the gradient of $w + u \geq 0$ and the search direction form an angle with each other that is greater than 90 degrees, implying $\Delta w + \Delta u < 0$. If this condition holds, then we need to solve for the largest step size a such that $-u^\star = w^\star$. The solution is given by

$$a = -(w + u)/(\Delta w + \Delta u). \tag{2.68}$$

The search direction $(\Delta w, \Delta u)$ points toward the constraint boundary $w^\star \leq u^\star$ if the gradient of $u - w \geq 0$ and the search direction form an angle greater than 90 degrees; *i.e.* $\Delta u - \Delta w < 0$. If this holds, we solve for a such that $w^\star = u^\star$, obtaining

$$a = -(w - u)/(\Delta w - \Delta u). \tag{2.69}$$

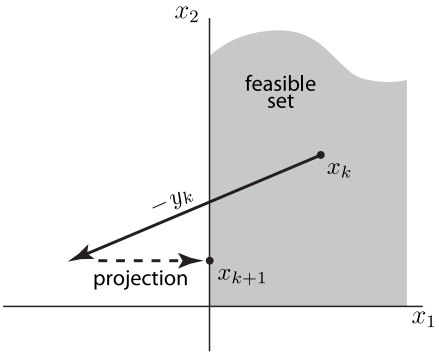

Figure 2.4: Illustration of the projection operator for the constraints $x \geq 0$.

I'd like to conclude this section by pointing out an immediate advantage of the interior-point method over other methods mentioned in this chapter: the derivation of the L_1-regularized delta rule presented here extends without too much effort to more sophisticated regularized schemes that employ, for instance, group norms.

2.5.5 Projected gradient

The projected gradient algorithm is founded on a very simple idea: whenever the iterate lies outside the feasible set, project back to the closest point within the feasible set (Bertsekas, 1982). For the constrained formulation with bound constraints (2.61), this projection is easily done: each new iterate x_{k+1} is taken to be the maximum of the stochastic gradient update (2.1), and 0. This projection operator is illustrated in Fig. 2.4 for a small problem with two variables. The feasible set is depicted by the shaded region. In projected gradient, we first follow the negative gradient. Since the new point violates the constraint $x_1 \geq 0$, we project the iterate to the closest point within the feasible set, which is simply $x_1 = 0$. This example illustrates why projected gradient is well-suited to the task of learning with L_1 regularization: unlike the interior-point method, it sets the regression coefficients exactly to zero. This example also illustrates a potential drawback: when we are near the boundary of the feasible set, the projection operator may induce bias, leading to slow progress. Projected gradient does have asymptotic guarantees of convergence for convex stochastic approximation problems (Bertsekas, 1999; Poljak, 1978).

Formally, the projection operator consists of computing the point $x^\star$ that solves

$$\begin{array}{ll} \text{minimize} & \|x - x^\star\|^2 \\ \text{subject to} & c(x^\star) \leq 0, \end{array} \tag{2.70}$$

where x is the new point that may or may not be feasible. For the bound-constrained formulation, this works out to be a simple maximum operation, as I illustrated above. The projection operator for the alternative constrained formulation (2.64) is also not very hard to derive. For more complicated forms of constraints, however, there may be no easy or closed-form solution to (2.70). Specialized algorithms have been developed

for specific types of constraints—see, for instance, Duchi et al. (2008) and van den Berg et al. (2008)—but in general the projection may introduce a great deal of overhead. The algorithm of Dykstra (1983) applies to any feasible set formed by the intersection of arbitrary convex constraints, but it may be computationally intensive to execute because it works by iteratively projecting onto the individual convex sets.

Another significant drawback is that projected gradient may not work well for poorly-scaled problems. Second-order information can be incorporated into the projected Robbins-Monro updates, but only incompletely (Gafni & Bertsekas, 1984). By contrast, second-order information is easily incorporated into the regularized delta rule derived from primal-dual interior-point methods. See Schmidt et al. (2007) for more details on projected gradient and two-metric projection algorithms applied to learning problems with L_1 regularization.

2.5.6 Sub-gradient method

The sub-gradient method is a direct application of the necessary and sufficient first-order optimality conditions for the L_1-regularized objective. From Schmidt et al. (2007), the optimality conditions are

$$\begin{aligned} \tfrac{\partial f}{\partial w_j} &= -\lambda \quad \text{if } w_j > 0, \\ \tfrac{\partial f}{\partial w_j} &= \lambda \quad \text{if } w_j < 0, \\ \left|\tfrac{\partial f}{\partial w_j}\right| &\leq \lambda \quad \text{if } w_j = 0. \end{aligned} \tag{2.71}$$

In the stochastic version of the algorithm, of course, the partial derivatives are replaced by noisy estimates.

The orthant-wise sub-gradient algorithm of Andrew and Gao (2007) allows us to incorporate second-order information to a limited degree. The orthant-wise steps also have the effect of promoting sparsity, which may be useful in certain cases. Variations on the basic sub-gradient approach are discussed in detail in Schmidt et al. (2007).

2.5.7 Experiments

I ran four small experiments to assess the reliability and shrinkage effect of the interior-point stochastic gradient method for linear regression with L_1 regularization; refer to Fig. 2.1 and the primal-dual Newton step (2.62,2.63). I also studied four alternatives to the proposed method: 1) the sub-gradient method described in Sec. 2.5.6, 2) the projected gradient method described in Sec. 2.5.5, 3) a smoothed, unconstrained approximation to (2.56), and 4) the augmented Lagrangian approach described in Wang and Spall (2003). See Schmidt et al. (2007) for a description o the smoothed approximation, and an in-depth discussion of the merits of applying the first three optimization approaches to L_1 regularization. All these methods have a per-iteration cost on the order of the number of features.

Method. For the first three experiments, I simulated 20 data sets following the procedure described in Sec. 7.5 of Tibshirani (1996). Each data set had $n = 100$

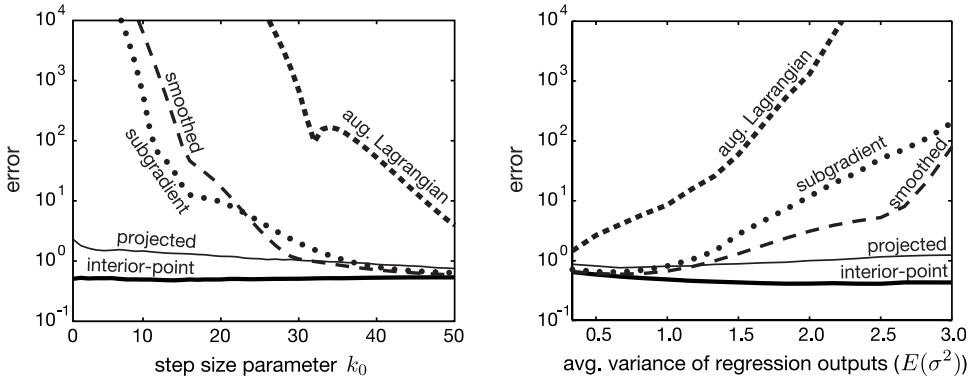

Figure 2.5: *(left)* Performance of constrained stochastic gradient methods for different step size sequences. *(right)* Performance of methods for increasing levels of variance in the dimensions of the training data. Error is measured in terms of the difference between the exact solution and the on-line estimate. Note the logarithmic scale in the vertical axis.

observations with $m = 40$ features. The observations were given by $x_{ij} = z_{ij} + z_i$, where each z_i was drawn from the standard normal, and each z_{ij} was drawn independently and identically from the normal distribution with variance σ_i^2. The variances σ_i^2 were, in turn, drawn from the inverse Gamma with shape 2.5 and scale $\nu = 1$. Note that the mean of the inverse Gamma variance σ_i^2 is proportional to ν. The true regression coefficients were defined to be $w = (0, \ldots, 0, 2, \ldots, 2, 0, \ldots, 0, 2, \ldots, 2)^T$ with 10 repeats in each block of zeros and twos. Outputs were generated according to $y_i = w^T x_i + \epsilon$ with standard Gaussian noise ϵ. Each stochastic gradient method was executed with a single pass on the data (100 iterations) with maximum step sizes $\hat{a}_k = 1/(k_0 + k)$, where $k_0 = 50$ by default. I chose the L_1 penalty parameter to be $\lambda/n = 1.25$, which tended to produce about 30% zero coefficients at the solution to (2.56). The augmented Lagrangian method required a sequence of penalty terms r_k that converge to zero; after some trial and error, I chose $r_k = 50/(k_0 + k)^{0.1}$. The control variables of Experiments 1, 2 and 3 were, respectively, the step size parameter k_0, the inverse Gamma scale parameter ν, and the L_1 penalty parameter λ. In the fourth and final experiment, each example y_i in the training set had $m = 8$ features, and the true coefficients were set to $w = (0, 0, 2, -4, 0, 0, -1, 3)^T$.

Results. Fig. 2.5 shows the results of Experiments 1 and 2. The vertical axis in each plot is the error measure $\frac{1}{n}\|w^{\text{exact}} - w^{\text{on-line}}\|_1$ averaged over the 20 data sets, in which w^{exact} is the solution to (2.56), and $w^{\text{on-line}}$ is the estimate obtained after 100 iterations of the on-line or stochastic gradient method. With a large enough k_0, almost all the methods converged close to w^{exact}. The stochastic interior-point method, however, always came closest to w^{exact} and, for the range of values I tried, its solution was insensitive to the step size sequence and level of variance in the observations. These results suggest that the interior-point steps are the most stable in the face of noisy gradient measurements. These results also suggest that projecting

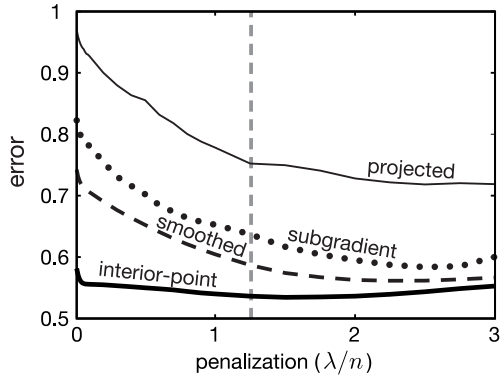

Figure 2.6: Performance of the methods for various choices of the L_1 penalty. Error is the difference between the exact solution and the on-line estimate.

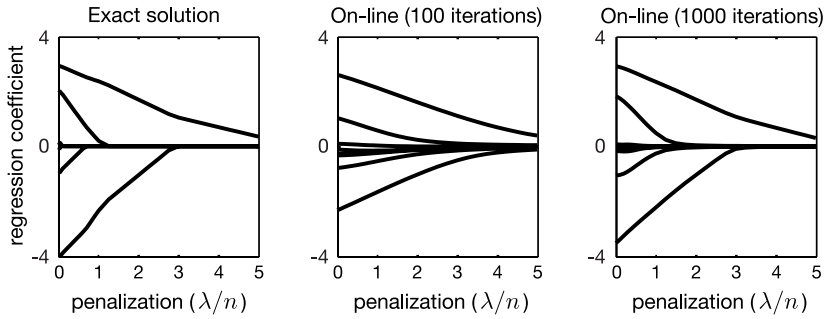

Figure 2.7: Shrinkage effect for different choices of the L_1 penalty parameter.

the gradient onto the feasible set, when possible, appears to be a reasonable second choice, though it is still significantly worse than the interior-point method. Fig. 2.6 shows results from Experiment 3. Results at $\lambda/n = 1.25$ (the dotted vertical line) correspond to $k_0 = 50$ and $E(\sigma^2) = 2/3$ in the left and right plots of Fig. 2.5. Even with well-chosen step sizes for all methods, the stochastic interior-point method still best approximates the exact solution, and its performance does not degrade when λ is small or large.

Fig. 2.7 shows the regularized estimates from Experiment 3 as a function of λ/n. The plot on the left is the exact regularized path. The plot in the middle is the on-line regularization path after one pass through the training data, and the rightmost plot is the on-line regularization path after 10 passes through the data. After one pass through the data—equivalent to a *single* iteration of an exact solver—the interior-point stochastic gradient method shrank some of the data components, but didn't quite discard irrelevant features altogether. After 10 visits to the training data, the stochastic algorithm exhibited feature selection close to what we would normally expect from the Lasso *(left)*.

2.5.8 Filtering spam

Classifying email as spam or not is most faithfully modeled as an on-line learning problem in which supervision is provided *after* each email has been designated for the inbox or trash (Cormack, 2006; Cormack & Bratko, 2006). An effective filter is one that minimizes misclassification of incoming messages. Throwing away a good email is, of course, considerably more deleterious than incorrectly placing a spam in the inbox. Without any prior knowledge as to what spam looks like, any filter will be error prone at initial stages of deployment.

Spam filtering necessarily involves lots of data and an even larger number of features, so a sparse, stable model is essential. I adapted the L_1-regularized delta rule (2.62,2.63) to the spam filtering problem, in which the loss function is the negative log-likelihood of the training data with respect to a binary logistic regression model (Hastie et al., 2001).

The spam filter I chose to use is a discriminative classifier similar to the regression model described in Sec. 2.5, except that the outputs y_i are now binomial random variables, whereby spam is labeled as $y_i = 1$. As such, it makes sense to model the binary labels as simple coin tosses:

$$p(y_i \,|\, a_i) = \phi(a_i, w)^{y_i}(1 - \phi(a_i, w))^{1-y_i}, \tag{2.72}$$

where the function $\phi(a_i, w)$ represents the probability that y_i equals 1. The logistic regression approach is to construct this probability with a sigmoid function, with the sigmoid input parameterized by a linear decision rule:

$$\phi(a_i, w) = 1/(1 + \exp(-a_i^T w)). \tag{2.73}$$

The sigmoid function will always give us a number between 0 and 1.

The learning problem is to find the linear decision rule that maximizes the log-likelihood. This yields the following loss function.

$$\ell(w) = -\textstyle\sum_i y_i \log d_i - \sum_i (1 - y_i) \log(1 - d_i). \tag{2.74}$$

where I've introduced the shorthand notation $d_i \equiv \phi(a_i, w)$. The partial derivatives have a very similar form to those obtained for linear regression:

$$\nabla \ell = -A^T(y - d), \tag{2.75}$$

where d is a vector of length n containing the entries d_i. The matrix of second-order partial derivatives is

$$\nabla^2 L = A^T D A, \tag{2.76}$$

where D is the matrix with entries $d_i(1 - d_i)$ along its diagonal. The on-line gradient estimates are nearly identical to those for the linear regression model (2.59), the only difference being that the expectation over quantities $a_i^T w$ is replaced by an expectation

over responses $\phi(a_i, w)$. Just as we found for the linear regression model in Sec. 2.5.1, the on-line learning updates for logistic regression have a direct connection to the Robbins-Monro recursion.

I incorporated second-order information from the diagonal of the on-line estimate of the Hessian—without sacrificing the linear cost of each iteration—because the optimization problem can be poorly scaled due to the shape of the logistic function. To my knowledge, no one has investigated this approach to on-line spam filtering, though there is some work on logistic regression plus the Lasso for batch classification in text corpora (Genkin et al., 2007). Needless to say, batch learning is completely impractical in this setting.

Method. I simulated the on-line spam filtering task on the TREC2005 corpus (Cormack & Lynam, 2005). It contains email messages from the legal investigations of Enron corporation (Klimt & Yang, 2004). I compared the proposed on-line classifier (with parameter settings $\lambda = 1.5$, $\hat{a}_i = 1/(1+i)$) to two open-source software packages, SpamBayes version 1.0.3 and Bogofilter 0.93.4.[15] A full comparison of on-line learning methods is certainly beyond the scope of this thesis; see Cormack and Lynam (2007) for a comprehensive evaluation. I represented each email as a vector a_i of normalized word frequencies (Joachims, 2002, p. 21). I used the Python interface of SpamBayes to extract the word tokens from each email.

In the end, the data set for the on-line learning problem consisted of $n = 92189$ documents and $m = 823470$ features. The feature responses are extremely sparse so it is easy to store the data. There is, however, no obvious way to take advantage of this sparsity structure to train the classification model, so calculating the gradient (2.74) at each point w would require on the order of $n \times m \approx 7.6 \times 10^{10}$ floating-point operations. It is in this sense that batch learning is impractical.[16]

Results. Following Cormack and Lynam (2007), I present the results of the on-line spam filtering experiment in contingency tables (Table 2.1). The top-right entry in each contingency table is the number of misclassified spam, and bottom-right entry is the number of misclassified non-spam. Let me stress that evaluation of the spam filter was conducted on-line. I tagged an email for deletion only if the logistic model was at least 98% certain it was spam. The proposed L_1-regularized spam filter dominated SpamBayes on the TREC2005 corpus, and performed comparably to Bogofilter—one of the best spam filters to date (Cormack & Lynam, 2007). The computational expense of my method was only slightly greater than the other two. As I found in Sec. 2.5.7, assessing the level of sparsity of the on-line solution was more difficult than in the exact case. I can say, however, that removing the 43% smallest entries of w resulted in almost no (< 0.001) change to the probabilities $p(y_i = \text{spam})$.

[15] These packages are publicly available at spambayes.sourceforge.net and bogofilter.sourceforge.net.

[16] When the entire data set is known in advance, there is a way to exploit the sparsity of the $m \times n$ feature matrix. The solution described by Sha et al. (2007) is to cast L_1-regularized logistic regression as a non-negative quadratic program and implement the multiplicative updates without explicitly constructing the feature matrix.

		true	
		not spam	spam
pred.	not spam	39382 (42.72%)	3291 (3.56%)
	spam	17 (0.02%)	49499 (53.69%)

Results for SpamBayes

		true	
		not spam	spam
pred.	not spam	39393 (42.73%)	5515 (5.98%)
	spam	3 (0.00%)	47275 (51.28%)

Results for Bogofilter

		true	
		not spam	spam
pred.	not spam	39384 (42.72%)	2438 (2.64%)
	spam	15 (0.02%)	50352 (54.62%)

Results for Logistic + L1

Table 2.1: Contingency tables for on-line spam filtering task on the TREC2005 data.

2.6 Conclusions and discussion

In summary, Robbins and Monro (1951) proposed a method for finding the root of a nonlinear equation $F(x) = 0$ when we only have access to noisy measurements of $F(x)$. I described an algorithm that solves a constrained optimization problem by finding the root of a moving target $F_\mu(x, z)$, and this target is given by the primal-dual system that arises in interior-point methods. Both theoretical and empirical evidence presented in this chapter show that the interior-point stochastic approximation algorithm is a significant improvement over other methods. It is effective because it is able to take large steps without violating the constraints. The interior-point approach also has the virtue of being much more general, and our analysis guarantees that it will be numerically stable. Since the proposed method imposes few restrictions formulation of the constraints, my hope is that the research presented in this chapter suggests new ways to approach problems in machine learning.

There are still many immediate issues that remain, in principle, open to investigation: the choice of centering sequence $\{\sigma_k\}$, gradient-free extensions, guarantees on rates of convergence, and scaling of the gradient descent direction. Also, much remains to be understood in terms of the behaviour of stochastic approximation methods in the particular context of maximum likelihood estimation subject to L_1 regularization.

In this chapter, on-line learning with L_1 regularization served as the principle motivation for solving a stochastic approximation problem subject to constraints. It

remains to be seen whether the on-line learning approaches presented here extend to more challenging domains. For instance, sparse coding (Olshausen & Field, 1997) is an important application of L_1 regularization to the unsupervised learning setting. Applications of sparse coding methods are limited in part by the size of the data set (Lee et al., 2007), so it would be of great interest to develop on-line learning methods for sparse coding. One major obstacle, however, is that the sparse coding objective is no longer convex. One potential solution would be a convex relaxation obtained via a sparse implementation of semidefinite programming methodology (Alizadeh et al., 1998; Vandenberghe & Boyd, 1996).

I'd like to note to the reader, in closing, that the search direction in finite-precision arithmetic may be biased even when the search direction in rational arithmetic is unbiased. What this means to the practitioner is that it may be too much to expect the stochastic approximation algorithm to converge to the solution, even in the limit; we may have to settle for convergence within a small region of the solution, where the size of the region depends on the variance of the gradient measurements and the numerical properties governing computation of the search direction. This observation pertains to almost any stochastic approximation method that employs a search direction other than the steepest descent direction.

Chapter 3

A stochastic approximation method for probabilistic inference

In this chapter, I describe a new approach to inference in probabilistic models for inference problems in which the exact solution is intractable. My new approach adopts the methodology of variational inference, but unlike existing variational methods such as mean field (Blei et al., 2003; Jordan et al., 1998), expectation propagation (Minka, 2001a) and approximate message passing algorithms (Aji & McEliece, 2001; Yedidia et al., 2005), it imposes no structural or analytical approximations.

Recall, in the introduction I discussed how the key idea behind variational inference is to come up with a family of approximating distributions $q(x;\theta)$ that have "nice" analytic properties, then to optimize some criterion in order to find the distribution parameterized by θ that most closely matches the target posterior $p(x)$. All variational inference algorithms, including belief propagation and its generalizations (Yedidia et al., 2005), expectation propagation (Minka, 2001b), mean field (Blei et al., 2003; Jordan et al., 1998), and even EM (Neal & Hinton, 1998), can be derived from a common objective, the Kullback-Leibler divergence (Cover & Thomas, 1991). The major drawback of variational methods is that the best approximating distribution may still impose an unrealistic or questionable factorization, leading to excessively biased estimates (see Fig. 3.1, left-hand side).

In this chapter, I describe a new variational method that does not have this limitation: it adopts the methodology of variational inference without being restricted to tractable classes of approximate distributions (see Fig. 3.1, right-hand side). The catch is that the variational objective—that is, the Kullback-Leibler divergence—is difficult to optimize because its gradient cannot be computed exactly. So to descend along the surface of the variational objective, I propose to employ stochastic approximation (Robbins & Monro, 1951) with Monte Carlo estimates of the gradient, and update these estimates over time with sequential Monte Carlo, or SMC (Del Moral et al., 2006)—hence, *a stochastic approximation method for probabilistic inference*. A common criticism of Monte Carlo methods is that they are computationally intensive, but the approach I describe can actually be less expensive than alternative variational methods, as I discuss.

Large gradient descent steps may quickly lead to a degenerate sample, so I introduce a mechanism that safeguards the variance of the Monte Carlo estimate at each iteration (Sec. 3.3.8). This variance safeguard mechanism is unlike existing variance-control techniques from the SMC and particle filter literature because it does not make the standard *effective sample size* (ESS) approximation (Doucet et al., 2000b), hence I claim that it more accurately monitors the variance of the sample.

Indirectly, the variance safeguard mechanism provides a way to obtain an estimator that has low variance at the expense of a biased solution (Steen, 1982). Bias

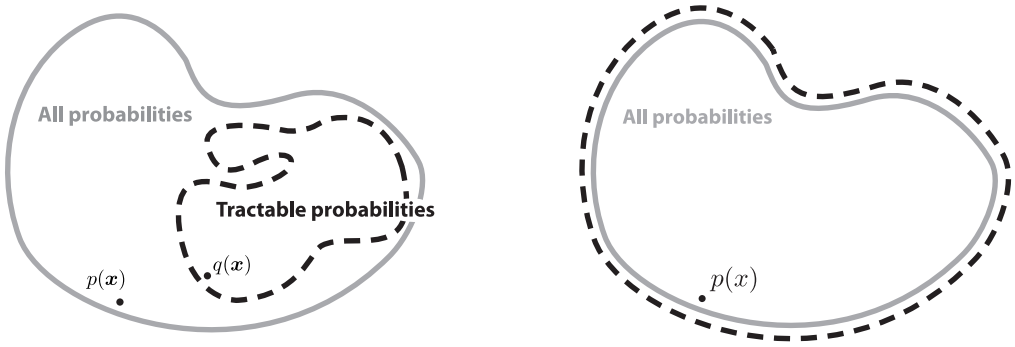

Figure 3.1: The guiding principle behind standard variational methods *(left)* is to find the approximating density $q(x;\theta)$ parameterized by θ that is closest to the distribution of interest $p(x)$, yet remains within the defined set of tractable probability distributions. In my approach *(right)*, the class of approximating probability densities always coincides with the target $p(x)$.

in no way disqualifies an estimator, for the small variance of an estimator can compensate for bias. It is important to note that achieving this trade-off between bias and variance in no way precludes Bayesian inference; we still obtain a distribution over the quantities of interest, not a point estimate. To my knowledge, the stochastic approximation algorithm is the first general means of achieving such a trade-off and, in so doing, it draws meaningful connections between two divergent philosophies on approximate inference, Monte Carlo simulation and variational methods.

The advantage of the stochastic approximation method with respect to other variational methods is clear: it does not restrict the class of variational densities by making assumptions about their structure. The advantage of my approach compared to sequential Monte Carlo methods such as annealed importance sampling (AIS) (Neal, 2001) is less obvious. One advantage is that there is no need to design the sequence of SMC distributions as it is a direct product of the algorithm's derivation (Sec. 3.3). That is, the algorithm both samples from a sequence of distributions, and knows what those distributions should be. I discuss the merits of this adaptive sequence of distributions relative to other SMC schemes in Sec. 3.2 of this chapter.

Importantly, the stochastic approximation method applies to a large set of intractable probabilistic inference problems. Examples in this set include hidden Markov models (Murphy, 2002), mixture models (Celeux et al., 2000), continuous-time Markov processes (Holmes & Rubin, 2002), and Markov random fields defined on planar lattices which are used extensively in the field of computer vision (Carbonetto et al., 2004; Li, 1994; Sun et al., 2008). The proposed inference framework can be used whenever expectation maximization (EM) or the two-stage Gibbs sampler can be used. (Note that the scope could be extended without too much difficulty to a much larger set of probabilistic models, but for the purpose of this study it would add needless complications.)

The immense flexibility of this new approximate inference framework can be both

a blessing and a curse: it leaves open a great deal of opportunity for exploiting the structure of the model, but at the same time it bestows on the user responsibility for designing an effective parameterization of the variational approximation. Note that the variance control mechanism can be sensitive to the number of simulation steps. Addressing this limitation is an important open question. These points will become more clear as I describe the stochastic approximation algorithm in detail in Sec. 3.3.

I investigate the behaviour of the proposed algorithm in two open probabilistic inference problems, one in statistical physics and another in population genetics. The latter task is to simultaneously infer the latent structure of a population and the ancestry of individuals based on their genetic profiles. The difficulty of obtaining reliable inferences continues to pose an obstacle to empirical studies of the genetic basis of evolution. As I show in Sec. 3.5, the existing Markov chain Monte Carlo (MCMC) methods are prone to producing very different answers in independent simulations, and they fail to adequately capture the uncertainty in its solutions. For many population genetics applications such as wildlife conversation (Coltman, 2007), it is crucial to accurately characterize the confidence in a solution.

In both applications, I compare my algorithm to a special instance of sequential Monte Carlo called annealed importance sampling (Jarzynski, 1997; Neal, 2001). Annealed importance sampling (AIS) derives the artificial sequence of distributions from a tempering scheme. Since AIS is a close relative to the proposed stochastic approximation method, it serves as the main point of comparison throughout the experiments. Experiments in Section 3.5 show that when existing methods do not provide satisfactory answers, my new algorithm provides a compelling alternative. The experiments demonstrate reductions in variance and improvements in accuracy at a comparable cost.

In summary, I propose a new stochastic approximation method for probabilistic inference, so called because it employs stochastic approximation, with Monte Carlo estimates of the gradient to descend along the surface of the variational objective. It is also a population-based Monte Carlo method (Fearnhead, 2008; Jasra et al., 2007) because it uses the machinery of sequential Monte Carlo to update the estimate of the gradient at each iteration. This new algorithmic framework for probabilistic inference has four main ingredients: one, a variational optimization problem framed using the Kullback-Leibler "distance" measure; two, the implementation of a sequential Monte Carlo method for constructing stochastic estimates of the gradient of the variational objective; three, a stochastic approximation method for finding a solution to the variational optimization problem; and four, a way to safeguard the variance of the importance weights at each iteration of the stochastic approximation algorithm. Sec. 3.3 describes each of these ingredients in detail.

Before diving into the full derivation of the stochastic approximation algorithm, I first take a couple paragraphs (Sec. 3.1) to argue for the originality of my work. Then, in Sec. 3.2, I contrast my work to existing work on sequential Monte Carlo.

3.1 Related work

The interface of optimization and simulation strategies for inference has been explored before. However, none of the previous approaches resemble the one proposed in this paper. De Freitas et al. (2001) use a variational approximation to formulate a Metropolis-Hastings proposal. I also use a variational proposal, but in my work the proposal is adapted over time. Recent work on adaptive MCMC (Andrieu & Moulines, 2006; Haario et al., 2001) combines ideas from both stochastic approximation and MCMC to automatically learn better proposal distributions. I, too, use stochastic approximation to adapt the proposal distribution, but my algorithm does not inherit some of the limitations of MCMC. Confusingly, this thesis chapter is also unrelated to Liang et al. (2007), despite having very similar titles. In that paper, stochastic approximation is applied to improving the Wang-Landau algorithm (Landau et al., 2004). Younes (1991) employs stochastic approximation to estimate the parameters of an undirected graphical model that maximize the likelihood, in which the stochastic gradient measurement at each iteration is obtained via MCMC simulation. Younes (1991) adopts stochastic approximation for maximum likelihood learning of undirected graphical models. My goal is not to compute a single mode, but rather to adopt the "Bayesian" approach (Fienberg, 2006; Tipping, 2004) and estimate the full distribution. Also, Younes (1991) uses MCMC to estimate the gradient, hence is bound by its limitations. Stochastic approximation has also been used in the context of variational inference for updating the parameters in an on-line fashion (Sato, 2001). This work, however, builds on traditional variational inference strategies that make assumptions regarding the conditional independence structure of the model.

Related work on the *cross-entropy method* (de Boer et al., 2005; Rubenstein & Kroese, 2004) deserves special attention: like my work, it uses importance sampling and optimization to infer an expectation, but it exhibits no similarities beyond that. Work on the cross-entropy method focuses on estimating of the probability of rare events (Heidelberger, 1995). Naive Monte Carlo simulation techniques may have to run for a very long time before coming up with an accurate answer, so the cross-entropy provides an appealing solution by progressively adapting the sampling distribution to increase the efficiency of the Monte Carlo estimator. The key limitation of the cross-entropy method is that samples must be drawn directly the auxiliary and target distributions; we make no such assumption, hence our work can handle a much broader class of inference problems. Another key difference is that the cross-entropy method involves solving a succession of optimization problems (with exact measurements of the gradient) whereas our method optimizes the same objective throughout (with noisy estimates of the gradient). It is quite possible the cross-entropy method is complementary to our own approach, and its innovations might enhance the applicability of our framework to the simulation of rare events.

3.2 Relationship to other SMC methods

Annealed importance sampling (Neal, 2001), hot coupling (Hamze & de Freitas, 2006), the stochastic approximation method presented here, and *conditional mean field*—the algorithm presented in the next chapter—are all algorithms for approximate probabilistic inference that (intentionally or not) adopt the sequential Monte Carlo framework of Del Moral et al. (2006). The principal difference between all these inference algorithms is that they each propose to conduct importance sampling over a different sequence of target distributions converging to the target: annealed importance sampling proposes a *tempered* sequence of distributions; hot coupling proposes a sequence of distributions by removing potentials, or factors, from the undirected graphical model and then successively reintroducing them; in this chapter, I propose a sequence of distributions that is constructed adaptively over time by optimizing a variational objective; and the conditional mean field also adaptively constructs a path of target distributions, but it does so via a sequence of objectives formulated using *conditionally-specified* variational mean field approximations. It is my conjecture that automatically selecting the sequence of distributions by optimizing the variational objective, when combined with the variance safeguard, is more efficient (in terms of variance of the final importance sampling estimator) than setting the sequence by hand, say, via tempered transitions. The experiments in Sections 3.4 and 3.5 do provide some empirical support for this claim. This automatic selection, however, can introduce non-negligible computational costs to inference, so it is important to examine this claim further.

All strategies for selecting the sequence of distributions in SMC do have one aspect in common. They all start with a distribution that is easy to sample from, then gradually approach the distribution of interest. The emphasis here is on the word *gradual*: if we approach the target distribution too quickly, then the Markov transition kernel will be unable to effectively update the samples at each iteration, leading to a degenerate importance sampling estimator. On the other hand, if the changes are too gradual, it will take too long to converge to the target. One way to improve upon the situation is to design the initial, easy-to-sample distribution so that the distance to the target is not quite so large.

For most applications, the tempering or "annealing" strategy is considered the naive approach. The basic idea, whose roots lie in statistical physics (Hukushima & Nemoto, 1996; Jarzynski, 1997; Marinari & Parisi, 1992), is to obtain a distribution that is easy to sample from by metaphorically heating the potentials in the graphical model, then gradually cooling the global temperature until we've recovered the original target distribution. Tempering is a naive strategy primarily because the initial distribution may look nothing like the target, hence a lot of effort must be taken to migrate the samples through the sequence of distributions. Typically a very gradual cooling of the temperature is needed to prevent the sample from degenerating too quickly in successive iterations of SMC.

An alternative strategy is to prune potentials or factors from the probabilistic model, then gradually reinsert them—which can be understood as adding edges to

the underlying undirected graphical model—until we've recovered the target. This is the "hot coupling" approach described in Hamze and de Freitas (2006). The principal advantage of this approach is that it is natural to use a spanning tree of the undirected graph as an initial sampling distribution. Since many of the factors of the target are already present in the spanning tree, it is expected that we won't have travel quite so far to recover the target. The main drawback of the hot coupling strategy is that it is meant to work for graphical models with pairwise factors—it is not obvious how this strategy can be appropriately adapted to larger, more arbitrary factors that arise in, say, mixture models.

The conditional mean field strategy presented in Chapter 4 parallels the hot coupling method, in that it also successively adds edges back to the undirected graphical model until the target is recovered. However, the way in which it recovers the edges is different than hot coupling: it does so by fitting a sort of mean field approximation—precisely, a *conditionally-specified* mean field approximation—to the target. Because the initial distribution is given by a naive mean field approximation, conditional mean field is to some extent bound by the limitations of variational mean field methods. If the naive mean field approximation is poor—meaning that it shows little resemblance to the target distribution—then conditional mean field will fare little better, if not worse, than the annealing strategy. The population genetics model described in Sec. 3.5 of this chapter is a prominent example where the naive mean field approximation serves as a very poor initial distribution.

The stochastic approximation method can potentially overcome some of the drawbacks of the strategies I've mentioned so far. This ability hinges on two ideas: one, the variance safeguard; two, the adaptive sequence of distributions.

The variance safeguard ensures that the sample does not degrade too much, and it accomplishes this by scaling back large changes between the current distribution and the next distribution in the sequence. Note that there is no inherent reason why the variance safeguard could not be used in conjunction with the annealing strategy, or with hot coupling.[1] However, in practice I've observed that the variance safeguard contributes little to either strategy because it only succeeds in slowing down progress toward the target.

With the stochastic approximation method, the hope is that it is able to recover a distribution that closely resembles the target, yet is sufficiently close to the initial distribution that it saves SMC from having to migrate the samples a great distance, and in so doing preserves the quality of the sample. This idea is illustrated in Fig. 3.2, an imagined scenario in which the distribution $q^\star(x)$ closely approximates the target posterior $p(x)$. Of course, if there exists no distribution that is closer to the initial distribution than the target (the target $p(x)$ lies in a deep and narrow well on the surface of the Kullback-Leibler divergence), then this adaptive strategy will not provide any advantage over the other SMC strategies. The extensive experience with variational methods in the machine learning, however, suggests that this is rarely the case. The onus on the practitioner is then to solve the stochastic approximation

[1] My supervisor, Nando de Freitas, and my external examiner, Max Welling, pointed this out.

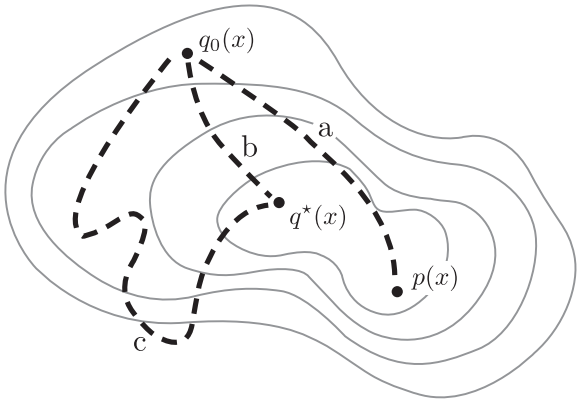

Figure 3.2: A hypothetical scenario for variational inference. The grey lines represent level curves of the variational objective, *i.e.* the Kullback-Leibler divergence. Since the variational approximation $q^\star(x)$ and the target posterior $p(x)$ lie approximately along the same level curve, the implication is that there is not much difference between the two distributions. Thus, rather than construct a sequence of distributions that follows the dashed line circumscribed by (a), a sensible strategy would be to instead follow the path (b) so as to reduce the total distance covered by the sequence of distributions in SMC. If, however, the stochastic approximation is inefficient in its approach toward point $q^\star(x)$—e.g. the dashed line (c)—then it might be better to adopt the non-adaptive sequence of distributions represented by (a). The $q_0(x)$ above represents the initial distribution in the sequence.

problem so as to find as quickly and as efficiently as possible this distribution; for example, the sequence of distributions should lie along the dashed line (b) in Fig 3.2. If the solver instead meanders along the surface of the variational objective, as illustrated by the path (c) in the figure, then the algorithm risks traveling a much further distance than the naive approach given by path (a), resulting in a subpar solution. See Sec. 3.3.10 for a less hypothetical illustrative example. I provide some hints in Sec. 3.3 for designing an effective solver.

A key consideration is computational cost. Suppose, for instance, that the adaptive sequence of distributions is better than a tempered sequence, but that a single iteration of the stochastic approximation method inovlves the prohibitively expensive task of factorizing a large, dense matrix, then we are probably better off sticking with the naive annealing strategy. In general, it is not possible to known in advance whether the stochastic approximation method will introduce significant computational costs over and above other SMC methods; it depends on the number of parameters needed to specify the variational approximation, and it also depends on the choice of search strategy (gradient descent, quasi-Newton, *etc.*). For the problems investigated in this chapter, the number of variational parameters is small enough that the optimization step at each iteration of SMC contributes a relatively modest computational expense. I return to this point in the experiments.

The intention of this chapter is to substantiate the claims made in this section. This aim has only been partly met; more studies are needed to better understand the advantages and potential pitfalls of adaptiving selecting the sequence of distributions using stochastic approximation.

3.3 Description of algorithm

In this section, I formally state the problem and describe my algorithm. To describe my algorithm, I will need to provide mathematical background on a series of topics, starting with the exponential family.

The overall goal is to calculate the expectation of $\varphi(x)$ with respect to a density function $p(x)$ that specifies a probability for every possible configuration $x \in \Omega$, or small interval when X is a continuous random variable. The space of possible configurations Ω may be of very high dimension. We use $\theta^\star$ to refer to the set of parameters that specify the target distribution, and this is written as

$$p(x) = p(x; \theta^\star). \tag{3.1}$$

We want to estimate the expectation

$$E[\varphi(X)] = \int \varphi(x)\, p(x)\, dx. \tag{3.2}$$

Today, such a problem most commonly arises when we are interested in making a prediction with a distribution that takes into account both empirical observations and prior information by multiplying probabilities; in other words, a posterior derived from Bayes' rule. In most cases of interest, it is infeasible to compute the posterior probability $p(x)$ at any point x, and there is no obvious way to draw samples directly from the posterior. Thus, in order to solve such an inference problem we will need to come up with a sensible approximation to the expectation (3.2).

First consider an approximation to (3.2) based on the importance sampling estimator. We can construct an unbiased estimate of the expectation $E[\varphi(X)]$ by drawing n samples $x^{(s)}$ from a proposal density $q(x)$ and evaluating importance weights $w(x^{(s)}) = p(x^{(s)})/q(x^{(s)})$. The Monte Carlo estimator is

$$U_n \equiv \tfrac{1}{n} \textstyle\sum_{s=1}^{n} w(x^{(s)})\, \varphi(x^{(s)}). \tag{3.3}$$

This estimator is based on the elementary importance sampling identity,

$$E_{p(\,\cdot\,)}[\varphi(X)] = E_{q(\,\cdot\,)}[w(X)\, \varphi(X)], \tag{3.4}$$

where the left-hand expectation is taken with respect to the target density $p(x)$, and the right-hand expectation is evaluated with respect to the proposal distribution $q(x)$. (Usually, $p(x)$ can only be evaluated up to a normalizing constant, in which case the asymptotically unbiased *normalized* importance sampling estimator would be used instead.) The main attraction of importance sampling is its flexibility—virtually

any proposal may be used provided it satisfies a few weak conditions (Robert & Casella, 2004). That being said, some proposal distributions are better than others; a poor proposal will lead to an importance sampling estimator U_n with high variance. In other words, independent simulations (with finite n) will produce very different answers. A good proposal is one has a small variance with respect to n independent draws (Andrieu et al., 2003):

$$\mathrm{Var}[U_n] = \tfrac{1}{n}\mathrm{Var}[w(X)\,\varphi(X)] = \tfrac{1}{n}E_{q(\,\cdot\,)}[(w(X)\,\varphi(X))^2] - \tfrac{1}{n}E_{p(\,\cdot\,)}[\varphi(X)]^2. \tag{3.5}$$

Notice that the second term on the right-hand side does not depend on the choice of proposal $q(x)$. A straightforward application of Jensen's inequality shows that the optimal proposal distribution is given by

$$q(x) = \frac{p(x)\,|\varphi(x)|}{\int p(x)\,|\varphi(x)|\,dx}. \tag{3.6}$$

See Andrieu et al. (2003). This result is, of course, only of theoretical interest. Unless great care is taken is in designing the proposal $q(x)$, the Monte Carlo estimator U_n will exhibit astronomically high variance (3.5) for all but the smallest problems.

Instead, we construct a Monte Carlo estimate (3.3) *by replacing* $p(x;\theta^\star)$ *with an alternate target* $p(x;\theta)$ *that resembles it, so that all importance weights are evaluated with respect to this alternate target. This new estimator is biased, but we minimize the bias by solving a variational optimization problem.*

Our algorithm has a dual interpretation: it can be interpreted as a stochastic approximation algorithm for solving a variational optimization problem, in which the iterates are the parameter vectors θ_k, and it can be equally viewed as a sequential Monte Carlo method, in which each distribution $p(x;\theta_k)$ in the sequence is chosen dynamically based on samples from the previous iteration. We conjecture that following the surface of the variational objective is more efficient in terms of variance than a deterministic (e.g. tempered) sequence, provided we have a good search direction. It is difficult to verify this conjecture directly, but we do provide indirect confirmation by comparing to a sequential Monte Carlo method with a tempered sequence of distributions, namely annealed importance sampling. Since SMC is effectively a framework for conducting importance sampling over a sequence of distributions, we describe a "variance safeguard" mechanism (Sec. 3.3.8) that directly regulates increases in variance at each step by preventing the iterates θ_k from moving too quickly. It is in this manner that we achieve a trade-off between bias and variance.

We spend the remainder of this section deriving the algorithm. The derivation builds on theory from variational inference, Monte Carlo methods, stochastic approximation, and the exponential family.

3.3.1 The exponential family

A probability density belonging to the exponential family is written as

$$p(x;\theta) = \exp\{\langle a(x), b(\theta)\rangle - c(\theta) - d(x)\}, \tag{3.7}$$

where $\langle\cdot,\cdot\rangle$ is an inner product, and the vector-valued function $a(x)$ is called the *statistic* of x.[2] Writing the probability density in this way highlights the symmetry, or *conjugacy* (Gelman et al., 2003), between the variables x and the model parameters θ. In this chapter, I restrict the discussion to random vectors X that admit a distribution written in the "natural" or "canonical" parameterization $b(\theta) = \theta$. In which case, we get the simpler "standard" form

$$p(x;\theta) = \exp\{\langle a(x), \theta\rangle - c(\theta)\} \tag{3.8}$$

by incorporating $d(x)$ into the inner product. The log-normalization factor $c(\theta)$ ensures that $p(x;\theta)$ defines a valid probability density. It is given by

$$c(\theta) = \log \int \exp\langle a(x), \theta\rangle dx. \tag{3.9}$$

The restriction to the exponential family provides us with a particularly convenient analysis of the variational objective derived in the next section. This is not at all a strong restriction, as most of the important and classically-studied distributions belong to the exponential family. And, for that matter, the exponential family plays a pivotal role in the development of Bayesian models.[3] My results can likely be extended without too much effort to the exponential family in non-standard form. The Ising spin glass (Sec. 3.4) and latent Dirichlet allocation (Sec. 3.5) are both examples of exponential family distributions that can be expressed in standard form.

The function $a(x)$ is also the vector of *sufficient statistics* of x for the simple reason that it is sufficient for uniquely determining the parameter vector θ provided the representation is not overdetermined. Thus, an important feature of the exponential family is that we can obtain sufficient statistics by inspection.

Another appealing feature of the exponential family is that we can obtain moments of the distribution by taking derivatives of the log-normalization function $c(\theta)$; the first derivatives of $c(\theta)$ produce the mean of the sufficient statistics, and the second-order derivatives give us the covariance of the sufficient statistics. These results can be deduced from the simple fact that $p(x;\theta)$ must represent a proper probability (Dobson, 2002). Taking derivatives of the identity $\int p(x;\theta)dx = 1$ with respect to the parameter vector θ, and reversing the order of integration and differentiation, we get

$$\int \nabla p(x;\theta)\, dx = \int p(x;\theta)(a(x) - \nabla c(\theta))\, dx = 0.$$

[2] In the classical treatment of Brown (1986), x itself is the statistic. It is generally more convenient, hence more widely adopted in recent literature, to use a function $a(x)$ instead.

[3] The Pitman-Koopman lemma provides a succinct case for this claim, because the class of conjugate distributions is contained by the exponential family (Robert & Casella, 2004, Sec. 1.6.1). The exponential family is also crucial to the study of variational methods (Ghahramani & Beal, 2001).

By rearranging the terms in this equation, we get

$$\nabla c(\theta) = E[a(X)]. \tag{3.10}$$

So the partial derivatives of the log-normalization factor are the entries of the expected statistic. Following a similar derivation, the second-order partial derivatives are

$$\int p(x;\theta)(a(x) - \nabla c(\theta))(a(x) - \nabla c(\theta))^T dx - \int p(x;\theta)\nabla^2 c(\theta)\, dx = 0.$$

where $\nabla^2 c(\theta)$ is the matrix of second-order partial derivatives of $c(\theta)$. Rearranging terms and from the previously established identity for the first derivatives, we find that the matrix of second derivatives is equal to the covariance matrix of $a(x)$:

$$\nabla^2 c(\theta) = E[a(X)\, a(X)^T] - E[a(X)]E[a(X)]^T = \mathrm{Var}[a(X)]. \tag{3.11}$$

Since the covariance must always be symmetric positive-definite, it follows that $c(\theta)$ must be convex in its arguments.

These exponential family identities play a fundamental role in the study of how the properties of thermodynamic systems (e.g. pressure, magnetization) at equilibrium respond to changes in θ (e.g. volume, temperature), where $p(x)$ represents the Boltzmann distribution and $c(\theta)$ represents the Helmholtz free energy (Newman & Barkema, 1999). These identities will prove to be extremely useful for subsequent analysis of the variational objective. The exponential family and its properties, particularly properties that pertain to maximum likelihood estimation, has been extensively studied as "information geometry" (Amari, 1998; Efron, 1978; Wainwright, 2002).

3.3.2 The variational objective

One of the predominant ideas of machine learning is that it is possible to cast probabilistic inference as an optimization problem. This is the basic idea behind variational inference. The departure point for all variational inference methods, including my own, is the Kullback-Leibler (K-L) divergence. It is given by

$$F \equiv \int q(x) \log \frac{q(x)}{p(x)}\, dx, \tag{3.12}$$

where $q(x)$ is an artificial distribution and $p(x)$ is the distribution of interest. Under most circumstances, it is only possible to evaluate the density $p(x)$ up to a normalizing constant.[4] Writing $p(x) = f(x)/Z$, where $Z = \int f(x)\, dx$, the K-L divergence is

$$F = \int q(x) \log \frac{q(x)}{f(x)}\, dx + \log Z. \tag{3.13}$$

[4]Note that there is no meaningful connection between the variational approximation in this section and the proposal distribution in the previous section. I use the same notation for both distributions simply to follow standard practice.

This is the *variational objective*.

The strategy is then to compute a sequence of iterates $q_k(x)$ that progress toward the minimum of (3.13). Since $\log Z$ is not affected by the choice of $q(x)$, it plays no role in the optimization. If we are given free reign in the choice of $q(x)$, the optimal choice must be $p(x) = q(x)$. This result is derived by formulating the constrained optimization problem, writing down the Karush-Kuhn-Tucker optimality conditions, then solving for the primal variables and the Lagrange multipliers. This solution is unique because the Hessian of F is positive-definite; see Cover and Thomas (1991).

The variational objective as it is stated here is of little practical value because for most models no tractable expression exists for the integral appearing in (3.13). It is, however, possible to restrict the choice of approximating distribution $q(x)$ in such a way that the integral can be evaluated in a tractable fashion. This concept is illustrated in Fig. 3.1. This is the conventional approach to designing variational inference algorithms (Jordan et al., 1998; Wainwright & Jordan, 2003a), but it is not the the approach adopted here.

Interpretation as a variational lower bound. It is instructive to derive the variational objective by constructing a lower bound on the logarithm of Z. This is easily accomplished using Jensen's inequality (Boyd & Vandenberghe, 2004). One version of Jensen's inequality states that

$$f(E[X]) \leq E[f(X)] \tag{3.14}$$

for any convex function $f(x)$. By using the fact that $-\log(x)$ is convex (*i.e.* tangent lines underestimate the negative of the logarithm), and by introducing artificial distribution $q(x)$, we obtain the variational lower bound

$$\begin{aligned}
\log Z &= \log \textstyle\int f(x)\, q(x)/q(x)\, dx \\
&= \log E[f(x)/q(x)] \\
&\geq E[\log(f(x)/q(x))] \\
&= \textstyle\int q(x) \log f(x)\, dx - \int q(x) \log q(x)\, dx \\
&= -\textstyle\int q(x) \log(q(x)/f(x))\, dx,
\end{aligned} \tag{3.15}$$

where all the expectations above are with respect to the artificial distribution $q(x)$. This negation of this expression is precisely the *variational free energy* in statistical mechanics, , which is the departure point for deriving approximate message passing algorithms such as loopy belief propagation (Aji & McEliece, 2001; Yedidia et al., 2005). The first integral appearing in (3.15) is the *average energy*, and the second integral is the Boltzmann-Shannon entropy (Cover & Thomas, 1991). Let me emphasize that the lower bound holds for *any* arbitrary distribution $q(x)$. Clearly, some lower bounds of the form (3.15) are better than others, so the optimization problem is to find a $q(x)$ that leads to the tightest bound. Maximizing the lower bound (3.15) is precisely the same as minimizing the K-L divergence (3.13).

Interpretation as a Bregman divergence. When $p(x)$ is a member of the

exponential family, the interpretation of the variational objective (3.13) as a Bregman divergence lends more insight. For strictly convex, continuously differentiable, vector-valued function $\phi(x)$, the Bregman divergence of $\phi(x)$ is defined to be

$$D_\phi(x, y) \equiv \phi(x) - \phi(y) - \langle \nabla\phi(y), x - y \rangle. \tag{3.16}$$

A Bregman divergence has many of the properties of distance metrics, although it does not satisfy the triangle equality, and it is evidently not symmetric. You can think of a Bregman divergence as the distance given by the difference between the point $\phi(x)$ and tangent line, or first-order approximation, to $\phi(x)$ about the point y. Supposing that $\phi(x)$ is convex, the tangent line will always be an underestimate to the true value of the function, hence (3.16) will always be positive. To achieve a non-trivial minimum of the Bregman divergence, the function responses at x and y must be equal, and y must be the stationary point of the convex function.

Now, suppose that the target and approximating densities belong to the exponential family, and furthermore, they share the same statistic $a(x)$ and log-normalization function $c(\theta)$, so that the only difference between them is their parameterizations. From now on, I dispense of the previous notation for the variational approximation, and instead reserve $\theta^\star$ for the parameter vector that specifies the target density $p(x) = p(x; \theta^\star)$, and θ for the parameter vector that specifies the approximating density. Inserting the expression of the standard form (3.8) into (3.13), the variational objective reduces to

$$F(\theta) = \langle E[a(X)], \theta - \theta^\star \rangle + c(\theta^\star) - c(\theta), \tag{3.17}$$

where the expectation is with respect to the approximating density. Intriguingly, we get this exact same expression if we substitute into the Bregman divergence (3.16) x for $\theta^\star$, y for θ, and we replace $\phi(x)$ with the log-normalization factor $c(\theta)$.[5] From inspection, the Bregman divergence is convex in its first argument x. But it is not necessarily convex in its second argument y, even when $\phi(x)$ is convex, as is the case here.[6] Recall that the K-L divergence, as it was originally presented in (3.13), is convex with respect to $q(x)$. However, the K-L divergence is not convex with respect to its parameterization θ—in fact, the Hessian involves the skewness, or third moment—hence we can only hope to recover a local minimum to the variational objective. By using what we know about the theory of duality, however, we can frame the variational optimization problem in such a way that it becomes a convex optimization problem.

Interpretation via convex duality. In Chapter 2, I explored the notion of duality in constrained optimization, starting with the Lagrangian function and the Lagrange dual. The Lagrange dual is closely related to the *conjugate* of a function.

[5] This derivation comes from Sec. 3 of Lafferty (1999). The remaining derivations in this section largely follow Johnson (2002).

[6] More properties of Bregman divergences are given in Appendix A of Banerjee et al. (2006).

The conjugate of a real-valued function $f(x)$ is defined to be

$$f^*(u) \equiv \sup_x \{\langle x, u\rangle - f(x)\}. \tag{3.18}$$

When $f(x)$ is differentiable, the conjugate is also called the *Legendre transform* (Boyd & Vandenberghe, 2004). The supremum is used instead of the maximum because $\langle x, u\rangle - f(x)$ may not be bounded, in which case the supremum is, by definition, equal to infinity. The conjugate is defined for *any* function; it need not be convex. Whether or not the function is convex, the conjugate $f^*(u)$ is convex because it is the pointwise supremum of a family of convex (and affine) functions of u. When $f(x)$ is convex, however, the conjugate possesses the special property that there is a one-to-one correspondence between a dual point y and the maximizer x, simply due to the fact that $f(x)$ being convex implies $\langle x, u\rangle - f(x)$ is concave, and so the supremum is attained at a unique value of x. (These observations have strong parallels to the notions of weak and strong duality studied in Chapter 2.) Therefore, we can write the conjugate of a convex function as

$$f^*(u) = \langle x, u\rangle - f(x), \tag{3.19}$$

where it is understood that x and u are the *duals* of each other. How does this result apply to the Kullback-Leibler divergence? As we did earlier with the Bregman divergence, we can apply this result to the K-L divergence by replacing the function $f(x)$ with the log-normalization function $c(\theta)$. The conjugate of $c(\theta)$ is written as

$$c^*(\mu) = \sup_\theta \{\langle \theta, \mu\rangle - c(\theta)\}, \tag{3.20}$$

where μ takes the place of u. The maximum must be at a point for which the gradient of the right-hand side vanishes, *i.e.* when $\mu = \nabla c(\theta)$. But we also know that the first derivatives of the log-normalization factor $c(\theta)$ are equal to the expectation of the sufficient statistics, and that $c(\theta)$ is convex, so we arrive at a fundamentally important conclusion: every dual point μ is the first moment of the sufficient statistics with respect to the density parameterized by θ, and each μ is in a one-to-one correspondence with a point θ. The relationship between the *mean statistic*

$$\mu = E[a(X)] \tag{3.21}$$

and the parameter vector θ is invertible because the log-normalization factor is convex, and a (strictly) convex function possesses one-to-one relationship between the position and its first derivative.

We already know that $c(\theta)$ is convex, and it is also *closed* because its domain is the real line, so every sublevel set is closed. As a result, we can take the conjugate of the conjugate, which recovers the original function $c(\theta)$; see Sec. 3.3.2 of Boyd and Vandenberghe (2004). What we have uncovered in a rather circuitous manner is the variational objective, and the same variational lower bound on the factor $c(\theta^\star)$ as

before. The lower bound is a direct application of Fenchel's inequality,

$$f(x) + f^*(u) \geq \langle x, u \rangle. \tag{3.22}$$

But this result happens to be more evocative than the derivation via Jensen's inequality (3.15) because the lower bound here is tight. In other words, the conjugate of the conjugate guarantees existence of an approximating distribution that recovers the true log-normalization factor exactly, and this result applies (for the most part) whenever the target and approximating distributions are part of the exponential family and they share the same statistic $a(x)$.

The discovery of the invertible relationship between θ and its dual μ motivate the "moment parameterization", in which the mean statistics μ act as the parameters of the approximating distribution. Wainwright and Jordan (2003b) characterize the set of valid mean statistics as the set of vectors μ that correspond to the stationary point for some θ, and describe this set as the intersection of a finite number of halfspaces—hence the term *marginal polytope.* By substituting the dual identity

$$c^*(\mu) = \langle \theta, \mu \rangle - c(\theta) \tag{3.23}$$

into the expression (3.17), we arrive at the dual form of the K-L divergence:

$$F(\mu) = c(\theta^\star) + c^*(\mu) - \langle \mu, \theta^\star \rangle. \tag{3.24}$$

The dual form is particularly attractive because it is convex with respect to the moment parameters μ. The conjugate function $c^\star(\mu)$ does not immediately appear to have an explicit form, but it can be shown that it is equal to the negative entropy of the distribution specified by θ (Wainwright, 2002). That being said, the dual form is difficult to work with because the set of valid moment parameters is difficult to characterize, and because it may not be easy to discern the expression for the entropy for a given μ. Special cases exist whereby the distribution does have a closed-form expression for the entropy, and there has been some recent and compelling work that exploits such special cases to develop approximate algorithms for optimizing the dual variational objective.[7] I do not work with the moment parameterization, but it is worth understanding why it has attracted interest in the machine learning community. It may also be a fruitful avenue to explore in future research.

3.3.3 The stochastic gradient

The fact that we cannot compute the factor $c(\theta)$ for every point θ poses no obstacle to optimizing the variational objective (3.13); through application of basic properties of the exponential family—namely (3.10) and (3.11)—the gradient vector works out

[7] For instance, Wainwright et al. (2005) exploit the tractable nature of tree-structured probability distributions, as per the *junction tree theorem* (Paskin, 2004, Corollary 2.2), and uses convex combinations of trees to construct variational approximations to undirected probabilistic graphical models with discrete random variables.

to be the matrix-vector product

$$\begin{aligned}\nabla F(\theta) &= \nabla_\theta E[a(X)](\theta - \theta^\star) + E[a(X)] - \nabla c(\theta) \\ &= \mathrm{Var}[a(X)](\theta - \theta^\star),\end{aligned} \tag{3.25}$$

where $\mathrm{Var}[a(X)]$ is the covariance matrix of the sufficient statistics vector with respect to the approximating distribution $p(x;\theta)$. Since the covariance matrix is positive-definite, the only non-trivial (finite variance) stationary point $\nabla F(\theta) = 0$ is obtained when $\theta = \theta^\star$; that is, when the variational approximation is *unbiased*. Thus, there is reason to expect that the stochastic approximation algorithm will converge to an unbiased solution, even if the Hessian $\nabla^2 F(\theta)$ is not positive-definite. In contrast, existing variational inference algorithms do not guarantee that a solution to the optimization problem will be unbiased.

The real obstacle is the presence of an integral in (3.25) that is most likely intractable. Supposing we have a collection of samples $x^{(s)}$, for $s = 1, \ldots, n$, with normalized importance weights $w^{(s)}$ that approximate $p(x;\theta)$, then a reasonable Monte Carlo estimate of (3.25) is

$$\nabla F(\theta) \approx \sum_{s=1}^{n} w^{(s)} (a(x^{(s)}) - \bar{a})(a(x^{(s)}) - \bar{a})^T (\theta - \theta^\star), \tag{3.26}$$

where $\bar{a}$ denotes the Monte Carlo estimate of the mean statistic,

$$\bar{a} \equiv \sum_{s=1}^{n} w^{(s)} a(x^{(s)}). \tag{3.27}$$

When there are many parameters to tweak, the Monte Carlo estimate of the gradient is much easier to compute than the full matrix-vector product in (3.25). Since we no longer possess an exact value for the gradient, I appeal to the theory of stochastic approximation to formalize the variational inference algorithm. I will not elaborate on stochastic approximation here as I already devoted a chapter to the topic.

Steepest descent turns out to be an extremely poor search direction. I do not, however, advise following the Newton step because the stochastic Hessian has high variance, and may be indefinite. Instead, I recommend using the damped quasi-Newton approximations developed in Sec. 2.4. The quasi-Newton methods developed by Schraudolph et al. (2007b) for on-line learning will not work here because the variational objective may have negative curvature, and because there is no way of "controlling" consecutive measurements of the gradient.

Positivity constraints are needed for the variational inference algorithm in Sec. 3.5 to guarantee a valid probability density. To handle constraints, I used the the primal-dual interior point stochastic approximation method described in Chapter 2. In experiments, I found that separate primal and dual step sizes dramatically improved performance. Note that I also tried to solve the constrained problem using two-metric projection (see Sec. 2.5.5), but it was actually too slow to be of use.

3.3.4 Rao-Blackwellized stochastic gradient

The algorithm's performance hinges on a good search direction, so it is worth our while to reduce the variance of the gradient measurements when possible. By the Theorem of Rao and Blackwell, we can always reduce the variance of our estimates by analytically computing the expectations when it is possible to do so (Casella & Robert, 1996). This fact is easy to demonstrate. Suppose we have an estimator $f(x, y)$ that depends on two random variables X and Y. From the *variance decomposition lemma*, the variance of the estimator $f(x, y)$ is given by

$$\mathrm{Var}[f(X,Y)] = \mathrm{Var}\{E[f(X,Y)\,|\,Y]\} + E\{\mathrm{Var}[f(X,Y)\,|\,Y]\}, \tag{3.28}$$

where $E[X|y]$ is the conditional expectation of X given $Y = y$. It follows directly from (3.28) that the *Rao-Blackwellized* estimator

$$f^\star(y) = E[f(X,Y)\,|\,y] \tag{3.29}$$

is guaranteed to have lower variance than $f(x, y)$.

In many cases, the Rao-Blackwellized gradient estimate

$$\nabla F(\theta) \approx \sum_{s=1}^{n} w^{(s)} E[(a(x_A^{(s)}, X_B) - \bar{a})(a(x_A^{(s)}, X_B) - \bar{a})^T]\,(\theta - \theta^\star) \tag{3.30}$$

can be computed whenever the responses of the conditionals $p(x_B\,|\,x_A)$ are available. The vector $\bar{a}$ is now the Rao-Blackwellized estimate of the mean statistics,

$$\bar{a} = \sum_{s=1}^{n} w^{(s)} E[a(x_A^{(s)}, X_B)], \tag{3.31}$$

The expectations in both (3.30) and (3.31) and are taken with respect to the conditional of X_B given a sample x_A. The covariance doesn't decompose as nicely as the mean—it is easy to forget that the variance of a linear combination is *not* a linear combination of variances—so Rao-Blackwellized estimates of the covariance may or may not be too expensive to compute, depending on the conditional independence structure of the target distribution. A Rao-Blackwellized estimate of the mean, by contrast, is easy to compute regardless of the conditional independence structure.

The following formula that I derived is generally useful for constructing Rao-Blackwellized estimates of the covariance. Suppose we have three random variables X, Y and Z, and we would like to construct an estimate of $\mathrm{Cov}[X, Y]$, the covariance between X and Y. The Rao-Blackwellized Monte Carlo estimator, denoted by $C[X, Y]$, works out to be

$$C[X,Y] = \sum_{s=1}^{n} w^{(s)} (E[X] - \bar{x})(E[Y] - \bar{y}) + \sum_{s=1}^{n} w^{(s)} \mathrm{Cov}[X,Y], \tag{3.32}$$

in which all expectations are defined on the conditionals of X and Y given samples $z^{(s)}$. When X and Y are conditionally independent given Z, $E[XY]$ is equal to $E[X]E[Y]$, and so the covariance estimator resolves to

$$C[X,Y] = \sum_{s=1}^{n} w^{(s)}(E[X] - \bar{x})(E[Y] - \bar{y}). \tag{3.33}$$

3.3.5 Choice of parameterization

Recall our goal: draw samples that closely follow the target distribution $p(x)$, so that the Monte Carlo estimate of the expectation matches the true expectation (3.2). The first step toward accomplishing this goal is to design a family of approximating distributions $p(x;\theta)$ parameterized by θ. The practitioner has nearly complete freedom in the design of the variational approximation. At this stage, there is no way to say *a priori* what constitutes a "good" parameterization. There are, however, some important restrictions to keep in mind:

1. Every variational distribution $p(x;\theta)$ should belong to the exponential family, and should be expressed in standard form (3.8).
2. There must be at least one choice of θ that yields a sampling distribution.
3. There must be at least one parameter vector, written $\theta^\star$, that recovers the target; *i.e.* $p(x) = p(x;\theta^\star)$.

In the experiments below, I study study a variety of designs that satisfy these criteria.

3.3.6 Sequential Monte Carlo

After having taken a step along the stochastic gradient (scaled by the quasi-Newton approximation of the Hessian), the weighted samples must be updated to reflect the new variational approximation $p(x;\theta_{k+1})$. To accomplish this feat, I borrow from Del Moral et al. (2006) a general methodology for sampling from a sequence of distributions.

Sequential Monte Carlo (SMC) is an instance of a *population-based* inference algorithm, as characterized by Iba (2001) and Jasra et al. (2007). The common feature of population-based methods is that they simulate a collection of samples, or particles, in parallel. These methods can be divided into two broad categories: those based on MCMC, and those that are derivatives of importance sampling. Population-based MCMC methods sample from a product of auxiliary distributions, in which one of the distributions in the product represents the target. Much of the work on population-based MCMC, including the early incarnations of simulated tempering (Marinari & Parisi, 1992) and replica exchange Monte Carlo (Swendsen & Wang, 1986), grew out of research in statistical physics. These tempering schemes were developed to more effectively simulate "frustrated" thermodynamic systems such as spin glasses (see Sec. 3.4), the key idea being that simulations at high temperatures broaden the sampling of the state space, thus aiding the Markov chain to escape from local minima. More recent work incorporates sophisticated strategies for exchanging information

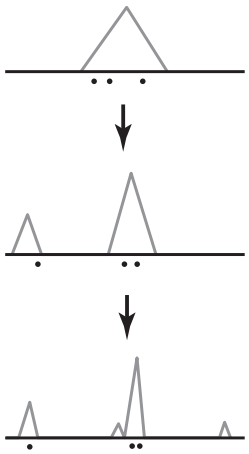

Figure 3.3: Schematic of sequential Monte Carlo in action.

among the parallel Markov chains; see Geyer and Thompson (1995), Goswami and Liu (2007), Iba (2001) and Liang and Wong (2001) for some of these advances.

Another way to approach parallel simulation of stochastic models is to apply importance sampling to a sequence of auxiliary distributions. Chopin (2002), Jarzynski (1997), Liang (2002), and Neal (2001) all investigated this idea for probabilistic inference, although all these developments can be understood as special cases of SMC. The idea of constructing a sequence of distributions has a strong tradition in the literature, dating back to work on simulating the behaviour of polymer chains (Rosenbluth & Rosenbluth, 1955), protein folding (Frauenkron et al., 1998; Grassberger, 2002), and counting and integration problems (Häggström, 2002; Jerrum & Sinclair, 1996). Tempering is perhaps the most widely used strategy, due to its ease of implementation and intuitive appeal.[8] At early stages, high global temperatures smooth the modes and allow easy exploration of the state space. Afterward, the temperature is progressively cooled until the original distribution is recovered. The problem is that the variance of the importance weights tends to degenerate around a system's critical range of temperatures, as observed in Hamze and de Freitas (2006). An entirely different approach is to remove constraints (or factors) from the original model, then incrementally reintroduce them. This has been a fruitful approach for approximate counting, simulation of protein folding, and inference in Ising spin glasses (Hamze & de Freitas, 2006). If, however, a reintroduced constraint has a large effect on the distribution, the particles may again rapidly deteriorate.

The basic idea behind all sequential Monte Carlo methods is to iteratively shift particles toward successively refined artificial distributions, until the refinements converge to the target. This idea is illustrated in Fig. 3.3. Depicted is a sequence of three artificial distributions, where the bottom one represents the target. At the top of the figure, the initial distribution is chosen in such a way that it is easy to draw samples from it. It might be helpful to think of the initial distribution as a smoothed version

[8] See Sec. 2.7.1 of Jasra et al. (2007) for a comparison of tempering strategies.

of the target. A kernel designed by the user shifts the samples—the black dots in the figure—toward regions of strong support in the second artificial distribution. This process continues until the samples have been shifted more or less successfully toward the distribution of interest.

As with earlier work on particle filters (Doucet et al., 2000b; Doucet et al., 2001), what one typically finds is that performing ordinary importance sampling (Andrieu et al., 2003) over time is prohibitive because it involves computing large integrals. This problem is resolved most naturally by designing a sequence of distributions $\tilde{p}_k(x_{1:k})$ on an expanding state space $\Omega \times \cdots \times \Omega = \Omega^k$, and then simulating particles so that they represent paths in this expanding state space.[9] Provided we are smart about designing the sequence of distributions, the expressions for the importance weights on this expanding state space are manageable.

The general strategy of SMC is to devise an initial proposal distribution $q_0(x)$ that is easy to sample from, then at every subsequent step to extend the path of each particle toward the next artificial distribution in the sequence via a Markov transition kernel $K_k(x' \,|\, x)$.[10] In the standard setting, the last distribution in the sequence should recover the target, $p(x)$. I denote the kth artificial distribution in the sequence by $p_k(x)$. In my proposed probabilistic inference algorithm, the sequence of artificial distributions is precisely the sequence of iterates $p_k(x) = p(x; \theta_k)$ obtained from executing stochastic approximation on the variational objective (3.13).

As I mentioned earlier, SMC actually conducts importance sampling on the distributions $\tilde{p}_0(x_0), \tilde{p}_1(x_0, x_1), \tilde{p}_2(x_0, x_1, x_2)$ and so on, such that $\tilde{p}_k(x_{1:k})$ is a distribution defined on the product space Ω^k. To ensure that SMC does the right thing, at time step k the marginal distribution over x_k should be equal to $p_k(x_k)$, the kth artificial target. I denote the kth proposal distribution on the ever expanding configuration space by $\tilde{q}_k(x_{1:k})$.

In the first step, samples $x_0^{(s)} \in \Omega$ are drawn from proposal density $q_0(x)$ and assigned importance weights $w_0(x) = p_0(x)/q_0(x)$. Provided a few weak conditions are satisfied (Geweke, 1989; Robert & Casella, 2004), the weighted samples $\{x_0^{(s)}, w_0^{(s)}\}$ are guaranteed to recover the artificial target $p_0(x)$ in the limit, where $w_0^{(s)}$ is shorthand for $w_0(x_0^{(s)})$. In many circumstances, the initial proposal and initial artificial target are the same, in which case the importance weights are uniform.

In the second step, a Markov transition kernel $K_1(x' \,|\, x)$ shifts each sample toward the next target in the sequence, $p_1(x)$, and the importance weights $\tilde{w}_1(x, x')$ compensate for any failure to do so. In effect, the second step consists of extending the path of each particle $x_0^{(s)} \in \Omega$ onto the joint space $\Omega \times \Omega$. The importance weights

[9] Alternatives to this strategy do exist; see, for instance, Klaas et al. (2006).

[10] Previously in this chapter, I had used $q(x)$ to denote a density obtained from a variational approximation. From now on, I will use $q(x)$ instead to refer to an importance sampling proposal distribution, since this is standard notation in the literature.

on the joint space are given by

$$\tilde{w}_1(x, x') = \frac{\tilde{p}_1(x, x')}{\tilde{q}_1(x, x')} = \frac{L_0(x \,|\, x')\, p_1(x')}{K_1(x' \,|\, x)\, p_0(x)} \times w_0(x), \tag{3.34}$$

where $\tilde{p}_1(x, x') = L_0(x \,|\, x')\, p_1(x')$ is the artificial distribution over the joint space, $\tilde{q}_1(x, x') = K_1(x' \,|\, x)\, q_0(x)$ is the proposal distribution over the joint space, and $L_0(x \,|\, x')$ is the "backward-in-time" kernel, or "backward kernel" for short. The expectation is that $K_1(x' \,|\, x)$ have invariant distribution $p_1(x)$, though this is not required (Del Moral et al., 2006).

Since each of the transition kernels $K_k(x' \,|\, x)$ is Markov—that is, it only takes into account the sample at the previous time step, and not the full history—the proposal distribution after k steps is

$$\tilde{q}_k(x_{1:k}) = K_k(x_k \,|\, x_{k-1}) \cdots K_1(x_1 \,|\, x_0)\, q_0(x_0). \tag{3.35}$$

The major insight of Del Moral et al. (2006) is that if we choose the artificial targets $\tilde{p}_k(x_{1:k})$ on the product space in a clever way, we can update the importance weights $\tilde{w}_k(x_{1:k}) \equiv \tilde{p}_k(x_{1:k})/\tilde{q}_k(x_{1:k})$ without having to look back at the entire history, just the components at the current and previous time steps. This special construction is

$$\tilde{p}_k(x_{1:k}) = L_0(x_0 \,|\, x_1) \cdots L_{k-1}(x_{k-1} \,|\, x_k)\, p_k(x_k), \tag{3.36}$$

where we've introduced a series of backward kernels $L_k(x \,|\, x')$. It is easy to show that this construction admits $p_k(x_k)$ as its marginal. It is also easy to show that the importance weights on the joint space at time k become

$$\tilde{w}_k(x_{1:k}) = \frac{L_{k-1}(x_{k-1} \,|\, x_k)\, p_k(x_k)}{K_k(x_k \,|\, x_{k-1})\, p_{k-1}(x_{k-1})} \times \tilde{w}_{k-1}(x_{1:k-1}). \tag{3.37}$$

Notice that the components from the current and previous time steps are the only components needed to evaluate the new importance weights. What this construction allows us to do is pretend that we have an evolving collection of particles with importance weights that converge toward the target, without having to worry about the fact that each sample actually represents a path residing in an expanding product space.

Normalized estimator. What I've described so far isn't entirely practical; for all the probabilistic inference problems studied in this thesis, it is only plausible to compute the probabilities $p_k(x)$ up to a normalizing constant. That is, only the $f_k(x)$ are known pointwise, where $p_k(x) = f_k(x)/Z_k$. This issue is easily resolved by adopting the *normalized importance sampling estimator* (Robert & Casella, 2004). The normalized importance sampling estimator yields importance weights

$$\tilde{w}_k(x_{1:k}) = \frac{\hat{w}_k(x_{1:k})}{\sum_{s=1}^{n} \hat{w}_k(x_{1:k}^{(s)})}, \tag{3.38}$$

where the *unnormalized* importance weights $\hat{w}_k(x_{1:k})$ remain the same as (3.34), except that $p_{k-1}(x)$, $p_k(x)$ are replaced by their unnormalized counterparts $f_{k-1}(x)$ and $f_k(x)$. This estimator is biased, but the bias quickly becomes negligible (Andrieu et al., 2003). The normalized estimator recovers a Monte Carlo estimate of the normalizing constant at time step k via the recursion

$$\begin{aligned}
Z_k &= \textstyle\int f_k(x_k)\, dx_k \\
&= \textstyle\int L_{k-1}(x_{k-1} \,|\, x_k)\, f_k(x_k)\, dx_{1:k} \\
&= \textstyle\int L_{k-1}(x_{k-1}|x_k) f_k(x_k) \dfrac{\hat{w}_k(x_{1:k})}{\tilde{w}_{k-1}(x_{1:k-1})} \times \dfrac{K_k(x_k|x_{k-1}) f_{k-1}(x_{k-1})}{L_{k-1}(x_{k-1}|x_k) f_k(x_k)} \times \dfrac{Z_{k-1}}{Z_{k-1}} dx_{1:k} \\
&= Z_{k-1} \times \textstyle\int \hat{w}_k(x_{1:k})\, K_k(x_k \,|\, x_{k-1})\, \tilde{q}_{k-1}(x_{1:k-1})\, dx_{1:k} \\
&= Z_{k-1} \times \textstyle\int \hat{w}_k(x_{1:k})\, \tilde{q}_k(x_{1:k})\, dx_{1:k} \\
&\approx Z_{k-1} \times \textstyle\sum_s \hat{w}_k(x_{1:k}^{(s)}),
\end{aligned} \tag{3.39}$$

provided we already have a good estimate of Z_{k-1}. This recursion can of course be used to estimate the log-normalization factor at the target $\theta^\star$.

Resampling. Recall that particles are paths in the product space Ω^k, so importance sampling is fundamentally conducted in higher and higher dimensions as the sequence grows. Over long sequences, the population naturally degenerates to a single particle (with an importance weight of 1). The measures presented in Sec. 3.3.8 for controlling the variance of the importance weights will, to some degree, prevent this phenomenon. Despite these measures, it is still occasionally helpful to employ the standard variance-reduction technique—resampling. Resampling reduces variance of the marginal estimator while perhaps sacrificing some of the diversity of the population. I use the stratified resampling scheme detailed in Fearnhead (1998).

Choice of forward and backward kernels. Sequential Monte Carlo provides us with a very general framework for sampling from a sequence of distributions. In this chapter, the sequence of distributions is given by the iterations of the stochastic approximation method. There remain two degrees of freedom: the choice of forward kernel $K_k(x' \,|\, x)$ at each time step k, and the choice of backward kernel $L_k(x \,|\, x')$ at each time k. All the models studied in this paper meet the necessary criteria to calculate the conditionals $p(x_A \,|\, x_B)$ and $p(x_B \,|\, x_A)$, so a natural choice for the forward kernel is the two-stage Gibbs sampler,

$$K_k(x' \,|\, x) = p_k(x'_A \,|\, x'_B)\, p_k(x'_B \,|\, x_A), \tag{3.40}$$

in which we first draw a sample of x_B given x_A, then update x_A conditioned on x_B. As for the backward kernel, Del Moral et al. (2006) and Peters (2005) suggest

$$L_{k-1}(x \,|\, x') = \frac{K_k(x' \,|\, x)\, p_{k-1}(x)}{\int K_k(x' \,|\, x)\, p_{k-1}(x)\, dx}. \tag{3.41}$$

This backward kernel has a distinct advantage over the sub-optimal kernels used in

other frameworks because it can potentially correct for proposal distributions that fail to "dominate" the target density. This property can be seen by examining the expression for the importance weights on the joint space. Following (3.37), they are

$$\tilde{w}_k(x_{1:k}) = \frac{p_k(x_k)}{\int K_k(x_k \,|\, x_{k-1})\, p_{k-1}(x_{k-1})\, dx_{k-1}} \times \tilde{w}_{k-1}(x_{1:k-1}). \tag{3.42}$$

If the transition kernel increases the mass of the proposal in regions where $p_{k-1}(x)$ is weak relative to $p_k(x)$, the backward kernel (3.41) will rectify some of the problems caused by a skewed or overconfident proposal. The problem with (3.41) is that it is difficult to find a forward kernel $K_k(x' \,|\, x)$ that leads to a tractable integral in the denominator. The two-stage Gibbs sampler is, unfortunately, not a kernel that meets this criterion. However, if we ignore the second stage of the two-stage Gibbs sampler, then we arrive at a convenient expression for the importance weights:

$$\tilde{w}_k(x_{1:k}) = \frac{p_k(x_A)}{p_{k-1}(x_A)} \times \tilde{w}_{k-1}(x_{1:k-1}), \tag{3.43}$$

in which x_A is the component from time step $k-1$ restricted to the set A, and $p_k(x_A) = \int p_k(x)\, dx_B$. From the assumptions I made earlier, it follows that it is always possible to compute the marginals $p(x_A)$ up to a normalizing constant. Notice that there is no need to store the sample configurations on set B.

It is interesting to note that this expression is a rare instance of an importance weight update that can be derived from the backward kernel (3.41), and can be equally derived from a Rao-Blackwellized version of the sub-optimal backward kernel

$$L_{k-1}(x \,|\, x') = \frac{K_k(x' \,|\, x)\, p_k(x)}{p_k(x')}. \tag{3.44}$$

The sequential Monte Carlo estimates with importance weights (3.43) will thus share the properties of other Rao-Blackwellized estimators. Note that the derivation of the Rao-Blackwellized sequential update (3.43) is unrelated to the derivation of the Rao-Blackwellized particle filter (Martinez-Cantin et al., 2007; Murphy, 2002).

The derivation of the importance weight update exposes a major drawback of SMC relative to its predecessor, the particle filter: in dynamic models, there is a natural transition from one joint distribution to the next, and so effective importance weight updates are easy to come by, whereas in SMC this matter is complicated by the absence of any natural dynamics.

3.3.7 Computing the variational lower bound

In the previous section, I demonstrated that SMC can be used to estimate the log-normalization factor $c(\theta^\star)$. Earlier, in Sec. 3.3.2, I showed how to obtain a variational lower bound on $c(\theta^\star)$ in three different ways: via the K-L divergence, via Jensen's inequality, and via the conjugate dual. Written in exponential family form, the vari-

ational lower bound is

$$c(\theta^\star) \geq c(\theta) - \langle E[a(X)], \theta - \theta^\star \rangle, \tag{3.45}$$

where the expectation is with respect to the approximating density $p(x;\theta)$. This clearly cannot be computed exactly. However, at any step of the stochastic approximation algorithm, we can compute a Monte Carlo approximation to the variational lower bound by maintaining an estimate of $c(\theta_k)$ via the recursion

$$c(\theta_k) \approx c(\theta_{k-1}) + \log \textstyle\sum_s \hat{w}_k^{(s)}, \tag{3.46}$$

following (3.39). When possible, it is advisable to use Rao-Blackwellized Monte Carlo estimates for the intractable expectation. A Monte Carlo estimate of the variational lower bound (3.45) typically converges more rapidly to the the target $c(\theta^\star)$ than the standard Monte Carlo estimate, hence the variational lower bound serves as a more robust estimator.

In this fashion, we can obtain reliable estimates of Bayes factors (Kass & Raftery, 1995). Bayes factors are widely used for hypothesis testing (e.g. whether or not recombination has occurred in a DNA sequence) and model comparison (e.g. to determine the best number of components for a mixture model). In a different but related context, the variational lower bounds derived here might also be used for model evaluation in maximum likelihood training (Murray & Salakhutdinov, 2009). Many methods have been devised for computing Bayes factors from MCMC simulations (DiCiccio et al., 1997; Gelman & Meng, 1998), but experience has shown that importance sampling and sequential Monte Carlo approaches are far superior.

3.3.8 Safeguarding the variance

A key component of the stochastic approximation algorithm is a method that enables the practitioner to regulate the variance of the importance weights and, by extension, the variance of the Monte Carlo estimate of $E[\varphi(X)]$. The proposed method consists of two parts. The first part safeguards the variance of the estimator at each iteration of the algorithm, preventing the variance from degrading too rapidly. It is a "myopic" variance safeguard, in that it only considers the variance incurred from one iteration of SMC. On its own, the variance safeguard is insufficient because it will accept further degradation in the quality of the estimator so long as it does not degrade too rapidly; hence, the behaviour of the algorithm will be sensitive to the number of iterations. I also propose a "global" variance regulation method that penalizes solutions to the variational objective that have a high variance. Since the global variable safeguard raises some issues that have yet to be resolved—notably, how to pick the strength of the variance penalty term—I focus on the local variance safeguard in the experiments in Sections 3.4 and 3.5.

To begin, we need to address a basic question: how can we effectively monitor the variance of the estimator? Suppose, purely hypothetically, we are interested in monitoring the variance of the importance sampling estimator (3.3), say, for the

purpose of comparing two different proposal distributions. Clearly, we cannot directly use (3.5) to compare the two proposal distributions. An obvious solution then is to formulate a Monte Carlo estimate of $\mathrm{Var}[U_n]$. There are certainly potential pitfalls to this approach—the Monte Carlo estimator of the variance may itself have high variance—but overall the results of my experiments indicate that this approach can be a reasonable one to take.

The basic importance sampling estimator is really only useful for the simplest cases. In more realistic scenarios, we can only expect to evaluate the target density $p(x) = f(x)/Z$ if we disregard the normalizing constant Z. This motivates the normalized importance sampling estimator, as I discussed in Sec. 3.3.6. The normalized estimator $\hat{U}_n$ is based on a different identity:

$$E_{p(\,\cdot\,)}[\varphi(X)] = \frac{E_{q(\,\cdot\,)}[\hat{w}(X)\,\varphi(X)]}{E_{q(\,\cdot\,)}[\hat{w}(X)]}. \tag{3.47}$$

where the unnormalized importance weights are $\hat{w}(x) = f(x)/q(x)$. This identity is often written as (3.4), in which the weights $w(x)$ are now defined to be the normalized importance weights $w(x) = \hat{w}(x)/\int q(x)\,\hat{w}(x)\,dx$. The obvious advantage of this identity is that it does not involve the normalizing constant of the target. A disadvantage of the normalized estimator is that it is difficult to characterize its mean and variance, because they both involve expectations of ratios. This difficulty can be resolved with the *delta method* (Casella & Berger, 2002; Rice, 1988).

Unless the function $f(x)$ is linear, there is no general way to obtain its mean response, at least without approximating the expectation through Monte Carlo techniques. Suppose we form a first-order Taylor series approximation of the function $f(x)$ about some point $\bar{x}$. Then the expected value is given by

$$\begin{aligned} E[f(X)] &\approx E[f(\bar{x}) + \nabla f(\bar{x})(X - \bar{x})] \\ &= f(\bar{x}) + \nabla f(\bar{x})(E[X] - \bar{x}) \end{aligned} \tag{3.48}$$

The delta method chooses $\bar{x}$ to be $E[X]$, so that the approximation becomes simply

$$E[f(X)] \approx f(\bar{x}). \tag{3.49}$$

When $f(x)$ is indeed linear, then this of will of course be exact. Provided the function resembles a linear function, then the delta method approximation will be a good one. Following a similar derivation, the tangent line approximation to the variance of $f(x)$ about the point $\bar{x} = E[X]$ is given by

$$\mathrm{Var}[f(X)] \approx (\nabla f(\bar{x}))^2 \mathrm{Var}[X]. \tag{3.50}$$

In a similar fashion, we can also apply the delta method to the variance of the normalized importance estimator. This is the basis for the "rule of thumb" of importance sampling, the effective sample size (ESS). The ESS commonly used to gauge the "efficiency" of an importance sampling procedure (Bergman, 1999; Kong et al.,

1994; Liu, 1996). The derivation of the ESS begins with a linear approximation to the variance of a ratio:

$$\mathrm{Var}[X/Y] \approx \left(\frac{\bar{x}}{\bar{y}}\right)^2 \left\{ \frac{\mathrm{Var}[X]}{\bar{x}^2} + \frac{\mathrm{Var}[Y]}{\bar{y}^2} - \frac{2\mathrm{Cov}[X,Y]}{\bar{x}\bar{y}} \right\}, \tag{3.51}$$

where $\bar{x} = E[X]$ and $\bar{y} = E[Y]$. See p. 245 of Casella and Berger (2002) or p. 126 of Robert and Casella (2004) for the derivation. It can then be shown that the delta method leads to the following approximate characterization of the variance of the normalized importance sampling estimator:

$$\begin{aligned} \mathrm{Var}[\hat{U}_n] &\approx \tfrac{1}{n}\mathrm{Var}_{p(\,\cdot\,)}[\varphi(X)] \times (1 + \mathrm{Var}_{q(\,\cdot\,)}[w(X)]) \\ &= \tfrac{1}{n}\mathrm{Var}_{p(\,\cdot\,)}[\varphi(X)] \times E_{q(\,\cdot\,)}[w(X)^2]. \end{aligned} \tag{3.52}$$

We get the second equality because the expected value of the normalized importance weights is 1. Notice it is the *normalized* importance weights $w(x)$ that appear in the above expression. See Doucet et al. (2000b) for a derivation of this result.

The ESS is widely used, but it is still a poor measure of the true variance because the tangent line is usually an unsuitable approximation to the ratio (3.47), the possible exception being when the normalized importance weights are all located near their mean (*i.e.* the case when the variance is close to zero). Since the variance on the right-hand side is evaluated with respect to $p(x)$, the ESS rule (3.52) tells us that the variance of a normalized estimator $\hat{U}_n$—as a function of the proposal—is proportional to the expected value of the squared importance weights. According to the delta method, the variance of $\hat{U}_n$ must always be greater than the variance of the estimator with sampling distribution $q(x) = p(x)$, yet we've already established that $p(x)$ is not the optimal proposal distribution! What this tells us is that the ESS approximation to the variance can be misleading.

I avoid the pitfalls of the ESS by adopting the following strategy: one, I construct a quadratic model of the variance which is likely to be a better than a first-order approximation; and two, the optimization algorithm tends not to visit locations where the quadratic approximation is poor because the Taylor series is expanded about the point θ_{k-1}, and the new iterate θ_k is unlikely is stray far from θ_{k-1}.

Stepwise variance safeguard. The trouble with taking a full Robbins-Monro step is that the Gibbs kernel may be unable to effectively migrate the particles toward the new target, in which case the the importance weights will overcompensate for this failure, quickly leading to a degenerate population. The remedy I propose is to find a step size a_k that satisfies

$$\beta \times \mathrm{Var}_{\tilde{q}_k(\,\cdot\,)}[\tilde{w}_k(X_{1:k})] \leq \mathrm{Var}_{\tilde{q}_{k-1}(\,\cdot\,)}[\tilde{w}_k - 1(X_{1:k-1})], \tag{3.53}$$

where β is a number between 0 and 1. (I've left out $\varphi(x)$ from the condition to simplify the analysis and implementation.) This condition prevents the new iterate θ_k from increasing the variance by more than a factor of β at each step, where a β near 1 leads to a stringent variance safeguard. We cannot, of course, expected to compute

the variances in (3.53) exactly, but at time step k we have samples drawn from the proposal $\tilde{q}_{k-1}(x_{1:k-1})$ and so reasonable sample approximation of the variance safeguard condition is

$$\beta \sum_{s=1}^{n} (\tilde{w}_k(x_{1:k}^{(s)}) - \tfrac{1}{n})^2 \leq \sum_{s=1}^{n} (\tilde{w}_{k-1}(x_{1:k-1}^{(s)}) - \tfrac{1}{n})^2. \tag{3.54}$$

Note that the average normalized importance weight is $1/n$ because the normalized importance weights always sum to 1. The simplest way to find a step size that satisfies a sample estimate of (3.54) is to run backtracking line search (Nocedal & Wright, 2006). The problem is that this approach can be incredibly inefficient if the acceptable step size is small.[11] A better approach is to analytically compute the safeguarded step size a_k that satisfies (3.54) by constructing a Taylor series approximation to (3.54). Given that the expression for the variance of the normalized importance weights is composed from several nonlinear operators, it is not clear whether this is a good strategy. A tangent line approximation to variance was found to be extremely unstable, as one would expect. A quadratic approximation of the Rao-Blackwellized importance weights (3.43) was also quite poor, as it occasionally recommended strange step sizes. (For that matter, the second derivative of the Rao-Blackwellized importance weights can be prohibitive to compute because it involves a covariance matrix.) Instead, I constructed a second-order Taylor-series approximation to the naive importance weight update obtained from the sub-optimal backward kernel (3.44) without Rao-Blackwellization. Its first and second derivatives are easy to compute, and in my experience it yielded a robust bound on the variance safeguard condition (3.54).[12]

The goal is to capture how the importance weights and, in turn, the variance of the importance weights, vary as a function of the new parameter vector θ_k. In more general circumstances, we would also have to worry about the effect of θ_k on the proposal distribution $\tilde{q}_k(x_{1:k})$, and specifically the kernel $K_k(x' \mid x)$ that generates the new samples. Fortunately, the analysis is simplified by the fact that the importance weight updates, recalling 3.43) do not depend on the value of the new sample. (This also explains why the importance weights may be updated prior to running the two-stage Gibbs sampler in Fig. 3.4.) I use

$$S_k(\theta_k) \equiv \textstyle\sum_{s=1}^{n} (\tilde{w}_k(x_{1:k}^{(s)}) - \tfrac{1}{n})^2 \tag{3.55}$$

to denote the sample variance at the kth iteration of the algorithm (multiplied by the number of samples, n), so that the variance safeguard condition is written as

$$\beta S_k(\theta_k) \leq S_{k-1}(\theta_{k-1}). \tag{3.56}$$

[11] A special note with regards to implementation of backtracking line search for the variance safeguard: it is important that simulations within the line search be deterministic, otherwise there is no guarantee that line search will terminate.

[12] Many exponential family distributions already have variances that are quadratic functions (Morris, 1982). Such cases lend further support to the Taylor series approximation.

We want to find the step length a_k that satisfies the nonlinear condition (3.56) with equality. Forming a Taylor-series expansion with second-order terms about $a_k = 0$, *i.e.* when the new artificial distribution is the same as the previous one, we obtain the following quadratic approximation:

$$\tfrac{1}{2}\Delta\theta_k^T \nabla^2 S_k(\theta_{k-1})\Delta\theta_k a_k^2 + \Delta\theta_k^T \nabla S_k(\theta_{k-1}) a_k - \tfrac{1-\beta}{\beta} S_{k-1}(\theta_{k-1}) = 0. \tag{3.57}$$

where $\Delta\theta_k$ is the search direction at iteration k. Note that each new importance weight $\tilde{w}_k(x_{1:k})$ is a function of the normalized importance weight from the previous step, the parameter vector θ_k specifying the new distribution, and the unnormalized importance weights $\hat{w}_k(x_{1:k}^{(s)})$, for $s = 1, \ldots, n$. As we will see, the first and second-order coefficients in the quadratic equation above can be can be computed efficiently—computation time is linear in both the number of particles and the size of the parameter vector. The roots a_k of the above equation are given by the familiar quadratic formula.

Thanks to the special properties of the exponential family, the derivatives $\nabla S_k(\theta)$ and $\nabla^2 S_k(\theta)$ in the above Taylor expansion work out nicely. To simplify the expressions a little bit, I do the following: 1) use x as shorthand for the last coordinate in the current sample path $x_{1:k-1}$, 2) refer to the importance weights from the previous iteration by $w(x)$, making explicit their dependence on coordinate $k-1$ only, 3) refer to the new, unnormalized importance weights by $\hat{W}(x)$, and 4) refer to the final, updated importance weights by $W(x)$. The normalized importance weights are recovered according to

$$W(x) = \hat{W}(x) / \textstyle\sum_{s=1}^{n} \hat{W}(x^{(s)}). \tag{3.58}$$

Following the naive update equation—akin to the expression (3.43) derived earlier—the formula for the new unnormalized importance weights is quite simply

$$\begin{aligned} \hat{W}(x) &= p_k(x)/p_{k-1}(x) \times w(x) \\ &= \exp\langle a(x), \theta_k - \theta_{k-1}\rangle \times w(x). \end{aligned} \tag{3.59}$$

The gradient of the unnormalized importance weights with respect to the new parameter vector θ_k is then simply

$$\nabla_\theta \hat{W} = \hat{W} a(x), \tag{3.60}$$

and the derivatives of the normalized importance weights are given by

$$\nabla_\theta W = W(a(x) - \bar{a}), \tag{3.61}$$

where $\bar{a}$ was defined as in (3.27). From these results, the gradient of the sample variance $S_k(\theta_k)$ works out to be the sample covariance between the sufficient statistics

and the importance weights:

$$\nabla S_k(\theta_k) = 2\sum_{s=1}^{n} W(x^{(s)})(W(x^{(s)}) - \bar{W})(a(x^{(s)}) - \bar{a}), \tag{3.62}$$

where $\bar{W}$ is the sample mean

$$\bar{W} \equiv \textstyle\sum_{s=1}^{n} W(x^{(s)})^2. \tag{3.63}$$

Finally, after quite a bit of algebra, the second-order derivatives of $S(\theta)$ turns out to involve the importance sampling estimate of a matrix of third moments:

$$\nabla^2 S_k(\theta_k) = 2\sum_{s=1}^{n} W(x^{(s)})(2W(x^{(s)}) - \bar{W})(a(x^{(s)}) - \bar{a})(a(x^{(s)}) - \bar{a})^T. \tag{3.64}$$

Note that the second derivative times a vector can be calculated without having to form the matrix of third moments.

The variance safeguard approach I have just described has the appearance of being similar to the delta method or effective sample size approach. After all, both my approach and the delta method formulate Taylor series approximations to the variance. However, there is a fundamental difference: the effective sample size uses a linear approximation to the quantity of interest—the normalized importance weights—about the mean of X, whereas I treat the variance as a function of the model parameters, and formulate a quadratic approximation about, θ_{k-1}, the variational approximation from the previous step. Crucially, I introduce no approximation into the importance weights as a function of X.

Safeguarding the ESS. There is a fundamental problem with this approach to safeguarding the variance: at the initial iteration, and at any iteration following a resample move, the variance of the importance weights will be zero, so no positive step size will satisfy (3.54). I propose then to also allow for step sizes that do not drive the effective sample size—the inverse of the expectation of the squared importance weights—below a certain factor $\xi \in (0, 1)$ from the optimal sample. The Monte Carlo approximation to the ESS safeguard condition is written as

$$\xi \textstyle\sum_{s=1}^{n} (\tilde{w}_k(x_{1:k}^{(s)}))^2 \leq \frac{1}{n}. \tag{3.65}$$

In the best case, the normalized importance weights are uniform, and the ESS is equal to the number of samples, n. In the worst case, we have a degenerate population with a single importance weight of 1, and so the ESS resolves to 1. Since the ESS is already a linearization of the variance, it is no surprise that the ESS safeguard condition has the same derivatives (3.62) and (3.64).

Penalizing the variance. The "global" variance safeguard works by introducing a penalty term into the variational objective. From (3.17), the penalized variational

objective is

$$F_\sigma(\theta) \equiv \langle E[a(X)], \theta - \theta^\star \rangle + c(\theta^\star) - c(\theta) + \sigma \mathrm{Var}[\tilde{w}(X)], \tag{3.66}$$

where $\sigma \geq 0$ is the strength of the variance penalty term. At the end of the kth iteration, the stochastic gradient of the penalized K-L divergence is given by

$$\nabla F_\sigma(\theta_k) \approx \sum_{s=1}^{n} \tilde{w}_k^{(s)} (a(x_k^{(s)}) - \bar{a})(a(x_k^{(s)}) - \bar{a})^T (\theta_k - \theta^\star) + \nabla S_k(\theta_k). \tag{3.67}$$

Over the course of the algorithm, it may eventually happen that the variance overwhelms the K-L divergence, and so the gradient no longer pushes the iterates in the direction of a solution near $\theta^\star$. It is still important to optimize the penalized variational objective (3.66) with stepwise variance safeguards (3.54) in order to prevent the algorithm from accumulating too much variance early on in the algorithm's execution. At this point, one major impediment to implementation is that it is not well-understood how to choose σ.

Despite these measures, resampling will occasionally be necessary over long sequences to prevent the population from degenerating to a single particle. Resampling, however, makes it difficult to keep track of the variance of the importance weights. A resample move resets the importance weights to zero, but the resampling procedure itself has the effect of introducing variance into the final estimate, so it is necessary to keep track of this additional variance when evaluating the variance penalty. Keeping track of the variance in an exact fashion is not plausible, since it would necessitate keeping all the particles that were *not* retained during the resample move. However, a reasonable upper bound on the variance of the importance weights can be obtained by assuming that the importance weights of the discarded samples are small relative to the samples that were retained by the resample move.

3.3.9 Algorithm summary

The basic algorithm is summarized in Fig. 3.4. The output of the algorithm is a collection of n samples $x^{(s)}$ and corresponding importance weights $w^{(s)}$ which can be used to estimate $E[\varphi(X)]$. The input $\theta^\star$ represents the parameterization of the target distribution $p(x) = p(x; \theta^\star)$, and the input θ_0 represents the parameterization of the initial approximating distribution $p(x; \theta_0)$. The practitioner must choose θ_0 in such a way that it is possible to draw samples directly from the joint distribution. Step 6 is the Robbins-Monro update (2.1) in which the steepest descent direction is replaced by the quasi-Newton search direction. The search direction is negative of H, the quasi-Newton approximation to inverse Hessian, times g_k, the Monte Carlo approximation to the gradient of the variational objective $F(\theta)$. When there are constraints on the parameters θ (as in Sec 3.5), the Robbins-Monro updates are replaced by the the interior-point stochastic approximation method described in Chapter 2. Most of the remaining steps of the main loop are standard parts of an SMC procedure, except for Step 5, the calculation of the variance-safeguarded step size α_k. There are many

- Let n, θ_0, $\theta^\star$, A, B be given.
- Draw samples $x^{(s)} \sim p(x; \theta_0)$.
- Set importance weights $w^{(s)} = 1/n$.
- Initialize inverse Hessian H to the identity matrix.
- **for** $k = 1, 2, 3, \ldots$
  1. Set maximum step size $\hat{a}_k$.
  2. Compute gradient estimate $g_k \approx \nabla F(\theta_{k-1})$ using samples $x^{(s)}$ and importance weights $w^{(s)}$; see (3.26).
  3. **if** $k > 1$, **then** modify H using damped quasi-Newton update (Sec 2.4), in which $\Delta\theta = a_{k-1}\Delta\theta_{k-1}$ and $\Delta g = g_k - g_{k-1}$.
  4. Compute search direction $\Delta\theta_k = -H \times g_k$.
  5. Compute variance-safeguarded step size $a_k \leq \hat{a}_k$ for given search direction $\Delta\theta_k$; see Sec. 3.3.8.
  6. Set $\theta_k = \theta_{k-1} + a_k\Delta\theta_k$.
  7. Update importance weights $w^{(s)}$ following (3.43).
  8. Run the two-stage Gibbs sampler:
     - Draw samples $x_B^{(s)} \sim p(z \,|\, x_A^{(s)}; \theta_k)$.
     - Draw samples $x_A^{(s)} \sim p(x \,|\, x_B^{(s)}; \theta_k)$.
  9. Resample the particles, if necessary.

Figure 3.4: The stochastic approximation algorithm for probabilistic inference.

conceivable improvements and extensions to the basic algorithm in Fig. 3.4.

Non-adaptive SMC algorithms, such as the hot coupling and AIS algorithms I discussed earlier, also implement steps 7, 8 and 9, so most of the additional computational cost for adaptating the sequence of distributions arises in step 2 (computing the gradient estimate) and step 4 (computing the search direction). Step 4 in particular may be prohibitive for large parameter vectors θ, though in my experience following the steepest descent direction instead of the quasi-Newton search direction by replacing H with the identity matrix always turned out to be a bad idea. Thus, it is best to make the space of variational parameters as small as possible.

3.3.10 A small example

Before moving on to the experiments, I illustrate the behaviour of the stochastic approximation algorithm for a small inference problem.

Explain more clearly what I infer by this example.

Consider an Ising model defined on a 12×12 lattice with periodic boundary conditions, and simulated at a low equilibrium temperature of $T = 1/10$. (If this technical language is unfamiliar to you, please note that I will cover the Ising model in the next section.) Each site i in the lattice corresponds to a magnetic dipole x_i that is in one of two spin states, $x_i = +1$ or $x_i = -1$. Due to the periodic boundary

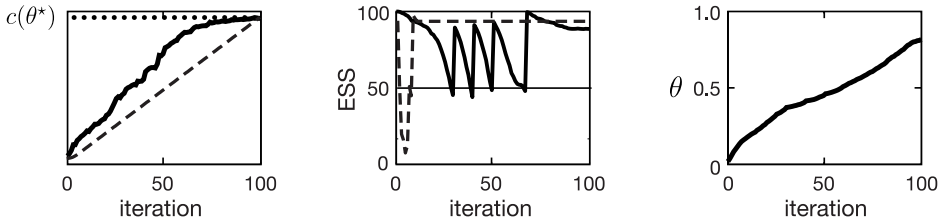

Figure 3.5: Small demonstration of AIS (dashed line) and the stochastic approximation algorithm (solid line) on a 12×12 Ising model.

conditions, each site i in the lattice—and, correspondingly, in the undirected graphical model—has exactly four neighbours. The simulation temperature is probably far enough away from the critical temperature that standard MCMC methods will work well for inferring thermodynamic properties of the Ising model (Newman & Barkema, 1999)—this small demonstration is not meant to serve as motivation on the new inference algorithm.

The target distribution is given by

$$p(x) = \exp\left\{\sum_{(i,j)} \theta^\star x_i x_j - c(\theta^\star)\right\}, \tag{3.68}$$

where the summation is over all unordered pairs of neighbouring sites (i, j) in the lattice. The interaction strengths are uniformly $\theta^\star = 1/T$. It is easy to see that the Ising model (3.68) is a member of the exponential family in standard form, in which the statistic $a(x)$ is simply the sum over all pairwise products $x_i x_j$. I've chosen the approximating distributions to also be of the same form, with $\theta^\star$ replaced by θ.

First, I ran the annealed importance sampling (AIS) method (Neal, 2001). AIS is a SMC algorithm in which the sequence of distributions follows a tempering scheme. The kth artificial distribution had bond strengths of $\theta_k = \gamma_k \theta^\star$, and $\{\gamma_k\}$ was an increasing sequence of inverse temperatures converging to 1, with $\gamma_0 = 0$. I simulated 100 particles on a sequence of 100 artificial tempered distributions. The results are depicted by the dashed lines in the left and middle plots of Fig. 3.5. It is clear from the left-hand plot in Fig. 3.5 that the AIS particles successfully recovered the log-normalization factor of the 12×12 Ising model at $\theta^\star$. The ESS quickly degenerated at initial stages, but subsequently stabilized. However, it is disconcerting that AIS failed to recover the two modes of the ferromagnetic Ising model; at the end of the simulation, every particle $x^{(s)}$ consisted entirely of "down" spins $x_i = -1$, but it should be noted that uniform configurations of "up" spins are also equally likely.

The results of running the stochastic approximation algorithm (Fig. 3.4) are shown by the solid lines in Fig. 3.5. I set the maximum step sizes to $\hat{a}_k = 1/(1+k)^{0.4}$. The left-hand plot shows that the Monte Carlo estimate of the variational lower bound (the solid line) recovered the log-partition function more rapidly than AIS. This is a somewhat surprising result, given that the stochastic approximation algorithm did so with a θ that was still far away from the target value of $\theta^\star = 10$; see the right-most plot of Fig. 3.5. From the point-of-view of the variational objective, what you will

find is that there is very little difference between 12×12 Ising models at temperatures of $T \approx 1$ and $T = 1/10$. This insensitivity of the variational objective makes sense from known properties of the Ising model: both temperatures of $T = 1/10$ and $T = 1$ are well below the *critical temperature* $T \approx 2.269$ (Newman & Barkema, 1999), so the "density of states" (Landau & Binder, 2005) should exhibit a similar shape at both temperatures. (Note that the log-normalization factors can be very different even when the corresponding densities of states are similar.) This outcome relates rather vividly to the scenario sketched in Fig. 3.2): the stochastic approximation algorithm has discovered an approximating distribution that resides along approximately the same level curve as the target, the advantage being that it lies closer to the distribution from which we initially drew the samples. (Refer to the discussion in Sec. 3.2.)

The effective sample size of the particle population in the stochastic approximation method degraded more gradually thanks to the variance safeguards (with parameter settings $\beta = 0.75$ and $\xi = 0.9$). Thanks to the discovery of a good approximate target, the algorithm was more successfully able capture the bimodality of the Ising model; about three quarters of the particles ended up in in the "up" spin mode, and the remaining quarter were in the "down" spin mode. The correct modes were discovered, although their relative volumes were not correctly estimated.

What can we infer about the stochastic approximation method from this small experiment? What we saw was that the stochastic approximation method was able to recover an Ising model with interaction strength $\theta \approx 1$ (corresponding to the $q^\star(x)$ in Fig. 3.2) that was much closer to the initial variational approximation ($\theta = 0$) yet retained the key features of the target distribution with interaction strength $\theta^\star = 10$. As a result, the stochastic approximation method was able to preserve the diversity of the sample because it did not need to travel as great a distance, and hence did not need to make a large transitions as the annealed importance sampling method. Thus, what we observed was a scenario similar to the one hypothesized in Sec. 3.2.

3.4 Application to statistical physics

In this section, I employ the 2-d spin glass as a test bed to investigate the viability of the proposed inference algorithm. Spin glasses are real magnetic substances in which random interactions below a certain temperature give rise to "frustration," the inability of the system to find an ordered state satisfying all interacting neighbours (Fischer & Hertz, 1991). A wide range of materials have been characterized as spin glasses. In the 1970s, Edwards and Anderson postulated a simple model of a spin glass that exhibited the critical properties of randomness, frustration, and long-range magnetic behaviour. Their model has helped advance the discovery phase transitions in these materials, a field of study in and of itself (Binder & Young, 1986; Landau & Binder, 2005; Sherrington, 2007). Attempts to come up with analytic solutions have yielded only sparse results, so Monte Carlo methods are extremely valuable to understanding the behaviour of spin glasses.

Spin glasses have also presented some of the toughest challenges to simulation, particularly for inferring thermodynamic properties at low temperatures. Depending

on how one starts the simulation, the system can find its way into a different energy "basin," so giving entirely different measurements of thermodynamic quantities. This phenomenon occurs because spin glasses contain metastable basins with low energy states surrounded by states with much higher energy (Newman & Barkema, 1999). Below the critical temperature, conventional Markov chain approaches do not work.

While the macroscopic properties of spin glasses are fairly well-understood, more fine-grained properties, such as the spin glass order parameter, have been difficult to identify—only recently have researchers made significant progress in this question (Landau et al., 2004). Great difficulties arise because a spin glass will possess a large number of thermodynamic states with the same macroscopic properties, such as the configuration energy, but with different microscopic properties. The order parameter of a spin glass—the statistical tool used to detect phase transitions, or changes in the system from one regime to another—is the amount of "spin overlap" between independent samples,

$$q \equiv \frac{1}{n} \sum_{i=1}^{n} x_i x_i', \tag{3.69}$$

where x and x' are two independently drawn spin configurations, and n is the number of magnetic dipoles in the spin glass (Bhatt & Young, 1985; Kawashima & Young, 1996). By convention, the order parameter is written as q. Due to the stated difficulty of sampling the states of the Edwards-Andersen spin glass, it is no surprise that estimating a distribution over q is a daunting task.

The challenge of predicting phase transitions in spin glasses has motivated the development of many advanced Monte Carlo methods such as parallel tempering (Earl & Deem, 2005; Hukushima et al., 1996), multicanonical Monte Carlo (Iba, 2001), and simulation methods that directly estimate the density of states (Wang & Landau, 2001). See Iba (2000) for a survey of these developments. The stochastic approximation method for variational inference is not expected to be outperform methods such as the Wang-Landau algorithm that are specifically designed for spin glass models.

3.4.1 Algorithmic development for spin glasses

The Edwards-Anderson spin glass is an Ising model with random, short-range spin interactions. Consider a random vector X with possible configurations $x = (x_1, \ldots, x_n) \in \Omega$. Dipoles or spins are placed on the sites $i \in \{1, \ldots, n\}$ of a lattice. Each spin can (somewhat unrealistically) take on one of two values, $x_i = +1$ for the "up" position, or $x_i = -1$ for "down." For a lattice with n sites, the system has 2^n possible configurations. The scalar θ_{ij} at every edge $(i, j) \in \mathcal{E}$ in the lattice defines the interaction between sites i and j. Setting $\theta_{ij} > 0$ causes attraction between spins, and $\theta_{ij} < 0$ induces repulsion. The probability distribution at equilibrium without an external magnetic field is

$$p(x) = \exp\left\{\sum_{(i,j)\in\mathcal{E}} \theta_{ij}^{\star} x_i x_j - c(\theta^{\star})\right\}, \tag{3.70}$$

in which I've absorbed the equilibrium temperature into the parameters $\theta^\star$. This probability measure is also known in statistical mechanics as the *Boltzmann distribution*. It comes from the equilibrium occupation probability of a finite state Hamiltonian, which was first derived by J. W. Gibbs in 1902. The distribution $p(x)$ is easily written in exponential family form (3.8), in which the entries of $a(x)$ are the products $x_i x_j$.

The approximating distributions $p(x;\theta)$ used in the experiments below are of the exact same form as (3.70), or a slight variation thereof. A sampling density is obtained simply by setting all the θ_{ij}'s to zero.

An undirected, labeled graph $G = (V, \mathcal{E})$, where $V \equiv \{1, \ldots, n\}$, can be used to represent the conditional independence structure of $p(x)$. Graph separation in G is equivalent to conditional independence: there is no edge between i and j if and only if X_i and X_j are conditionally independent given values at all other points of the graph. The two-stage Gibbs sampler is a valid Markov kernel for any distribution of the form (3.70) as long as the lattice can be partitioned into two acyclic subgraphs. To sample from the conditional distributions $p(x_A \,|\, x_B)$ and $p(x_B \,|\, x_A)$ defined on the acyclic subgraphs of G, I use a variation of the forward-filtering backward-sampling algorithm (Chib, 1996; Godsill et al., 2004) generalized to trees (Hamze & de Freitas, 2004). I performed all simulations on 2-d lattices with periodic boundary conditions (all dipoles had exactly four neighbours). I used a simple "checkerboard" partition to divide the undirected graphical model G into two trivially acyclic subgraphs.

Calculating the Rao-Blackwellized importance weights (3.43) requires computing the marginal probabilities $p(x_A)$ up to a normalizing constant, which in turn involves evaluation of an integral over all configurations x_B. This most efficient way to evaluate this integral is to run belief propagation on the conditional distribution $p(x_B \,|\, x_A)$, then evaluate the variational free energy, which is precisely equal to the normalizing constant of $p(x_A)$.

Due to the structure of the undirected graphical model, computing the Rao-Blackwellized estimates of the stochastic gradient is prohibitive for all but the smallest lattices because the covariance is an integral over a quadratic with $O(n^2)$ terms. Instead, I use the basic Monte Carlo gradient estimates (3.26).

3.4.2 Experiments with the Ising ferromagnet

To begin, I ran some experiments on the Ising ferromagnet, a simple—if not particularly realistic—model of a magnet that has occupied physicists for close to a century (for this reason, the Ising model has been called the "fruit fly" of statistical mechanics). Ernst Ising developed his model in order to explain the phenomenon of "spontaneous magnetization" in magnets. (Although he had to wait over ten years before he was able to verify his hypothesis!) The probability distribution is recovered from (3.70) by setting $\theta_{ij} = 1/T$, where T is the temperature at equilibrium. The Ising model is not of great interest to us given that MCMC works well except near the critical temperature, the temperature at which the phase transition takes place (Newman & Barkema, 1999). But its properties on 2-d finite lattices are known analytically through series expansions (Baxter, 1982), and can be used to verify the

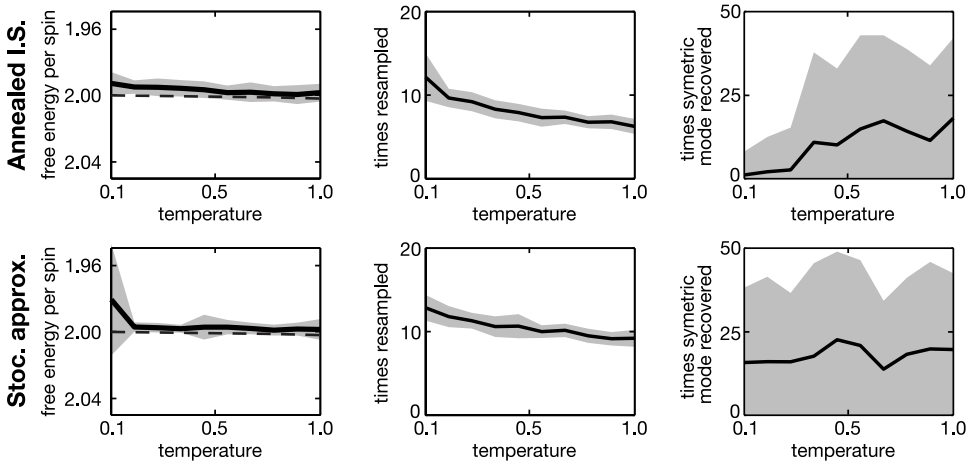

Figure 3.6: Simulations of annealed importance sampling *(top)* and the stochastic approximation method *(bottom)* on a 20×20 Ising model at a range of temperatures. The dashed line is the exact value of the free energy per spin, and the shaded region is the 90% confidence interval.

answers obtained via simulation.

Method. At each temperature $T = \frac{1}{10}, \frac{2}{10}, \ldots, 1$, I ran 20 independent simulations on a 20×20 lattice with periodic boundary conditions. I ran annealed importance sampling for 250 iterations with 100 particles and a uniform tempering schedule. The ESS threshold for resampling was set to 50. The settings for the stochastic approximation method were identical, with additional safeguard settings $\beta = 0.75, \xi = 0.9$, step sizes $a_k = 1/(1+k)^{0.65}$, and damping factor $\gamma = 0.75$ for the quasi-Newton updates. In my experience, the stochastic approximation method was insensitive to the choice of step size sequence. Unlike the target distribution, the variational distribution $p(x; \theta)$ was allowed to have non-uniform interactions θ_{ij}, and so the number of parameters was equal to the number of edges in the lattice.

The stochastic approximation method will take longer to execute than AIS due to the additional cost of computing the Monte Carlo gradient estimate and in computing the search direction. Since we have one parameter for every edge in the undirected graphical model, the computational cost can be rather significant for large systems. Thus, the improvements in accuracy for large systems should be measured up against these costs.

Results. The outcomes of the first experiment are shown in Fig. 3.6. The solid line in the left-hand column is the estimate of the Helmholtz free energy per spin—a quantity proportional to the log-normalizing factor $c(\theta^\star)$—averaged over the independent trials. The shaded region is the 90% confidence interval, and the dashed line is the exact value following the calculations of Ferdinand and Fisher (1969). The middle column shows the number of times at each temperature the ESS fell below the threshold, a rough measure of the quality of the sample. The critical result is in the rightmost column: at low temperatures, the distribution produced by annealed

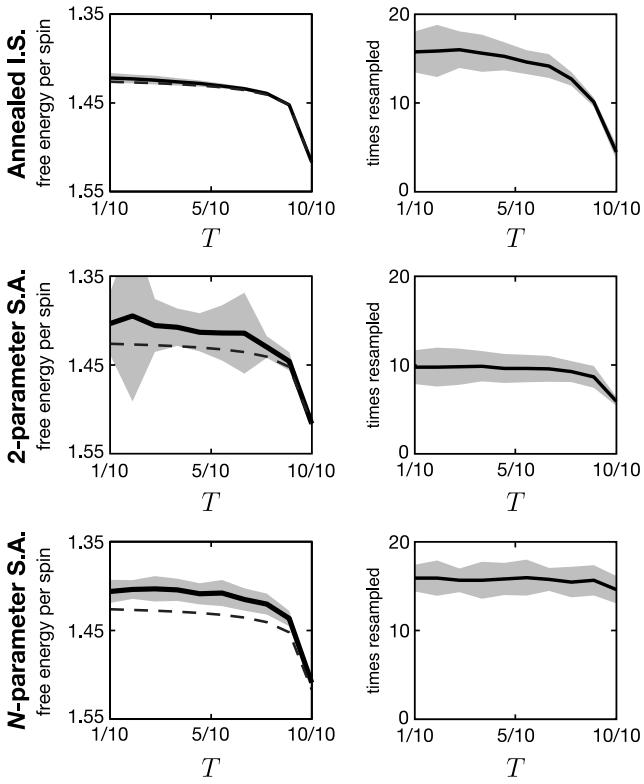

Figure 3.7: Simulations of annealed importance sampling *(top)* and stochastic approximation *(middle, bottom)* on a 20×20 spin glass at various temperatures. The dashed line is the exact value of the free energy per spin, and the shaded region is the 90% confidence interval.

importance sampling degenerated to a single point, whereas stochastic approximation much more adeptly recovered the two modes—all up spins or all down spins—of the Ising model. A perfect result would be 50 out of 100 samples, since both modes are equally likely. What is not shown in Fig. 3.6 is that our method tended to overestimate the variance (or "specific heat") of the spin configurations. The high variance did not appear to be supported by the final approximating distribution, so it must have been an artifact of importance sampling. Finally, it is rather intriguing to observe that the stochastic approximation method succeeded in discovering a spin glass that approximates the Ising ferromagnet.

3.4.3 Experiments with spin glasses

I ran a second set of experiments on random-interaction spin glass models exhibiting frustration. The bonds θ_{ij}, following the Edwards-Anderson model, were set to $-1/T$ or $+1/T$ uniformly at random.

Method. The second experiment was similar to the first, with a few key differences. By running the simulations at temperatures $T = \frac{1}{10}, \frac{1}{9}, \ldots, 1$, I was able to use

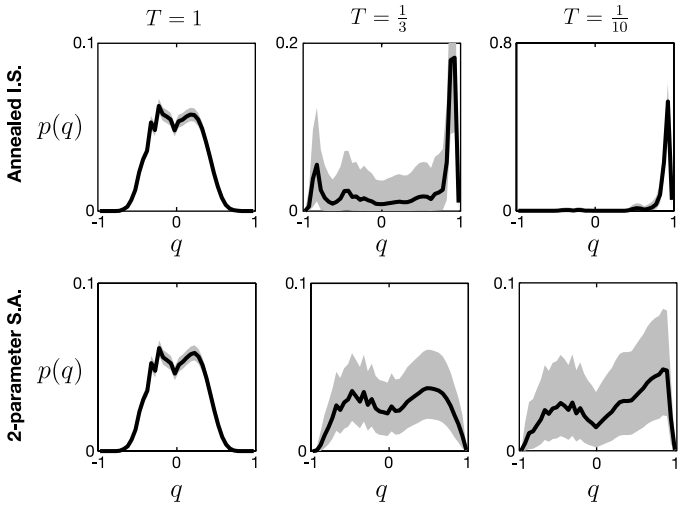

Figure 3.8: Estimated distribution of the order parameter q at three temperatures. The shaded region is the 90% confidence interval.

the deterministic, modular arithmetic algorithm of Galluccio et al. (2000) to obtain exact solutions for the normalization factor and free energy.[13] Since their polynomial time algorithm is valid for all toroidal lattices, it clearly applies to 2-d lattices with periodic boundary conditions, although it quickly becomes impractical for lattices much larger than 20×20. See Saul and Kardar (1993) for background on computing the ground state of an Ising spin glass. Note that the algorithm of Galluccio et al. (2000) cannot be used to compute the distribution of the order parameter q. I increased the number of particles to 1000 in order to obtain a fine-grained estimate of q. I compared AIS to the stochastic approximation method with two different parameterizations of the variational distribution: one in which every edge had its own parameter ("N-parameter"), and another with only two parameters ("2-parameter"), one for the attractive edges and another for the repulsive edges.[14]

Results. The results of the spin glass experiments are shown in Figures 3.7 and 3.8. The dashed line is the ground truth, and the shaded region is the 90% confidence interval. From the plots in Fig. 3.7, it appears that AIS best infers properties of the 20×20 spin glass. But the free energy is a macroscopic property, hence it is comparatively easy to estimate. The order parameter q—instrumental for predicting phase transitions in spin glasses—is, by contrast, notoriously difficult to infer. As the simulation temperature was lowered, the annealed importance sampling estimate

[13] All the intermediate computations in the algorithm of Galluccio et al. (2000) are performed in exact arithmetic on Galois fields (Moon, 2005). What is rather remarkable is that the algorithm reports the number of states at every energy level with no round-off error. For instance, for the 20×20 spin glass used in my experiments, at a temperature of $T = 1$ there are precisely (in hexadecimal) E70 86DA03B4 4DAAA1E4 2168EF59 E05C5087 7C529FA6 129BE377 C2DE37F2 BE810F99 226C5AE1 58BCC61E F42C411C 83984144 configurations that have an energy of zero.

[14] Note that a 1-parameter approximation to the spin glass would behave little differently from the AIS method.

degenerated to a point (see the top row of Fig. 3.8). On the other hand, in the bottom row of Fig. 3.8 we see that the estimates from the 2-parameter variational approximation method exhibited some variance, but they still managed to retain the symmetry of the order parameter distribution. The estimates of the N-parameter scheme (not shown) were actually no better than the AIS estimates. My suspicion is that this was due to an ill-conditioned variational objective.

3.5 Application to population genetics

The development of the polymerase chain reaction in the early 1980s opened the door for techniques to sequence highly variable polymorphic genetic markers. Of particular importance are the generic markers found in noncoding regions of the genome—internal regions that are not translated into polypeptides, the building blocks of proteins. A large portion of noncoding regions is made up of repeated sequences. These repeated sequences, called *microsatellites* or *short tandem repeats*, may be present in different numbers, and it is their variation that has formed the basis for detecting differences among individuals in a population (Ellegren, 2004). Microsatellite genetic markers can reveal high levels of diversity much more adeptly than traditional DNA sequencing due to their higher mutation rate and their ability to genotype a large number of individuals quickly and cheaply (Pearse & Crandall, 2004). Microsatellites are also *co-dominant* markers, meaning that they are able to distinguish homozygotes and heterozygotes. The most reliable microsatellites are not linked to loci that undergo selection, since most statistical models used by geneticists assume *neutrality.*

For geneticists, these molecular tools are indispensable, and are routinely used for hypothesizing relationships among organisms. For example, microsatellites have been used to determine the genealogy of human populations (Pritchard et al., 2000b); to identify the parental population of individuals that have unusual origin, say, to enforce wildlife poaching laws (Manel et al., 2002); and to assess the ancestry of individuals in inferring disease risks (Hartl & Clark, 2007). An early application was to fisheries management, where the problem was to determine the species being harvested, and to detect gene flow within the fishery (Davies et al., 1999). The power of genetics is that one can identify significant differences between individuals, and exploit these differences to infer evolutionary history, even with small amounts ($< 1\%$) of genetic variation (Hartl & Clark, 2007).

The problem is that all the aforementioned tasks require defining a priori population structure. Population subdivision arises from the separation of mating individuals, and it can be caused by geographical phenomena, migration and social stratification. But it can be difficult to identify. The classic approach via Wright's F_{ST} statistic calculates the reduction in heterozygous genotypes relative to the expected frequencies under Harvey-Weinberg random mating (Hartl & Clark, 2007). However, this approach can be misleading because many patterns of subdivision can lead to the same response (Hedrick, 2005; Nei, 1973; Neigel, 2002; Weir & Cockerham, 1984). Also, genetic differences between individuals can be small and subtle.

The Bayesian model of Pritchard et al. (2000a) offers a solution to this conun-

drum by simultaneously identifying both patterns of population subdivision and the ancestry of individuals from highly variable genetic markers. It divides a population into K islands, or *demes*, and assumes Hardy-Weinberg random mating within each one. The demes represent smaller populations, such as a flocks or herds, within which random mating can take place. Population stratification leads to *inbreeding*. Therefore, genetic clusters can be identified by shared ancestry among their genomes. Hardy-Weinberg equilibrium assumes uniformly random mating and an infinite-sized population, so that *genetic drift*—divergence due to random changes in allele frequencies—does not occur (Gillespie, 2004). (This means that Hardy-Weinberg equilibrium cannot apply to sex-linked genes.) The basic model (Pritchard et al., 2000a) also assumes no linkage between loci.[15] Since the assumption is that all linkage disequilibrium or inbreeding is caused by population substructure, the model can be understood as trying to minimize inbreeding.[16] The model of Pritchard et al. (2000a) has made important contributions to the difficult problem of detecting population stratification in humans (Rosenberg et al., 2002)—difficult due to the high level (95%) of variation within groups—and has lead to a better scientific understanding of race and ethnicity (Burchard et al., 2003).

This model, however, can be frustrating to work with because independent MCMC simulations can produce remarkably different answers for the same data, even with simulations millions of samples long. The problem of simultaneously inferring the admixture proportions and population stratification that bests fits the genotype data is intimately related to the problem of fitting a mixture model to data, a classical problem in Bayesian statistics and machine learning (Celeux et al., 2000). In such problems, MCMC can do a poor job exploring the hypothesis space when there are several divergent hypotheses that explain the data. These obstacles are also well-noted in the genetics and evolutionary biology literature (Huelsenbeck et al., 2002; Rannala, 2002). For large studies, the computational demands are so great that computer clusters have been set up at Washington State and Cornell Universities to carry out population structure simulations. Our ability to acquire highly variable genetic markers is quickly outstripping our capacity for making statistical inferences.

Note that parallel tempering variants of MCMC have been used for related phylogenetic inference problems (Huelsenbeck et al., 2002), but my study suggests that they are ineffective.

3.5.1 Algorithmic development for LDA

In this section, I investigate how the stochastic approximation algorithm can be used to make accurate inferences in latent Dirichlet allocation, or LDA (Blei et al., 2003). LDA is nearly the same as the the population structure model of Pritchard et al. (2000a) assuming fixed Dirichlet priors. Approximate methods are needed due to the

[15] At a molecular level, when two genes are *linked* the loci of the genes are on the same chromosome, and so the genes assort independently at meiosis (Griffiths et al., 2008).

[16] Note that linkage disequilibrium could also be due to other confounding factors, such as selection, selective mating, or self-pollenation.

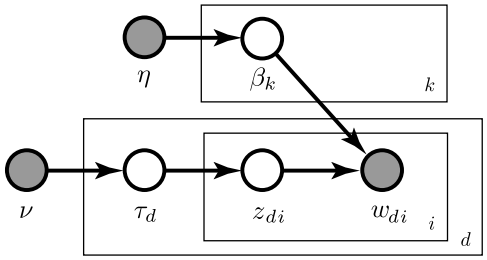

Figure 3.9: The directed graphical model with plates for LDA. Shaded nodes represent observations or fixed quantities.

presence of an intractable integral in the denominator of the posterior.

Even though the LDA model is only a few years old, it has already been adopted for a wide array of machine learning applications, including modeling scientific topics and information retrieval (Griffiths & Steyvers, 2004; Rosen-Zvi et al., 2004), object recognition (Barnard et al., 2003; Sudderth et al., 2005), and social network analysis (McCallum et al., 2005). Latent semantic analysis (Landauer et al., 1998; Hofmann, 2001), which is related to LDA, has been around for some time, but having a Bayesian model and a probabilistic interpretation is important in order to resolve queries in a principled manner.

Latent Dirichlet allocation is a generative model of a collection of documents, or *corpus*. Its two key features are: the order of the words is unimportant, and each document is drawn from a mixture of topics. The corpus consists of a collection of D documents, and each document is expressed as a "bag" of word tokens. I'll assume all the documents are of the same length N. Each word $w_{di} = j$ refers to a vocabulary item indexed by $j \in \{1, \ldots, W\}$, where W is the size of the vocabulary.[17] The ith word in the dth document is associated with a latent topic indicator z_{di}. Each topic indicator takes a value a from from 1 to K, where K is a fixed value representing the number of topics. Observing the jth vocabulary item in the kth topic occurs with probability $p(w_{di}{=}j \,|\, z_{di}{=}k, \beta) = \beta_{kj}$. The vector β_k represents the word proportions for the kth topic. The word proportions must represent proper probabilities, so it is required that $\sum_j \beta_{kj} = 1$ for all $k = 1, \ldots, K$, and $\beta_{kj} \geq 0$. Since there are K topics and W vocabulary items, this process is fully specified by $K \times W$ parameters. It would be difficult to specify these parameters by hand, so we learn them by treating them as unobserved variables. The word proportions for the kth topic are generated according to a Dirichlet distribution with fixed prior η_k:

$$p(\beta_k \,|\, \eta_k) = \frac{\Gamma(\sum_j \eta_{kj})}{\prod_j \Gamma(\eta_{kj})} \prod_{j=1}^{W} \beta_{kj}^{\eta_{kj}-1}. \tag{3.71}$$

The latent topic indicators are generated independently according to $p(z_{di}{=}k \,|\, \tau_d) =$

[17] I realize that my notation here conflicts with my notation for importance weights, but it should be obvious from the context which symbol refers to a word, and which refers to an importance weight.

τ_{dk}, and the vector of topic proportions τ_d for document d in turn follows a Dirichlet with fixed prior ν_d. Ordinarily, uniform and exchangeable Dirichlet priors are assumed by setting all the entries of η and ν to the same value. In summary, LDA assumes the following generative process:

1. For every topic k, draw $\beta_k \sim p(\cdot \,|\, \eta_k)$.
2. For every document d, draw $\tau_d \sim p(\cdot \,|\, \nu_d)$.
3. For every document d:
   - For every word i, draw $z_{di} \sim p(\cdot \,|\, \tau_d)$.
   - For every word i, draw $w_{di} \sim p(\cdot \,|\, z_{di}, \beta)$.

The generative process I have just described defines a joint distribution over the observed data w and unknowns $x = \{\beta, \tau, z\}$ given the hyperparameters $\{\eta, \nu\}$. The joint density is given by

$$\begin{aligned} p(w, x \,|\, \eta, \nu) &= \prod_{k=1}^{K} p(\beta_k \,|\, \eta_k) \times \prod_{d=1}^{D} p(\tau_d \,|\, \nu_d) \times \prod_{d=1}^{D} \prod_{i=1}^{N} p(w_{di} \,|\, z_{di}, \beta)\, p(z_{di} \,|\, \tau_d). \\ &\propto \prod_{k=1}^{K} \prod_{j=1}^{W} \beta_{kj}^{\eta_{kj} + m_{kj} - 1} \prod_{d=1}^{D} \prod_{k=1}^{K} \tau_{dk}^{\nu_{dk} + n_{dk} - 1}, \end{aligned} \tag{3.72}$$

where $m_{kj} \equiv \sum_d \sum_i \delta_k(z_{di})\, \delta_j(w_{di})$ counts the number of times the jth word is assigned to the kth topic, and $n_{dk} \equiv \sum_i \delta_k(z_{di})$ counts the number of words assigned to the kth topic in the dth document. The directed graphical model with plates (Buntine, 1994) for LDA with exchangeable priors is shown in Fig. 3.9. To simplify the expressions written in the passages below, I omit the ranges of products and summations. I use the following logical variables throughout: d is an index that ranges over the documents, i ranges over the words in each document d, j ranges over items in the vocabulary, and k ranges over the topics.

If you were to imagine for a moment that each document is written in multiple languages, the correspondence between LDA and the population structure model of Pritchard et al. (2000a) is:

$$\begin{aligned} \text{documents} &\Leftrightarrow \text{individuals} \\ \text{topics} &\Leftrightarrow \text{demes} \\ \text{languages} &\Leftrightarrow \text{loci} \\ \text{vocabulary items} &\Leftrightarrow \text{alleles}. \end{aligned}$$

The extension to multiple languages is easily achieved by adding plates to Fig. 3.9 so that we have a set of words (or alleles) w and word proportions β for each language (or locus).

The modeling assumptions of LDA have a direct interpretation in evolutionary biology. The multinomial distribution $p(w \,|\, z, \beta)$ over observed alleles w in each population directly corresponds directly to Harvey-Weinberg equilibrium; see Hedrik

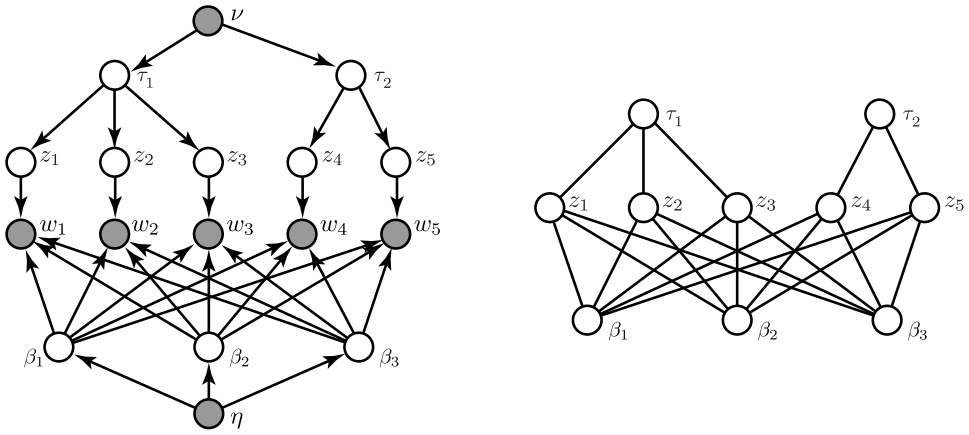

Figure 3.10: Directed graphical model *(left)* and undirected graphical model *(right)* for latent Dirichlet allocation with three topics and two documents.

(2005). The number of vocabulary items W at each locus represents the arity of the microsatellite or single nucleotide polymorphism (SNP). And the representation of a document d as a distribution of topics τ_d allows for the possibility of mixed ancestry, or *admixture*, meaning that each genotyped allele can come from one of the K populations. In this sense, the clusters are not only an aggregate of individuals, but are also an aggregate of allele frequencies.[18]

Extensions to the basic model account for linked loci (Falush et al., 2003), dominant genetic markers such as amplified fragment length polymorphisms (Falush et al., 2007), and uncertainty in the number of demes via a Dirichlet process (Huelsenbeck & Andolfatto, 2007). Including within-group correlations via linkage disequilibrium typically gives the model greater power to detect similar but distinct populations, but requires a more careful study design (Rosenberg et al., 2005). Pritchard et al. (2000a) caution that the number of clusters, K, has no straightforward biological interpretation, so I do not explore estimation of K in experiments.

Standard algorithms for inference in LDA include MCMC (Griffiths & Steyvers, 2004; Pritchard et al., 2000a), variational inference with mean field approximations (Blei et al., 2003; Teh et al., 2007b), and expectation propagation (Minka & Lafferty, 2002). Before developing the stochastic approximation algorithm, it will be helpful to examine how MCMC and variational mean field are implemented for LDA.

Two-stage Gibbs sampler. The graphical structure of LDA allows us to implement a two-stage Gibbs sampler in which we alternately draw samples from the word proportions β and the topic proportions τ given the latent variables z, then draw z given β and τ. To understand why the graphical structure allows us to do this, it is helpful to examine the undirected graphical model for an example corpus with three

[18]This is precisely the mathematical abstraction of genetic variation envisaged by Fisher (1953): "the frequencies with which the different genotypes occur define the gene ratio characteristic of the population, so that it is often more convenient to consider a natural population not so much as an aggregate of individuals, but rather as an aggregate of gene ratios."

topics and two documents, shown in Fig. 3.10.

In the first stage of the Gibbs sampler, the word proportions for the kth topic are drawn from the Dirichlet distribution

$$p(\beta_k \mid z, \eta) \propto \prod_{j=1}^{W} \beta_{kj}^{\eta_{kj}+m_{kj}-1}. \tag{3.73}$$

The topic proportions for the dth document are drawn from the Dirichlet density

$$p(\tau_d \mid z, \nu) \propto \prod_{k=1}^{K} \tau_{dk}^{\nu_{dk}+n_{dk}-1}. \tag{3.74}$$

In the second stage, each latent topic indicator z_{di} is drawn from the multinomial proportional to

$$p(z_{di}=k \mid \beta, \tau, w_{di}=j) \propto \tau_{dk}\beta_{kj}. \tag{3.75}$$

This is the effectively the MCMC algorithm described by Pritchard et al. (2000a).

The collapsed Gibbs sampler. The Gibbs sampler of Griffiths and Steyvers (2004) is derived in a different manner by integrating out two of the unknowns, the topic proportions τ and word proportions β, so that only discrete variables z remain. This is conceivably a better strategy than the two-stage Gibbs sampler, because the approximate inference only involves discrete variables, and because conditional expectations over the continuous variables can be evaluated exactly.

The posterior over the latent indicators is

$$\begin{aligned} p(z \mid w, \eta, \nu) &= \iint p(x \mid w, \eta, \nu)\, d\beta\, d\tau \\ &\propto \prod_d \int \prod_k \tau_{dk}^{\nu_{dk}+n_{dk}-1} d\tau_d \times \prod_k \int \prod_j \beta_{kj}^{\eta_{kj}+m_{kj}-1} d\beta_k. \end{aligned} \tag{3.76}$$

The integrals in (3.76) are in a familiar form: the product of a Dirichlet times a multinomial. From the Dirichlet integral of type 1 (Jeffreys & Swirles, 1956), the first integral in (3.76) becomes

$$\int \prod_k \tau_{dk}^{\nu_{dk}+n_{dk}-1} d\tau_d = \frac{\Gamma(\sum_k \nu_{dk})}{\Gamma(\sum_k \nu_{dk} + N)} \times \prod_k \frac{\Gamma(\nu_{dk} + n_{dk})}{\Gamma(\nu_{dk})}. \tag{3.77}$$

Analytic expressions for the other integrals are obtained in a similar manner. As a result, we have an analytic expression for the posterior:

$$\begin{aligned} p(z \mid w, \eta, \nu) = &\prod_d \frac{\Gamma(\sum_k \nu_{dk})}{\Gamma(\sum_k \nu_{dk} + N)} \prod_k \frac{\Gamma(\nu_{dk} + n_{dk})}{\Gamma(\nu_{dk})} \\ &\times \prod_k \frac{\Gamma(\sum_j \eta_{kj})}{\Gamma(\sum_j \eta_{kj} + t_k)} \prod_j \frac{\Gamma(\eta_{kj} + m_{kj})}{\Gamma(\eta_{kj})}, \end{aligned} \tag{3.78}$$

where $t_k \equiv \sum_d \sum_i \delta_k(z_{di})$ is defined to be the number of times a word is assigned to the kth topic. We now have a posterior solely over the discrete variables z.

While it isn't possible to obtain samples directly from (3.78), it is possible to sample the latent indicator at a single site given values at all the other sites. Denoting z_{-di} to be the set of all indicator variables in the corpus except z_{di}, we have

$$\begin{aligned} p(z_{di}=k \mid z_{-di}, w, \eta, \nu) &= \frac{p(z_{di}=k, z_{-di} \mid w, \eta, \nu)}{p(z_{-di} \mid w, \eta, \nu)} \\ &\propto (\nu_{dk} + n_{dk} - 1) \times \frac{\eta_{kj} + m_{kj} - 1}{\sum_j \eta_{kj} + t_k - 1}, \end{aligned} \tag{3.79}$$

such that $w_{di}=j$. This is equivalent to the expression given in Griffiths and Steyvers (2004). The first line comes from the definition of conditional probability, and we arrive at the second line after eliminating common terms in the numerator and denominator. For instance, when we set z_{di} to topic k, the term

$$\Gamma(\nu_{dk} + n_{dk}) = (\nu_{dk} + n_{dk} - 1)\Gamma(\nu_{dk} + n_{dk} - 1)$$

would be the almost the same as if we chose z_{di} to be instead some other topic $k' \neq k$, the one exception being $(\nu_{dk} + n_{dk} - 1)$, which is precisely the term that appears in the expression (3.79). Once we have samples z, it is easy to obtain samples for the word proportions β and the topic proportions τ. Mean Rao-Blackwellized estimates of β and τ work out to be

$$E[\tau_{dk}] \approx \sum_{s=1}^{n} \frac{\nu_{dk} + n_{dk}^{(s)}}{\sum_{k'} \nu_{dk'} + N}, \tag{3.80}$$

$$E[\beta_{kj}] \approx \sum_{s=1}^{n} \frac{\eta_{kj} + m_{kj}^{(s)}}{\sum_{j'} \eta_{kj'} + t_k^{(s)}}. \tag{3.81}$$

The posterior (3.78) is only over discrete random variables, so it has strong ties to the spin glass models explored earlier in this chapter. Thus, an MCMC algorithm that draws iteratively from the full conditionals of LDA experiences computational difficulties similar to the Gibbs sampler for the Ising spin glass—both can get easily stuck in local modes.

Mean field theory. In Sec. 3.3.2, I used Jensen's inequality to derive a variational lower bound (3.15). Maximizing this lower bound gave us the variational principle. In this section, following Blei et al. (2003), I design the variational approximation $q(x)$ so that the expectations in (3.15) have a closed form. This is an example of a *mean field approximation.*[19] As reported in Buntine and Jakulin (2004), mean field estimates can be biased and often quite poor. Still, it represents a good starting point for variational inference.

In the variational formulation of Blei et al. (2003), the approximating distribution

[19]This is not the only choice of mean field approximation. Teh et al. (2007a) proposed an approximating family inspired by the collapsed Gibbs sampler. Their factorization assumptions are weaker than those made by Blei et al. (2003), so the resulting approximations are arguably better.

is fully-factorized, which means it decomposes as a product of terms

$$q(x;\theta) = \prod_k q(\beta_k;\xi_k) \times \prod_d q(\tau_d;\gamma_d) \times \prod_d \prod_i q(z_{di};\phi_{di}), \tag{3.82}$$

where $\theta = \{\xi, \gamma, \phi\}$ comprises the set of variational parameters. Unlike the variational methods I proposed in Sec. 3.3, the mean field approximation does not have the same form as the target $p(x)$, so I must use $q(x;\theta)$ to refer to the approximating distribution. A sensible approximating distribution is one that exhibits a similar form to that of the target, so the individual factors in (3.82) are defined to be

$$\begin{aligned} q(\beta_k;\xi_k) &= \text{Dirichlet}(\beta_k \,|\, \xi_k) \\ q(\tau_d;\gamma_d) &= \text{Dirichlet}(\tau_d \,|\, \gamma_d) \\ q(z_i{=}k;\phi_{di}) &= \phi_{dik}. \end{aligned}$$

I now proceed to derive a coordinate ascent algorithm for optimizing the variational lower bound (3.15) under the specified class of approximating distributions.

Expanding terms in the variational objective (3.13), we obtain

$$\begin{aligned} F(\theta) = &-\textstyle\sum_k E[\log p(\beta_k \,|\, \eta_k)] - \sum_d E[\log p(\tau_d \,|\, \nu_d)] \\ &- \textstyle\sum_d \sum_i E[\log p(w_{di} \,|\, z_{di}, \beta)] - \sum_d \sum_i E[\log p(z_{di} \,|\, \tau_d)] \\ &+ \textstyle\sum_k E[\log q(\beta_k;\xi_k)] + \sum_d E[\log q(\tau_d;\gamma_d)] + \sum_i \sum_d E[\log q(z_{di};\phi_{di})]. \end{aligned} \tag{3.83}$$

where all the expectations are defined with respect to $q(x;\theta)$. The first four integrals comprise the average energy, and the remaining three integrals comprise the negative entropy of $q(x)$. I now proceed to investigate each of these integrals. The expectation with respect to $\log p(\tau_d \,|\, \nu_d)$ for document d resolves to

$$\begin{aligned} E[\log p(\tau_d \,|\, \nu_d)] = &\log \Gamma(\textstyle\sum_k \nu_{dk}) - \sum_k \log \Gamma(\nu_{dk}) \\ &+ \textstyle\sum_k (\nu_{dk} - 1) \int q(\tau_d;\gamma_d) \log \tau_{dk} \, d\tau_d. \end{aligned} \tag{3.84}$$

The integral on the right-hand side of (3.84) is the expectation of $\log \tau_{dk}$ with respect to a Dirichlet distribution on the vector τ_d. We can exploit the properties of the exponential family (3.7) to solve this integral.

Written in standard form, the Dirichlet density with parameter θ is

$$\text{Dirichlet}(x \,|\, \theta) = \exp\left\{\textstyle\sum_k (\theta_k - 1) \log x_k + \log \Gamma(\sum_k \theta_k) - \sum_k \log \Gamma(\theta_k)\right\}. \tag{3.85}$$

From this equation, it is easy to see that the Dirichlet distribution is a member of the exponential family with natural parameters $\theta_k - 1$, log-normalization factor $c(\theta) = \sum_k \log \Gamma(\theta_k) - \log \Gamma(\sum_k \theta_k)$, and sufficient statistics $\log x_k$, which are analogous to the quantities of interest $\log \tau_{dk}$. In fact, the integrals in (3.84) are precisely the entries of the mean statistic. Recall our earlier result that the mean statistic is recovered by taking derivatives of $c(\theta)$ with respect to the natural parameters.

Therefore,

$$E[\log x_k] = \frac{\partial c(\theta)}{\partial \theta_k} = \Psi(\theta_k) - \Psi(\textstyle\sum_k \theta_k), \tag{3.86}$$

where $\Psi(\,\cdot\,)$ is the digamma function, the first derivative of the logarithm of the Gamma function. From this result, (3.84) becomes

$$\begin{aligned} E[\log p(\tau_d \mid \nu_d)] = {} & \log \Gamma(\textstyle\sum_k \nu_{dk}) - \sum_k \log \Gamma(\nu_{dk}) \\ & + \textstyle\sum_k (\nu_{dk} - 1)(\Psi(\gamma_{dk}) - \Psi(\sum_{k'} \gamma_{dk'})). \end{aligned} \tag{3.87}$$

It is then a straightforward application of (3.86) to obtain an expression for the expectation of $\log q(\tau_d; \gamma_d)$:

$$\begin{aligned} E[\log q(\tau_d; \gamma_d)] = {} & \log \Gamma(\textstyle\sum_k \gamma_{dk}) - \sum_k \log \Gamma(\gamma_{dk}) \\ & + \textstyle\sum_k (\gamma_{dk} - 1)(\Psi(\gamma_k) - \Psi(\sum_{k'} \gamma_{dk'})). \end{aligned} \tag{3.88}$$

The expectation of $\log q(z_{di}; \phi_{di})$ is easily derived to be

$$E[\log q(z_{di}; \phi_{di})] = \textstyle\sum_k \phi_{dik} \log \phi_{dik}. \tag{3.89}$$

The expectation of $\log p(z_{di} \mid \tau_d)$ resolves to

$$E[\log p(z_{di} \mid \tau_d)] = \textstyle\sum_k \phi_{dik} \left(\Psi(\gamma_{dk}) - \Psi(\sum_k \gamma_{dk})\right). \tag{3.90}$$

The expectation of $\log q(\beta_k; \xi_k)$ for each topic k is

$$\begin{aligned} E[\log q(\beta_k; \xi_k)] = {} & \log \Gamma(\textstyle\sum_j \xi_{kj}) - \sum_j \log \Gamma(\xi_{kj}) \\ & + \textstyle\sum_j (\xi_{kj} - 1)(\Psi(\xi_{kj}) - \Psi(\sum_{j'} \xi_{kj'})). \end{aligned} \tag{3.91}$$

And in a very similar manner can we expand on the expectation of $\log p(\beta_k \mid \eta_k)$, arriving at the expression

$$\begin{aligned} E[\log p(\beta_k \mid \eta_k)] = {} & \log \Gamma(\textstyle\sum_j \eta_{kj}) - \sum_j \log \Gamma(\eta_{kj}) \\ & + \textstyle\sum_j (\eta_{kj} - 1)(\Psi(\xi_{kj}) - \Psi(\sum_{j'} \xi_{kj'})). \end{aligned} \tag{3.92}$$

Finally, the expectation of $\log p(w_{di} \mid z_{di}, \beta)$ with respect to the approximating distribution $q(x; \theta)$ is

$$E[\log p(w_{di} \mid z_{di}, \beta)] = \textstyle\sum_k \sum_j \delta_j(w_{di})\, \phi_{dik} (\Psi(\xi_{kj}) - \Psi(\sum_{j'} \xi_{kj'})). \tag{3.93}$$

With these derivations in hand, we can construct the full expression for $F(\theta)$.

For a large corpus, optimizing the variational objective (3.83) with mean field approximation $q(x)$ by taking gradient descent steps is infeasible (let alone Newton steps) because we will need to keep track of a vector θ of size $WK + DK + DNK$. Blei et al. (2003) describe a clever way to minimize the variational objective without

- Choose an initial value for ϕ.
- **Repeat** until the convergence criterion is met:
  1. **For** $k = 1, \ldots, K$ and $j = 1, \ldots, W$:
     - Set $\xi_{kj} = \eta_{kj} + \sum_d \sum_i \delta_j(w_{di})\, \phi_{dik}$.
  2. **For** $d = 1, \ldots, D$ and $k = 1, \ldots, K$:
     - Set $\gamma_{dk} = \nu_{dk} + \sum_i \phi_{dik}$.
  3. **For** $d = 1, \ldots, D$ and $i = 1, \ldots, N$:
     - **For** $k = 1, \ldots, K$, set $\phi_{dik} = \exp\big\{\Psi(\xi_{kj}) - \Psi(\sum_{j'} \xi_{kj'}) + \Psi(\gamma_{dk}) - \Psi(\sum_{k'} \gamma_{dk'})\big\}$, such that $w_{di} = j$.
     - **For** $k = 1, \ldots, K$, normalize ϕ_{dik}.

Figure 3.11: Coordinate descent for optimizing the LDA mean field approximation.

having to keep track of all the parameters at once. The key is to derive a coordinate descent algorithm akin to the two-stage Gibbs sampler, in which steps are iteratively taken along ξ and γ with ϕ fixed, then along ϕ with ξ and γ fixed. The coordinate descent algorithm is summarized in Fig. 3.11. In the third step of the main loop, observe that we do not need store the values ϕ_{dik} for every d, i and k—we only need to keep track of the sums that appear in the coordinate descent updates for ξ and γ. The drawback of the coordinate descent algorithm is that its convergence to a stationary point is typically very slow.

The coordinate descent directions are derived as follows. The optimization problem is constrained since the parameters ϕ_{di} for each document d must lie on the probability simplex, and the Dirichlet parameters γ and ξ must be positive. Necessary and sufficient conditions for optimality in a constrained optimization problem can be obtained by introducing Lagrange multipliers and constructing the Lagrangian function (Nocedal & Wright, 2006). Assuming that iterates in the optimization algorithm lie in the strict interior of the feasible region, the Lagrange multipliers associated with inequality constraints are always inactive, hence they can be ignored during coordinate descent. Denoting the Lagrange multipliers associated with the equality constraints by υ_{di}, the Lagrangian $L(\theta, \upsilon)$ is given by

$$\begin{aligned} L(\theta, \upsilon) = {} & \textstyle\sum_k \big\{ \sum_j \log\Gamma(\eta_{kj}) - \log\Gamma(\sum_j \eta_{kj}) \big\} + \sum_d \big\{ \sum_k \log\Gamma(\nu_{dk}) - \log\Gamma(\sum_k \nu_{dk}) \big\} \\ & + \textstyle\sum_k \big\{ \log\Gamma(\sum_j \xi_{kj}) - \sum_j \log\Gamma(\xi_{kj}) \big\} + \sum_d \big\{ \log\Gamma(\sum_k \gamma_{dk}) - \sum_k \log\Gamma(\gamma_{dk}) \big\} \\ & + \textstyle\sum_d \sum_k \big\{ \Psi(\sum_{k'} \gamma_{dk'}) - \Psi(\gamma_{dk}) \big\} \big(\nu_{dk} + \sum_i \phi_{dik} - \gamma_{dk} \big) \\ & + \textstyle\sum_k \sum_j \big\{ \Psi(\sum_{j'} \xi_{kj'}) - \Psi(\xi_{kj}) \big\} (\eta_{kj} + \sum_d \sum_i \delta_j(w_{di})\, \phi_{dik} - \xi_{kj}) \\ & + \textstyle\sum_d \sum_i \sum_k \phi_{dik} \log \phi_{dik} - \sum_d \sum_i \upsilon_{di} \{ \sum_k \phi_{dik} - 1 \}. \end{aligned} \tag{3.94}$$

Descent directions along individual coordinates are obtained by finding the roots of the derivatives of the Lagrangian (3.94) with respect to the variational parameters.

Taking the derivative of the Lagrangian (3.94) with respect to γ_{dk}, we obtain

$$\begin{aligned}\frac{\partial L}{\partial \gamma_{dk}} &= \Psi'(\textstyle\sum_{k'}\gamma_{dk'})\sum_{k'}(\nu_{dk'} + \sum_i \phi_{dik'} - \gamma_{dk'}) \\ &\quad - \Psi'(\gamma_{dk})(\nu_{dk} + \textstyle\sum_i \phi_{ik} - \gamma_{dk}),\end{aligned} \tag{3.95}$$

where $\Psi'(\,\cdot\,)$ is the trigamma function, or equivalently the derivative of the digamma function. By setting

$$\gamma_{dk} = \nu_{dk} + \textstyle\sum_i \phi_{dik}, \tag{3.96}$$

for all $k = 1, \ldots, K$, the K equations (3.95) resolve to zero. Thus, (3.96) is an descent direction for the topic proportions for each document. This update is fairly intuitive; it promotes topic proportions that have more words assigned to them.

Next, consider descent along ξ. The derivative of $L(\theta, \upsilon)$ with respect to ξ_{kj} is

$$\begin{aligned}\frac{\partial L}{\partial \xi_{kj}} &= \Psi'(\textstyle\sum_{j'}\xi_{kj'})\sum_{j'}(\eta_{kj'} + \sum_d \sum_i \delta_{j'}(w_{di})\,\phi_{dik} - \xi_{kj'}) \\ &\quad - \Psi'(\xi_{kj})(\eta_{kj} + \textstyle\sum_d \sum_i \delta_j(w_{di})\,\phi_{dik} - \xi_{kj}).\end{aligned} \tag{3.97}$$

If we set

$$\xi_{kj} = \eta_{kj} + \textstyle\sum_d \sum_i \delta_j(w_{di})\,\phi_{dik}, \tag{3.98}$$

for all $j = 1, \ldots, W$, the W equations (3.97) resolve to zero, and thus the update (3.98) specifies a proper descent direction for ξ_k. The coordinate descent directions (3.96) and (3.98) ensure that the variational parameters remain within the strict interior of the feasible set as long as all the η, ν and ϕ are already positive.

Finally, let's work out the coordinate descent update for ϕ. The derivative of the Lagrangian with respect to ϕ_{dik} is

$$\frac{\partial L}{\partial \phi_{dik}} = \Psi(\textstyle\sum_{j'}\xi_{kj'}) - \Psi(\xi_{kj}) + \Psi(\sum_{k'}\gamma_{dk'}) - \Psi(\gamma_{dk}) + \log \phi_{dik} + 1 - \upsilon_{di}. \tag{3.99}$$

such that $w_{di} = j$. Setting this derivative to zero, we obtain the root

$$\phi_{dik} \propto \exp\big\{\Psi(\xi_{kj}) - \Psi(\textstyle\sum_{j'}\xi_{kj'}) + \Psi(\gamma_{dk}) - \Psi(\sum_{k'}\gamma_{dk'})\big\}, \tag{3.100}$$

We recover the normalizing constant, hence the Lagrange multiplier υ_{di}, in (3.100) by dividing by the updated, unnormalized parameters ϕ_{dik} for all $k = 1, \ldots, K$.

Equations (3.96), (3.98) and (3.100) are all the ingredients we need for the mean field coordinate descent algorithm. The full recipe was given earlier in Fig. 3.11.

Stochastic approximation. Unlike the variational method I just described, my variational inference algorithm does not need clever coordinate-wise updates to optimize the variational objective. Gradient or quasi-Newton steps are feasible because the number of parameters invariant to the size of the corpus. Unlike the stochastic approximation algorithm for Ising spin glasses (Sec. 3.4), my implementation for LDA

is able to use Rao-Blackwellized stochastic gradient estimates.

The first step in development of the stochastic approximation algorithm is the design of the variational approximation to the posterior. I chose a class of approximating densities of the form

$$p(x;\theta) = \exp\big\{\textstyle\sum_d \sum_k (\nu^\star_{dk} + n_{dk} - 1)\log\tau_{dk} + \sum_k \sum_j (\eta_{kj} - 1)\log\beta_{kj} \\ + \phi\textstyle\sum_k \sum_j m_{kj}\log\beta_{kj} + \gamma\sum_k\sum_j (c_j - m_{kj})\log\beta_{kj} - c(\theta)\big\}, \quad (3.101)$$

where $c_j \equiv \sum_d \sum_i \delta_j(w_{di})$ is the number of times jth vocabulary item is observed, and the variational parameters are $\theta = \{\eta, \phi, \gamma\}$. From now on, I denote the target Dirichlet priors by $\eta^\star$ and $\nu^\star$. I've written the variational distribution in such a way that the connection to the exponential family is obvious; the natural parameters are ϕ, γ and η_{kj} for every topic k and vocabulary item j (along with a parameter that remains constant), and the corresponding entries of the statistic $a(x)$ are

$$\eta_{kj} \Leftrightarrow \log\beta_{kj} \quad (3.102)$$

$$\phi \Leftrightarrow \textstyle\sum_k\sum_j m_{kj}\log\beta_{kj} \quad (3.103)$$

$$\gamma \Leftrightarrow \textstyle\sum_k\sum_j (c_j - m_{kj})\log\beta_{kj}. \quad (3.104)$$

Next, I need to show that my chosen family of approximating distributions satisfies the criteria given in Sec. 3.3.5. It is already clear that $p(x;\theta)$ is a member of the exponential family. By setting $\phi = 1$, $\gamma = 0$ and $\eta = \eta^\star$, the target posterior is recovered. And whenever $\phi = \gamma$, the counts m_{kj} disappear from (3.101), and $p(x;\theta)$ becomes a sampling density with a tractable, closed form expression for $c(\theta)$:

$$c(\theta) = \textstyle\sum_k\sum_j \log\Gamma(\eta_{kj} + c_j\phi_{kj}) - \sum_k \log\Gamma(\sum_j \eta_{kj} + c_j\phi_{kj}) \\ + \sum_d\sum_k \log\Gamma(\nu_{dk}) - \sum_d \log\Gamma(\textstyle\sum_k \nu_{dk}). \quad (3.105)$$

The derivation of this expression is very similar to the derivation of the marginal posterior (3.78) over the discrete variables z.

At this point, there are two ways to go about designing the algorithm: either we integrate out the random variables $\{\beta, \tau\}$, in which case we only need to keep track of the samples $z^{(s)}$, or we integrate out the discrete variables z, in which case we only need to store the samples $(\beta^{(s)}, \tau^{(s)})$. The former strategy turns out to be the better choice because it leads to Rao-Blackwellized estimates of the gradient that are efficient to calculate thanks to the structure of the graphical model. Conditioned on z, much of the graphical model becomes disconnected, so the majority of the entries in the covariance matrix $\mathrm{Var}[a(x)]$ are given by the outer product (3.33).

Observe that all sufficient statistics (3.102-3.104) are either given by responses $\log\beta_{kj}$, or linear combinations of them. Therefore, the expectations of X and Y that appear in (3.32) and (3.33) follow directly from the identity (3.86) derived earlier.

For every topic k and for every sample s, we need to evaluate $\mathrm{Var}[\log\beta_k]$ with respect to the conditional Dirichlet density $p(\beta_k \mid z^{(s)};\theta)$. The entries of this covariance

matrix are derived, as before, by appealing to special properties of the exponential family, namely that $\mathrm{Var}[a(X)] = \nabla^2 c(\theta)$; see (3.11). So from (3.86), we have that

$$\mathrm{Cov}[\log x_k, \log x_{k'}] = \frac{\partial^2 c(\theta)}{\partial \theta_k \partial \theta_{k'}} = \begin{cases} \Psi'(\theta_k) - \Psi'(\sum_{k''} \theta_{k''}) & \text{if } k = k', \\ -\Psi'(\sum_{k''} \theta_{k''}) & \text{otherwise.} \end{cases} \tag{3.106}$$

The remaining entries of the covariance matrix involve sufficient statistics that are composed of linear combinations of responses $\log \beta_{kj}$. The Rao-Blackwellized estimates for these covariances are derived in a similar fashion by again appealing to the identity (3.32) and properties of the Dirichlet distribution (Narayanan, 1991).

Constraints of the form $\theta \geq 0$ must be imposed to guarantee that $p(x; \theta)$ is a valid probability density. I implemented the stochastic primal-dual interior point method described in the previous chapter. The gradient descent or quasi-Newton search direction is replaced by the primal-dual search direction,

$$\begin{aligned} (B + S)\Delta\theta &= -(g - \mu/\theta) \\ \Delta\lambda &= \mu/\theta - \lambda - S\Delta\theta, \end{aligned} \tag{3.107}$$

where λ is the vector of dual variables, μ is the barrier parameter, g is the stochastic gradient, B is the quasi-Newton approximation to the Hessian, S is the matrix with λ/θ along its diagonal, and the division operator is element-wise. I found that separate primal and dual step sizes significantly improved performance. I also tried a simpler approach via projection (Gafni & Bertsekas, 1984), but it was too slow to be of use.

The expressions for the conditionals $p(\beta, \tau \mid z; \theta)$ and $p(z \mid \beta, \tau; \theta)$ in the two-stage Gibbs sampler look very much like (3.73), (3.74) and (3.75). And the marginal density $p(z; \theta)$, needed to compute the Rao-Blackwellized importance weights (3.43), is very similar to the expression for the marginal posterior (3.78) derived earlier.

3.5.2 Experiments

The goal is to obtain posterior estimates that are both accurate—that is, they recover the "true" population structure—and in high agreement over independent simulations. To ascertain the extent to which the stochastic approximation method achieves these aims, I designed a series of experimental comparisons with annealed importance sampling (AIS), the two-stage Gibbs sampler as it is implemented in the software STRUCTURE[20], and the collapsed Gibbs sampler described in Griffiths and Steyvers (2004), and I developed two statistical measures to rigorously assess the accuracy of the estimates.

Method. I used the software package CoaSim (Mailund et al., 2005) to simulate three synthetic data sets according to a non-equilibrium coalescent process. A *coalescent* is a lineage of alleles in a sample traced backward in time to their common ancestor allele (Gillespie, 2004). The coalescent process is the stochastic process that generates the genealogy. It assumes an approximation to the Wright-Fisher neutral model of evolution, which is a good approximation so long as the sample sizes are

[20] Available at http://pritch.bsd.uchicago.edu/structure.html.

small relative to the population size (Hudson, 1991). In the absence of selection, the mutational process and the transmission of genes from one distribution to the next are independent processes. Thus, a sample configuration of genes can be simulated by first generating a random genealogical history for a segment of a chromosome then, conditional on this genealogy, randomly placing mutations according to a specified mutation model (Hein et al., 2005; Hudson, 2002). The standard mutation model is a constant-rate Poisson distribution over the number of mutations. Under this model, the number of mutations is completely dependent on the number of generations since the most recent common ancestor. This property is exploited in coalescent simulation techniques. The coalescent process applies equally to haploids (e.g. mitochondrial genomes) and diploid organisms, such as humans.

I simulated the coalescent process with divergence events at coalescent times 2, 1 and 0.5.[21] I used a finite sites model to simulate 10 microsatellite loci with a maximum of 30 alleles at each locus and a scaled mutation rate of 5, 2 and 0.5 for each of the three data sets.[22] The rate of recombination is determined by the relative distance between the markers. I used CoaSim's Scheme interface to randomly generate microsatellite markers and simulate the genealogical process. Here the Scheme code (Abelson et al., 1996) used to generate the first data set:

```
(let* ((markers (make-random-ms-markers 10 5 30))
       (seqs (simulate-sequences markers
       '(population 4
                  (merge 2.0
                    (population 1 (sample 30))
                    (population 3
                      (merge 1.0
                        (population 1 (sample 30))
                        (population 2
                          (merge 0.5
                            (population 1 (sample 30))
                            (population 1 (sample 30))))))))))
  (display seqs)
  (newline)
  (call-with-output-file "output.txt" (sequences-printer seqs)))
```

CoaSim outputs an ancestral recombination graph, but I only needed the genetic sequences at the leaves the graph. In the end, each data set consisted of genotypes of 15 diploid individuals sampled from each of the 4 subpopulations, for a total of 60 samples.

For each data set, and for each K from 2 to 6, I carried out 20 independent trials of the three methods. For fair comparison, I ran the methods with the same number of sampling events: for both MCMC methods (the two-stage Gibbs sampler and the collapsed Gibbs sampler) I ran a Markov chain of length 50000 with a burn-in of 10000,

[21] Calendar time = 2 × effective population size × generation length × coalescent time.

[22] Under neutrality, the scaled mutation rate is usually written as $\theta = 4N\mu$, where N is the effective population size and μ is the probability of mutation at a single locus (Gillespie, 2004).

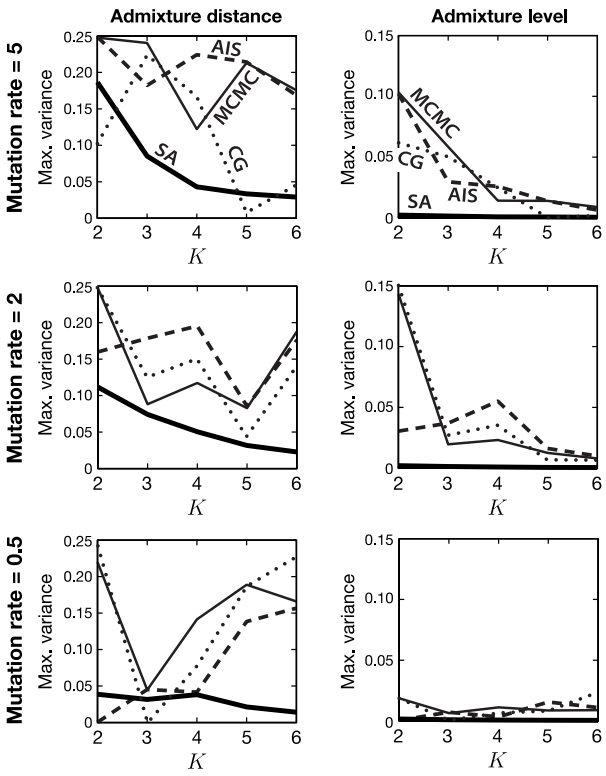

Figure 3.12: Variance in estimates of the invariant statistics taken over 20 trials. The dashed line is annealed importance sampling ("AIS"), the thin solid line is the two-stage Gibbs sampler ("MCMC"), the dotted line is the collapsed Gibbs sampler ("CG"), and the thick solid line is the stochastic approximation method ("SA").

and for both SMC methods (AIS and my stochastic approximation method) I used 100 particles and 500 iterations. A more honest comparison of these inference algorithms would fix the computational cost instead of the number of sampling events, but in practice such comparison of computing time can be highly questionable, particularly in the MATLAB programming environment. In my experience, I found the running times of all the algorithms to be quite comparable. For instance, the stochastic approximation algorithm took only about twice as long to run as the two-stage Gibbs sampler under the same number of sampling events.

Additional algorithm settings included an ESS threshold of 50, maximum step sizes $\hat{a}_k = 1/(1+k)^{0.6}$, centering parameters $\sigma_k = 1/k^{0.9}$, safeguards $\beta = 0.95$ and $\xi = 0.9$, and a quasi-Newton damping factor of 0.75. I set the initial iterate of stochastic approximation to $\phi = \gamma = \eta = \eta^\star$. I used uniform Dirichlet priors $\eta^\star = 0.1$ and $\nu^\star = 0.1$ throughout.

As with most mixture models, LDA suffers from the "label-switching problem" (Stephens, 2002), so it is inappropriate to aggregate estimates over independent trials or particles. (It is usually safe in practice to average over a Markov chain because it

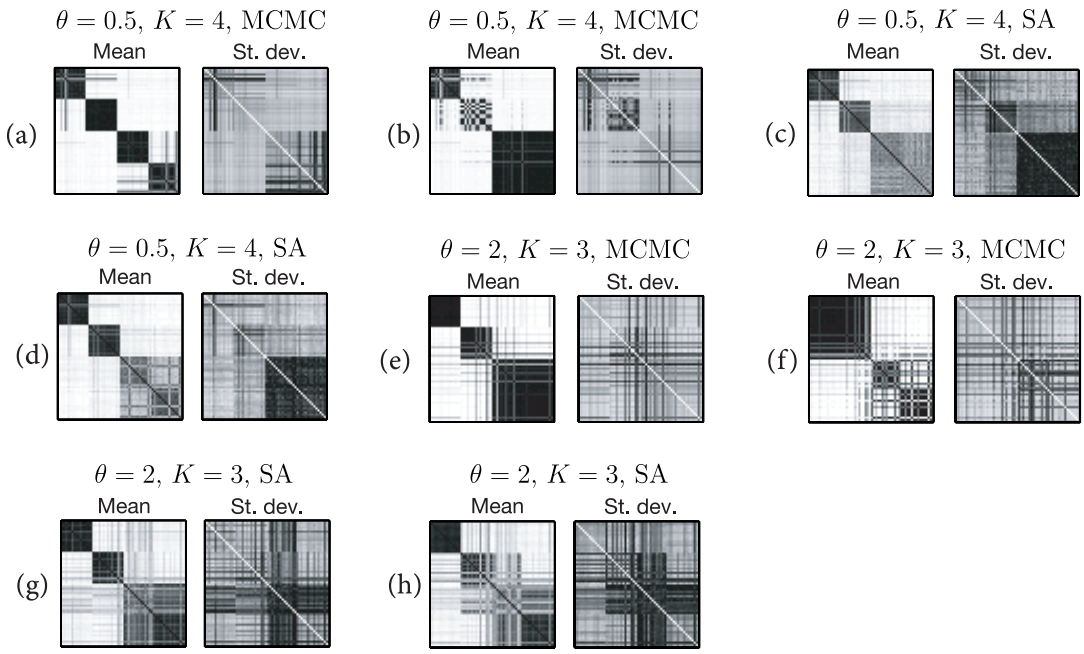

Figure 3.13: Estimated mean and standard deviation of the admixture distance statistic for a range of simulation settings. "SA" stands for stochastic approximation, and "MCMC" refers to the two-stage Gibbs sampler.

rarely visits the posterior's symmetric modes). To resolve this issue, I developed two informative statistics that are invariant to label ordering: the *admixture distance*, a number between 0 and 1 that measures the extent to which two individuals differ in their ancestry, and the *admixture level*, a number between 0 and 1, where 0 means an individual's alleles all come from a single subpopulation, and 1 means its ancestry is shared equally among the K subpopulations. The admixture distance is defined to be the total variational distance

$$\varphi_{\text{admixture distance}}(d, d') \equiv \tfrac{1}{2} \textstyle\sum_{k=1}^{K} |\tau_{dk} - \tau_{d'k}|, \tag{3.108}$$

and the admixture level is defined to be

$$\varphi_{\text{admixure level}}(d) \equiv 1 - \tfrac{K}{2(K-1)} \textstyle\sum_{k=1}^{K} \left|\tau_{dk} - \tfrac{1}{K}\right|. \tag{3.109}$$

For approaches to summarizing and visualizing the output of MCMC, see Rosenberg (2004) and Huelsenbeck et al. (2002).

Results. A summary of the experimental results is shown in Fig. 3.12. The stochastic approximation method significantly reduced the variance across independent trials in almost all cases, while AIS offered little advantage over the MCMC algorithms. To produce these plots, I took the individual or pair of individuals that exhibited the most variance in the admixture level or distance.

To assess the accuracy of the answers, I show some the anecdotal evidence in

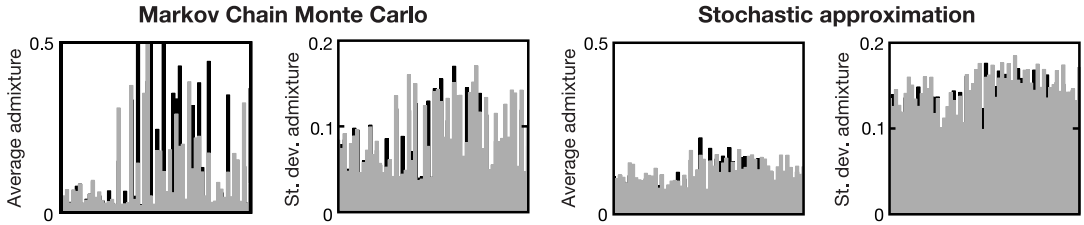

Figure 3.14: Mean and standard deviation of the admixture level, according to two independent runs (black, grey) of the two-stage Gibbs sampling implementation of MCMC *(left)* and stochastic approximation *(right)*.

Fig. 3.13. Each pair of images shows the mean and standard deviation of the admixture distance in a different experimental setting. The 60 rows and columns in each image correspond to individuals sorted by their true population label. A darker pixel in the left image means a smaller expected admixture distance, and in the right image it means a lower confidence. The top row shows the two-stage Gibbs sampler ("MCMC") (a,b) and stochastic approximation (c,d) trials on the data set with $K = 4$ and a scaled mutation rate of $\theta = 0.5$. In (a), the MCMC algorithm for the most part correctly assigned the individuals to their coalescent subpopulations. Contrast this result with (c) and (d): stochastic approximation failed to distinguish the most recently diverged subpopulations. Since the last divergence event is less than the the expected time for genes to find their common ancestor ("coalescence"), the model has difficulty clustering the most recent (rightmost) subpopulations. However, the empirical results suggest that (c) and (d) are more representative of the true posterior, because MCMC trials (a,b) are much more inconsistent and occasionally spurious (b). This trend is repeated in the bottom row, which shows results from the MCMC (e,f) and stochastic approximation (g,h) simulations in a more challenging scenario with $K = 3$ and a scaled mutation rate of $\theta = 2$: the stochastic approximation method exhibited some variance in its estimates (g,h) of the admixture distance, but MCMC tends to place disproportionate support on very accurate (e) or very strange (f) solutions.

To further demonstrate these tendencies, I plotted in Fig. 3.14 the admixture levels of the 60 sampled individuals for the two trials that produced the least similar estimates. Stochastic approximation again demonstrated much higher agreement. While not shown in Fig. 3.14, annealed importance sampling and the collapsed Gibbs sampler both exhibited similar results to the two-stage gibbs sampler.

3.6 Conclusions and discussion

In this chapter, I proposed a new stochastic approximation approach to probabilistic inference, and explored its merits on two important and particularly challenging problems, one in statistical physics and another in population genetics. I spent much of this chapter developing a formalism that merges ideas from optimization, Monte

Carlo methods, information theory, variational methods, and stochastic approximation. The results of the experiments show that the proposed method offers distinct advantages for problems that are poorly tackled with existing methods.

Many open questions remain.

This study suggests that the success of my approach hinges on a good search direction. The quasi-Newton search direction, however, may not be viable for large problems due to the $O(n^2)$ complexity of maintaining the matrix factorization. It remains to be seen whether limited-memory quasi-Newton methods (Nocedal & Wright, 2006) can be properly adapted to the stochastic setting.

One of the challenges in quantifying the behaviour of the stochastic approximation algorithm is separating the effect of the variational approximation from the effects of SMC; biased results may be an artifact of the forward and backward kernels, not the final variational approximation. More in-depth studies are needed to explore this question.

We are faced with a great deal of choice in the design of variational approximation, and it not clear *a priori* whether one given parameterization is better than another. More parameters introduce greater flexibility at the cost of a more difficult optimization problem, but there is certainly a need to develop more useful design guidelines.

Also, as I mentioned previously, the performance of the algorithm with a stepwise variance safeguard can be sensitive to the number of iterations. A more coherent strategy is needed that strikes a balance between the variance of the sample and the quality of the variational approximation. I proposed one possible approach in Sec. 3.3.8, but there remain some impediments to its implementation. One glaring issue is the mismatch between the formulation of the Kullback-Leibler divergence and the variance penalty term. Nevertheless, the results of this chapter do hint at a new algorithmic perspective on probabilistic inference, one in which algorithms are specifically designed to monitor and control for the variance of the final estimate. Ultimately, we want to trade-off three competing objectives: variance, bias, and computational complexity. This trade-off has been addressed in studies on the value of computation for specific probabilistic models. It remains to be seen whether the framework I have proposed might lead to new insights on the value of computation is less restricted settings.

One drawback of the Kullback-Leibler divergence as a variational objective, as witnessed in Sec. 3.3.10, is that it is not sensitive to the choice of query $E[\varphi(X)]$. An interesting and potentially fruitful direction to explore is the design of more flexible variational objectives that take into account the user's query.

The experiments in this chapter called to attention only a small subset of the many aspects of the novel algorithmic framework for probabilistic inference. Future work in this direction includes tackling more challenging problems by introducing structural assumptions, making improvements to existing variational methods, such as loopy belief propagation, that are derived via the dual objective, and developing algorithms for on-line learning that are more robust than stochastic expectation maximization (Delyon et al., 1999; Sato, 2000).

Finally, this new approach to inference also suggests novel ways to learn the potentials of undirected graphical models, or Markov random fields, when it is difficult or impossible to compute the log-normalization factor. Possible applications include learning Boltzmann machines and deep belief networks (Hinton, 2002; Hinton et al., 2006; Salakhutdinov et al., 2007), learning models for low-level vision and optical flow (Sun et al., 2008; Szeliski, 1989), and learning to recognize objects in scenes by taking into account the context (Carbonetto et al., 2004).

Chapter 4

Additional improvements to inference: conditional mean field

The aim of this chapter is to motivate and develop a new approach to probabilistic inference that overcomes the limitations inherent in conventional Monte Carlo methods and the problems caused by variational factorization assumptions, leading to more accurate, more reliable predictions.

The proposed scheme is to construct increasingly flexible variational classes by recovering edges left out from the previous factorization, then optimizing a weighted combination of mean field objectives to the conditionals. This idea of recovering dependencies in the target model, first proposed by Hamze and de Freitas (2006), is particularly appropriate in models that are explicitly formulated as undirected graphical models or Markov random fields. However, this scheme may be less appropriate for undirected or directed graphical models with large, non-pairwise factors, such as latent Dirichlet allocation (Sec. 3.5.1) and other mixture models.

In this chapter, I develop a new approach that preserves the generality of variational mean field, so that it can be applied to a very broad set of probabilistic inference problems. I refer to this new approach as *conditional mean field.* This new approach may incur greater computational expenses than competing variational or MCMC schemes, but these extra expenses may be warranted for solving difficult inference problems.

Several Monte Carlo methods have been proposed to correct for the discrepancy between the factorized variational approximations and the target distribution. These methods include importance sampling (Ghahramani & Beal, 2000; Muyan & de Freitas, 2003) and adaptive MCMC (de Freitas et al., 2001). However, none of these techniques scale well to general, high-dimensional state spaces because the variational approximations tend to be too restrictive when used as a proposal distribution. This is corroborated by experimental results in those papers as well as theoretical results (Sadowsky & Bucklew, 1990).

I propose an entirely new approach that overcomes these drawbacks by constructing a sequence of variational approximations that converges to the target distribution. To accomplish this, I derive a new class of *conditionally-specified* mean field approximations, and use SMC to obtain samples from them. SMC acts as a mechanism to migrate particles from an easy-to-sample distribution (naive mean field) to a difficult-to-sample one (the distribution of interest), through a sequence of artificial distributions. Each artificial distribution is a *conditional mean field* approximation, designed in such a way that it is at least as sensible as its predecessor because it recovers dependencies left out by mean field. In summary, I propose an SMC algorithm in which each artificial distribution is the solution to a conditionally-specified variational optimization problem. Sec. 4.3 explains these ideas thoroughly.

Like the previous chapter, I test the performance of the conditional mean field algorithm on the Ising spin glass model (Newman & Barkema, 1999). Conditional mean field is a rather generic algorithm, so I don't expect it to perform as well as algorithms designed specifically for the spin glasses. Surprisingly, the experiments in Sec. 4.4 show that conditional mean field achieves performance comparable to algorithms that are among the best available.

Sections 4.1 and 4.2 serve as background for the presentation of the main contribution in Sec. 4.3.

4.1 Mean field theory

Consider a slight extension to the Edwards-Anderson spin glass (3.70) with a non-uniform external magnetic field. Following the notation used in Sec. 3.4.1, scalars θ_i define the effect of the external magnetic field on the energy of the system. The probability density is then

$$p(x) = \exp\left\{\sum_{i\in V}\theta^\star_i x_i + \sum_{(i,j)\in\mathcal{E}}\theta^\star_{ij}x_i x_j - c(\theta^\star)\right\}. \tag{4.1}$$

From (3.15), the variational lower bound for the target (4.1) can be written as

$$F(\theta) = \sum_{i\in V}\theta^\star_i\mu_i(\theta) + \sum_{(i,j)\in\mathcal{E}}\theta^\star_{ij}\mu_{ij}(\theta) + H(\theta). \tag{4.2}$$

where $H(\theta)$ is the entropy of the approximating distribution $p(x;\theta)$, and the *mean statistics* are defined to be $\mu(\theta) \equiv E[a(X)]$, so that $\mu_i(\theta)$ and $\mu_{ij}(\theta)$ are the expectations of single spins i and pairs of spins (i,j), respectively.[1] The problem, as I discussed in the thesis introduction and in Sec. 3.3.2, is that the variational principle is of little practical use because no tractable expressions exist for the entropy and the mean statistics; both involve integrals that, in the case of the Ising model, reduce to exponentially large, finite summations over all the possible configurations. Superficially, we appear to have an optimization problem, but unless we do something clever we still have solve an integral that is intractable (or worse). The good news is that there exist restricted subsets of θ whereby it is possible to compute both the entropy and the mean statistics in a tractable fashion. In this section, I give background on one particular tractable family, *naive mean field*, in the context of the Ising spin glass.

Naive mean field requires that the approximating density $p(x;\theta)$ be written as a product of factors defined solely on the spins x_i at individual sites i. The factored representation can thus be expressed in the form

$$p(x;\theta) = \exp\left\{\sum_{i\in V} f_i(x_i;\theta) - c(\theta)\right\}. \tag{4.3}$$

For the Ising spin glass, this means that the variational parameters θ must belong to the set $\{\theta \,|\, \forall\,(i,j) \in \mathcal{E}, \theta_{ij} = 0\}$. We now can compute the variational lower

[1] To be clear, (4.2) is precisely the negative of the K-L divergence (3.17), plus a constant. In the previous chapter, the goal was to *minimize* the K-L divergence. Therefore, the stated optimization problem in this chapter must be to *maximize* $F(\theta)$.

bound (4.2) for any θ belonging to this subset because we have tractable expressions for the mean statistics and entropy. The mean of x_i is given by

$$\mu_i(\theta) \equiv \int x_i \, p(x;\theta) \, dx = \frac{\sum_{x_i} x_i \exp\{\theta_i x_i\}}{\sum_{x_i} \exp\{\theta_i x_i\}} = \tanh(\theta_i), \tag{4.4}$$

and the mean of the pairwise sufficient statistic $x_i x_j$ is

$$\begin{aligned} \mu_{ij}(\theta) &\equiv \int x_i \, x_j \, p(x;\theta) \, dx \\ &= \frac{\sum_{x_i} \sum_{x_j} x_i \, x_j \exp\{\theta_i x_i + \theta_j x_j\}}{\sum_{x_i} \sum_{x_j} \exp\{\theta_i x_i + \theta_j x_j\}} \\ &= \mu_i(\theta) \, \mu_j(\theta). \end{aligned} \tag{4.5}$$

Given the mean field factorization, the entropy decomposes as

$$H(\theta) = \textstyle\sum_{i \in V} p(x_i = -1) \log p(x_i = -1) + \sum_{i \in V} p(x_i = +1) \log p(x_i = +1).$$

For ease of presentation, I omitted the marginals' dependence on θ. Using the fact

$$\mu_i = \textstyle\sum_{x_i} x_i p(x_i) = -p(x_i = -1) + p(x_i = +1) \tag{4.6}$$

and

$$p(x_i = -1) = 1 - p(x_i = +1), \tag{4.7}$$

together yielding the simple identity

$$p(x_i;\theta) = \big(1 + x_i \mu_i(\theta)\big)/2, \tag{4.8}$$

the final expression for the entropy is given by

$$H(\theta) = -\sum_{i \in V} \left(\tfrac{1-\mu_i(\theta)}{2}\right) \log \left(\tfrac{1-\mu_i(\theta)}{2}\right) - \sum_{i \in V} \left(\tfrac{1+\mu_i(\theta)}{2}\right) \log \left(\tfrac{1+\mu_i(\theta)}{2}\right). \tag{4.9}$$

The standard way to proceed (Wainwright & Jordan, 2003a; Yedidia, 2001) is to derive coordinate ascent updates by equating the derivatives $\partial F / \partial \mu_i$ to zero and solving directly for μ_i. Since the variables μ_i must be valid mean statistics, they are constrained to lie within an envelope known as the *marginal polytope* (Wainwright & Jordan, 2003b). The coordinate ascent algorithm for optimizing the objective (4.2) is shown in Fig. 4.1. Alternatively, one can solve the optimization problem with respect to the unconstrained variational parameters θ. It is not possible to obtain the fixed-point equations by isolating the roots. Instead, one can run an unconstrained, nonlinear solver, such as a modified version of Newton's method (Nocedal & Wright, 2006) that maintains a quadratic approximation of the objective with a positive definite Hessian. It is easy to compute the gradient and Hessian to the mean field

1. Choose an initial value for μ.
2. Repeat until the convergence criterion is met:
   - Repeat for every $i \in V$:
     - Set $H_i = \exp\big\{ -2\big(\theta_i + \sum_{j \in N(i)} \theta_{ij}\mu_j\big)\big\}$.
     - Set $\mu_i = \frac{1-H_i}{1+H_i}$.

Figure 4.1: Coordinate ascent algorithm for optimizing the mean field approximation to the Ising spin glass model. The output vector μ contains the mean statistic. Above, $N(i)$ denotes the set of variable nodes adjacent to vertex i in the undirected graph.

variational objective.

4.2 Sequential Monte Carlo

A detailed overview of sequential Monte Carlo was presented in Sec. 3.3.6. Here, I give some additional background pertinent to the conditional mean field algorithm.

Mean field tends to be overconfident in its estimates. Loosely speaking, this means that if $p_{k-1}(x)$ were to be a mean field approximation, then it would likely have lighter tails than the target distribution $p_k(x)$. If we were to use the sub-optimal backward kernel (3.44), the importance weights would simplify to

$$\tilde{w}_k(x_{1:k}) = \frac{p_k(x_{k-1})}{p_{k-1}(x_{k-1})} \times \tilde{w}_{k-1}(x_{1:k-1}), \tag{4.10}$$

duly noting that $\tilde{w}_{k-1}(x_{1:k-1})$ will usually be replaced by unnormalized importance weights $\hat{w}_{k-1}(x_{1:k-1})$ following the description of the normalized importance sampling estimator in Chapter 3. Implicitly, this is the choice of backward kernel made in earlier sequential frameworks (Jarzynski, 1997; Neal, 2001). Since the mean field approximation $p_{k-1}(x)$ might very well fail to "dominate" the target $p_k(x)$, the expression (4.10) risks having unbounded variance. This is a problem because the weights may change abruptly from one iteration to the next, or give too much importance to too few values x (Robert & Casella, 2004). Instead, Del Moral et al. (2006) suggest approximating the optimal backward-in-time kernel by (3.41). The resulting importance weights on the joint space are (3.42). If the transition kernel increases the mass of the proposal in regions where $p_{k-1}(x)$ is weak relative to $p_k(x)$, the backward kernel (3.41) will rectify the problems caused by an overconfident proposal.

The drawback of the backward kernel (3.41) is that it limits the choice of transition kernel $K_k(x' \,|\, x)$, a crucial ingredient to a successful simulation. For instance, we can't use Metropolis-Hastings Because its transition kernel involves an integral that does not admit a closed form (Robert & Casella, 2004). One transition kernel which fits our requirements and is widely applicable is a mixture of kernels based on the random-scan Gibbs sampler (Robert & Casella, 2004). Denoting $\delta_y(x)$ to be the Dirac measure

at location y, the transition kernel with invariant distribution $p(x)$ is

$$K(x' \,|\, x) = \textstyle\sum_i \rho_i \, p(x'_i \,|\, x_{-i}) \, \delta_{x_{-i}}(x'_{-i}), \tag{4.11}$$

where $p(x_i \,|\, x_{-i})$ is the conditional density of x_i given values at all other sites, and ρ_i is the probability of shifting the samples according to the Gibbs kernel at site i. Following (3.42) and the conditional probability identity, the importance weights are

$$\begin{aligned}
\tilde{w}_k(x_{1:k}) &= \left\{ \sum_i \rho_i \int p_k(x'_i \,|\, x'_{-i}) \times \frac{p_{k-1}(x_i, x'_{-i})}{p_k(x'_i, x'_{-i})} \, dx_i \right\}^{-1} \times \tilde{w}_{k-1}(x_{1:k-1}) \\
&= \left\{ \sum_i \rho_i \times \frac{\int p_{k-1}(x_i, x'_{-i}) \, dx_i}{\int p_k(x_i, x'_{-i}) \, dx_i} \right\}^{-1} \times \tilde{w}_{k-1}(x_{1:k-1}) \\
&= \left\{ \sum_i \rho_i \times \frac{p_{k-1}(x'_i, x'_{-i})}{p_k(x'_i, x'_{-i})} \times \frac{p_k(x'_i \,|\, x'_{-i})}{p_{k-1}(x'_i \,|\, x'_{-i})} \right\}^{-1} \times \tilde{w}_{k-1}(x_{1:k-1}) \\
&= \frac{p_k(x')}{p_{k-1}(x')} \left\{ \sum_i \rho_i \frac{p_k(x'_i \,|\, x'_{-i})}{p_{k-1}(x'_i \,|\, x'_{-i})} \right\}^{-1} \times \tilde{w}_{k-1}(x_{k-1}),
\end{aligned} \tag{4.12}$$

in which x' is the component from current step k, and x is the component from the previous step. This formula is used as a departure point for subsequent derivations.

In practice, of course, we use the normalized importance sampling estimator. The importance weight update (4.12) remains the same as before, except that $p_{k-1}(x)$ is substituted for the unnormalized response $f_{k-1}(x)$, and $p_k(x)$ is substituted for $f_k(x)$.

4.3 Conditional mean field

I begin the presentation of the main contribution by deriving a new class of conditionally-specified variational approximations by extending the theory presented in the previous two sections. After that, I show how successively refined conditional mean field approximations are incorporated into an SMC algorithm.

The starting point is a partition R, or equivalence relation, of the set of vertices V. Elements of R, which I denote with the capital letters A and B, are disjoint subsets of V.[2] The strategy is to come up with a good naive mean field approximation to the target conditional density $p(x_A \,|\, x_{-A}; \theta^\star)$ for every equivalence class $A \in R$, and then again for every configuration x_{-A}. That is, I adapt naive mean field theory to conditional densities, forcing each approximating conditional $p(x_A \,|\, x_{-A}; \theta)$ to decompose as a product of marginals $p(x_i \,|\, x_{-A}; \theta)$. Here, I've denoted x_A to be the configuration x restricted to set $A \subseteq V$, and x_{-A} to be the restriction of x to $V \setminus A$.

The crux of the matter is that for any point θ, the functions $p(x_A \,|\, x_{-A}; \theta)$ *only* represent valid conditional densities if they correspond to some unique joint. Condi-

[2]The assumption is that every subset $A \in R$ is non-empty and connected. By connected, I mean that for every pair of indices $i, j \in A$, there exists a path from i to j in the graph G that visits only nodes in A.

tions for guaranteeing the consistency of conditional densities have been extensively studied in multivariate (Arnold et al., 1999; Arnold et al., 2001) and spatial statistics (Besag, 1974). It so happens that under the Ising spin glass, the terms $p(x_A \,|\, x_{-A}; \theta)$ represent valid conditionals for *any* parameterization θ. What we have is a slight generalization of the auto-logistic model (Besag, 1974), for which the joint is always known.[3] In the statistical mechanics literature, the Ising model is cast in terms of its joint through an equilibrium process (and the Hammersley-Clifford theorem). It can also be generated in an equally natural way via its conditional probabilities (Besag, 1974). Although conditionally-specified distributions, or *auto-models*, have arguably greater intuitive appeal (Arnold et al., 2001; Besag, 1974), they are also subject to highly unobvious constraints, hence remain relatively unpopular in Bayesian analysis.

Conditional mean field requires that each conditional decompose as a product of marginals. As a result, θ_{ij} must be zero for every edge $(i,j) \in \mathcal{E}(A)$, where $\mathcal{E}(A) \equiv \{(i,j) \,|\, i \in A, j \in A\}$ is the set of edges contained by the vertices in subset A. Notice that we have free variational parameters θ_{ij} defined on the edges (i,j) that straddle subsets of the partition. Formally, these are the edges that belong to $C_R \equiv \{(i,j) \,|\, \forall A \in R, (i,j) \notin \mathcal{E}(A)\}$. I call C_R the set of "connecting edges".

The variational formulation on partition R consists of competing objectives, since the conditionals $p(x_A \,|\, x_{-A}; \theta)$ share a common set of parameters. Commonly, one frames multiple competing objectives as a nonlinear least-squares problem (Nocedal & Wright, 2006). Least-squares, however, does not apply to our case because the quantities of interest, the log-normalizing factor of each conditional, are clearly not observed. I formulate the final objective function as a linear combination of *conditional objectives*. A conditional mean field optimization problem with respect to graph partition R and linear weights λ is of the form

$$\begin{array}{ll} \text{maximize} & \sum_{A \in R} \sum_{x_{N(A)}} \lambda_A(x_{N(A)}) F_A(\theta, x_{N(A)}) \\ \text{subject to} & \theta_{ij} = 0, (i,j) \in \mathcal{E} \setminus C_R. \end{array} \tag{4.13}$$

I extend the notion of neighbours to sets, so that $N(A)$ is the *Markov blanket* of A (Murphy, 2002). The non-negative scalars $\lambda_A(x_{N(A)})$ are defined for every equivalence class $A \in R$ and configuration $x_{N(A)}$. Each conditional objective $F_A(\theta, x_{N(A)})$ represents a naive mean field lower bound to the log-normalizing factor of the conditional density $p(x_A \,|\, x_{-A}; \theta) = p(x_A \,|\, x_{N(A)}; \theta)$. For the Ising model, $F_A(\theta, x_{N(A)})$ follows from the exact same steps used in the derivation of the naive mean field lower bound, except that *we replace the joint by a conditional*. We obtain the expression

$$\begin{aligned} F_A(\theta, x_{N(A)}) = {} & \textstyle\sum_{i \in A} \theta_i \mu_i(\theta, x_{N(A)}) + \sum_{(i,j) \in \mathcal{E}(A)} \theta_{ij} \mu_{ij}(\theta, x_{N(A)}) \\ & + \textstyle\sum_{i \in A} \sum_{j \,\in\, (N(i) \cap N(A))} \theta_{ij} x_j \mu_i(\theta, x_{N(A)}) + H_A(\theta, x_{N(A)}), \end{aligned} \tag{4.14}$$

[3] As remarked by Besag (2001): "although this is derived classically from thermodynamic principles, it is remarkable that the Ising model follows necessarily as the very simplest non-trivial binary Markov random field."

with the *conditional mean statistics* for $i \in A, j \in A$ given by

$$\begin{aligned}\mu_i(\theta, x_{N(A)}) &\equiv \textstyle\int x_i \, p(x_A \,|\, x_{N(A)}; \theta) \, dx \\ &= \tanh\left(\alpha_i + \textstyle\sum_{j \in (N(i) \cap N(A))} \alpha_{ij} x_j\right) \qquad (4.15) \\ \mu_{ij}(\theta, x_{N(A)}) &\equiv \textstyle\int x_i \, x_j \, p(x_A \,|\, x_{N(A)}; \theta) \, dx \\ &= \mu_i(\theta, x_{N(A)}) \, \mu_j(\theta, x_{N(A)}). \qquad (4.16)\end{aligned}$$

The entropy is identical to (4.9), except that the mean statistics are replaced by their conditional counterparts:

$$\begin{aligned}H_A(\theta, x_{N(A)}) = &- \sum_{i \in A} \left(\tfrac{1-\mu_i(\theta, x_{N(A)})}{2}\right) \log\left(\tfrac{1-\mu_i(\theta, x_{N(A)})}{2}\right) \\ &- \sum_{i \in A} \left(\tfrac{1+\mu_i(\theta, x_{N(A)})}{2}\right) \log\left(\tfrac{1+\mu_i(\theta, x_{N(A)})}{2}\right). \qquad (4.17)\end{aligned}$$

Notice the appearance of the new terms in (4.14). These terms account for the interaction between the random variables on the border of the partition. Also note that $N(i) \cap N(A)$ is the set of neighbours of node i that are outside the subset A. The mean statistics can no longer be optimized following the standard approach because the $\mu_i(\theta, x_{N(A)})$ cannot be treated as independent variables for all $x_{N(A)}$ since the solution would no longer define an Ising model (or even a valid probability density). Instead, I optimize the objective with respect to the parameters θ. Denoting the objective function in (4.13) by $F_{R,\lambda}$, I derive expressions for the derivatives $\nabla F_{R,\lambda}(\theta)$ and $\nabla^2 F_{R,\lambda}(\theta)$, and run an unconstrained, nonlinear solver. Note that a conditional mean field approximation is neither an upper nor a lower bound on $c(\theta^\star)$.

From (4.14), the gradient vector entries $\nabla F_{R,\lambda}(\theta)$ associated with sites $i \in V$ are

$$\begin{aligned}\frac{\partial F_{R,\lambda}}{\partial \theta_i} &= \sum_{x_{N(A)}} \lambda_A(x_{N(A)}) \frac{\partial F_A(x_{N(A)})}{\partial \mu_i(x_{N(A)})} \times \frac{d\mu_i(x_{N(A)})}{d\tilde{\theta}_i(x_{N(A)})} \times \frac{\partial \tilde{\theta}_i(x_{N(A)})}{\partial \theta_i} \\ &= \sum_{x_{N(A)}} \lambda_A(x_{N(A)}) \operatorname{sech}^2\big(\tilde{\theta}_i(x_{N(A)})\big) \\ &\quad \times \big(\tilde{\theta}_i^\star(x_{N(A)}) - \tilde{\theta}_i(x_{N(A)}) + \textstyle\sum_{j \in (N(i) \cap A)} \theta_{ij}^\star \mu_j(x_{N(A)})\big), \qquad (4.18)\end{aligned}$$

where I define

$$\tilde{\theta}_i(x_{N(A)}) \equiv \theta_i + \textstyle\sum_{j \in (N(i) \cap N(A))} \theta_{ij} x_j.$$

From this definition, it is easy to see that $\mu_i(x_{N(A)})$ in (4.15) reduces to $\tanh(\theta_i)$. For ease of presentation, I omit the functions' explicit dependence on θ. The gradient entries corresponding to pairs $(i, j) \in C_R$ are

$$\frac{\partial F_{R,\lambda}}{\partial \theta_{ij}} = \sum_{x_{N(A)}} \lambda_A(x_{N(A)}) \frac{\partial F_A(x_{N(A)})}{\partial \theta_{ij}} + \sum_{x_{N(B)}} \lambda_B(x_{N(B)}) \frac{\partial F_B(x_{N(B)})}{\partial \theta_{ij}}, \qquad (4.19)$$

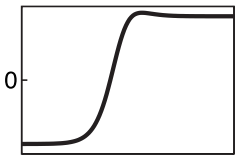

Figure 4.2: The conditional objective on a subset consisting of a single node.

where $i \in A$ and $j \in B$; that is, edge (i, j) straddles two subsets A and B of the partition R. Since (i, j) is a connecting edge, it is necessarily the case than $A \neq B$, $i \in N(B)$ and $j \in N(A)$. The partial derivative on the left-hand side of (4.19) is

$$\begin{aligned}\frac{\partial F_A(x_{N(A)})}{\partial \theta_{ij}} &= \frac{\partial F_A(x_{N(A)})}{\partial \mu_i(x_{N(A)})} \times \frac{d\mu_i(x_{N(A)})}{d\tilde{\theta}_i(x_{N(A)})} \times \frac{\partial \tilde{\theta}_i(x_{N(A)})}{\partial \theta_{ij}} \\ &= x_j \operatorname{sech}^2\big(\tilde{\alpha}_i(x_{N(A)})\big) \\ &\qquad \times \big(\tilde{\theta}^\star_i(x_{N(A)}) - \tilde{\theta}_i(x_{N(A)}) + \textstyle\sum_{k \in (N(i) \cap A)} \theta^\star_{ik} \mu_k(x_{N(A)})\big). \end{aligned} \tag{4.20}$$

The right-hand partial derivative in (4.19) follows in a similar manner.

Unlike standard naive mean field, conditional mean field also optimizes over the pairwise interactions θ_{ij} defined on the connecting edges $(i, j) \in C_R$. In certain settings, however, this will be not be feasible. Consider the case when when $|R| = n$, so that each subset $A \in R$ consists of a single node. Each iterate of gradient descent requires the computation of $|V| + |\mathcal{E}|$ derivatives. When the model is dense or completely connected, each iterate of gradient descent will require at least $O(n^2)$ operations (and this is to say nothing of the matrix of second derivatives). In such case, it will be more reasonable to fix the edge parameters to $\theta_{ij} = \theta^\star_{ij}$. This choice brings two additional benefits. First, the objective is separable on the subsets of the partition, which bestows computational advantages on our sequential Monte Carlo algorithm (as we shall see shortly). Second, the conditional objective of a singleton subset has a unique maximum at the point $\theta_i = \theta^\star_i$, so any solution to (4.13) is guaranteed to recover the original distribution when $|R| = n$. See Fig. 4.2 for an illustration of this fact. What we sacrifice by fixing the edge parameters is that the lower bounds on the conditionals may not be as tight as they could be.

I have yet to address the question: how to select the scalars λ? A large $\lambda_A(x_{N(A)})$ bestows a greater reward for finding a good approximation to $p(x_A \,|\, x_{N(A)}; \theta^\star)$, so it stands to reason that we should place greater emphasis on those conditionals that are realized more often, and set $\lambda_A(x_{N(A)}) \propto p(x_{N(A)}; \theta^\star)$. Of course, these probabilities aren't available. Equally problematic is the fact that (4.13) may involve nearly as many terms as there are possible worlds. A greedy choice resolves both issues. Supposing that we are at some intermediate stage in the SMC algorithm (which will be described shortly), a greedy but not unreasonable choice is to set $\lambda_A(x_{N(A)})$ to be the

current Monte Carlo estimate of the marginal $p(x_{N(A)}; \theta^\star)$:

$$\lambda_A(x_{N(A)}) = \sum_{s=1}^{n_s} w^{(s)} \delta_{x_{N(A)}^{(s)}}(x_{N(A)}). \tag{4.21}$$

Happily, the number of terms in (4.13) is now on the order of the number of samples.

Next, I describe an SMC algorithm that produces progressively refined particle estimates of the mean statistics. The initial auxiliary distribution is obtained by solving (4.13) for $R = \{V\}$, which amounts to the naive mean field approximation. Subsequent steps consists of iteratively solving (4.13) using the scalar weights (4.21), updating the estimates of the mean statistics by reweighting (see equation 4.23) and occasionally resampling the particles, then splitting the partition until it cannot be split it anymore, at which point $|R| = n$ and the target $p(x; \theta^\star)$ is recovered. Like the stochastic approximation algorithm from the previous chapter, conditional mean field is an SMC algorithm in which the next distribution in the sequence is constructed dynamically according to the particle approximation from the previous step.

It is easy to draw samples from the initial fully-factorized distribution, as the single-site marginals are given by (4.8). It is also easy to compute $c(\theta)$, as

$$\begin{aligned} c(\theta) &= \log\left(\textstyle\sum_x \exp\left(\sum_{i \in V} \theta_i x_i\right)\right) \\ &= \textstyle\sum_{i \in V} \log\left(\sum_{x_i} \exp(\theta_i x_i)\right) \\ &= \textstyle\sum_{i \in V} \log\left(2\cosh(\theta_i)\right). \end{aligned} \tag{4.22}$$

Note that (4.22) is not a variational lower bound.

Now, suppose we are at some intermediate step in the algorithm. We currently have a particle estimate of the R-partition conditional mean field approximation $p(x; \theta)$ with samples $x^{(s)}$ and marginal importance weights $w^{(s)}$. To construct the next artificial distribution $p(x; \theta^{(\text{new})})$ in the sequence, we choose a finer partitioning of the graph, $R^{(\text{new})}$, set the weights $\lambda^{(\text{new})}$ according to (4.21), and use an unconstrained nonlinear solver to find a local minimum $\theta^{(\text{new})}$ to (4.13). The solver is initialized to $\theta_i^{(\text{new})} = \theta_i^\star$. We require that the new graph partition satisfy that for every $B \in R^{(\text{new})}$, $B \subseteq A$ for some $A \in R$. In this manner, we ensure that the sequence is progressing toward the target (provided $R \neq R^{(\text{new})}$), and that it is always possible to evaluate the importance weights. It is not understood how to tractably choose a good sequence of partitions, so I select them in an arbitrary manner. Next, I execute the random-scan Gibbs sampler (4.11) to shift the particles toward the new distribution, where the Gibbs sites correspond to the subsets $B \in R^{(\text{new})}$. I set the mixture probabilities of the Markov transition kernel to $\rho_B = |B|/n$. Following (3.43), the

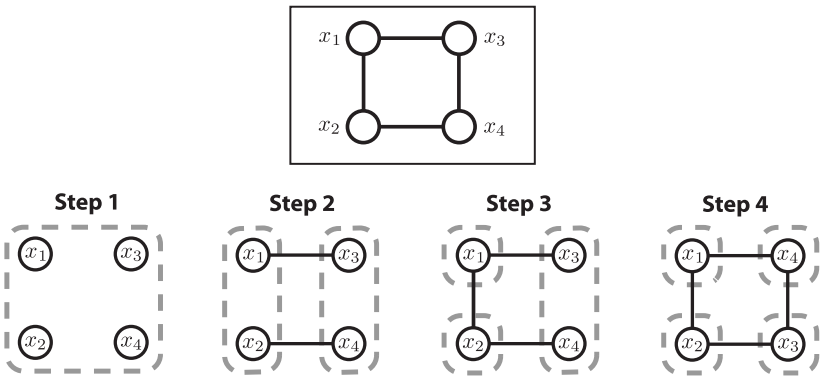

Figure 4.3: *(Top)* The undirected graphical model of the small Ising model. *(Bottom)* Steps 1 through 4 depict the Markov properties of the successive conditional mean field approximations. The last step recovers the Markov properties of the target. The dashed lines represent the subsets of the graph partition.

expression for the unnormalized importance weights is

$$\hat{w}_k(x_{1:k}) = \frac{\exp\left(\sum_i \theta_i^{(\text{new})} x_i' + \sum_{(i,j)} \theta_{ij}^{(\text{new})} x_i' x_j'\right)}{\exp\left(\sum_i \theta_i x_i' + \sum_{(i,j)} \theta_{ij} x_i' x_j'\right)} \times \left\{ \sum_{B \in R'} \rho_B \prod_{i \in B} \frac{p(x_i' \,|\, x'_{N(B)}; \theta^{(\text{new})})}{p(x_i' \,|\, x'_{N(A)}; \theta)} \right\}^{-1} \times \tilde{w}_{k-1}(x_{1:k-1}), \tag{4.23}$$

where the single-site conditionals are given by

$$p(x_i \,|\, x_{N(A)}; \theta) = (1 + x_i \mu_i(\theta, x_{N(A)}))/2, \tag{4.24}$$

and $A \in R$ is the unique subset containing $B \in R^{(\text{new})}$. The SMC estimate of $c(\theta)$ is updated according to (3.46). Let's look at a small example to see how this works.

Example. Consider a very small Ising model with 4 vertices. It has conditional independence properties depicted by the undirected graphical model at the top of Fig. 4.3. The target parameters are $\theta^\star_{1:4} = \frac{1}{10}(4, 3, -5, -2)$, $\theta^\star_{13} = \theta^\star_{24} = \theta^\star_{34} = +\frac{1}{2}$ and $\theta^\star_{12} = -\frac{1}{2}$. I assume there are enough particles to recover the distributions almost perfectly. In the first step, the graph partition is set to $R = \{V\} = \{\{1, 2, 3, 4\}\}$. The first artificial distribution is the naive mean field solution in which all the edges are removed from the undirected graphical model (see the bottom of Fig. 4.3). The solution is $\theta_{1:4} = (0.09, 0.03, -0.68, -0.48)$ with $c(\theta) = 3.10$. Knowing that the true mean statistics are $\mu_{1:4} = (0.11, 0.07, -0.40, -0.27)$, and $\text{Var}(X_i) = 1 - \mu_i^2$, it is easy to see naive mean field largely underestimates the variance of the spins. In the second step, I split the partition into $R = \{\{1, 2\}, \{3, 4\}\}$, and all the edges from the original graph that span subsets of the partition are recovered, as illustrated in Fig. 4.3. The solution of new conditional mean field objective is $\theta_{1:4} = (0.39, 0.27, -0.66, -0.43)$, with potentials $\theta_{13} = \theta^\star_{13}$, $\theta_{24} = \theta^\star_{24}$ on the connecting edges C_R. After shifting

1. Set $k = 0$.
2. Compute solution θ_0 to the naive mean field objective.
3. Draw samples $x^{(s)}$ from the naive mean field approximation with parameters θ_0.
4. Set the importance weights $w^{(s)}$ to $1/n_s$.
5. Set the graph partition to be $R_0 = \{V\}$.
6. Repeat until $|R_k| = |V|$:
   - Set $k \leftarrow k + 1$.
   - Split the partition R_{k-1} and save the result as R_k.
   - Obtain a new variational approximation θ_k by optimizing the conditional mean field objective with graph partition $R^\star$ and scalars $\lambda_A(x_{N(A)}) = \sum_s \delta_{x^{(s)}}(x_{-A})$.
   - Shift each sample $x^{(s)}$ toward the new variational approximation according to the random-scan Gibbs sampler with sites corresponding to subsets of the partition R_k.
   - Compute the new unnormalized importance weights $\hat{w}^{(s)}$ following (4.23).
   - Normalize the importance weights; $w^{(s)} = \hat{w}^{(s)} / \sum_s \hat{w}^{(s)}$.
   - Optionally, resample the particles $\{x^{(s)}, w^{(s)}\}$.

Figure 4.4: Conditional mean field algorithm for the Ising spin glass.

the particles toward this distribution with the random-scan Gibbs sampler, I update the estimate of the log-normalizing factor, obtaining 3.26. Step 3 then splits subset $\{1, 2\}$, and the solution is $\theta = (0.40, 0.30, -0.64, -0.42)$ by setting λ according to the weighted samples from step 2. Notice that $\theta_1 = \theta_1^\star$, $\theta_2 = \theta_2^\star$. The new estimate of the log-normalizing factor is 3.37. Step 4 recovers the original distribution, at which point the estimate of the log-normalizing factor is nearly the exact solution, $c(\theta^\star) = 3.37$.

The procedure shown in Fig. 4.4 encapsulates all the steps of the conditional mean field algorithm for the Ising model. The algorithm takes three inputs: the undirected graph $G = (V, \mathcal{E})$, the target parameterization $\theta^\star$, and the number of samples n_s. The outputs are the collection of samples $x^{(s)}$ with normalized weights $w^{(s)}$, and the estimate of the log-normalizing constant. Lines 1-5 constitute the initial phase of the algorithm, in which samples are drawn from the naive mean field approximation. After that, the algorithm repeatedly refines the variational distribution and updates the particle approximation until all the dependencies are recovered. The pseudocode in Fig. 4.4 may be modified to allow for smoother transitions from one distribution in the sequence to the next; see the discussion below.

In order to get a better understanding of the computational aspects of the conditional mean field algorithm, consider the case in which the graph is fully-connected,

we do not optimize over the edge potentials, and all the subsets of the graph partition are split in half at every step. Further suppose that the complexity of optimizing the mean field objective is $O(n^2)$, which might be the case when employing a quasi-Newton solver with a dense, quadratic approximation. In the second step, I partition the graph G in half and run the solver separately on the conditional mean field objective for each subset of the partition; the complexity is $O((\frac{n}{2})^2)$. Continuing in this manner, the accumulated computation works out to

$$\begin{aligned} &O(n^2) + 2O\big(\left(\tfrac{n}{2}\right)^2\big) + 4O\big(\left(\tfrac{n}{4}\right)^2\big) + 8O\big(\left(\tfrac{n}{8}\right)^2\big) + \dots \\ &\quad = O(n^2) + \tfrac{1}{2}O(n^2) + \tfrac{1}{16}O(n^2) + \tfrac{1}{64}O(n^2) + \dots \\ &\quad = O(n^2). \end{aligned}$$

So the asymptotic computational complexity is no greater than the cost of the original mean field approximation.

Let's examine a second scenario, in which the graph is sparse (the number of undirected edges is on the order of the number of vertices, so that each vertex has $O(1)$ neighbours). I optimize over the edge potentials, and as before I use a quasi-Newton solver which respects the sparsity of the Hessian. At every step, the cost of computing and maintaining the Hessian is $O(n)$. If the partition is split at each step in the same manner as before, then this requires $O(\log n)$ iterations of SMC, so the total cost is $O(n \log n)$.

The random-scan Gibbs sampler can mix poorly, especially on a fine graph partition. This can result in degradation of the importance weights if the support of $p(x; \theta_{k-1})$ is significantly different from the support of $p(x; \theta_k)$. Gradually changing the parameters with tempered artificial distributions (Del Moral et al., 2006) $p(x; (1-\gamma)\theta_{k-1} + \gamma\theta_k)$, with an increasing sequence of scalars $\gamma \in [0, 1]$, gives the transition kernel more opportunity to correctly migrate the samples to the next distribution. In the final steps of the sequence, however, it is unlikely that tempered transitions will have much effect on the Gibbs sampler's poor mixing properties. This consideration raises an important design question: is it better to spend effort mitigating degradation of the importance weights, or is that effort better spent finding the tightest possible variational approximation at each step of sequential Monte Carlo through optimization of the edge potentials?

4.4 Experiments

I conduct experiments on two spin glasses, one defined on a 12×12 grid, and the other on a fully-connected graph with 26 nodes. The model sizes approach the limit of what we can compute exactly for the purposes of evaluation.

Method. The magnetic fields are generated by drawing each θ_i uniformly from $[-1, 1]$ and drawing θ_{ij} uniformly from $\{-\frac{1}{2}, +\frac{1}{2}\}$. Both models exhibit strong and conflicting pairwise interactions, so it is expected that rudimentary MCMC methods such as Gibbs sampling will get "stuck" in local modes (Hamze & de Freitas, 2006).

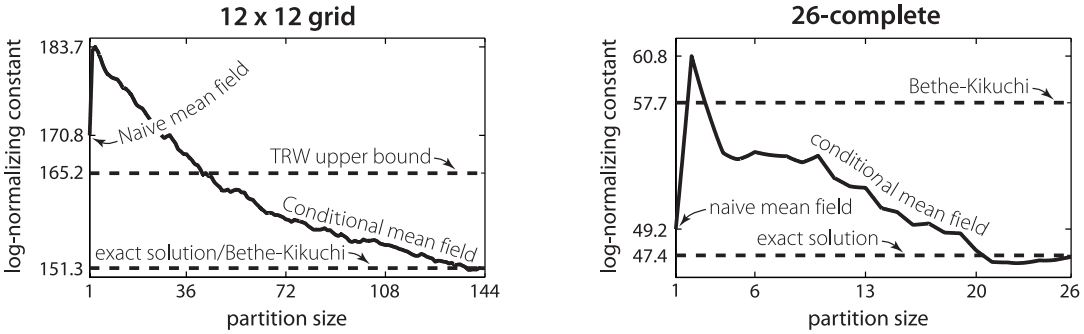

Figure 4.5: *(left)* Estimate of the 12×12 grid log-partition function for each iteration of SMC. *(right)* Same, for the fully-connected graph with 26 nodes. I omitted the tree-reweighted upper bound because it is way off the map. Note that these plots will vary slightly for each simulation.

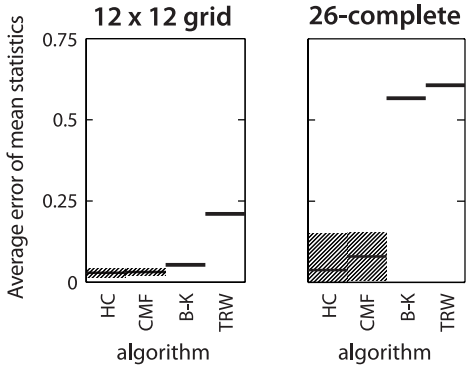

Figure 4.6: Average error of the mean statistics according to the hot coupling (HC), conditional mean field algorithm (CMF), Bethe-Kikuchi variational approximation (B-K), and tree-reweighted upper bound (TRW) estimates. The maximum possible average error is 2. For the HC and CMF algorithms, 95% of the estimates fall within the shaded regions according to a sample of 10 simulations.

The algorithm settings are as follows. I use 1000 particles (as with most particle methods, the running time is proportional to the number of particles), and I temper across successive distributions with a linear inverse temperature schedule of length 100. The particles are resampled when the effective sample size drops below 500. I compare the conditional mean field algorithm to the "hot coupling" SMC algorithm described in Hamze and de Freitas (2006) (using the same algorithm settings), and with two sum-product methods based on Bethe-Kikuchi approximations (Aji & McEliece, 2001) and tree-reweighted upper bounds (Wainwright et al., 2005). I adopt the simplest formulation of both methods in which the regions (or junction graph nodes) are defined as the edges $\mathcal{E}$. Since loopy belief propagation failed to converge for the complete graph, I implemented the convergent double-loop algorithm of Heskes et al. (2003).

Results. The results of the experiments are summarized in Figures 4.5 and 4.6. The plots on the left and right of Fig. 4.5 show that the estimate of the log-partition function, for the most part, moves to the exact solution as the graph is partitioned into smaller and smaller pieces. Both Bethe-Kikuchi approximations and tree-reweighted upper bounds provide good approximations to the grid model. Indeed, the former recovers the log-partition function almost perfectly. However, these approximations break down as soon as they encounter a dense, frustrated model. This is consistent with the results observed in other experiments (Hamze & de Freitas, 2006; Wainwright et al., 2005). The SMC algorithms proposed here and in (Hamze & de Freitas, 2006), by contrast, produce significantly improved estimates of the mean statistics. It is surprising that conditional mean field achieves similar performance with hot coupling (Hamze & de Freitas, 2006), given that it does not exploit the tractability of sum-product messages in the Ising model, which would offer guaranteed improvements due to the Rao-Blackwell theorem.

4.5 Conclusions and discussion

I presented an SMC algorithm in which each artificial distribution is the solution to a conditionally-specified mean field optimization problem. While all the experiments were conducted on probabilistic graphical models with binary random variables and pairwise potentials or factors, the ideas in this chapter can be naturally extended to other discrete and continuous random variables and to different types of factors.

Perhaps the most significant contribution of this chapter is the idea that one can apply variational methodology to the *conditionals* of a target distribution instead of the joint; that is, instead of finding some artificial distribution that is as close as possible to the target joint, the objective is to find some artificial distribution that matches the conditional densities of the target as well as possible. While not useful on its own, the idea of optimizing the conditionals of a variational density can be used to come up with a sensible sequence of distributions for SMC.

My claim was that the extra expense of optimization at each step is warranted in the long run because the conditional mean field method holds promise in solving more difficult inference problems, problems where Monte Carlo and variational methods alone perform poorly. I hypothesize that our approach is superior methods that "prune" constraints on factors, but further investigation is needed to verify this theory. One potentially serious misgiving is that mean field approximations, as I discussed earlier, may serve poorly as proposal distributions as they can be very peaked. Conditional mean field may not work well on inference problems for which the initial mean field approximation is very poor. Conceivably, the quality of the conditional mean field approximations could be improved in these cases by adopting more sophisticated tractable families such as structured mean field (Saul & Jordan, 1996) or a mixture of mean field approximations (Bishop et al., 1998; Zhang & Fossorier, 2006).

Chapter 5

Learning a contingently acyclic probabilistic relational model

The previous two chapters showed how we can devise new algorithms that exploit the strengths of variational and Monte Carlo methods, leading to better approximations to difficult inference problems. The final contribution of my thesis follows a different but related theme: it is a study in comparing different probabilistic representations, and what are the advantages of these representations in terms of inference and in terms of modeling power. The main contribution of this chapter is not a new algorithm or a new framework for probabilistic inference. Rather, the two major contributions are: one, a novel directed probabilistic graphical model of a social network that incorporates special latent variables to guarantee acyclicity; two, investigation of the inference and learning challenges entailed by a new probabilistic model—I show how existing ideas from variational inference can be used to conduct inference and learning in this *contingently acyclic* model. Perhaps the most intruiging result contained in this chapter is that the model has comparable performance to conventional undirected graphical models, yet also retains the advantages of directed representations.

It has long been known in the sciences that social context matters. Epidemiologists, for instance, have studied the spread of infectious diseases like HIVs with social networks, while sociologists have studied how risk-taking behaviours are learned in social peer groups (Pearson & Michell, 2000). Broadly speaking, social network analysis is concerned with the nature of relationships, and how the structure of relationships influences other processes. The point of departure for statistical representations of social network structure is the class of models proposed by Frank and Strauss (1986) and Besag (1974), now known as *exponential random graph models* or $p^\star$ (Carrington et al., 2005). This modeling approach relies on the (famously unpublished) theorem of Hammersley and Clifford to provide the necessary link between a preliminary dependency analysis and the final probabilistic model. The argument for this approach is that social relations are intrinsically interdependent with no obvious form of causation, so the aim is to develop models that hypothesize possible forms of interdependence, or *autocorrelation.*

This class of models, also known to many researchers as *undirected probabilistic graphical models* or *Markov random fields*, has witnessed a resurgence of popularity in other well-explored domains, notably computer vision and collaborative filtering. The key, again, is that such formalisms naturally represent interdependence, such as constancy of motion in neighbouring image pixels (Sun et al., 2008), hydrogen bond interactions in secondary protein structure (Muñoz & Eaton, 1999) or similar tastes in movies (Salakhutdinov et al., 2007), and they can incorporate simple factors to form rich, predictive models without having to worry about avoiding cycles in the underlying graph.

The undirected formalism is not without its problems, however. First, the difficulties of learning the model parameters—for instance, by maximizing the likelihood of the model given the data—are well-noted (Hunter et al., 2008). Another possible approach is to compute the maximum likelihood estimator via stochastic approximation (Younes, 1991), but this may involve repeated, computationally intensive simulations of a Markov chain. In some cases, the contrastive divergence approximation provides a more realistic alternative (Hinton, 2002). Due to these difficulties, the *pseudo-likelihood* approximation still appears in the literature, despite severe criticism of its use (Snijders, 2002). Furthermore, undirected graphical models scale poorly to large social networks, including models that boast compact first-order representations, because inferring the result of a query always implicates every node in the network—even parts of the network for which we have no information. Because of this, it is argued, undirected models can be poorly suited for prediction in the presence of missing data (Marlin, 2008).

The main contribution of this chapter is to show through experiment that a directed probabilistic model (Spiegelhalter et al., 1993) is an equally viable representation of "interdependent" relations in a non-trivial social network domain, in addition to having several important advantages, as we discuss below. We formulate a directed model that explains how people alter their smoking habits within their social network (Sec 5.1), and in a series of experiments (Sec. 5.3) we compare it to an undirected model—to be precise, a Markov logic network (Richardson & Domingos, 2006).[1] We introduce special latent random variables (related to the hypothesis variables in *multinets*; see Geiger & Heckerman, 1996) to ensure that the directed graph is *contingently acyclic* (Poole, 2000), a notion which is grounded on *context-specific independence* (Boutilier et al., 1996). (These variables also have an interesting interpretation in the social network domain, as we explain in Sec. 5.1.) The introduction of cycle-resolving latent variables allows us to surpass the representational limitations of directed graphical models caused by the need to avoid cycles.

Three main advantages of our directed representation over existing undirected social network models are that it is easier to learn (for instance, under complete information the maximum likelihood solution is easily obtained), the probabilities have a local interpretation as conditionals, and irrelevant nodes can be pruned from the directed graph (Shachter, 1998). For example, whenever we have no information about two individuals, we can prune the friendship relation between them. This is an important step for extending statistical models to large or infinite-sized domains. An alternative way to capture interdependencies in a directed model is to permit cycles, as in Relational Dependency Networks (Neville & Jensen, 2007), although this approach still shares the learning difficulties of undirected models. We use the Independent Choice Logic (Poole, 1997) to define our model, though it could also be written as a program in BLOG (Milch et al., 2005), for instance.

Since the cycle-resolving variables are not observed, we use the expectation maxi-

[1] We do not compare to discriminative undirected models because they explain entity attributes given social links (Taskar et al., 2002) and link existence given entity attributes (Taskar et al., 2004), but not both simultaneously.

mization (EM) algorithm to learn the parameters of the social network model (Sec. 5.2). When all friendship and smoking relations are observed, the corresponding factor graph is highly interconnected so we must approximate inference in the E-step. (To be clear, a directed graph without cycles can still correspond to an undirected graph or factor graph with cycles.) One possible strategy would be to simulate a Markov chain in the E-step and replace the intractable expectations with Monte Carlo estimates; *i.e.* a stochastic EM algorithm (Delyon et al., 1999). In fact, the interior-point stochastic approximation method developed in Chapter 2 could then be used to optimize the parameters of the model. (Since the model parameters all represent probabilities, they are constrained to lie within the probability simplex.) But given the dense structure of the graphical model, it is likely that it would be extremely challenging to obtain unbiased estimates via Monte Carlo estimulation, particularly in large networks. We study an alternative strategy that is is not necessarily better but certainly less computationally intensive. Due to the particular structure of our network, a variational approximation based on the Bethe decomposition of the free energy is well-suited for approximating the E-step (Heskes, 2006; Yedidia et al., 2005). An EM algorithm based on this variational approximation is formally presented in Heskes et al. (2004).

Finally, our study leads to unsolved problems that would be of interest to people researching new and better tractable inference and learning algorithms. We elaborate on these issues in Sec. 5.4.

5.1 Description of the model

We describe an idealized relational probabilistic model of the relationship between smoking habits and the formation of friendships ("link existence"), a prototypical example of a relational domain where individuals influence each other. Our intent is to investigate the modeling and inference challenges that arise from studying a social network domain, not to construct a scientifically plausible analysis of smoking and risk-taking behaviour.

The conditional independence structure of our directed graphical model cannot be captured as a belief network (Pearl, 1988) because we don't know beforehand what is the set of parents of a random variable. In this capacity, our model fits within the definition of a *contingent Bayesian network* (Milch, 2006). Also, our representation is at a first-order level; we reason about relationships regarding collections of individuals. Since the independence relationships are only known when conditioned on certain random variables, and since the Independent Choice Logic of Poole (1997) naturally and compactly captures contingent (or context-specific) independencies at a first-order level, we define our model in ICL.

5.1.1 Preliminaries

Preliminaries. We follow the Prolog convention and write logical variables in upper case, and predicate symbols in lower case. Throughout, X and Y refer to individuals; that is, they are logical variables whose domain is the set of people. The predicate

smokes(X) = true if and only if X smokes, and friends(X, Y) = true if and only if X and Y are friends. Given an assignment of individuals to the logical variables, the predicates correspond to Boolean random variables. We define friendship as a symmetric, irreflexive relation, and enforce this constraint via some arbitrary total ordering $X \prec Y$ on the individuals.

In the social network, interdependencies arise between friendship and smoking. For example, a non-smoker might convince a friend to quit smoking, or the similar lifestyle choices of two smokers might make them more likely to become friends. A causal, temporal model might form an accurate description of these interdependencies, but it would be unwise to attempt to infer the history of events leading up to the present state.

In our directed probabilistic model, we regulate the direction of influence through a hidden predicate ind(X), and learn a distribution over it. For each individual X, ind(X) tells us, loosely speaking, whether X's decision to smoke is based on social factors, or whether it is governed by other factors that are not captured by our model (*i.e.* X makes an *independent* decision to smoke). When ind(X) = true, X can persuade others to smoke (or not to smoke), but X cannot be persuaded. This is a coarse-grained depiction of influence, and there are many alternatives for analyzing interdependencies at a propositional level; for instance, Alice could influence Bob only if Bob does not influence Alice, either directly or indirectly through other people. However, it is inordinately difficult to learn propositional rules such as this one, and they may not be useful in new situations. Our first-order rules are simple, easily transferable and, as we show, work reasonably well.

We now proceed to define our ICL theory for the social network domain. An ICL theory consists of two parts: a deterministic controller specified as a logic program, and noisy inputs that comprise the *choice space*. (Virtually all probabilistic programming languages could be described as a combination of deterministic controller and noisy inputs.) The logic program consists of a set of clauses, and each clause is either an *atom*—for our purposes, an atom is of the form $r(t_1, t_2, \ldots)$ where r is a predicate symbol and each t_i is either a logical variable or a constant—or a rule of the form $h \leftarrow a_1 \wedge \ldots \wedge a_k$, where h is an atom and each a_i is either an atom or its negation. ICL requires that the logic program be *contingently acyclic* (Poole, 2000).

The noisy inputs are called *atomic choices* in ICL, denoted as ground instances of $\phi_k(X)$ or $\phi_k(X, Y)$ in the clauses below. Each $\phi_k(X)$ or $\phi_k(X, Y)$ can appear in the body of a rule, but not the head of a clause. Of particular interest are the atomic choices $\phi_0(X)$, introduced above as ind(X). Our social network model has a very simple choice space, so we do not introduce ICL's general syntax for choice spaces.

5.1.2 Logic program

The rules for friendship are as follows. When ind(X) and ind(Y) are true, smokes(X) and smokes(Y) can legitimately be parents of friends(X, Y) without creating a cycle

in the directed graph, so we define clauses

$$\begin{aligned}
\text{friends}(X,Y) \leftarrow & X \prec Y \wedge \text{ind}(X) \wedge \text{ind}(Y) \\
& \wedge \neg\,\text{smokes}(X) \wedge \neg\,\text{smokes}(Y) \wedge \phi_1(X,Y) \\
\text{friends}(X,Y) \leftarrow & X \prec Y \wedge \text{ind}(X) \wedge \text{ind}(Y) \\
& \wedge \neg\,\text{smokes}(X) \wedge \text{smokes}(Y) \wedge \phi_2(X,Y) \\
\text{friends}(X,Y) \leftarrow & X \prec Y \wedge \text{ind}(X) \wedge \text{ind}(Y) \\
& \wedge \text{smokes}(X) \wedge \neg\,\text{smokes}(Y) \wedge \phi_2(X,Y) \\
\text{friends}(X,Y) \leftarrow & X \prec Y \wedge \text{ind}(X) \wedge \text{ind}(Y) \\
& \wedge \text{smokes}(X) \wedge \text{smokes}(Y) \wedge \phi_3(X,Y).
\end{aligned} \tag{5.1}$$

For those more familiar with Bayesian networks, it is instructive to see how the clauses above correspond to a conditional probability table (CPT). The clauses state that if both $\text{ind}(X)$ and $\text{ind}(Y)$ are true, then the corresponding entries of the CPT for $\text{friends}(X,Y)$ are

$$p(\text{friends}(X,Y) = T \,|\, \text{smokes}(X), \text{smokes}(Y), \text{ind}(X) = \text{true}, \text{ind}(Y) = \text{true})$$

$$= \begin{cases}
\text{smokes}(X) & \text{smokes}(X) & p \\
T & T & \theta_1 \\
F & T & \theta_2 \\
T & F & \theta_2 \\
F & F & \theta_3.
\end{cases} \tag{5.2}$$

(The binomial probabilities θ_k will be defined in the next part on semantics.) The remaining cases for friendship are covered by the clauses

$$\begin{aligned}
\text{friends}(X,Y) \leftarrow & X \prec Y \wedge \text{ind}(X) \wedge \neg\,\text{ind}(Y) \wedge\ \neg\,\text{smokes}(X) \wedge \phi_4(X) \\
\text{friends}(X,Y) \leftarrow & X \prec Y \wedge \text{ind}(X) \wedge \neg\,\text{ind}(Y) \wedge\ \text{smokes}(X) \wedge \phi_5(X),
\end{aligned} \tag{5.3}$$

and the analogous clauses with X and Y switched, and by the clauses

$$\text{friends}(X,Y) \leftarrow X \prec Y \wedge \neg\,\text{ind}(X) \wedge \neg\,\text{ind}(Y) \wedge \phi_6(X,Y). \tag{5.4}$$

when both X and Y can be influenced by others. The second set of clauses (5.3) says that whenever exactly one of the individuals is not influenced by others, then the corresponding entries of the CPT are given by

$$p(\text{friends}(X,Y) = T \,|\, \text{smokes}(X), \text{smokes}(Y), \text{ind}(X) = \text{true}, \text{ind}(Y) = \text{false})$$

$$= \begin{cases}
\text{smokes}(X) & p \\
T & \theta_4 \\
F & \theta_5.
\end{cases} \tag{5.5}$$

The clause (5.4) specifies CPT entries when $\text{ind}(X)$ and $\text{ind}(Y)$ are both false:

$$p(\text{friends}(X,Y)=T \mid \text{smokes}(X), \text{smokes}(Y), \\ \text{ind}(X)=\text{false}, \text{ind}(Y)=\text{false}) = \theta_6. \tag{5.6}$$

We also include a rule to enforce symmetry of friendship:

$$\text{friends}(X,Y) \leftarrow Y \prec X \wedge \text{friends}(Y,X). \tag{5.7}$$

The clauses (5.1), (5.3), (5.4) and (5.7) in combination with the choice space define conditional probability distributions (CPDs) for $\text{friends}(X,Y)$ given values for, $\text{smokes}(X)$, $\text{smokes}(Y)$, $\text{ind}(X)$ and $\text{ind}(Y)$. It is quite apparent that the specification of the CPT for $\text{friends}(X,Y)$ defined by (5.2), (5.5) and (5.6) provides a much more compact representation of friendship than a CPT over all 2^5 possible assignments to the random variables. Note that the conditional probability is deterministic conditioned on atomic choices $\phi_1(X,Y)$ through $\phi_5(X,Y)$.

Rules for smoking habits are as follows. The simplest case occurs when X's friends have no bearing on X's decision to smoke:

$$\text{smokes}(X) \leftarrow \text{ind}(X) \wedge \phi_7(X). \tag{5.8}$$

To determine whether X smokes when $\text{ind}(X) = \text{false}$, we aggregate "advice" from smokers and non-smokers through hidden predicates smoking-advice(X) and non-smoking-advice(X), or $\text{sa}(X)$ and $\text{nsa}(X)$ for short,

$$\text{sa}(X) \leftarrow \exists Y\ \text{friends}(X,Y) \wedge \text{ind}(Y) \wedge \text{smokes}(Y) \wedge \phi_8(X,Y) \tag{5.9}$$

$$\text{nsa}(X) \leftarrow \exists Y\ \text{friends}(X,Y) \wedge \text{ind}(Y) \wedge \neg\,\text{smokes}(Y) \wedge \phi_9(X,Y), \tag{5.10}$$

and then combine the advice through the clauses

$$\begin{array}{l} \text{smokes}(X) \leftarrow \neg\,\text{ind}(X) \wedge \neg\,\text{sa}(X) \wedge \neg\,\text{sa}(X) \wedge \phi_{10}(X) \\ \text{smokes}(X) \leftarrow \neg\,\text{ind}(X) \wedge \neg\,\text{sa}(X) \wedge \text{nsa}(X) \wedge \phi_{11}(X) \\ \text{smokes}(X) \leftarrow \neg\,\text{ind}(X) \wedge \text{sa}(X) \wedge \neg\,\text{nsa}(X) \wedge \phi_{12}(X) \\ \text{smokes}(X) \leftarrow \neg\,\text{ind}(X) \wedge \text{sa}(X) \wedge \text{nsa}(X) \wedge \phi_{13}(X). \end{array} \tag{5.11}$$

Clauses (5.9) and (5.10) together with the choice space form a noisy-or aggregation over all smoking and non-smoking friends respectively for which $\text{ind}(X)$ is turned on. Following Pearl (1988), the noisy-or for advice from smokers is given by

$$P(\text{ps}(X) = F \mid \{\text{friends}(X,Y), \text{ind}(Y), \text{smokes}(Y)\}) \\ = (1-\theta_8)^{\text{num-smoking-friends}(X)}, \tag{5.12}$$

where num-smoking-friends(X) is defined to be the number of individuals Y such that Y is a smoker, X and Y are friends, and Y is either an independent thinker or Y is before X in the total ordering ($Y \prec X$). Rules (5.8-5.11) along with the choice

space define CPDs for smokes(X) given values for latent variables sa(X), sa(X) given values for friends(X, Y), smokes(Y) and ind(Y) for all individuals Y, and likewise for nsa(X).

5.1.3 Semantics

Our ICL theory consists of the collection of clauses (5.1), (5.3), (5.4), (5.7) and (5.8-5.11), and the choice space. In our model, it suffices to say that when individuals are assigned to all logical variables, each ground instance of an atomic choice $\phi_k(X)$ follows a simple Bernoulli distribution $p_k(\nu)$ over $\nu \in \{\phi_k(X), \neg\, \phi_k(X)\}$ with probability of success θ_k. (The same then goes for each $\phi_k(X, Y)$.) These θ_k are the parameters θ of our model.

The semantics is given in terms of possible worlds. A *total choice* for choice space is a selection of exactly one atomic choice from each ground instance of $\{\phi_k(X), \neg\, \phi_k(X)\}$ or $\{\phi_k(X, Y), \neg\, \phi_k(X, Y)\}$ in the choice space. There is a *possible world* for each total choice. What is true in a possible world is defined by the atoms chosen by the total choice together with the logic program. The measure of a possible world is the product of values $p_k(\nu)$ for each ν selected by the total choice. The probability of the proposition is the sum of the measures of the possible worlds in which the proposition is true.

5.1.4 Acyclicity

We now elaborate on how the collection of clauses in the ICL theory above forms a contingently acyclic logic program. The propositional directed graphical model corresponding to the theory has cycles, but context-specific independence saves us: when we've assigned values to all ground instances of ind(X), the graphical model on the remaining random variables becomes acyclic.[2] As a result, our theory defines the joint probability for each configuration x of the random variables as the product

$$\begin{aligned} p(x \,|\, \theta) = & \textstyle\prod_X p(\text{smokes}(X) \,|\, \text{ind}(X), \text{sa}(X), \text{nsa}(X), \theta) \\ & \times \textstyle\prod_X p(\text{sa}(X) \,|\, \{\text{friends}(X, Y), \text{smokes}(Y), \text{ind}(Y)\}, \theta) \\ & \times \textstyle\prod_X p(\text{nsa}(X) \,|\, \{\text{friends}(X, Y), \text{smokes}(Y), \text{ind}(Y)\}, \theta) \times \prod_X p(\text{ind}(X) \,|\, \theta) \\ & \times \textstyle\prod_{X,Y} p(\text{friends}(X, Y) \,|\, \text{smokes}(X), \text{smokes}(Y), \text{ind}(X), \text{ind}(Y), \theta). \end{aligned} \quad (5.13)$$

Note that x, our notation for an assignment of all the random variables to binary values, has no relation to the logical variable X. This notion of contingent acyclicity is handled naturally in ICL, as the logic program becomes acyclic when values of all ground instances of ind(X) are known. We illustrate how the latent variables ind(X) ensure that we obtain a directed, acyclic graph with the following example.

[2] We caution that in general an acyclic logic program does not correspond to an acyclic directed graphical model.

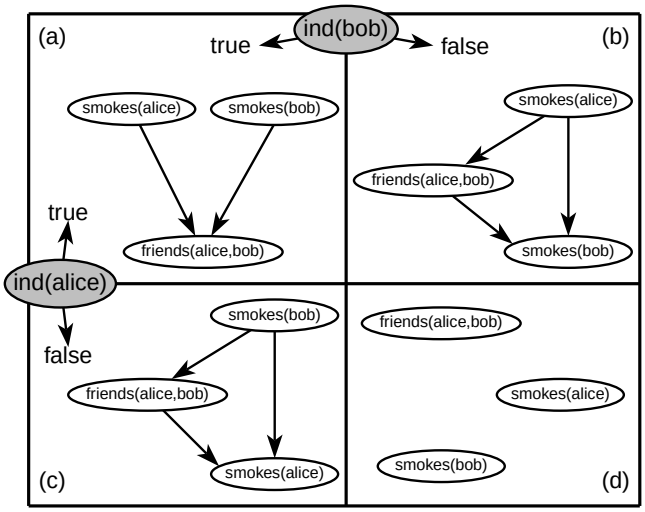

Figure 5.1: Illustration of how ind(X) works.

5.1.5 An example

Refer to Fig. 5.1.[3] We focus our attention on two individuals, Alice and Bob, within a larger social network domain, such that alice $\prec$ bob. The predicates ind(alice) and ind(bob) have four possible configurations, as depicted in Fig. 5.1. When ind(alice) = true and ind(bob) = true (Fig. 5.1a), their friendships hold no influence over their smoking habits, so their habits are allowed to influence the probability of becoming friends. In Fig. 5.1b, ind(alice) = true and ind(bob) = false. Whether Bob smokes depends on whether he is friends with Alice, and whether Alice smokes. Alice decides independently to smoke, which in turn affects her propensity to form a relationship with Bob. Note that Bob's decision to smoke also depends on other friends of his whom we haven't mentioned. Fig. 5.1c is the opposite case when ind(alice) = false and ind(bob) = true. Finally, in Fig. 5.1d, both ind(alice) and ind(bob) are false. Neither Alice nor Bob have influence over each other's smoking habits, although their smoking habits can still be influenced by other friends X for which ind(X) = true.

5.2 Learning the model

During training, we observe friendships and smoking habits, and the objective is to find a collection of model parameters that maximizes the likelihood of the evidence. Since we have random variables that are not observed during training, we follow the expectation maximization (EM) recipe, which consists of iteratively choosing the parameters θ that maximize the *expected complete log-likelihood*, then computing the posterior distribution $p(x_U \mid x_E, \theta)$ of the unobserved random variables x_U given the observations or evidence x_E. In our case, x_E corresponds to the values smokes(X) for every X, and friends(X, Y) for every pair (X, Y). We clarify what are the unobserved

[3]Illustration courtesy of Jacek Kisyński.

variables x_U in Sec. 5.2.1. Note that each entry x_i of the random vector x corresponds to a ground instance of some predicate. The conditional independence structure does not allow us to compute expectations with respect to the posterior in a reasonable amount of time, so we adopt the approximate EM framework of Heskes et al. (2004).

If x_U could be observed, then the maximum likelihood estimator would amount to the vector θ that maximizes $\log p(x_E, x_U \mid \theta)$. Since we do not observe the x_U's, a seemingly sensible course of action would be to optimize the *incomplete log-likelihood*

$$\log \textstyle\sum_{x_U} p(x_E, x_U \mid \theta). \tag{5.14}$$

This, however, will be difficult to optimize because it is not clear how to exploit the conditional independence structure of the model. Suppose instead we average over the unobserved variables, and instead work with the *expected complete log-likelihood:*

$$\ell(\theta) \equiv \textstyle\sum_{x_U} q(x_U) \log p(x_E, x_U \mid \theta), \tag{5.15}$$

where $q(x_U)$ is the "averaging distribution." If the averaging distribution is chosen correctly—precisely, if $q(x_U)$ is equal to the distribution of x_U conditioned on the evidence x_E and the parameter vector θ—then the stationary points of the expected log-likelihood (5.15) are the same as the the stationary points of (5.14). In other words, optimizing the incomplete log-likelihood or optimizing the expected log-likelihood (with the right averaging distribution) amount to the same solution.

EM can actually be understood, following Neal and Hinton (1998), as coordinate descent on the variational free energy

$$F(\theta, q) \equiv -\textstyle\sum_{x_U} q(x_U) \log p(x_E, x_U \mid \theta) + \sum_{x_U} q(x_U) \log q(x_U). \tag{5.16}$$

It is effectively the Kullback-Leibler divergence (Cover & Thomas, 1991) between the target posterior $p(x_U \mid x_E, \theta)$ and some distribution $q(x_U)$ that approximates the posterior (refer to Sec. 3.3.2). The term on left is the negative of the expected log-likelihood, and the term on the right is the negative entropy. From (5.16), the M-step reduces to finding a θ that maximizes the the expected complete log-likelihood, and the E-step reduces to finding a distribution $q(x_U)$ that best matches the posterior $p(x_U \mid x_E, \theta)$; see Heskes et al. (2004). The main difficulty lies in the E-step: direct minimization of F is infeasible due to an intractable entropy. One strategy is to restrict the class of distributions $q(x_U)$ to those that factorize in an analytically convenient fashion (Neal & Hinton, 1998). An alternative strategy is to approximate the intractable entropy by a collection of entropies on small clusters of variables. This yields belief propagation (Yedidia et al., 2005). If we choose these clusters wisely, we will obtain a tractable E-step (see Sec. 5.2.2), and the approximate M-step may resemble the true maximum likelihood estimator.

5.2.1 Maximization step

It is difficult to compute the maximum likelihood solution of the noisy-or aggregation factor for $\mathrm{sa}(X)$ when it is written as (5.12), following the standard prescription (Pearl, 1988), because maximization roughly amounts to finding the root of a polynomial. ICL directly provides us with a solution to this conundrum through the atomic choices $\phi_8(X, Y)$ that appear in the aggregation (5.9). We name these atomic choices $\mathrm{is}(X, Y)$, short for "influences to smoke" because Y is counted in X's decision to smoke when $\mathrm{is}(X, Y) = \text{true}$. These latent variables act as noisy versions of the aggregated causes; they are generated according to the choice space, and the final aggregation is achieved with a deterministic factor (5.9). (Note this variable is not symmetric like friendship; $\mathrm{is}(X, Y) = \mathrm{is}(Y, X)$ does not necessarily hold.) Similarly, in the aggregation for non-smokers we write $\phi_9(X, Y)$ as influences-not-to-smoke(X, Y), or $\mathrm{ins}(X, Y)$ for short. Thus, the atomic choices of importance are $\mathrm{is}(X, Y)$, $\mathrm{ins}(X, Y)$, and the cycle-resolving latent variables $\mathrm{ind}(X)$. These are precisely the unobserved variables x_U. The remaining atomic choices can be ignored because they are easily summed out from the disjoint rules.

Each random variable indexed by i is generated by some CPD $p(x_i \mid x_{\pi[i]})$, where $\pi[i]$ is the set of parents of node i in the directed graph. We separate the vertices of the directed graph into two sets: 1) the set of variables A that are generated by deterministic noisy-or aggregation factors, namely instances of $\mathrm{sa}(X)$ and $\mathrm{nsa}(X)$, and 2) the remaining variables B. From the ICL semantics, $p(x_i = \text{true} \mid x_{\pi[i]}) = \theta_k$ and $p(x_i = \text{false} \mid x_{\pi[i]}) = 1 - \theta_k$ for all CPDs that are not aggregation factors. Given the factorization (5.13), the expected complete log-likelihood works out to be simply

$$\ell(\theta) = \sum_{i \in B} \sum_{x_{vars[i] \cap U}} q(x_{vars[i] \cap U}) \log p(x_i \mid x_{\pi[i]}, \theta) + \text{constant}, \tag{5.17}$$

where $vars[i]$ is defined to be the intersection of $\pi[i]$ and $\{i\}$, which is precisely all the variables implicated in the ith conditional probability. The deterministic aggregation factors do not matter in the M-step because they are not affected by the choice of θ.

Since each θ_k represents a binomial success rate, we introduce uniform Beta priors

$$p(\theta_k \mid \alpha, \beta) \propto \theta_k^{\alpha-1}(1 - \theta_k)^{\beta-1}, \tag{5.18}$$

and compute the *maximum a posteriori* solution to the penalized log-likelihood. Taking partial derivatives of the penalized objective and equating them to zero, we obtain

roots $\theta_k = a_k/b_k$, where

$$a_k = \alpha - 1 + \sum_{i \in B \cap E} \sum_{x_{\pi[i] \cap U}} \mathbb{I}[k, i, x]\, \delta_{\text{true}}(x_i)\, q(x_{\pi[i] \cap U}) + \sum_{i \in B \cap U} \sum_{x_{\pi[i] \cap U}} \mathbb{I}[k, i, x]\, q(x_i = \text{true}, x_{\pi[i] \cap U}) \tag{5.19}$$

$$b_k = \alpha + \beta - 2 + \sum_{i \in B} \sum_{x_{vars[i] \cap U}} \mathbb{I}[k, i, x]\, q(x_{vars[i]}), \tag{5.20}$$

and where the delta-Dirac function $\delta_y(x) = 1$ if and only if $x = y$, and $\mathbb{I}[k, i, x] = 1$ if and only if $p(x_i = \text{true} \mid x_{\pi[i]})$ is a function of θ_k.

5.2.2 Expectation step

The missing quantities in the M-step are the marginal probabilities $q(x_i = \text{true}, x_{\pi[i] \cap U})$ for every $i \in U$, and the marginal probabilities $q(x_{\pi[i] \cap U})$ for every $i \in E$. We now explain how to estimate these marginals.

The best known tractable solution is to frame the inference problem—the problem of computing the marginals $q(x_{vars[i] \cap U})$ and $q(x_{\pi[i] \cap U})$—as an optimization problem using variational methodology, then to approximate the optimization problem using a *region-based approximation* (Yedidia et al., 2005) so we can compute the marginals efficiently. Let's look at this approximate solution in detail.

Factor graphs. The probability distribution of interest can be described in general terms as a product of non-negative functions $f_C(x_C)$ called *factors*. The probability of the configuration x_U is written as

$$p(x_U \mid x_E, \theta) = \frac{1}{Z} \prod_C f_C(x_C), \tag{5.21}$$

Each C refers to a subset of U, so that x_C represents the restriction of configuration x_U to the subset C. The normalizing constant Z is designed to ensure that $p(x_U)$ represents a valid probability; the probabilities of all configurations must sum to one.

A *factor graph* is used to express the factorization structure of the probability distribution (Kschischang et al., 2001). A factor graph has two sets of vertices: variable nodes and factor nodes. Ordinarily, variables nodes are drawn as circles and factor nodes are depicted as squares. An edge connects a variable node i to a factor node C if and only if x_i is one of the arguments of the factor ($i \in C$). The symbol C serves two roles: to index a factor as in f_C, and to refer to a collection of variable nodes as in x_C. It is assumed that no two factors are defined on the same subset C.

Each unobserved variable introduces a variable node i to the factor graph, and each CPT $p(x_i \mid x_{\pi[i]})$ introduces a factor node C, which is then linked to the variable nodes in $vars[i] \cap U$. The factor graph of the posterior $p(x_U \mid x_E, \theta)$ is nearly fully connected, as each CPT for friends(X, Y) introduces a factor between ind(X) and

ind(Y). Also, each aggregation rule creates a large factor over latent variables sa(X) and is(X, Y) for all Y that are smoking friends of X, or over the variable nsa(X) and latent causes ins(X, Y). Discovering useful substructure is a daunting task.

The Bethe method. The strategy described here, whose roots lie in the early work of Bethe (1935) and Kikuchi (1951) in statistical physics, is to approximate the intractable sums in the variational free energy F by a linear combination of more manageable terms F_R. The R represents a "region" or "cluster" of the undirected graphical model, and is a subset of U. Bethe (1935) proposed an approximation to the variational free energy F by forcing the entropy to decompose as a product of entropy terms on the sets C and singleton sets $\{i\}$. This approximation is generally referred to as the *Bethe free energy.* The junction graph method is a natural generalization of the Bethe method in which the large regions (the sets C) and the small regions (the singletons $\{i\}$) can be chosen with greater freedom. A *junction graph* is ordinarily used to formalize these notions; see Aji and McEliece (2001).

A *region graph* is a graph with directed edges and labeled vertices. It generalizes the notion of the junction graph. Each vertex is labeled by a *region* of the target factor graph. A region is defined to be a collection of variable nodes and factor nodes, with the single restriction that if a factor belongs to a region, then all its arguments also belong to the region. We denote a region by the capital letter R. Depending on the context, the symbol R may alternately refer to a collection of variable nodes, a collection of factor nodes, or a node of the region graph. In this manner, we may use x_R to denote the configuration x restricted to the set $R \subseteq U$, we may use the notation $C \in R$ to refer to factors C that are members of region R, and we also say that $q_R(x_R)$ denotes the marginal density function defined at region R.

Given a region graph, its corresponding *region free energy* is defined to be

$$\tilde{F}(q) \equiv \sum_R c_R \bar{U}_R - \sum_R c_R H_R, \tag{5.22}$$

where the average energy and entropy of region R are, respectively,

$$\bar{U}_R = -\sum_{x_R} \sum_{C \in R} q_R(x_R) \log f_C(x_C) \tag{5.23}$$

$$H_R = -\sum_{x_R} q_R(x_R) \log q_R(x_R). \tag{5.24}$$

We define $q_R(x_R)$ to be the marginal probability defined on region R, and c_R to be the "counting" number (also called the "overcounting" number) for region R. If the counting numbers are well-chosen, then decomposition of the average energy is exact.

Yedidia et al. (2005) give a recipe for coming up with reasonable counting numbers c_R for a given region graph. A good choice of numbers c_R ensures that we only count the contribution of each subset once in $\tilde{F}$. This insight is the basis for the *cluster variation method.* The observation made by McEliece and Yildirim (2002) is that this recipe is connected to results in combinatorial mathematics and, in particular,

the theory of partially ordered sets. By introducing a *partial ordering* on the regions, we can treat the collection of regions as a partially ordered set, or *poset*, where the partial ordering is the set inclusion relation, and we can then draw the regions as vertices in a Hasse diagram (Rosen, 2007). Since we have described the regions R as elements of a poset, we can frame the choice of counting numbers as a counting problem on the poset, and use the principle of inclusion and exclusion for partially ordered sets—otherwise known as the *Möbius inversion principle*—to come up with the answer (Bogart, 1990). Fortunately, the region graph construction won't be quite this complicated for the factor graph induced by the social network model when all the smoking habits and friendship relations are observed.

Suppose we define two sets of regions. The regions in the first set correspond to the maximal subsets C, and their counting numbers are set to 1. The regions in the second set correspond to the singletons $\{i\}$. We set their counting numbers to be equal to $1 - d_i$, where the degree d_i of the ith variable node is defined to be the number of neighbouring factor nodes in the factor graph, or the number of factors with which the ith variable participates. Provided all friends(X, Y) and smokes(X) are observed, a region graph defined in this way ensures that the average energy is exact, and that the contribution of every subset of variable nodes is only counted once in $\tilde{F}$. This is so because: 1) every non-empty intersection of two regions is a member of the region graph, and 2) the counting numbers are equivalent to those obtained as a solution to the Möbius inversion principle. This particular region-based approximation is equivalent to the Bethe approximation.

Expanding and simplifying (5.22), the Bethe approximation to the variational free energy is given by

$$\begin{aligned}\tilde{F}(q) = &- \sum_C \sum_{x_C} q_C(x_C) \log f_C(x_C) + \sum_C \sum_{x_C} q_C(x_C) \log q_C(x_C) \\ &+ \sum_i (1 - d_i) \sum_{x_i} q_i(x_i) \log q_i(x_i),\end{aligned} \tag{5.25}$$

where $q_i(x_i)$ and $q_C(x_C)$ are the *pseudo-marginals* defined on the variable nodes and factor nodes of the factor graph.

Solution to the Bethe method. The object is now to come up with marginals $q_C(x_C)$ and $q_i(x_i)$ that minimize the approximate variational free energy (5.25). The immediate form of the objective appears to be problematic because it could involve a summation over a large number of configurations x_C when $f_C(x_C)$ is an aggregation factor. We will address this concern shortly.

The optimization problem is to minimize $\tilde{F}(q)$ subject to three types of constraints: 1) the pseudo-marginals must be non-negative, 2) they must sum to one, and 3) the pseudo-marginals on neighbouring regions should agree. Thus, the constrained, nonconvex program is to minimize $\tilde{F}$ subject to non-negativity constraints

$$q_C(x_C) \geq 0 \quad \text{and} \quad q_i(x_i) \geq 0, \tag{5.26}$$

normalization constraints

$$\sum_{x_C} q_C(x_C) = 1 \quad \text{and} \quad \sum_{x_i} q_i(x_i) = 1, \tag{5.27}$$

and consistency constraints

$$\sum_{x_{C\setminus\{i\}}} q_C(x_C) = q_i(x_i), \tag{5.28}$$

for every factor node C, for every neighbouring variable node $i \in C$, and then again for every configuration x_i.

The standard course of action is to use results in duality to locate solutions. This leads to the familiar sum-product updates (Yedidia et al., 2005). The Lagrangian function for the constrained optimization problem is

$$\begin{aligned}\tilde{L}(q,\gamma,\lambda) = \tilde{F}(q) &+ \sum_C \gamma_C\big\{\textstyle\sum_{x_C} q_C(x_C) - 1\big\} + \sum_i \gamma_i\big\{\textstyle\sum_{x_i} q_i(x_i) - 1\big\} \\ &+ \sum_C \sum_{i\in C} \sum_{x_i} \lambda_{C,i}(x_i)\big\{q_i(x_i) - \textstyle\sum_{x_{C\setminus\{i\}}} q_C(x_C)\big\},\end{aligned} \tag{5.29}$$

where the γ_i and γ_C are the Lagrange multipliers associated with the normalization constraints, and the $\lambda_{C,i}(x_i)$ are the Lagrange multipliers for the consistency constraints. It is assumed that all the probabilities are strictly positive so that the Lagrange multipliers associated with the non-negativity constraints vanish. For a candidate point to be optimal, the gradient of the Lagrangian with respect to the primal variables must vanish. The partial derivatives of the Lagrangian (5.29) with respect to the primal variables are given by

$$\frac{\partial \tilde{L}}{\partial q_i(x_i)} = (1 - d_i)(1 + \log q_i(x_i)) + \gamma_i + \sum_{C\in N(i)} \lambda_{C,i}(x_i), \tag{5.30}$$

$$\frac{\partial \tilde{L}}{\partial q_C(x_C)} = 1 + \log q_C(x_C) - \sum_C \log f_C(x_C) + \gamma_C - \sum_{i\in C} \lambda_{C,i}(x_i), \tag{5.31}$$

where $N(i)$ is the set of factor nodes adjacent to the ith variable node in the factor graph. We recover the coordinate ascent equations by equating the partial derivatives to zero and solving for $q_C(x_C)$ and $q_i(x_i)$:

$$q_i(x_i) \propto \textstyle\prod_{C\in N(i)} (\exp \lambda_{C,i}(x_i))^{\frac{1}{d_i - 1}} \tag{5.32}$$

$$q_C(x_C) \propto f_C(x_C) \textstyle\prod_{i\in C} \exp \lambda_{C,i}(x_i), \tag{5.33}$$

Next, by making the substitutions

$$\lambda_{C,i}(x_i) = \log m_{i\to C}(x_i) \tag{5.34}$$

$$m_{i\to C}(x_i) = \textstyle\prod_{C'\in N(i)\setminus\{C\}} m_{C'\to i}(x_i), \tag{5.35}$$

the expressions for the marginals become

$$q_i(x_i) \propto \textstyle\prod_{C \in N(i)} m_{C \to i}(x_i) \tag{5.36}$$

$$q_C(x_C) \propto f_C(x_C) \textstyle\prod_{i \in C} m_{i \to C}(x_i), \tag{5.37}$$

which give us the familiar expressions for the marginal beliefs. The message update from variable node i to factor node C is given in (5.35), so the remaining piece of the puzzle is the update equation for a message passed from C to i. Starting from (5.28), then plugging (5.36) and (5.37) into this identity, we obtain the sum-product rule

$$m_{C \to i}(x_i) \propto \sum_{x_{C \setminus \{i\}}} f_C(x_C) \prod_{j \in C \setminus \{i\}} m_{j \to C}(x_j). \tag{5.38}$$

In summary, the sum-product message updates represent descent directions of the Bethe free energy (5.25) subject to the constraint that the pseudo-marginals remain locally consistent. There is some concern that these updates will oscillate indefinitely, so we implemented an E-step that is guaranteed to converge by iteratively solving a convex relaxation of $\tilde{F}$ (Heskes, 2006). In the experiments (Sec. 5.3), we compared the quality of the solutions obtained from both convergent and non-convergent implementations of the E-step.

The Bethe approximation does not immediately lead to a tractable message-passing algorithm because we still have to deal with a potentially monstrous summation for any sum-product message sent from an aggregation factor. What we have is one of the simplest examples of a *causally independent* factor (Zhang & Poole, 1996), and this fact guarantees us an efficient way to compute the summation.

Causal independence. Zhang and Poole (1996) define causal independence as follows. Causal variables $x \equiv \{x_1, \ldots, x_n\}$ are causally independent with respect to aggregate variable e if there exists a commutative, associative binary operator $*$, a collection of random variables $\xi \equiv \{\xi_1, \ldots, \xi_n\}$ with the same set of realizations as x, and a conditional probability density $p(\xi \,|\, x)$ such that

1. $e = \xi_1 * \cdots * \xi_n$
2. $p(\xi_i \,|\, \xi_{-i}, x) = p(\xi_i \,|\, x_i)$,

where ξ_{-i} is defined to be the collection of all the introduced random variables except for ξ_i. A simple but useful result of causal independence is that the probability of e given x can be written as

$$p(e \,|\, x) = \textstyle\sum_{\xi} p(\xi_1 \,|\, x_1) \cdots p(\xi_n \,|\, x_n), \tag{5.39}$$

where the summation is over all realizations ξ such that $e = \xi_1 * \cdots * \xi_n$.

The definition of causal independence extends with little extra effort to factors: a causally independent factor $f(e, x)$ would be described as an arithmetic decomposition on factors $f_i(\xi_i, x_i)$. We can then show that this notion applies directly to the sum-product update (5.38), in which one of the random variables involved in the mes-

sage update is the aggregate variable $\mathrm{sa}(X)$ or $\mathrm{nsa}(X)$, and the remaining random variables are the causes $\mathrm{is}(X, Y)$ or $\mathrm{ins}(X, Y)$. A similar observation can be used to derive efficient message-passing updates for probabilistic decoding of low-density parity check codes; see Moon (2005).

To derive the efficient message update (5.38) for the case when $f_C(x_C)$ is a noisy-or factor, we need to consider two cases. In the first case, x_i is an aggregation variable. Rewriting the sum-product message update as

$$m_{C\to i}(x_i) \propto \sum_{x_{C\setminus\{i\}}} f_C(x_C) \prod_{j\in C} g_j(x_j), \tag{5.40}$$

then the message for $x_i = \text{false}$ is derived to be

$$m_{C\to i}(\text{false}) \propto \textstyle\prod_{j\in C} g_j(\text{false}), \tag{5.41}$$

and the message for $x_i = \text{true}$ is proportional to

$$m_{C\to i}(\text{true}) \propto g_i(\text{true}) \textstyle\prod_{j\in C\setminus\{i\}} \sum_{x_j} g_j(x_j) - g_i(\text{true}) \prod_{j\in C\setminus\{i\}} g_j(\text{false}), \tag{5.42}$$

In the second case, $x_{i'}$ is one of the causes (x_i is the aggregate variable). The message sent to variable node i' works out to be

$$\begin{aligned} m_{C\to i'}(\text{false}) = {} & (g_i(\text{false}) - g_i(\text{true})) \textstyle\prod_{j\in C\setminus\{i\}} g_j(\text{false}) \\ & + g_i(\text{true})\, g_{i'}(\text{false}) \textstyle\prod_{j\in C\setminus\{i,i'\}} \sum_{x_j} g_j(x_j). \end{aligned} \tag{5.43}$$

$$m_{C\to i'}(\text{true}) = g_i(\text{true})\, g_{i'}(\text{true}) \textstyle\prod_{j\in C\setminus\{i,i'\}} \sum_{x_j} g_j(x_j). \tag{5.44}$$

The inference strategy we have outlined in this section is not necessarily appropriate for making predictions about a social network when arbitrary friendships and smoking habits are unknown. When we only need to make a single prediction $\mathrm{smokes}(X)$ or $\mathrm{friends}(X, Y)$, however, a straightforward way to obtain a prediction is to estimate the Bayes factor (Kass & Raftery, 1995) from $\tilde{F}$ for two cases, when the query variable is true and when it is false.

5.2.3 Summary of EM algorithm

The approximate EM algorithm for training the model on social network data is summarized in Fig. 5.2. The inputs to the learning algorithm are the collection of evidence x_E regarding friendships and smoking habits, and the initial parameter settings $\theta^{(0)}$. After running the EM algorithm for a specified number of iterations, the output is the parameter vector θ approximating the maximum likelihood estimator. Steps 1 through 3 of the main loop comprise the E-step, and step 4 is the M-step. The E-step in Fig. 5.2 uses the non-convergent version of belief propagation.

The learning algorithm we have described does not strictly adhere to Bayesian principles, because we do not adjust the model to reflect evidence obtained after the training phase, and because we replace the integral over the model parameters θ with

- Let evidence x_E and initial parameters $\theta^{(0)}$ be given.
- **for** $t = 1, 2, 3, \ldots$
  1. **(E-step)** Initialize messages $m_{C \to i}(x_i)$ and $m_{i \to C}(x_i)$ to uniform for every factor node C and for every neighbouring variable node $i \in C$.
  2. **repeat** for a specified number of iterations
     - Update the variable-to-factor messages $m_{i \to C}(x_i)$ following (5.35).
     - Update the factor-to-variable messages $m_{C \to i}(x_i)$ following (5.38) when C is not an aggregation factor, and following (5.42,5.41) or (5.44,5.43) when C is an aggregation factor, using the factors $f_C(x_C)$ defined by the current parameters $\theta^{(t)}$.
  3. For every factor C that is not an aggregation factor, calculate the marginal belief $q_C(x_C)$ according to (5.37).
  4. **(M-step) for** $k = 0, 1, \ldots, 13$
     - Get counts a_k, b_k following (5.19) and (5.20) using the marginal beliefs $q_C(x_C)$.
     - Update parameters $\theta_k = a_k / b_k$.

Figure 5.2: Approximate EM algorithm for learning the social network model.

a single mode. However, this is standard practice for learning in graphical models.

5.3 Experiments

We ran three experiments to assess the behaviour of the proposed network model. For the first two, we used data generated from artificial processes. The third experiment comprised an actual social network analysis of smoking in grade school adolescents.

5.3.1 Experimental setup

In all the experiments, we trained our model with two versions of EM following the description of Sec. 5.2: one with a non-convergent E-step ("loopy" belief propagation), and another with an E-step based on a convergent message passing algorithm. The only real parameter to adjust was the Beta prior on the model parameters θ_k. We chose a weak, uniform prior $\alpha_k = 4, \beta_k = 4$.

We compared the performance of our model to an undirected probabilistic graphical model represented as a Markov logic network, or MLN (Richardson & Domingos, 2006). We used the software Alchemy (Kok et al., 2009) to learn weighted formulae of the form

$$\texttt{Smokes}(x) \wedge \texttt{Friends}(x, y) \Rightarrow \texttt{Smokes}(y) \tag{5.45}$$

for various non-redundant combinations of its atoms, and $\texttt{Friends}(x, y) \Rightarrow \texttt{Friends}(y, x)$ to enforce symmetry of friendship.[4] We tried more complex models that had more rules such as reflexivity and transitivity of friendship, but they offered no advantage. Alchemy implements the pseudo-likelihood approximation for learning, and includes a specialized satisfiability solver MC-SAT for inferring queries.

In one of the experiments, we compared to special cases of our model when $p(\text{ind}(X) = \text{true}) = 0$ (called the "independent friendship" model since all decisions regarding friendship are unaffected by smoking habits), and the "independent smokers" model when $p(\text{ind}(X) = \text{true}) = 1$.

Part 1. In the first set of experiments, we generated artificial social networks from our directed model with pre-specified model parameters. The control variable was the prior on $\text{ind}(X) = \text{true}$, which we varied from 1/10 (most friendships are generated randomly) to 9/10 (most smoking habits are generated randomly) in intervals of 1/10. Such an experiment may appear to be unfair, but it was unexpectedly challenging, probably due to the difficulty of recovering the latent behaviour of individuals. For each $p(\text{ind}(X) = \text{true})$, we ran 16 independent trials, and in each trial we generated training and test sets, each containing 8 isolated social networks with populations of size $n = 50$.

Part 2. In the second set of experiments, we generated data from a temporal process that bore little resemblance to the simple model we propose in this paper. In our simulation, individuals were occasionally pressured to change their smoking habits, they started or stopped smoking due to external factors, formed new friendships either by chance encounters or through mutual friends, or stopped being friends, sometimes because of friends' smoking habits. At any time step, an individual X might begin or stop smoking, depending on whether or not X's friends smoke. Furthermore, the structure of the network evolved over time: at any time step $\text{friends}(X, Y)$ may become true with some probability that depends on whether X and Y smoke. The likelihood of these interactions depended on proximity according to a latent location. Precisely, individuals were sampled at geographic locations, and people that lived close by were more likely to become friends than those that lived far away. Since none of the tested models possessed such details, we did not expect them to perform well. We ran three experiments with populations of size $n = 20$, 100 and 200. Each training and test set consisted of 5 separate populations. The data sets exhibited considerable variance in the number of friendships and smokers.

Part 3. In the final experiment, we learned a social network model from a year-long longitudinal study of smoking and drug use in a cohort of $n = 150$ teenagers attending a school in Scotland (Pearson & Michell, 2000). It is purported to be the first scientific study in the UK to adopt social network methodology for analyzing smoking and drug-taking behaviour. The authors only recorded reciprocal friendship links. They gathered other information, such as gender, and used this information to assess the strength of links. This information would have surely improved the quality

[4] In practice, we found that the number of smoking rules of the form (5.45) had a significant impact on accuracy, so we always tried to pick a set that worked well.

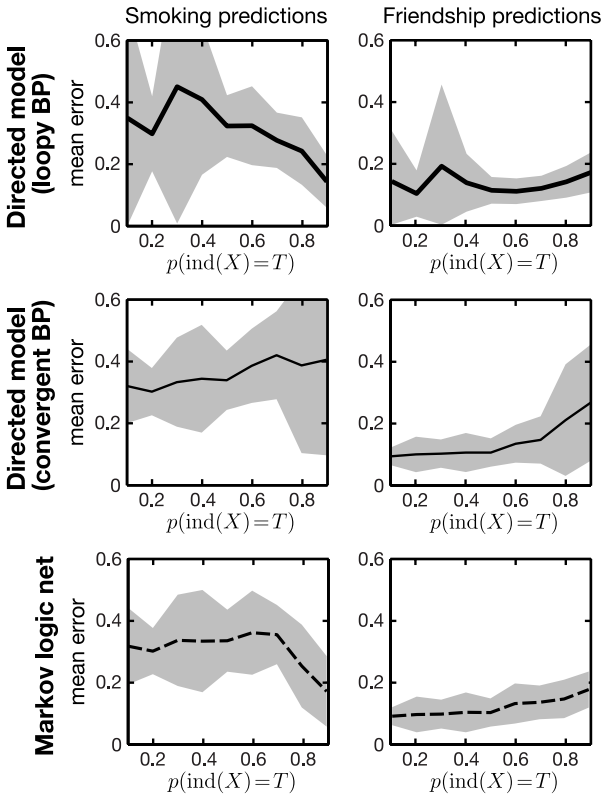

Figure 5.3: Average smoking and friendship prediction error from the directed model (trained with a loopy and convergent E-step) and the MLN for varying proportions of ind(X) = true, over 16 independent trials.

of predictions (e.g. girls tended to be friends with girls, smoking was less prevalent among boys due to perception that it affects performance in sports). We trained the models on the data collected when the students were in grade 2, and validated the models on the survey data from a year later in grade 3. These social networks were very sparsely connected and highly transitory, hence none of approaches tested here were able to to learn a useful model of friendship.

5.3.2 Experiment results

Part 1. The results of the first experiment are shown in Fig. 5.3. For each model, a single test consisted of computing the maximum a posteriori estimate of smokes(X) or friends(X, Y) for a particular individual X or pair of individuals (X, Y) given information regarding the habits and friendships of the remaining portion of the testing network. Fig. 5.3 was then obtained by taking the mean error of these tests. The shaded region is the 90% confidence interval. As expected, the accuracy of the MLN *(bottom)* and the directed model with non-convergent, loopy belief propagation *(top)* got better as more and more individuals were not influenced by their peers,

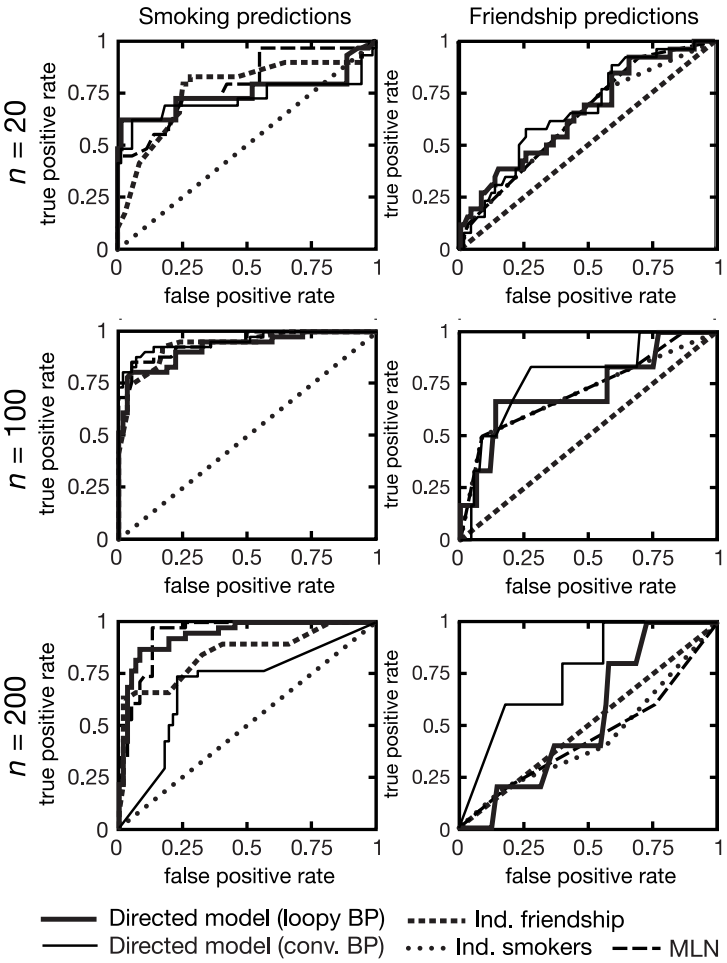

Figure 5.4: ROC curves for smoking and friendship predictions on test sets from the artificial temporal process.

but what is surprising is that the performance of the directed model with convergent belief propagation *(middle)* did not improve, and even degraded slightly—we currently have no explanation for this behaviour. The loopy implementation *(top)* was not completely satisfactory either as its performance varied considerably in networks with few $\text{ind}(X) = T$. At the right-most end of the spectrum, when $\text{ind}(X) = T$ for most individuals X, it is still possible for the model to make useful inferences about smoking habits by conditioning on observations about friendship. Within the confidence intervals, we obtained about the same level of performance for the directed and undirected models, barring the unexpected effects of an approximate E-step.

Part 2. Results of the second set of simulations are shown as receiver operating characteristic (ROC) curves in Fig. 5.4. Tests were done in the same manner as before: for each test, we left out one smoking or friendship observation. Unsurprisingly, these simple social network models did not quite capture the complexity of the artificial

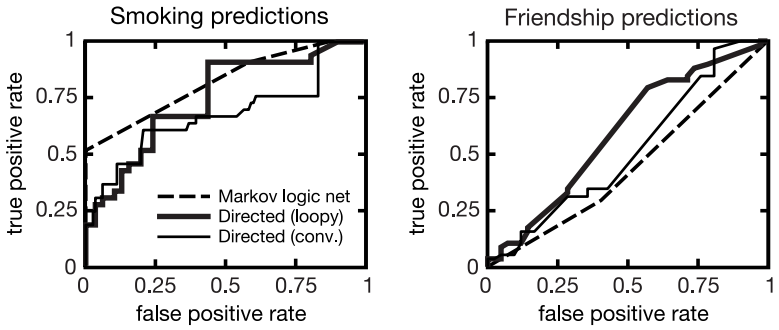

Figure 5.5: ROC curves for smoking and friendship predictions on the adolescent social network test data.

process, particularly in predicting friendships. We did not observe a degradation in the performance of the convergent implementation like we did in the first experiment, although it is interesting to note that it did much better at predicting friendships in the large ($n = 200$) network at the expense of poor prediction of smoking habits. As expected, the "independent smokers" and "independent friendship" models did no better than the worst possible (*i.e.* a straight line) at predicting, respectively, smokes(X) and friends(X, Y). It is significant that the directed, contingently acyclic model: 1) outperformed these two simple relational models on both predictions of smoking and friendship, and 2) tended to make predictions about as accurately as the Markov logic network model.

Part 3. Finally, we examine the results from the adolescent smoking and drug use study in Fig. 5.5. Overall, we observe trends similar to our previous experiments on synthetic data. The MLN displayed some advantage in accuracy of smoking habits, but did worse in predicting friendships. Friendship predictions were globally poor, as we forewarned. These results do nonetheless clearly suggest that more detailed expert knowledge must be inputed into the model to obtain useful scientific inferences.

5.4 Conclusions and discussion

Contrary to common practice, we developed a directed graphical model for social networks and an approximate EM algorithm for training the model. Our experiments on both synthetic and actual data of friendships and smoking habits showed that a directed model can predict interdependencies equally as well as a similarly expressive undirected model in a simple but challenging social network domain. Our experiments also highlight the need for more work into robust convergent message passing algorithms for belief propagation, or the development of entirely different inference approaches based on Monte Carlo and stochastic approximation, as suggested in the introduction to this chapter.

There are many open research questions in extending our ideas to larger and more challenging social network domains. One important open question is how to design directed graphical representations of social networks that transfer to populations of

different sizes. In so doing, one could learn the model parameters from a small network for which data has been collected, and use it to make predictions in much larger social networks. Another unresolved problem is how to efficiently handle queries with arbitrary sets of observations in large social networks—it is far from clear how to exploit such model structure for conducting inference at a first-order level (Poole, 2003), and for developing approximate sum-product message passing algorithms.

Chapter 6

Conclusions

In my thesis, I argued that many problems of real interest to scientists lead to difficult probabilistic inference problems. I also argued and demonstrated that the three well-explored approaches to designing approximate inference algorithms—Markov chain Monte Carlo, importance sampling and variational methods—demonstrate weaknesses in the face of difficult inference problems. In Chapters 3 and 4, I showed how we can devise new algorithms that exploit the strengths of variational and Monte Carlo methods, leading to better approximations to these difficult inference problems.

My work on the stochastic approximation method for probabilistic inference (Chapter 3) was initially an attempt to overcome some of the problems I observed with conditional mean field (Chapter 4). The unexpected outcome of this technical contribution is that it demonstrated how to effect a general trade-off in probabilistic inference between bias (due to variational approximation) and variance (due to importance sampling). Much work remains in understanding the advantages and limitations of this new approach to inference, and in adapting this approach to the wide range of probabilistic inference problems that are being investigated in the scientific community.

The stochastic approximation method for probabilistic inference is what initially motivated me to pursue the research presented in Chapter 2 on an interior-point stochastic approximation method, though in this chapter I provided further motivation by showing how the problem of learning a sparse regression model in an on-line fashion could be formulated as a constrained stochastic approximation problem. The unexpected outcome of this chapter was that it laid the groundwork for an intriguing connection between primal-dual interior-point methods—methods that were originally developed in the 1980s for linear programming—and stochastic approximation. It remains to be seen whether my interior-point stochastic approximation method opens the door to tackling other problems in machine learning that can be cast as constrained optimization problems with noisy estimates of the gradient.

Finally, in Chapter 5, I described a new *contingently acyclic* probabilistic model. I demonstrated how to adapt existing approximate inference techniques to this model, and the experiments I conducted show how directed graphical models can offer a viable alternative to the standard undirected modeling approach for representing interdependent relations (such as friendships and smoking habits) within a social network. I proposed one way of attacking inference in this contingently acyclic probabilistic model, but many there are still many open questions on how to make accurate and efficient inferences in this model.

Bibliography

Abelson, H., Sussman, G. J., & Sussman, J. (1996). *Structure and interpretation of computer programs*. MIT Press. 2nd edition.

Aji, S. M., & McEliece, R. J. (2001). The generalized distributive law and free energy minimization. *Proceedings of the 39th Allerton Conference* (pp. 672–681).

Aldous, D. J. (1985). Exchangeability and related topics. *Lecture Notes in Math., École d'été de probabilities de Saint-Flour XIII*, *1117*, 1–198.

Alizadeh, F., Haeberly, J.-P. A., & Overton, M. L. (1998). Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results. *SIAM Journal on Optimization*, *8*, 746–768.

Altman, E. (1999). *Constrained Markov decision processes*. Chapman and Hall/CRC.

Amari, S. (1998). Natural gradient works efficiently for learning. *Neural Computation*, *10*, 251–276.

Amestoy, P. R., Duff, I. S., L'Excellent, J.-Y., & Koster, J. (2001). A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, *23*, 15–41.

Andersen, K. D. (1996). An efficient Newton barrier method for minimizing a sum of Euclidean norms. *SIAM Journal on Optimization*, *6*, 74–95.

Andersen, K. D., Christiansen, E., Conn, A. R., & Overton, M. L. (2000). An efficient primal-dual interior-point method for minimizing a sum of Euclidean norms. *SIAM Journal on Scientific Computing*, *22*, 243–262.

Andrew, G., & Gao, J. (2007). Scalable training of L1-regularized log-linear models. *Proceedings of the 24th International Conference on Machine Learning* (pp. 33–40).

Andrieu, C., de Freitas, N., Doucet, A., & Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine Learning*, *50*, 5–43.

Andrieu, C., & Doucet, A. (2002). Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society*, *64*, 827–836.

Andrieu, C., & Moulines, E. (2006). On the ergodicity properties of some adaptive MCMC algorithms. *Annals of Applied Probability*, *16*, 1462–1505.

Andrieu, C., Moulines, E., & Priouret, P. (2005). Stability of stochastic approximation under verifiable conditions. *SIAM Journal on Control and Optimization*, *44*, 283–312.

Anitescu, M., Tseng, P., & Wright, S. J. (2007). Elastic-mode algorithms for mathematical programs with equilibrium constraints: global convergence and stationarity properties. *Mathematical Programming*, *110*, 337–371.

Arnold, B., Castillo, E., & Sarabia, J.-M. (1999). *Conditional specification of statistical models*. Springer.

Arnold, B. C., Castillo, E., & Sarabia, J.-M. (2001). Conditionally specified distribu-

tions: an introduction. *Statistical Science*, *13*, 249–274.

Banerjee, A., Merugu, S., Dhillon, I. S., & Ghosh, J. (2006). Clustering with Bregman divergences. *Journal of Machine Learning Research*, *6*, 1705–1749.

Barnard, K., Guygulu, P., Forsyth, D., de Freitas, N., Blei, D. M., & Jordan, M. I. (2003). Matching words and pictures. *Journal of Machine Learning Research*, *3*, 1107–1135.

Barzilai, J., & Borwein, J. M. (1988). Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, *8*, 141–148.

Baxter, J., & Bartlett, P. L. (2001). Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, *15*, 319–350.

Baxter, R. J. (1982). *Exactly solved models in statistical mechanics*. Academic Press.

Beal, M. J. (2003). *Variational algorithms for approximate Bayesian inference*. Doctoral dissertation, University College London.

Benveniste, A., Métivier, M., & Priouret, P. (1990). *Adaptive algorithms and stochastic approximations*. Springer-Verlag.

Benzi, M., Golub, G. H., & Liesen, J. (2005). Numerical solution of saddle point problems. *Acta Numerica*, *14*, 1–137.

Berger, M. (1990). Convexity. *American Mathematical Monthly*, *97*, 650–678.

Bergman, N. (1999). *Recursive Bayesian estimation: nagivation and tracking applications*. Doctoral dissertation, Dept. of Electrical Engineering, Linköping University.

Berteskas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-dynamic programming*. Athena Scientific.

Bertsekas, D. P. (1982). *Constrained optimization and Lagrange multiplier methods*. Academic Press.

Bertsekas, D. P. (1999). *Nonlinear programming*. Athena Scientific.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, *36*, 192–236.

Besag, J. (2001). Comment to "Conditionally specified distributions: an introduction". *Statistical Science*, *16*, 265–267.

Bethe, H. A. (1935). Statistical theory of superlattices. *Proceedings of the Royal Society of London*, *150*, 552–575.

Bhatt, R. N., & Young, A. P. (1985). Search for a transition in the three-dimensional +/-J Ising spin-glass. *Physical Review Letters*, *54*.

Binder, K., & Young, A. P. (1986). Spin glasses: experimental facts, theoretical concepts, and open questions. *Reviews of Modern Physics*, *58*, 801–976.

Bishop, C., Lawrence, N., Jaakkola, T., & Jordan, M. I. (1998). Approximating posterior distributiosn in belief networks using mixtures. In *Advances in neural information processing systems*, vol. 10.

Blei, D., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Bogart, K. P. (1990). *Introductory combinatorics*. Academic Press. 2nd edition.

Borkar, V. S. (2008). *Stochastic approximation: a dynamical systems viewpoint*. Cambridge University Press.

Bottou, L. (1998). Online learning and stochastic approximations. In *On-line learning in neural networks*, 9–42. Cambridge University Press.

Bottou, L. (2004). Stochastic learning. In O. Bousquet, von U. Luxburg and G. Rätsch (Eds.), *Advanced lectures on machine learning*, vol. 3176 of *Lecture Notes in Computer Science*, 146–168. Springer.

Boutilier, C., Friedman, N., Goldszmidt, M., & Koller, D. (1996). Context-specific independence in Bayesian networks. *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence* (pp. 115–123).

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Boykov, Y., Veksler, O., & Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*, 1222–1239.

Brown, L. D. (1986). *Fundamentals of statistical exponential families*. Institute of Mathematical Statistics.

Bunch, J. R., & Parlett, B. N. (1971). Direct methods for solving symmetric indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, *8*, 639–655.

Buntine, W., & Jakulin, A. (2004). Applying discrete PCA to data analysis. *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence* (pp. 59–66).

Buntine, W. L. (1994). Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, *2*, 159–225.

Burchard, E. G., Ziv, E., Coyle, N., Gomez, S. L., Tang, H., Karter, A. J., Mountain, J. L., Pérez-Stable, E. J., Sheppard, D., & Risch, N. (2003). The importance of race and ethnic background in biomedical research and clinical practice. *The New England Journal of Medicine*, *348*, 1170–1175.

Byrd, R. H., Nocedal, J., & Schnabel, R. B. (1994). Representations of quasi-newton matrices and their use in limited memory methods. *Mathematical Programming*, *63*, 129–156.

Cafieri, S., D'Apuzzo, M., Marino, M., Mucherino, A., & Toraldo, G. (2006). Interior-point solver for large-scale quadratic programming problems with bound constraints. *Journal of Optimization Theory and Applications*, *129*, 55–75.

Candès, E. J., Romberg, J., & Tao, T. (2006). Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, *52*, 489–509.

Carbonetto, P., & de Freitas, N. (2007). Conditional mean field. In *Advances in neural information processing systems*, vol. 19, 201–208. MIT Press.

Carbonetto, P., de Freitas, N., & Barnard, K. (2004). A statistical model for general contextual object recognition. *Proceedings of the 8th European Conference on Computer Vision* (pp. 350–362).

Carbonetto, P., Schmidt, M., & de Freitas, N. (2009). An interior-point stochastic approximation method and an L1-regularized delta rule. In *Advances in neural information processing systems*, vol. 21. MIT Press.

Carrington, P. J., Scott, J., & Wasserman, S. (Eds.). (2005). *Models and methods in social network analysis*. Cambridge University Press.

Casella, G., & Berger, R. L. (2002). *Statistical inference*. Thomson Learning. 2nd edition.

Casella, G., & Robert, C. P. (1996). Rao-Blackwellisation of sampling schemes. *Biometrika*, *83*, 81–94.

Celeux, G., Hurn, M., & Robert, C. P. (2000). Computational and inferential difficulties with mixture posterior distributions. *Journal of the American Statistical Association*, *95*, 957–970.

Chen, L., & Goldfarb, D. (2006). Interior-point L2-penalty methods for nonlinear programming with strong global convergence properties. *Mathematical Programming*, *108*, 1–36.

Chen, S., Donoho, D., & Saunders, M. (1999). Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, *20*, 33–61.

Chen, S. S., Donoho, D. L., & Saunders, M. A. (2001). Atomic decomposition by basis pursuit. *SIAM Review*, *43*, 129–159.

Chib, S. (1996). Calculating posterior distributions and modal estimates in Markov mixture models. *Journal of Econometrics*, *75*, 79–97.

Chib, S., & Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm. *The American Statistician*, *49*, 327–335.

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika*, *89*, 539–552.

Clark, J. S. (2005). Why environmental scientists are becoming Bayesians. *Ecology letters*, *8*, 2–14.

Coltman, D. W. (2007). Molecular ecological approaches to studying the evolutionary impact of selective harvesting in wildlife. *Molecular Ecology*, *17*, 221–235.

Cormack, G. V. (2006). TREC 2006 spam track overview. *Proceedings of the 15th Text Retrieval Conference*.

Cormack, G. V., & Bratko, A. (2006). Batch and on-line spam filter evaluation. *Proceedings of the 3rd Conference on Email and Anti-Spam*.

Cormack, G. V., & Lynam, T. R. (2005). Spam corpus creation for TREC. *Proc. 2nd CEAS*.

Cormack, G. V., & Lynam, T. R. (2007). Online supervised spam filter evaluation. *ACM Trans. Information Systems*, *25*.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). *Introduction to algorithms*. MIT Press. 2nd edition.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. Wiley.

Darwiche, A. (2001). Recursive conditioning. *Artificial Intelligence*, *126*, 5–42.

Davies, N., Villablanca, F. X., & Roderick, G. K. (1999). Determining the source of individuals: multilocus genotyping in nonequilibrium population genetics. *Trends in Ecology and Evolution*, *14*, 17–21.

de Boer, P.-T., Kroese, D. P., Mannor, S., & Rubinstein, R. Y. (2005). A tutorial on the cross-entropy method. *Annals of Operations Research*, *134*, 19–67.

de Freitas, N., Højen-Sørensen, P., Jordan, M. I., & Russell, S. (2001). Variational MCMC. *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence* (pp. 120–127).

Dechter, R., & Mateescu, R. (2007). AND/OR search spaces for graphical models. *Artificial Intelligence*, *171*, 73–106.

Del Moral, P., Doucet, A., & Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society*, *68*, 411–436.

Delyon, B., Lavielle, M., & Moulines, E. (1999). Convergence of a stochastic approximation version of the EM algorithm. *Annals of Statistics*, *24*, 94–128.

Dennis, J. E., & Moré, J. J. (1977). Quasi-Newton methods, motivation and theory. *SIAM Review*, *19*, 46–89.

Dennis, J. E., & Schnabel, R. B. (1996). *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM.

Derrick, W. R. (1984). *Complex analysis and applications*. Wadsworth, Inc. 2nd edition.

DiCiccio, T. J., Kass, R. E., Raftery, A., & Wasserman, L. (1997). Computing Bayes factors by combining simulation and asymptotic approximations. *Journal of the American Statistical Association*, *92*, 903–915.

Dobson, A. J. (2002). *An introduction to generalized linear models*. Chapman and Hall/CRC Press. 2nd edition.

Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, *52*, 1289–1306.

Doucet, A., de Freitas, N., & Gordon, N. J. (2001). *Sequential Monte Carlo methods in practice*. Series on Statistics for Engineering and Information Science. Springer-Verlag.

Doucet, A., de Freitas, N., Murphy, K., & Russell, S. (2000a). Rao-Blackwellised particle filtering for dynamic Bayesian networks. *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence* (pp. 176–183).

Doucet, A., Godsill, S., & Andrieu, C. (2000b). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, *10*, 197–208.

Duchi, J., Shwartz, S. S., Singer, Y., & Chandra, T. (2008). Efficient projections onto the L1-ball for learning in high dimensions. *Proceedings of the 25th International Conference on Machine Learning* (pp. 272–279).

Duff, I. S., Erisman, A. M., & Reid, J. K. (1986). *Direct methods for sparse matrices*. Oxford University Press.

Dvoretsky, A. (1956). On stochastic approximation. *Proceedings of the 3rd Berkeley*

Symposium on Mathematical Statistics and Probability (pp. 39–45).

Dykstra, R. L. (1983). An algorithm for restricted least squares regression. *Journal of the American Statistical Association*, *78*, 837–842.

Earl, D. J., & Deem, M. W. (2005). Parallel tempering: theory, applications, and new perspectives. *Physical Chemistry and Chemical Physics*, *7*, 3910–3916.

Efron, B. (1978). The geometry of exponential families. *Annals of Statistics*, *6*, 362–376.

Eisenstat, S. C., & Walker, H. F. (1994). Globally convergent inexact Newton methods. *SIAM Journal on Optimization*, *4*, 393–422.

El-Bakry, A. S., Tapia, R. A., Tsuchiya, T., & Zhang, Y. (1996). On the formulation and theory of the Newton interior-point method for nonlinear programming. *Journal of Optimization Theory and Applications*, *89*, 507–541.

Ellegren, H. (2004). Microsatellites: simple sequences with complex evolution. *Nature Reviews Genetics*, *5*, 435–445.

Falush, D., Stephens, M., & Pritchard, J. K. (2003). Inference of population structure using multilocus genotype data: linked loci and correlated allele frequencies. *Genetics*, *164*, 1567–1587.

Falush, D., Stephens, M., & Pritchard, J. K. (2007). Inference of population structure using multilocus genotype data: dominant markers and null alleles. *Molecular Ecology Notes*, *7*, 574–578.

Fearnhead, P. (1998). *Sequential Monte Carlo methods in filter theory*. Doctoral dissertation, University of Oxford.

Fearnhead, P. (2008). Computational methods for complex stochastic systems: a review of some alternatives to MCMC. *Statistics and Computing*, *18*, 151–171.

Ferdinand, A. E., & Fisher, M. E. (1969). Bounded and inhomogeneous Ising models I: specific-heat anomaly of a finite lattice. *Physical Review*, *185*, 832–846.

Fiacco, A. V., & McCormick, G. P. (1968). *Nonlinear programming: sequential unconstrained minimization techniques*. John Wiley and Sons.

Fienberg, S. E. (2006). When did Bayesian inference become "Bayesian"? *Journal of Bayesian Analysis*, *1*, 1–40.

Figueiredo, M. A. T. (2003). Adaptive sparseness for supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*, 1150–1159.

Fischer, K. H., & Hertz, J. A. (1991). *Spin glasses*. Cambridge University Press.

Fisher, R. (1953). Population genetics. *Proceedings of the Royal Society of London*, *141*, 510–523.

Fletcher, R. (2005). *On the Barzilai-Borwein method*, 235–256. Springer.

Fletcher, R., Leyffer, S., Ralph, D., & Scholtes, S. (2006). Local convergence of SQP methods for mathematical programs with equilibrium constraints. *SIAM Journal on Optimization*, *17*, 259–286.

Forsgren, A., Gill, P. E., & Wright, M. H. (2002). Interior methods for nonlinear optimization. *SIAM Review*, *44*, 525–597.

Frank, O., & Strauss, D. (1986). Markov graphs. *Journal of the American Statistical Association*, *81*, 832–842.

Frauenkron, H., Bastolla, U., Gerstner, E., Grassberger, P., & Nadler, W. (1998). New Monte Carlo algorithm for protein folding. *Physical Review Letters*, *80*, 3159–3152.

Frey, B. J., & MacKay, D. J. C. (1997). A revolution: belief propagation in graphs with cycles. In *Advances in neural information processing systems*, vol. 10. MIT Press.

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, *28*, 337–407.

Fung, R., & Chang, K.-C. (1989). Weighting and integrating evidence for stochastic simulation in Bayesian networks. *Proceedings of the 5th Workshop on Uncertainty in Artificial Intelligence* (pp. 112–117).

Gafni, E. M., & Bertsekas, D. P. (1984). Two-metric projection methods for constrained optimization. *SIAM Journal on Control and Optimization*, *22*, 936–964.

Galluccio, A., Loebl, M., & Vondrak, J. (2000). New algorithm for the Ising problem: Partition function for finite lattice graphs. *Physical Review Letters*, *84*, 5924–5957.

Garrigues, P., & Ghaoui, L. E. (2009). An homotopy algorithm for the Lasso with online observations. In *Advances in neural information processing systems*, vol. 21, 489–496.

Gay, D. M., Overton, M. L., & Wright, M. H. (1996). A primal-dual interior method for nonconvex nonlinear programming. In xiang Y. Yuan (Ed.), *Advances in nonlinear programming*, 31–56. Kluwer Academic Publishers.

Geiger, D., & Heckerman, D. (1996). Knowledge representation and inference in similarity networks and Bayesian multinets. *Artificial Intelligence*, *82*, 45–74.

Gelman, A., & Meng, X.-L. (1998). Simulating normalizing constants: from importance sampling to bridge sampling to path sampling. *Statistical Science*, *13*, 163–185.

Gelman, A. G., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2003). *Bayesian data analysis*. Chapman & Hall/CRC. 2nd edition.

Genkin, A., Lewis, D. D., & Madigan, D. (2007). Large-scale Bayesian logistic regression for text categorization. *Technometrics*, *49*, 291–304.

George, A. P., & Powell, W. B. (2006). Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. *Machine Learning*, *65*, 167–198.

Getoor, L., & Taskar, B. (Eds.). (2007). *Introduction to statistical relational learning*. MIT Press.

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, *57*, 1317–1339.

Geyer, C. J., & Thompson, E. A. (1995). Annealing Markov chain Monte Carlo with applications to ancestral inference. *Journal of the American Statistical Association*, *90*, 909–920.

Ghahramani, Z., & Beal, M. J. (2000). Variational inference for Bayesian mixtures of factor analysers. In *Advances in neural information processing systems*, vol. 12, 449–455. MIT Press.

Ghahramani, Z., & Beal, M. J. (2001). Propagation algorithms for variational bayesian learning. In *Advances in neural information processing systems*, vol. 13, 507–513. MIT Press.

Ghosh, J. K., & Ramamoorthi, R. V. (2003). *Bayesian nonparametrics*. Springer-Verlag.

Gill, P. E., Murray, W., & Wright, M. E. (1986). *Practical optimization*. Academic Press.

Gillespie, J. H. (2004). *Population genetics: a concise guide*. John Hopkins University Press. 2nd edition.

Godsill, S. J., Doucet, A., & West, M. (2004). Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, *99*, 156–168.

Goswami, G., & Liu, J. S. (2007). On learning strategies for evolutionary monte carlo. *Statistics and Computing*, *17*, 23–38.

Gould, N., Orban, D., & Toint, P. (2005). Numerical methods for large-scale nonlinear optimization. *Acta Numerica*, *14*, 299–361.

Gould, N. I. M. (1985). On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem. *Mathematical Programming*, 90–99.

Gould, N. I. M., Orban, D., & Toint, P. L. (2003). *An interior-point L1-penalty method for nonlinear optimization* (Technical Report). Rutherford Appleton Laboratory.

Grassberger, P. (2002). Go with the winners: a general Monte Carlo strategy. *Computer Physics Communications*, *147*, 64–70.

Greenberg, E. (2007). *Introduction to Bayesian econometrics*. Cambridge University Press.

Gregory, P. C. (2005). A Bayesian analysis of extrasolar planet data for HD 73526. *Astrophysics Journal*, *631*, 1198–1214.

Griffiths, A. J., Wessler, S. R., Lewontin, R. C., & Carroll, S. B. (2008). *Introduction to genetic analysis*. W. H. Freeman & Co. 9th edition.

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, *101*, 5228–5235.

Haario, H., Saksman, E., & Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, *7*, 223–242.

Häggström, O. (2002). *Finite markov chains and algorithmic applications*. Cambridge University Press.

Hamze, F., & de Freitas, N. (2004). From fields to trees. *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence* (pp. 243–250).

Hamze, F., & de Freitas, N. (2006). Hot Coupling: a particle approach to inference and normalization on pairwise undirected graphs. In *Advances in neural information*

processing systems, vol. 18, 491–498. MIT Press.

Hansson, O., & Mayer, A. (1989). Heuristic search as evidential reasoning. *Proceedings of the 5th Workshop on Uncertainty in Artificial Intelligence* (pp. 152–161).

Hartl, D. L., & Clark, A. G. (2007). *Principles of population genetics*. Sinauer Associates.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. Springer.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, *57*, 97–109.

Hazan, E., Agarwal, A., & Kale, S. (2007). Logarithmic regret algorithms for online convex optimization. *Machine Learning*, *69*, 169–192.

Hedrick, P. W. (2005). A standardized genetic differentiation measure. *Evolution*, *59*, 1633–1638.

Hedrik, P. W. (2005). *Genetics of populations*. Jones and Bartlett. 3rd edition.

Heidelberger, P. (1995). Fast simulation of rare events in queueing and reliability models. *ACM Transactions on Modeling and Computer Simulation*, *5*, 43–85.

Hein, J., Schierup, M. H., & Wiuf, C. (2005). *Gene genealogies, variation and evolution: a primer in coalescent theory*. Oxford University Press.

Heskes, T. (2006). Convexity arguments for efficient minimization of the Bethe and Kikuchi free energies. *Journal of Artificial Intelligence Research*, *26*, 153–190.

Heskes, T., Albers, K., & Kappen, B. (2003). Approximate inference and constrained optimization. *Uncertainty in Artificial Intelligence* (pp. 313–320).

Heskes, T., Zoeter, O., & Wiegerinck, W. (2004). Approximate expectation maximization. In *Advances in neural information processing systems*, vol. 16, 353–360. MIT Press.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, *14*, 1771–1800.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*, 1527–1554.

Hiriart-Urruty, J. (1977). *Algorithms of penalization-type and of dual type for the solution of stochastic approximation problems with stochastic constraints*, 183–221. North-Holland Publishers.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, *42*, 177–196.

Holmes, I., & Rubin, G. M. (2002). An expectation maximization algorithm for training hidden substitution models. *Journal of Molecular Biology*, *317*, 753–764.

Hudson, R. R. (1991). *Gene genealogies and the coalescent process*, vol. 7, 1–44. Oxford University Press.

Hudson, R. R. (2002). Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, *18*, 337–338.

Huelsenbeck, J. P., & Andolfatto, P. (2007). Inference of population structure under

a Dirichlet process model. *Genetics*, *175*, 1787–1802.

Huelsenbeck, J. P., Larget, B., Miller, R. E., & Ronquist, F. (2002). Potential applications and pitfalls of Bayesian inference of phylogeny. *Systematic Biology*, *51*, 673–688.

Huelsenbeck, J. P., Ronquist, F., Nielsen, R., & Bollback, J. P. (2001). Bayesian inference of phylogeny and its impact on evolutionary biology. *Science*, *294*, 2310–2314.

Hukushima, K., & Nemoto, K. (1996). Exchange Monte Carlo method and application to spin glass simulations. *Journal of the Physical Society of Japan*, *65*, 1604–1608.

Hukushima, K., Takayama, H., & Nemoto, K. (1996). Application of an extended ensemble method to spin glasses. *International Journal of Modern Physics C*, *7*, 337–344.

Hunter, D. R., Goodreau, S. M., & Handcock, M. S. (2008). Goodness of fit of social network models. *Journal of the American Statistical Association*, *103*, 248–248.

Iba, Y. (2000). Population Monte Carlo algorithms. *Journal of the Japanese Society for Artificial Intelligence*, *16*, 279–286.

Iba, Y. (2001). Extended ensemble Monte Carlo. *International Journal of Modern Physics*, *12*, 623–656.

Izmailov, A. F., & Solodov, M. V. (2004). Newton-type methods for optimization problems without constraint qualifications. *SIAM Journal on Optimization*, *15*, 210–228.

Jaakkola, T., Jordan, M. I., & Singh, S. P. (1994). On the convergence of stochastic iterative dynamic programming algorithms. *Neural Computation*, *6*, 1185–1201.

Jaakkola, T. S. (1997). *Variational methods for inference and learning in graphical models*. Doctoral dissertation, Massachusetts Institute of Technology.

Jarzynski, C. (1997). Nonequilibrium equality for free energy differences. *Physical Review Letters*, *78*, 2690–2693.

Jasra, A., Stephens, D., & Holmes, C. (2007). On population-based simulation for static inference. *Statistics and Computing*, *17*, 263–279.

Jeffreys, H., & Swirles, B. (1956). *Methods of mathematical physics*. Cambridge University Press.

Jerrum, M., & Sinclair, A. (1996). The Markov chain Monte Carlo method: an approach to approximate counting and integration. In *Approximation algorithms for NP-hard problems*, 482–520. PWS Publications.

Joachims, T. (2002). *Learning to classify text using support vector machines*. Kluwer/Springer.

Johnson, C. A., Seidel, J., & Sofer, A. (2000). Interior-point methodology for 3-D PET reconstruction. *IEEE Transactions on Medical Imagining*, *19*, 271–285.

Johnson, J. K. (2002). *Min-max Kullback-Leibler model selection* (Technical Report). Massachusetts Institute of Technology.

Jordan, M. I. (2004). Graphical models. *Statistical Science*, *19*, 140–155.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1998). An introduction to variational methods for graphical models. In M. I. Jordan (Ed.), *Learning in graphical models*, 105–161. MIT Press.

Kass, R. E., & Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, *90*, 773–795.

Kawashima, N., & Young, A. P. (1996). Phase transition in the three-dimensional +/-J Ising spin glass. *Physical Review B*, *53*.

Keller, E. F. (2000). *The century of the gene*. Harvard University Press.

Kiefer, J. C., & Wolfowitz, J. (1952). Stochastic estimation of a regression function. *Annals of Mathematical Statistics*, *23*, 462–466.

Kikuchi, R. (1951). A theory of cooperative phenomena. *Physical Review*, *81*, 988–1003.

Kim, S.-J., K. Koh, M. L., Boyd, S., & Gorinevsky, D. (2007). An interior-point method for large-scale L1-regularized least squares. *IEEE Journal of Selected Topics in Signal Processing*, *1*, 606–617.

Kindermann, R., & Snell, J. L. (1980). *Markov random fields and their applications*. American Mathematical Society.

King, G., Honaker, J., Joseph, A., & Scheve, K. (2001). Analyzing incomplete political science data: An alternative algorithm for multiple imputation. *American Political Science Review*, *95*, 49–69.

Kivinen, J. (2003). Online learning of linear classifiers. In S. Mendelson and A. J. Smola (Eds.), *Advanced lectures on machine learning*, vol. 2600 of *Lecture Notes in Computer Science*, 235–257. Springer-Verlag.

Kivinen, J., Smola, A. J., & Williamson, R. C. (2004). Online learning with kernels. *IEEE Transactions on Signal Processing*, *52*, 2165–2176.

Kivinen, K., & Warmuth, M. K. (1997). Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, *132*, 1–63.

Klaas, M., Briers, M., de Freitas, N., Doucet, A., Maskell, S., & Lang, D. (2006). Fast particle smoothing: if I had a million particles. *Proceedings of the 23rd international conference on Machine learning* (pp. 481–488).

Klimt, B., & Yang, Y. (2004). The Enron corpus: A new dataset for email classification research. *Proceedings of the 15th European Conference on Machine Learning* (pp. 217–226).

Kok, S., Sumner, M., Richardson, M., Singla, P., Poon, H., Lowd, D., Wang, J., & Domingos, P. (2009). *The Alchemy system for statistical relational AI* (Technical Report). Dept. of Computer Science and Engineering, University of Washington.

Kolmogorov, V., & Zabih, R. (2004). What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*, 147–159.

Kong, A., Liu, J. S., & Wong, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, *89*, 278–

288.

Körding, K., & Wolpert, D. M. (2006). Bayesian decision theory and sensorimotor control. *Trends in Cognitive Science*, *10*, 316–326.

Kschischang, F. R., Frey, B. J., & Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, *47*, 498–519.

Kushner, H. J., & Clark, D. S. (1978). *Stochastic approximation methods for constrained and unconstrained systems*. Springer-Verlag.

Kushner, H. J., & Yang, J. (1995). Analysis of adaptive step-size stochastic approximation algorithms for parameter tracking. *IEEE Transactions on Automatic Control*, *40*, 1403–1410.

Kushner, H. J., & Yin, G. G. (2003). *Stochastic approximation and recursive algorithms and applications*. Springer.

Lafferty, J. (1999). Additive models, boosting, and inference for generalized divergences. *Proceedings of the 12th Annual Conference on Computational Learning Theory* (pp. 125–133).

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: probabilistic models for segmenting and labeling sequence data. *Proceedings of the 18th International Conference on Machine Learning* (pp. 282–289).

Lancaster, T. (2004). *Introduction to modern Bayesian econometrics*. Viley-Blackwell.

Landau, D. P., & Binder, K. (2005). *A guide to Monte Carlo simulations in statistical physics*. Cambridge University Press.

Landau, D. P., Tsai, S.-H., & Exler, M. (2004). A new approach to Monte Carlo simulations in statistical physics: Wang-Landau sampling. *American Journal of Physics*, *72*, 1294–1302.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, *25*, 259–284.

Lawrence, G., Cowan, N., & Russell, S. (2003). Efficient gradient estimation for motor control learning. *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*.

Lawrence, N. D. (2000). *Variational inference in probabilistic models*. Doctoral dissertation, Cambridge University.

Lee, H., Battle, A., Raina, R., & Ng, A. Y. (2007). Efficient sparse coding algorithms. In *Advances in neural information processing systems*, vol. 19, 801–808. MIT Press.

Lee, H., Ekanadham, C., & Ng, A. Y. (2008). Sparse deep belief network model for visual area V2. In *Advances in neural information processing systems*, vol. 20, 881–888. MIT Press.

Leyffer, S., López-Calva, G., & Nocedal, J. (2006). Interior methods for mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, *17*, 52–77.

Li, S. Z. (1994). Markov random field models in computer vision. *Proceedings of the 3rd European Conference on Computer Vision* (pp. 361–370).

Liang, F. (2002). Dynamically weighted importance sampling in Monte Carlo computation. *Journal of the American Statistical Association*, *97*, 807–821.

Liang, F., Liu, C., & Carroll, R. J. (2007). Stochastic approximation in Monte Carlo computation. *Journal of the American Statistical Association*, *102*, 305–320.

Liang, F., & Wong, W. H. (2001). Real-parameter evolutionary Monte Carlo with applications to Bayesian mixture models. *Journal of the American Statistical Association*, *96*, 653–666.

Liu, J. S. (1996). Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, *6*, 113–119.

Liu, J. S., Wong, W., & Kong, A. (1994). Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and sampling schemes. *Biometrika*, *81*, 27–40.

Liu, J. S., & Wu, Y. N. (1999). Parameter expansion for data augmentation. *Journal of the American Statistical Association*, *94*, 1264–1274.

Ljung, L. (1977). Analysis of recursive stochastic algorithms. *IEEE Transactions on Automatic Control*, *22*, 551–575.

Ljung, L., & Söderström, T. (1983). *Theory and practice of recursive identification*. MIT Press.

Luenberger, D. G. (1979). *Introduction to dynamic systems*. John Wiley & Sons.

Lustig, I. J., Marsten, R. E., & Shanno, D. F. (1992). On implementing Mehrotra's predictor-corrector interior-point method for linear programming. *SIAM Journal on Optimization*, *2*, 435–449.

Mailund, T., Schierup, M. H., Pedersen, C. N. S., Mechlenborg, P. J. M., Madsen, J. N., & Schauser, L. (2005). CoaSim: a flexible environment for simulating genetic data under coalescent models. *BMC Bioinformatics*, *6*.

Manel, S., Berthier, P., & Luikart, G. (2002). Detecting wildlife poaching: identifying the origin of individuals with Bayesian assignment tests and multilocus genotypes. *Conservation Biology*, *16*, 650–659.

Marinari, E., & Parisi, G. (1992). Simulated tempering: A new Monte Carlo scheme. *Europhysics Letters*, *19*, 451–458.

Marlin, B. M. (2008). *Missing data problems in machine learning*. Doctoral dissertation, University of Toronto.

Marroquin, J. L. (1985). *Probabilistic solution of inverse problems*. Doctoral dissertation, Massachusetts Institute of Technology.

Martinez-Cantin, R., de Freitas, N., & Castellanos, J. A. (2007). Analysis of particle methods for simultaneous robot localization and mapping and a new algorithm: Marginal-SLAM. *Proceedings of the International Conference on Robotics and Automation* (pp. 2415–2420).

Mathews, V. J., & Xie, Z. (1993). A stochastic gradient adaptive filter with gradient adaptive step size. *IEEE Transactions on Signal Processing*, *41*, 2075–2087.

McCallum, A., Corrada-Emmanuel, A., & Wang, X. (2005). Topic and role discov-

ery in social networks. *Proceedings of the 19th International Joint Conference on Artificial Intelligence* (pp. 786–791).

McEliece, R. J., & Yildirim, M. (2002). Belief propagation on partially ordered sets. In D. Gilliam and J. Rosenthal (Eds.), *Mathematical systems theory in biology, communications, computation and finance*, 275–299. Springer.

Meier, L., de Geer, S. V., & Bühlmann, P. (2008). The group Lasso for logistic regression. *Journal of the Royal Statistical Society*, *70*, 53–71.

Métivier, M. (1982). *Semimartingales*. de Gruyter.

Métivier, M., & Priouret, P. (1984). Applications of a kushner and clark lemma to general classes of stochastic algorithms. *IEEE Transactions on Information Theory*, *30*, 140–151.

Metropolis, N., & Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association*, *44*, 335–341.

Meyn, S. P., & Tweedie, R. L. (1993). *Markov chains and stochastic stability*. Springer-Verlag.

Milch, B. (2006). *Probabilistic models with unknown objects*. Doctoral dissertation, University of California, Berkeley.

Milch, B., Marthi, B., Russell, S., Sontag, D., Ong, D. L., & Kolobov, A. (2005). BLOG: Probabilistic models with unknown objects. *Proceedings of the 19th International Joint Conference on Artificial Intelligence* (pp. 1352–1359).

Milch, B., Zettlemoyer, L. S., Kersting, K., Haimes, M., & Kaelbling, L. P. (2008). Lifted probabilistic inference with counting formulas. *Proceedings of the 23rd National Conference on Artificial Intelligence* (pp. 1062–1068).

Minka, T. (2001a). Expectation propagation for approximate Bayesian inference. *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence* (pp. 362–369).

Minka, T. (2001b). *A family of algorithms for approximate Bayesian inference*. Doctoral dissertation, Massachusetts Institute of Technology.

Minka, T., & Lafferty, J. (2002). Expectation-propagation for the generative aspect model. *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence* (pp. 352–359).

Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.

Monteiro, R. D. C., & Adler, I. (1989a). Interior path following primal-dual algorithms, part I: linear programming. *Mathematical Programming*, *44*, 27–41.

Monteiro, R. D. C., & Adler, I. (1989b). Interior path following primal-dual algorithms, part II: convex quadratic programming. *Mathematical Programming*, *44*, 43–66.

Moon, T. K. (2005). *Error correction coding: mathematical methods and algorithms*. Wiley-Interscience.

Morris, C. N. (1982). Natural exponential families with quadratic variance functions. *Annals of Statistics*, *10*, 65–80.

Muñoz, V., & Eaton, W. A. (1999). A simple model for calculating the kinetics of protein folding from three-dimensional structures. *Proceedings of the National Academy of Sciences*, *96*, 11311–11316.

Murphy, K. P. (2002). *Dynamic Bayesian networks: representation, inference and learning*. Doctoral dissertation, University of California, Berkeley, Computer Science Division.

Murray, I., & Salakhutdinov, R. (2009). Evaluating probabilities under high-dimensional latent variable models. In *Advances in neural information processing systems*, vol. 21. MIT Press.

Muyan, P., & de Freitas, N. (2003). A Blessing of dimensionality: measure concentration and probabilistic inference. *Proceedings of the 19th Workshop on Artificial Intelligence and Statistics*.

Narayanan, A. (1991). Algorithm AS 266: Maximum likelihood estimation of the parameters of the Dirichlet distribution. *Applied Statistics*, *40*, 365–374.

Neal, R., & Hinton, G. (1998). A view of the EM algorithm that that justifiies incremental, sparse, and other variants. In M. I. Jordan (Ed.), *Learning in graphical models*, 355–368. Kluwer Academic.

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, *11*, 125–139.

Nedic, A., & Bertsekas, D. P. (2001). Incremental subgradient methods for nondifferentiable optimization. *SIAM Journal on Optimization*, *12*, 109–138.

Nei, M. (1973). Analysis of gene diversity in subdivided populations. *Proceedings of the National Academy of Sciences*, *70*, 3321–3323.

Neigel, J. E. (2002). Is F_{ST} obsolete? *Conservation Genetics*, *3*, 167–173.

Neville, J., & Jensen, D. (2007). Relational dependency networks. In L. Getoor and B. Taskar (Eds.), *Introduction to statistical relational learning*, 239–268. MIT Press.

Newman, M. E. J., & Barkema, G. T. (1999). *Monte Carlo methods in statistical physics*. Oxford University Press.

Nocedal, J., & Wright, S. J. (2006). *Numerical optimization*. Springer. 2nd edition.

Olshausen, B. A., & Field, D. J. (1997). Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vision Research*, *37*, 3311–3325.

Opper, M., & Saad, D. (Eds.). (2001). *Advanced mean field methods, theory and practice*. MIT Press.

Park, D. K., Gelman, A., & Bafumi, J. (2004). Bayesian multilevel estimation with poststratification: state-level estimates from national polls. *Political Analysis*, *12*, 375–385.

Park, J. D., & Darwiche, A. (2003). Solving MAP exactly using systematic search. *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence* (pp. 459–468).

Paskin, M. A. (2004). *Exploiting locality in probabilistic inference*. Doctoral dissertation, Universeity of California, Berkeley.

Pearl, J. (1984). *Heuristics: intelligent search strategies for computer problem solving*. Addison-Wesley.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.

Pearse, D. E., & Crandall, K. A. (2004). Beyond FST: Analysis of population genetic data for conservation. *Conservation Genetics*, *5*, 585–602.

Pearson, M., & Michell, L. (2000). Smoke Rings: social network analysis of friendship groups, smoking and drug-taking. *Drugs: education, prevention and policy*, *7*, 21–37.

Peot, M. A., & Shachter, R. D. (1991). Fusion and propagation with multiple observations in belief networks. *Artificial Intelligence*, *48*, 299–318.

Peters, G. W. (2005). Topics in sequential Monte Carlo samplers. Master's thesis, University of Cambridge.

Poljak, B. T. (1978). Nonlinear programming methods in the presence of noise. *Mathematical Programming*, *14*, 87–97.

Poole, D. (1997). The Independent Choice Logic for modelling multiple agents under uncertainty. *Artificial Intelligence*, *94*, 5–57.

Poole, D. (2000). Abducting through negation as failure: Stable models with the Independent Choice Logic. *Journal of Logic Programming*, *44*, 5–35.

Poole, D. (2003). First-order probabilistic inference. *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 985–991.

Poupart, P. (2005). *Exploiting structure to efficiently solve large scale partially observable markov decision processes*. Doctoral dissertation, University of Toronto.

Powell, M. J. D. (1978). Algorithms for nonlinear constraints that use Lagrangian functions. *Mathematical Programming*, *14*, 224–248.

Pritchard, J. K., Stephens, M., & Donnelly, P. (2000a). Inference of population structure using multilocus genotype data. *Genetics*, *155*, 945–959.

Pritchard, J. K., Stephens, M., Rosenberg, N. A., & Donnelly, P. (2000b). Association mapping in structured populations. *American Journal of Human Genetics*, *67*, 170–181.

Punt, A., & Hilborn, R. (1997). Fisheries stock assessment and decision analysis: the Bayesian approach. *Reviews in Fish Biology and Fisheries*, *7*, 35–63.

Rannala, B. (2002). Identifiability of parameters in MCMC Bayesian inference of phylogeny. *Systematic Biology*, *51*, 754–760.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. MIT Press.

Ravikumar, P. (2007). *Approximate inference, structure learning and feature estimation in Markov random fields*. Doctoral dissertation, Carnegie Mellon University.

Ravikumar, P., Liu, H., Lafferty, J., & Wasserman, L. (2008). SpAM: sparse additive models. In *Advances in neural information processing systems*, vol. 20, 1201–1208. MIT Press.

Rees, T., & Greif, C. (2007). A preconditioner for linear systems arising from interior point optimization methods. *SIAM Journal on Scientific Computing*, *29*, 1992–2007.

Rice, J. A. (1988). *Mathematical statistics and data analysis*. Wadsworth & Brooks/Cole.

Richardson, M., & Domingos, P. (2006). Markov logic networks. *Machine Learning*, *62*, 107–136.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, *22*, 400–407.

Robert, C. P., & Casella, G. (2004). *Monte Carlo statistical methods*. Springer. 2nd edition.

Rosen, K. H. (2007). *Discrete mathematics and its applications*. McGraw-Hill.

Rosen-Zvi, M., Griffiths, T., Steyvers, M., & Smyth, P. (2004). The author-topic model for authors and documents. *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence* (pp. 487–494).

Rosenberg, N. A. (2004). DISTRUCT: a program for the graphical display of population structure. *Molecular Ecology Notes*, *4*, 137–138.

Rosenberg, N. A., Mahajan, S., Ramachandran, S., Zhao, C., Pritchard, J. K., & Feldman, M. W. (2005). Clines, clusters, and the effect of study design on the inference of human population structure. *PLoS Genetics*, *1*.

Rosenberg, N. A., Pritchard, J. K., Weber, J. L., Cann, H. M., Kidd, K. K., Zhivotovsk, L. A., & Feldman, M. W. (2002). Genetic structure of human populations. *Science*, *298*, 2381–2385.

Rosenbluth, M. N., & Rosenbluth, A. W. (1955). Monte Carlo calculation of the average extension of molecular chains. *Journal of Chemical Physics*, *23*, 356–359.

Ross, S. M. (2007). *Introduction to probability models*. Academic Press. 9th edition.

Roux, N. L., & Bengio, Y. (2000). Topmoumoute online natural gradient. In *Advances in neural information processing systems*, vol. 20, 849–856. MIT Press.

Rubenstein, R. Y., & Kroese, D. P. (2004). *The cross-entropy method: a unified approach to combinatorial optimization, Monte Carlo simulation and machine learning*. Springer-Verlag.

Rudin, L. I., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D*, *60*, 259–268.

Ruppert, D. (1985). A Newton-Raphson version of the multivariate Robbins-Monro procedure. *Annals of Statistics*, *13*, 236–245.

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). LabelMe: a database and web-based tool for image annotation. *Intl. Journal of Computer Vision*, *77*, 157–173.

Russell, S. J. (1997). Rationality and intelligence. *Artificial Intelligence*, *94*, 57–77.

Saad, D. (Ed.). (1998). *On-line learning in neural networks*. Cambridge University Press.

Saad, Y. (1996). *Iterative methods for sparse linear systems*. PWS Publishers.

Sadegh, P. (1997). Constrained optimization via stochastic approximation with a simultaneous perturbation gradient approximation. *Automatica*, *33*, 889–892.

Sadowsky, J. S., & Bucklew, J. A. (1990). On large deviations theory and asymptotically efficient Monte Carlo estimation. *IEEE Transactions on Information Theory*, *36*, 579–588.

Salakhutdinov, R., Mnih, A., & Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. *Proceedings of the 24th International Conference on Machine Learning* (pp. 791–798).

Samet, J. M., Dominici, F., Curriero, F. C., Coursac, I., & Zeger, S. L. (2000). Fine particulate air pollution and mortality in 20 u.s. cities, 1987-1994. *The New England Journal of Medicine*, *343*, 1742–1749.

Sardy, S., Bruce, A. G., & Tseng, P. (2000). Block coordinate relaxation methods for nonparametric wavelet denoising. *Journal of Computational and Graphical Statistics*, *9*, 361–379.

Sato, M. (2000). On-line EM algorithm for the normalized Gaussian network. *Neural Computation*, *12*, 407–432.

Sato, M.-A. (2001). Online model selection based on the variational Bayes. *Neural Computation*, *13*, 1649–1681.

Saul, L., & Kardar, M. (1993). Exact integer algorithm for the two-dimensional +/-J Ising spin glass. *Physical Review E*, *48*.

Saul, L. K., & Jordan, M. I. (1996). Exploiting tractable structures in intractable networks. In *Advances in neural information processing systems*, vol. 8, 486–492. MIT Press.

Scheber, T. (1973). Stochastic approximation: a survey. Master's thesis, Naval Postgraduate School, Springfield, Virginia.

Schenk, O., & Gärtner, K. (2006). On fast factorization pivoting methods for sparse symmetric indefinite systems. *Electronic Transactions on Numerical Analysis*, *23*, 158–179.

Schmidt, M., Fung, G., & Rosales, R. (2007). Fast optimization methods for L1 regularization. *Proceedings of the 18th European Conference on Machine Learning* (pp. 286–297).

Schraudolph, N. N., Yu, J., & Günter, S. (2007a). A stochastic quasi-Newton method for online convex optimization. *In Proceedings of the 11th Conference on Artificial Intelligence and Statistics* (pp. pp. 433–440).

Schraudolph, N. N., Yu, J., & Günter, S. (2007b). A stochastic quasi-Newton method for online convex optimization. *Proceedings of the 11th Conference on Artificial Intelligence and Statistics* (pp. 433–440).

Sha, F., Park, Y. A., & Saul, L. K. (2007). Multiplicative updates for l1-regularized linear and logistic regression. *Proceedings of the 7th International Symposium on Intelligent Data Analysis* (pp. 13–24). Springer.

Shachter, R. D. (1998). Bayes-Ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence* (pp. 480–487).

Shah, S. P., Lam, W. L., Ng, R. T., & Murphy, K. P. (2007). Modeling recurrent DNA copy number alterations in array CGH data. *Bioinformatics*, *23*.

Shalev-Shwartz, S., Singer, Y., & Srebro, N. (2007). Pegasos: primal estimated subgradient solver for SVM. *Proceedings of the 24th Intl. Conference on Machine learning* (pp. 807–814).

Shen, Y., Ng, A., & Seeger, M. (2006). Fast Gaussian process regression using KD-Trees. In *Advances in neural information processing systems*, vol. 18, 1225–1232. MIT Press.

Sherrington, D. (2007). *Spin glasses: a perspective*, 45–62. Springer.

Snijders, T. A. B. (2002). Markov chain Monte Carlo estimation of exponential random graph models. *Journal of Social Structure*, *3*.

Spall, J. C. (2000). Adaptive stochastic approximation by the simultaneous perturbation method. *IEEE Transactions on Automatic Control*, *45*, 1839–1853.

Spall, J. C. (2003). *Introduction to stochastic search and optimization*. Wiley-Interscience.

Spiegelhalter, D. J., Dawid, A. P., Lauritzen, S. L., & Cowell, R. G. (1993). Bayesian analysis in expert systems. *Statistical Science*, *8*, 219–247.

Steen, F. H. (1982). *Elements of probability and mathematical statistics*. Duxbury Press.

Stephens, M. (2002). Dealing with label switching in mixture models. *Journal of the Royal Statistical Society*, *62*, 795–809.

Strang, G. (1980). *Linear algebra and its applications*. Academic Press. Second edition.

Strang, G. (1991). *Calculus*. Wellesley-Cambridge Press.

Sudderth, E. B., Torralba, A., Freeman, W. T., & Willsky, A. S. (2005). Learning hierarchical models of scenes, objects, and parts. *Proceedings of the 10th International Conference on Computer Vision* (pp. 1331–1338).

Sun, D., Roth, S., Lewis, J., & Black, M. J. (2008). Learning optical flow. *Proceedings of the 10th European Conference on Computer Vision*.

Sutton, R. S., McAllester, D., Singh, S., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, vol. 12, 1057–1063. MIT Press.

Swendsen, R. H., & Wang, J.-S. (1986). Replica Monte Carlo simulation of spin-glasses. *Physical Review Letters*, *57*, 2607–2609.

Szeliski, R. (1989). *Bayesian modeling of uncertainty in low-level vision*. Kluwer Academic Publishers.

Taskar, B., Abbeel, P., & Koller, D. (2002). Discriminative probabilistic models for relational data. *Proceedings of the 18th Conference on Uncertainty in Artificial*

Intelligence (pp. 485–492).

Taskar, B., Wong, M.-F., Abbeel, P., & Koller, D. (2004). Link prediction in relational data. In *Advances in neural information processing systems*, vol. 16, 659–666. MIT Press.

Teh, Y. W., Newman, D., & Welling, M. (2007a). A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *Advances in neural information processing systems*, vol. 19. MIT Press.

Teh, Y. W., Newman, D., & Welling, M. (2007b). A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *Advances in neural information processing systems*, vol. 19, 1353–1360. MIT Press.

Tenenbaum, J. B., Griffiths, T. L., & Kemp, C. (2006). Theory-based Bayesian models of inductive learning and reasoning. *Trends in Cognitive Science*, *10*, 309–318.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society*, *58*, 267–288.

Tipping, M. E. (2004). Bayesian inference: An introduction to principles and practice in machine learning. In O. Bousquet, von U. Luxburg and G. Rätsch (Eds.), *Advanced lectures on machine learning*, vol. 3176 of *Lecture Notes in Computer Science*, 41–62. Springer.

Titterington, D. M. (1984). Recursive parameter estimation using incomplete data. *Journal of the Royal Statistical Society, Series B*, *46*, 257–267.

Trefethen, L. N., & Bau, D. (1997). *Numerical linear algebra*. SIAM.

Tropp, J. A. (2006). Just relax: Convex programming methods for identifying sparse signals. *IEEE Transactions on Information Theory*, *51*, 1030–1051.

van den Berg, E., Schmidt, M., Friedlander, M., & Murphy, K. (2008). *Group sparsity via linear-time projection* (Technical Report). University of British Columbia.

Vandenberghe, L., & Boyd, S. (1996). Semidefinite programming. *SIAM Review*, *38*, 49–95.

Vanderbei, R. J. (1995). Symmetric quasidefinite matrices. *SIAM Journal on Optimization*, *5*, 100–113.

Vishwanathan, S. V. N., Schraudolph, N. L., Schmidt, M., & Murphy, K. (2006). Accelerated training of conditional random fields with stochastic gradient methods. *Proceedings of the 23rd International Conference on Machine Learning*.

Wächter, A. (2002). *An interior point algorithm for large-scale nonlinear optimization with applications in process engineering*. Doctoral dissertation, Carnegie Mellon University.

Wainwright, M. J. (2002). *Stochastic processes on graphs with cycles: geometric and variational approaches*. Doctoral dissertation, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Wainwright, M. J., Jaakkola, T. S., & Willsky, A. S. (2005). A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, *51*, 2313–2335.

Wainwright, M. J., & Jordan, M. I. (2003a). *Graphical models, exponential families, and variational inference* (Technical Report). University of California, Berkeley.

Wainwright, M. J., & Jordan, M. I. (2003b). Variational inference in graphical models: the view from the marginal polytope. *Proceedings of the 41st Allerton Conference*.

Waltz, R. A., Morales, J. L., Nocedal, J., & Orban, D. (2006). An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Mathematical Programming*, *107*, 391–408.

Wang, F., & Landau, D. P. (2001). Determining the density of states for classical statistical models: A random walk algorithm to produce a flat histogram. *Physical Review E*, *64*.

Wang, I., & Spall, J. C. (2003). Stochastic optimization with inequality constraints using simultaneous perturbations and penalty functions. *Proc. 42nd IEEE Conf. Decision and Control*.

Wang, S. B., Quattoni, A., Morency, L.-P., Demirdjian, D., & Darrell, T. (2006). Hidden conditional random fields for gesture recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1521–1527).

Weir, B. S., & Cockerham, C. C. (1984). Estimating F-statistics for the analysis of population structure. *Evolution*, *38*, 1358–1370.

Wiegerinck, W. (2000). Variational approximations between mean field theory and the junction tree algorithm. *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence* (pp. 626–633).

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, *8*, 229–256.

Winn, J., & Shotton, J. (2006). The layout consistent random field for recognizing and segmenting partially occluded objects. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 37–44).

Wright, M. H. (1992). Interior methods for constrained optimization. *Acta Numerica*, 341–407.

Wright, M. H. (1994). Some properties of the Hessian of the logarithmic barrier function. *Mathematical Programming*, *67*, 265–295.

Wright, M. H. (1995). Why a pure primal Newton barrier step may be infeasible. *SIAM Journal on Optimization*, *5*, 1–12.

Wright, M. H. (1998). Ill-conditioning and computational error in interior methods for nonlinear programming. *SIAM Journal on Optimization*, *9*, 84–111.

Wright, S. J. (1997). *Primal-dual interior-point methods*. SIAM.

Wright, S. J. (2001). Effects of finite-precision arithmetic on interior-point methods for nonlinear programming. *SIAM Journal on Optimization*, *12*, 36–78.

Xing, E. P., Jordan, M. I., & Karp, R. M. (2001). Feature selection for high-dimensional genomic microarray data. *Proceedings of the 18th International Conference on Machine Learning* (pp. 601–608).

Xing, E. P., Jordan, M. I., & Russell, S. (2003). A generalized mean field algorithm

for variational inference in exponential families. *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence* (pp. 583–591).

Yedidia, J. S. (2001). An idiosyncratic journey beyond mean field theory. In M. Opper and D. Saad (Eds.), *Advanced mean field methods, theory and practice*, 21–36. MIT Press.

Yedidia, J. S., Freeman, W. T., & Weiss, Y. (2005). Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, *51*, 2282–2312.

Younes, L. (1991). Stochastic gradient estimation strategies for Markov random fields. *Proceedings of the Spatial Statistics and Imaging Conference*.

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society*, *68*, 49–67.

Zhang, J., & Fossorier, M. (2006). Mean field and mixed mean field iterative decoding of low-density parity-check codes. *IEEE Transactions on Information Theory*, *52*, 3168–3185.

Zhang, N. L., & Poole, D. (1994). A simple approach to Bayesian network computations. *Proceedings of the 10th Canadian Artificial Intelligence Conference* (pp. 171–178).

Zhang, N. L., & Poole, D. (1996). Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, *5*, 263–313.

Zheng, A. (2005). *Statistical software debugging*. Doctoral dissertation, University of California, Berkeley.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. *Proceedings of the 20th International Conference on Machine Learning*.